Processor Sharing Models for Bluetooth–Based File Sharing Systems

Urtzi Ayesta [*†] and Daniele Miorandi [†‡]

[*] France Telecom R&D, email: {urtzi.ayesta@francetelecom.com }
[†] INRIA Sophia Antipolis, email: {urtzi.ayesta,daniele.miorandi}@sophia.inria.fr
[‡] Dep. of Information Engineering, Univ. of Padova (Italy), email: daniele.miorandi@dei.unipd.it

***Abstract*— In this paper we analyze the performance of a file sharing system based on the Bluetooth technology. Files are stored in a server to which clients may connect by setting up a Bluetooth connection. Files are transmitted over the TCP/IP protocol suite, and users alternate between activity and idle periods. In the first part of the paper we concentrate on the simplest configuration, an isolated piconet, and propose a processor sharing model for the computation of the mean session delay for interactive users. Then, we focus on the more complex scatternet configuration, and propose a discriminatory processor sharing model which provides insight into the network performance. Based on the latter, we draw a connection to some classical results in the field of scheduling theory, which can provide useful guidelines for the design of effective scatternet management schemes.**

***Keywords*— Bluetooth, Processor Sharing, Discriminatory Processor Sharing, TCP, Scatternet Management**

I. Introduction

The Bluetooth technology, promoted by the Bluetooth Special Interest Group [1], aims at providing wireless connectivity to low–power low–cost devices.

In this paper we present a performance study of a file sharing system based on the Bluetooth radio interface. Various devices, such as PDAs or laptops, exploit wireless connections (provided by means of Bluetooth transceivers) in order to download files from a central server. Nodes alternate between activity periods, correspondent to the download of a file from the central server, and idle periods (referred to as thinking times), during which they keep silent. Files are transmitted using the standard TCP/IP protocol suite; a TCP session is open for each file transfer.

The first part of the paper focuses on the basic piconet configuration. We start by computing, in presence of k active connections and n slaves, the throughput experienced by any of the TCP connections under some typical segmentation–and–reassembly (SAR) policies [2]. The TCP thorughput analysis is then extended to an optimal SAR policy and to an optimal polling schemes, which upperbounds the performance of any polling algorithm. Then, such results are used to build a closed single–server processor sharing queue with state–dependent arrival and service rates, which is used to evaluate the network performance in terms of the mean session delay. Insensitivity properties of the model are discussed.

The analysis is then extended to multihop Bluetooth networks, commonly referred to as scatternets. In this case, we analysed the system performance by means of an equivalent single–server discriminatory processor sharing queue. On the basis of such equivalent model, we discuss how efficient algorithms for dynamic scatternet management can be designed. In particular, we show how some classical results in the field of scheduling theory could be used to design what we refer to as topology–based scatternet management schemes, a family of algorithms which dynamically modify the scatternet structure in order to enhance network performance.

The paper is organized as follows: in Sec. II some background on the Bluetooth technology and the processor sharing (and its discriminatory variant) scheduling techniques are presented. Sec. III presents the analysis of the network performance for the piconet case. In Sec. IV the use of a discriminatory processor sharing model of the scatternet is discussed. Sec. V concludes the paper pointing out some issues and directions for future research.

II. Background

A. The Bluetooth Technology

The basic Bluetooth network configuration is the so–called piconet, a group of no more than eight devices organised according to a master–slave architecture. In particular, the master controls the access to the channel in a centralized way, by polling the slaves and asking for data transmission. In such a way, a master–driven time division duplexing (TDD) mechanism for medium access control is provided. The standard does not specify the polling strategy to be employed, whose choice is left to the manufacturer. Although offering poor performance, limited–1 pure round robin is the current choice, due to its ease of implementation, low cost and low power consumption.

At the physical layer, a frequency–hopping spread spectrum scheme is employed, in order to obey the strict requirements, in terms of power spectrum, necessary for operations in the unlicensed 2.4 GHz–centered ISM band. Different piconets use different frequency hopping (FH) sequences, so that they can overlap in time and space without causing excessive interference, at least for a limited number of piconets [3]. Two or more piconets may be interconnected by sharing, on a time basis, a common device, forming what is commonly referred to as a scatternet. As an example, a piconet with 3 slaves and a simple two–level scatternet with a master/slave bridge are depicted in Fig. 1.

The baseband layer protocol encompasses two types of links. One,

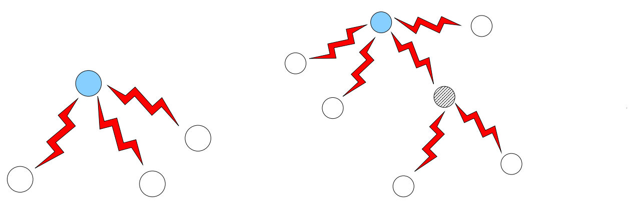

(a) A single piconet.   (b) A scatternet with a M/S bridge.

Fig. 1. Bluetooth network virtual topologies: two simple cases.

sycnhronous connection oriented (SCO), is aimed at the transport of real–time services (mainly voice), and is based upon a periodical reservation scheme. The other, asynchronous connectionless (ACL), is designed for the transport of elastic data traffic. Since in this paper we deal with data traffic only, we restrict our analysis to ACL links. The standard offers, for ACL links, six different packet formats, which differ in length (1, 3 of 5 time slots, where a slot duration is $T = 0.625$ ms) and in the presence/absence

Type	Slot occupancy	Max.payload length (bytes)	FEC rate
DM1	1	17	$2/3$
DM3	3	121	$2/3$
DM5	5	224	$2/3$
DH1	1	27	–
DH3	3	183	–
DH5	5	339	–

Tab. 1. Packet characteristics for ACL links

of forward error correction (a shortened $(15, 10)$ Hamming code). Packet formats are denoted as $D\xi x$, where $\xi = M$ for protected packets and $\xi = H$ otherwise, and $x = 1, 3, 5$ denotes the packet length (expressed in slots). A resume of the packet characteristics is reported in Tab. 1.

In this paper we will limit our analysis to the case of error-free channels, so that we will deal only with unprotected packets. Extensions of the results to lossy links will be discussed later.

Higher layer protocol multiplexing is provided by the logical link control and adaptation protocol (L2CAP), which acts on top of the baseband layer. L2CAP provides also segmentation–and–reassembly (SAR) capabilities; the SAR policy to be used is not defined in the standard and is left to the manufacturers choice.

Tipically, TCP–conveyed traffic is transported over PPP over RFCOMM, which provides serial cable emulation. However, the specification is open, and it is possible to implement IP directly over L2CAP. While such implementation choice may pose internet-working problems, it is able to achieve higher performance levels, due to the considerable overhead savings. Further, it represents a reasonable choice in a scenario as the one we are considering, where Bluetooth is used to implement a local area network. For such reasons, we assume that TCP/IP is layered directly over L2CAP.

B. Network scenario

The analysis will focus on a file–sharing systems, where Bluetooth–enabled devices are used to access and browse files which are hosted on a central server. Users are interactive, i.e. they alternate between activity periods (correspondent to the download of a file from the central unit) and thinking times (which correspond to the time to read a web page in the case of HTTP transfers or to listen to an MP3 file just downloaded).

Two configurations will be considered, a piconet and a two–level scatternet with master/slave bridges. In the first case, in order to optimize network performance, the server is chosen to act as master device [4]. In the scatternet case, many choices are possible in terms of network configuration. We opted for working with a tree structure, where the server act as root of the tree configuration. The devices used to interconnect the different piconets are passive devices (i.e. they do not generate any download request) and act as master/slave bridges. While such configuration choice is clearly simplistic, the analysis can nonetheless be extended to more complex configurations. The two–level tree structure lends itself to a simple analysis, enabling us to illustrate our model in details, and showing how a nice connection to scheduling theory can be drawn.

C. Processor Sharing and Discriminatory Processor Sharing

The Processor Sharing (PS) model was formally introduced and analyzed by Kleinrock [5] as the limiting discipline obtained by letting the quota go to zero in the a Round-Robin policy. In the PS model, all jobs present in the system obtain a fair share of the capacity, i.e., assuming a server with unit capacity, if at arbitrary time $t > 0$ there are N jobs in the system each job will be served at rate $1/N$.

Since the pioneering work by Heyman et al. [6], Processor Sharing based models have been proposed and studied in order to characterize the bandwidth sharing performed by the TCP protocol. In [6], the authors consider a bottleneck link shared by a fix number of sources. The state of the sources alternates between activity and thinking periods. Notably, they showed that mean expected performance (after unconditioning on the file size) shows the so-called *insensitivity* property, that is, depends on the distribution of the activity and think periods only through their mean values. Similar models assuming a Poisson arrival process have been studied in [7] and [8]. The model we develop in this paper for the single piconet topology (see Section III for more details) is largely based on the model presented in [6].

However, sometimes the assumption of fair bandwidth sharing is too unrealistic. For example, it is known that the bandwidth sharing performed by TCP is biased against long round-trip-times (RTT). In order to handle such asymmetric bandwidth-sharing, a Discriminatory Processor Sharing (DPS) model may be used. The DPS model was introduced by Kleinrock [9] as a generalization of the egalitarian Processor-Sharing scheduling discipline. In the DPS model, a single server is shared by M job classes. All jobs present in the system are served in parallel with a rate that is controlled by a vector of weights $\{g_k > 0; k = 1, \ldots, M\}$. More precisely, if at an arbitrary time there are N_j jobs of class j present in the system, $j = 1, \ldots, M$, each class k job gets served at rate $g_k / \sum_{j=1}^{M} g_j N_j$ times the server rate. When all the weights are equal, the DPS discipline is equivalent to Processor Sharing system. The interest of DPS lies on the flexibility provided by the vector of weights that controls the service rate of each class. We refer the reader to [10] for a comprehensive mathematical analysis of an open-loop DPS queue with Poisson arrivals. A closed-loop DPS queue is analyzed by Mitra and Weiss in [11]. The applicability of DPS based models for the modeling of session performance in the Internet has been discussed in [12], [13], [14].

In this paper, we apply the results obtained in [11] to model the mean transfer time experienced by Bluetooth users in a scatternet (see Section IV for more details)

III. Performance Analysis: a Processor Sharing Model for The Piconets

We start by analysing a single piconet, consisting of a master and n slaves, among which $k \leq n$ are downloading files from the server. The time it takes to transfer a TCP packet of length L_{TCP} depends on the SAR policy employed[1]. Let us assume that each TCP packet generates, under policy x, $m^{(x)}$ baseband packets of length $l_1, \ldots, l_m$ (taking also into account the L2CAP header). Conversely, a TCP ACK is fragmented into $p^{(x)}$ packets of length $j_1, \ldots, j_p$. We assume that all the nodes involved in the TCP connections have always a packet to send (this corresponds to neglecting the possibility of packet drops and TCP timeout expirations). The number of polling cycles needed to transmit a TCP packet is given by $\max\left\{m^{(x)}, p^{(x)}\right\} = m^{(x)}$ [2]. The average length of a typical cycle is $T\left[k\Psi^{(x)} + 2(n-k)\right]$, where:

$$\Psi^{(x)} = \frac{\sum_{i=1}^{m^{(x)}} l_i + \sum_{i=1}^{p^{(x)}} j_i + \left[m^{(x)} - p^{(x)}\right]}{m^{(x)}}, \qquad (1)$$

where the last term in the numerator accounts for the number of dummy NULL packets sent by the TCP receiver in order to answer to the master's polls.

[1]Note that in the following we limit ourselves to deterministic SAR mechanisms only.

[2]Since the SAR policy does not depend on the packet type, and since TCP data packets are longer than TCP ACKs, we clearly have $m^{(x)} \geq p^{(x)}$ $\forall x$.

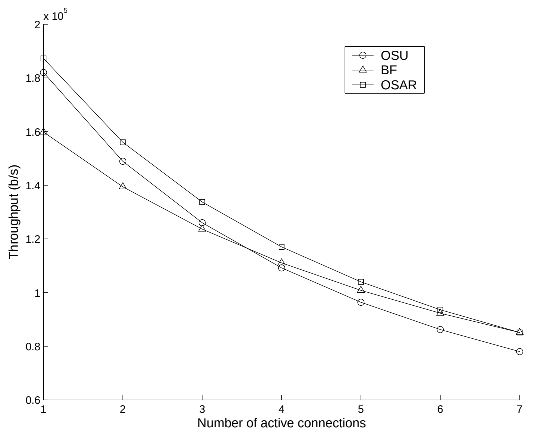


Fig. 2. Average TCP throughput vs. number of active connections for OSU, BF and OSAR SAR policies, $n = 7$.

Hence, the average throguhput experienced by any of the k TCP connections is given by:

$$S(k,n)^{(x)} = \frac{L_{TCP}}{m^{(x)}T\left[k\Psi^{(x)} + 2(n-k)\right]}. \tag{2}$$

We assume TCP data packets to be of fixed lenght equal to 1 Kbyte, and analyse two efficient SAR policies, Best Fit (BF) and Optimum Slot Utilization (OSU). BF tends to reduce the wasted bandwidth, by minimizing the unused packet payload. On the other hand, OSU aims at minimizing the transmission delay of higher layer packets by limiting the number of packets sent. In such a way, some bandwidth is wasted, but the overall number of packets generated is smaller than BF. The reader is referred to [2] for more details.
For the two policies considered, a simple analysis leads to:

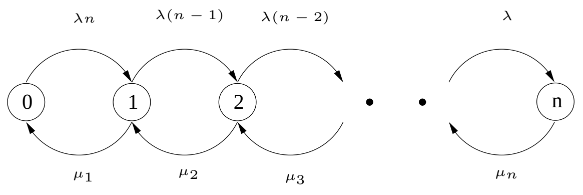
$$m^{(BF)} = 5 \qquad m^{(OSU)} = 4 \tag{3}$$
$$l_1 = l_2 = l_3 = 5 \quad l_1 = l_2 = l_3 = 5 \tag{4}$$
$$l_4 = l_5 = 1 \qquad l_4 = 3 \tag{5}$$
$$p^{(BF)} = 2 \qquad p^{(OSU)} = 1 \tag{6}$$
$$j_1 = j_2 = 1 \qquad j_1 = 3 \tag{7}$$
$$\Psi^{(BF)} = 6 \qquad \Psi^{(OSU)} = 4.4 \tag{8}$$

We also considered what we referred to as Optimum SAR (OSAR) policy. OSAR tends to maximize the throughput experienced by any TCP connections. Thus, from (1) and (2) we see that OSAR is the policy which minimizes

$$\Omega = k\left\{\sum_{i=1}^{m^{(x)}} l_i + \sum_{i=1}^{p^{(x)}} j_i - \left[m^{(x)} + p^{(x)}\right]\right\} + 2m^{(x)}n \quad \forall n, k \le n.$$

For a static policy (i.e. one which does not depend on n or k), and for a packet of length 1 Kbyte, the following values may be found: $m^{(OSAR)} = 4$, $l_1 = l_2 = l_3 = 5$, $l_4 = 3$, $p^{(OSAR)} = 2$, $j_1 = j_2 = 1$, $\Psi^{(OSAR)} = 5.5$.
In Fig. 2 we plotted the TCP throughput, for $n = 7$, as a function of k for all the three SAR policies considered. As it may be seen, if the number of active connections is small ($k \le 3$), OSU outperforms BF, while the roles swap for a number of active connections $k \ge 4$.
This simple model may be readily enlarged to obtain a bound on the performance of any polling policy. The optimal polling policy, in terms of throughput, is the one which avoids the transmissions of dummy POLL packets to slaves which are not downloading from the web server. For this Optimal MAC policy (OMAC), under SAR (x), we have:

$$S(k,n)^{x}_{OMAC} = S(k,k)^{(x)} \quad \forall k \le n. \tag{9}$$



$$\begin{tikzcd} 0 \arrow[r, bend left, "\lambda n"] & 1 \arrow[l, bend left, "\mu_1"] \arrow[r, bend left, "\lambda(n-1)"] & 2 \arrow[l, bend left, "\mu_2"] \arrow[r, bend left, "\lambda(n-2)"] & \bullet \quad \bullet \arrow[l, bend left, "\mu_3"] \arrow[r, bend left, "\lambda"] & n \arrow[l, bend left, "\mu_n"] \end{tikzcd}$$

Fig. 3. Transition probability diagram of the Markov chain $Y(t)$.

It is worth noting, however, that such performance cannot be achieved by a real polling scheme, since OMAC inhibits slaves from requiring the download of a file from the central server, thus causing a deadlock situation.
The obtained results, in terms of throughput, can then be used to get an estimate of the mean session delay. Due to the round robin polling scheme, we can readily construct a processor sharing model of the piconet.
For the moment, let us assume that both the file size (in bits) and the thinking time are exponentially distributed, with respective intensities $\nu = \frac{1}{E[F]}$ and λ, wher $E[F]$ is the mean file size. Then, the dynamic of the system may be represented by a simple birth–death process $Y(t)$, representing the number of active TCP connections at time t. It can be easily seen that Y is a Markov process over the state space $\mathcal{S} = \{0, 1, \ldots, n\}$, as represented in Fig. 3. Given k active connections, new connections are generated at rate $(n-k)\lambda$. In presence of k active customers, the service completion rate (i.e. the rate at which the transfers of files are completed) is given by $\mu_k = \nu S(k,n)$.
We can easily get the limiting distribution for the process Y, obtaining:

$$\pi_k = \frac{\frac{\lambda^k \cdot n(n-1)\ldots(n-k)}{\mu_1 \cdots \mu_k}}{\left(1 + \sum_{i=1}^{n} \frac{\lambda^i n(n-1)\ldots(n-i)}{\mu_1 \cdots \mu_i}\right)}, \quad k = 0, 1, 2, \ldots, n. \tag{10}$$

The mean session delay can be computed applying Little's formula:

$$E[T_s] = \frac{E[Y]}{\overline{\lambda}}, \tag{11}$$

where the mean number of active session is $E[Y] = \sum_{k=0}^{n} k\pi_k$ and the (mean) arrival intensity is $\overline{\lambda} = \lambda \sum_{k=0}^{n-1} (k+1)\pi_k$.
At this stage, we can apply insensitivity results for the $GI|GI|1$ PS queue with state–dependent arrival and service rates [6], [15]. The limiting distribution π_k depends on the file size and thinking time distributions only through their mean values. As a byproduct, we have also insensitivity of the mean session delay.
In Fig. 4 we reported the mean session delay, as a function of the number of slaves, for the three SAR policies considered; the mean file size has been taken equal to 30 Kbyte and the TCP packet size is 1 Kbyte; the mean thinking time is $\frac{1}{\lambda} = 100$ s. The performances obtained with the same parameters, but with an average thinking time equal to $\frac{1}{\lambda} = 10$ s are reported in Fig. 5, where we also plotted the results achievable with a joint use of OSAR and OMAC. As it may be seen, the difference in performance among the different SAR policies is hardly perceived by the final user. On the other hand, it turns out that the choice of an efficient polling algorithm may have a notable influence on the network performance.
Our model can be easily extended to account for the impact of fading channels. Indeed, by assuming that all master–slave links experience the same SNR statistics (which usually holds in such networks with very short range communications), we can easily apply the results of [16] in order to compute the throughput in presence of different fading statistics (Rayleigh, Rice and m–Nakagami), and then plug such results into our queueing model

in order to obtain the mean session delay.

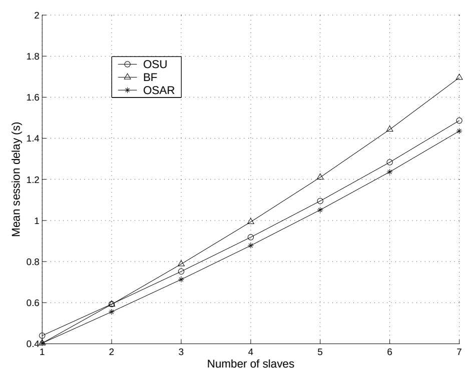


Fig. 4. Mean session delay as a function of the number of slaves, mean file size 30 Kbyte, $\lambda = 0.01\ s^{-1}$.

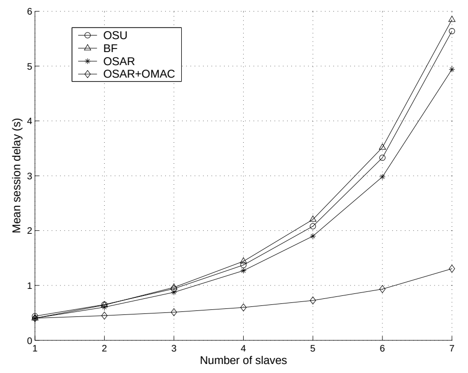


Fig. 5. Mean session delay as a function of the number of slaves, mean file size 30 Kbyte, $\lambda = 0.1\ s^{-1}$.

IV. Performance Analysis: a Discriminatory Processor Sharing Model for The Scatternet

While the processor sharing model presented in the previous section lends itself to a nice and easy analysis, showing an interesting insensitivity property, things become much more complicated when dealing with more complex network configurations. In this section, we aim at showing how, for a simple scatternet structure, a DPS model may be constructed in order to get estimates of the system performance. For the sake of simplicity, let us focus on the simple two–level scatternet structure depicted in Fig. 1. In such case, upon the assumption that the unit which acts as bridge is just a passive device, which stores and forward in– and out–going traffic, it is easy to see that the slaves can be classified into two different classes depending on the piconet they belong to. The ones closer to the server experience a higher throughput (i.e. they have a higher priority) with respect to the ones in the second–level piconet, which have to pass through the bottleneck represented by the shared device. Now, the mean performance of such scatternet can be modeled applying the results of [11] to the case of a 2–classes closed DPS queue.

For completeness we recall some results of [11]. We introduce first some notation. Let n_j be the total number of class j devices, $j = 1, 2$. It is assumed that both the mean active (ON) and think (OFF) periods are exponentially distributed with mean $1/\gamma_j$ and $1/\lambda_j$ respectively $j = 1, 2$. Further, for each class we define

$$b_j = \frac{n_j}{n_1 + n_2}, \qquad w_j = \frac{\gamma_j}{(n_1 + n_2)\lambda_j}.$$

Let $\{g_j > 0\}_{j=1,2}$ be the vector of DPS weights. Let $Y_j(t)$ denote the number of class j sessions active at time $t > 0$. It was proven in [11] that under the "heavy-usage condition" $\frac{n_1\lambda_1}{\gamma_1} + \frac{n_2\lambda_2}{\gamma_2} > 1$, the mean number of class j active sessions $E[Y_j]$ was given by

$$E[Y_j] = \frac{b_j}{1 + \frac{w_j g_j}{L}}, \quad j = 1, 2, \tag{12}$$

where L is the unique positive solution of

$$1 = \sum_{j=1}^{2} \frac{g_j b_j}{L + g_j w_j}. \tag{13}$$

Upon a saturation assumptions, the weights may be computed as $g_1 = \frac{2n_1+1}{2n_1(n_1+1)}$ and $g_2 = \frac{1}{2n_2(n_1+1)}$. Note that, in this case, the weights g_1 and g_2 represent the fraction of time the channel is dedicated to a single node. Further, let the file size be exponentially distributed, with cdf $F_j(\cdot)$, $j = 1, 2$, and let $1/\nu_j$ denote the mean file size.
Then, the mean active time can be computed as

$$1/\gamma_j = g_j/(\nu_j r_e), \quad \text{j=1,2},$$

where $1/\nu_j$ is equal to the mean file size and r_e is the effective service rate (b/s) of the Bluetooth channel. Considering that DH5 packets are used for the TCP segments and DH1 for the ACKs, a straightforward calculation gives $r_e = \frac{339*8}{(5+1)T} = 723200$ b/s, where T is the slot duration.

We will fix the total number of slaves, $n_1 + n_2$, and consider two possible situations. In the first one, $n_1 = n_2$, the mean file size is 40 kbyte for both classes and the mean think times are $1/\lambda_1 = 1/\lambda_2 = 0.08s$. In the second case, which mimes a dynamic scatternet management scheme, jobs transfer data in the high or low priority piconet depending on the size of the requested file. In particular, we consider a threshold of $th = 30kB$ and we assume that sessions smaller than th will be transferred in the high-priority piconet, whereas those larger than th sessions will be connected through the low priority one. As a consequence $1/\lambda_1 = 0.08F(th)$ and $1/\lambda_1 = 0.08(1 - F(th))$. The mean file sizes in the new scheme are given by $1/\nu_1 = 1/F(th)\int_0^{th} xdF(x) = 13kByte$ and $1/\nu_2 = 1/(1-F(th))\int_{th}^{\infty} xdF(x) = 70kByte$. The resulting performances are depicted in Fig. 6. As it may be seen, the second scheme is able to enhance the system performance, by considerably reducing the total mean number of active sessions is reduced even though the combined statistical properties of the requested file sizes remain the same. This means that also the average session delay is reduced.
This fact leads us to propose a novel scatternet management scheme, in which short sessions will get preferential treatment.

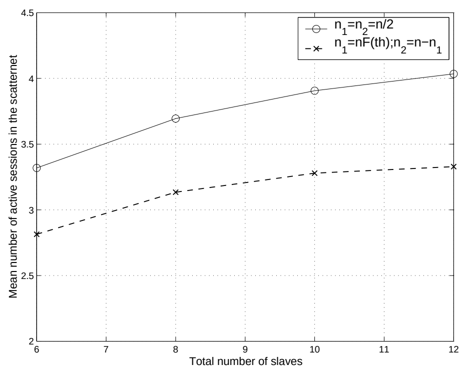


Fig. 6. Mean number of active sessions as a function of the total number of nodes for the two schemes considered.

Topology–based scheduling

The difference in the performance observed in Fig. 6 is not surprising. It suffices to recall that, in single–server queues, giving priority to shorter flows leads to an overall performance enhancement and to note that assigning the high-priority piconet to shorter jobs implicitly we are providing shorter jobs a preferential treatment. This fact lead us to draw a nice link between scatternet management schemes and scheduling policies.

One of the classical results says that the Shortest Remaining Processing Time disciplines minimizes the mean unconditional sojourn time in the system, and as consequence the average number of jobs present in the system [17]. The SRPT discipline, however, requires the knowledge of the remaining service times of all jobs. In practice, these remaining service times are not always known. However, even when the total lenght is not known, the attained service may usually be computed (for example using the TCP sequence numbers [18]). Kleinrock [5] gives a thorough overview of scheduling in such systems where only the attained service times are known together with the service time distribution. Righter and Shantikumar [19] showed the so called Foreground-Background FB scheduling discipline is optimal among this set of scheduling disciplines if the hazard rate of the required service time distribution is decreasing (DHR)[3]. The FB discipline gives priority to the job(s) that have least attained service.

These theoretical results provide us the motivation to consider Bluetooth scatternets in which nodes will be arranged based on the remaining/attained length of the TCP sessions. Indeed, by dinamically adapting the scatternet configuration, we may provide higher priority levels to jobs with the least attained service, by "moving" them closer to the central unit. In such a way, we provide a scheduling mechanism which works not in the classical time domain, but rather in the "virtual space" domain generated by scatternet configurations [20]. We coin this mechanism as *topology–based scatternet management scheme*.

From the theoretical point of view, it has been shown recently that even a simple two level time-sharing queue as the one considered here can reduce significantly the mean number of sessions [18].

V. Conclusion

In this paper we have presented a queueing model for assessing the performance of a Bluetooth–based file sharing system. For the case of an isolated piconet, we have showed how a processor sharing model with state dependente arrival and service rates may be used in order to derive the mean session delay for short–lived TCP flows. Insensitivity of the mean session delay to the think time and file size distribution has been discussed.

The flexibility of the proposed framework allows to consider different SAR policies, the impact of fading channels and also to obtain un upper bound on the performance of any polling algorithm.

In the case of a Bluetooth scatternet, we have showed how, similarly, a discriminatory processor sharing model may be used in order to get qualitatitve insight into the network behavior. In particular, closed form results have been derived under the heavy–usage condition of [11].

The scatternet analysis has been used to draw a link between scatternet management schemes and scheduling theory; in particular, it turns out that efficient scatternet management schemes may be designed on the basis of some classical results on jobs scheduling in a single–server queue.

Among the possible directions for future work, the most promising one seems the design and implementation of these topology–based scatternet management schemes; of particular interest appear the class of multi–level processor sharing schemes [18], due to their ease of implementation by means of a tree configuration.

[3]The hazard rate of a distribution function is given by $\mu(x) = \frac{f(x)}{\overline{F}(x)}$, where $f(x)$ is the density function and $\overline{F}(x)$ is the complementary distribution function. Widely used distributions that have decreasing hazard rate are Pareto, Weibull and Hyperexponential

VI. Acknowledgments

The work of D. Miorandi was partially supported by MIUR within the framework of the PRIMO project FIRB RBNE018RFY, by Fond. A. Gini and CRBV.

References

[1] Bluetooth special interest group. [Online]. Available: http://www.bluetooth.com

[2] A. Das, A. Ghose, A. Razdan, H. Saran, and R. Shorey, "Enhancing performance of asynchronous data traffic over the Bluetooth wireless ad hoc network," in *Proc. of Infocom*, Anchorage, Alaska, 2001.

[3] A. El-Hoiydi, "Interference between Bluetooth networks—upper bound on the packet error rate," *IEEE Comm. Lett.*, vol. 5, no. 6, pp. 245–247, Jun 2001.

[4] D. Miorandi and A. Zanella, "On the optimal topologies of Bluetooth piconets: roles swapping algorithms," in *Proc. of Med-Hoc-Net*, Chia (Italy), 2002.

[5] L. Kleinrock, *Queueing Systems, vol. 2*. John Wiley and Sons, 1976.

[6] D. P. Heyman, T. V. Lakhsman, and A. L. Neidhardt, "A new method for analysing feedback–based protocols with applications to engineering web traffic over the Internet," in *Proc. of Sigmetrics*, Seattle, 1997.

[7] G. D. Veciana, T.-J. Lee, and T. Konstantopoulos, "Stability and performance analysis of networks supporting services with rate control – could the Internet be unstable ?" in *Proc. of Infocom*, New York, 1999.

[8] L. Massoulié and J. Roberts, "Bandwidth sharing and admission control for elastic traffic," *Telecommunication Systems*, vol. 15, pp. 185–201, 2000.

[9] L. Kleinrock, "Time-shared systems: A theoretical treatment," *Journal of the Association for Computing Machinery*, vol. 14, no. 2, pp. 242–261, 1967.

[10] G. Fayolle, I. Mitrani, and R. Iasnongorodski, "Sharing a processor among many job classes," *J. ACM*, vol. 27, no. 3, pp. 519–532, 1980.

[11] D. Mitra and A. Weiss, "A closed network with a Discriminatory Processor-Sharing server," *Performance Evaluation Review*, vol. 17, pp. 200–209, 1989.

[12] T. Bu and D. Towsley, "Fixed point approximations for tcp behavior in an AQM network," in *SIGMETRICS*, San Diego, 2001.

[13] T. Bonald and L. Massoulié, "Impact of fairness on internet performance," in *SIGMETRICS*, San Diego, 2001.

[14] L. Guo and I. Matta, "Scheduling flows with unknown sizes: Approximate analysis," in *ACM Sigmetrics'02 (Extended Abstract)*, Marina del Rey, CA, June, 2002, extended version available as a Boston University Technical Report BU-CS-2002-009.

[15] T. Bonald and A. Proutière, "Insensitivity in processor–sharing networks," *Performance Evaluation*, vol. 49, pp. 193–209, 2002.

[16] D. Miorandi and A. Zanella, "Achievable rate regions for Bluetooth piconets in fading channels," in *Proc. IEEE VTC*, Milano, Italy, 2004 Spring.

[17] L. Schrage, "The queue M/G/1 with feedback to lower priority queues," *Management Science*, no. 13, pp. 466–471, 1967.

[18] K. Avrachenkov, U.Ayesta, P.Brown, and E.Nyberg, "Differentiation between short and long TCP flows: Predictability of the response time," in *Proceedings of INFOCOM*, Hong Kong, 2004.

[19] R. Righter and G. Shanthikumar, "Scheduling multiclass single server queueing systems to stochastically maximize the number of successful departures," *Prob. Eng. Inf Sciences*, vol. 3, pp. 323–334, 1989.

[20] D. Miorandi, A. Trainito, and A. Zanella, "On efficient topologies for Bluetooth scatternets," in *Proc. of PWC2003*, Venezia (Italy), 2003.