Performance of Bluetooth Bridges in Scatternets With Exhaustive Service Scheduling

Vojislav B. Mišić and Jelena Mišić

Abstract— **The performance of two Bluetooth piconets linked through a bridge device is analyzed using the tools of queueing theory. We analyze both the Master/Slave (MS) and Slave/Slave (SS) bridge topologies; in both cases, the piconet master polls its slaves according to the exhaustive scheduling policy. Analytical results are derived for the probability distribution of access delay (i.e., the time that a packet has to wait before being serviced) and end-to-end delay for both intra- and inter-piconet bursty traffic. SS bridge has been found to offer lower access delay and local end-to-end delay than its MS counterpart. MS bridge provides lower non-local end-to-end delay, due to the smaller number of hops (three, instead of four) for such traffic. All analytical results have been confirmed through simulations.**

Index Terms— **Bluetooth Bluetooth scatternet Master/Slave bridge Slave/Slave bridge queues with vacations exhaustive service scheduling**

I. Introduction

Among modern wireless LAN technologies, Bluetooth holds great promise – not only as a simple cable replacement solution, but also as a feasible solution for short range ad hoc networking [3], [13]. Bluetooth devices are organized in piconets, ad hoc structures that contain a master device and up to seven simultaneously active slave devices. The master sequentially polls the slaves by sending them packets with appropriate identification and (when available) data; slaves can talk back to the master only when addressed, and only immediately after being addressed by the master. A downlink transmission from master to slave, together with the subsequent uplink transmission from slave to master, is commonly known as a frame. Since both transmissions within a frame use the same frequency, this mechanism is known as Time Division Duplexing, or TDD for short.

More complex networks may be obtained when two or more Bluetooth piconets share a device, or several of them; such a network is called a scatternet. (A Bluetooth device may be a slave in several piconets, but it can be a master in at most one piconet.) Data packets may be relayed from one network to another through the shared device, which then functions as a bridge. Depending on the exact role of the bridge device in the piconets it links, two cases may be distinguished. The bridge device may act as the master in one piconet and a slave in the other one (Master/Slave or MS bridge, shown in Fig. 1(a)), or it can be a slave in both piconets it belongs to (Slave/Slave or SS bridge, shown in Fig. 2(a)).

These and other characteristics mean that Bluetooth networks operate in a rather different manner from other wireless networks [3]. Consequently, detailed analyses of the performance of such networks are necessary, if Bluetooth technology is to be accepted as a common standard for ad hoc voice and data networks, as is widely anticipated [13]. Yet, performance analyses of Bluetooth networks are still scarce, and most of them deal with simple networks – single piconets only. (This may be attributed, in part at least, to the fact that several important issues, such as scheduling policy and exact mode to be used for bridging, are left unspecified in the current version of the Bluetooth specification [3].) To the best of authors' knowledge, comparative analysis of the two scatternet topologies, which is certainly more interesting as a research topic, has not been attempted so far.

In this paper, we analyze and compare the performance of both Bluetooth bridge topologies with bursty traffic. Main performance indicators are mean values of various delay variables, namely, access delay when the packet is generated in the network and end-to-end delay for both local (i.e., intra-piconet) and non-local (inter-piconet) traffic. We investigate the impact of parameters such as packet burst arrival rate, traffic locality, and the time interval between bridge exchanges on the performance indicators of the piconet operating under exhaustive service scheduling. All analyses are based on the tools of queueing theory, in particular, theory of queues with vacations [2], [15], [16], and verified through simulations.

The paper is structured as follows. In Section II, we will discuss the operation of two different scatternet topologies, including scheduling policies, and explain some rather mild assumptions we have adopted. Section III presents the queueing theoretic analysis of the scatternet with an MS bridge, followed by an analogous analysis of the scatternet with an SS bridge in Sec. IV. Analytical and simulation results are shown in Sec V. Our findings are discussed and compared to those of other authors in Sec. VI, where some challenges for future research are presented as well.

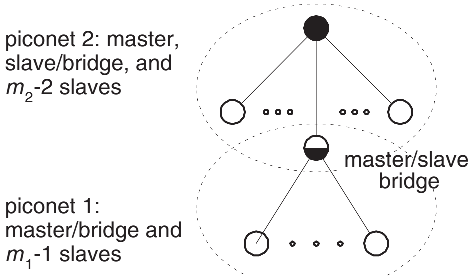
II. On topologies and scatternet operation

We will describe the operation of two scatternet topologies in more detail, first the one with an MS bridge and then the one with an SS bridge. We will also explain the performance indicators we have chosen for our analysis, and list some assumptions that are needed to obtain analytical solutions for the values of those indicators.

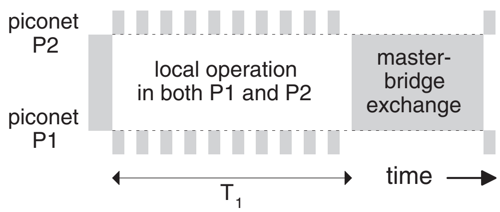
A. MS bridge operation

A generic topology of the scatternet with an MS bridge is shown in Fig. 1(a). During the time interval denoted with T_1, the bridge device assumes the role of piconet $P1$ master and services its slaves: i.e., it routes the packets from and to its slav

J. Mišić and V. B. Mišić are with The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong SAR, China.

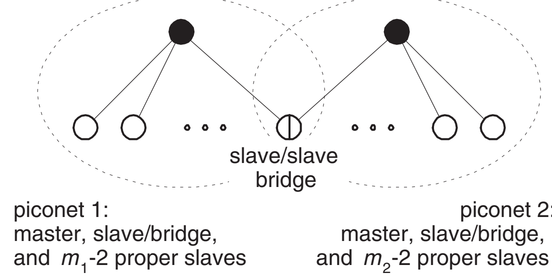


(a) Scatternet topology.

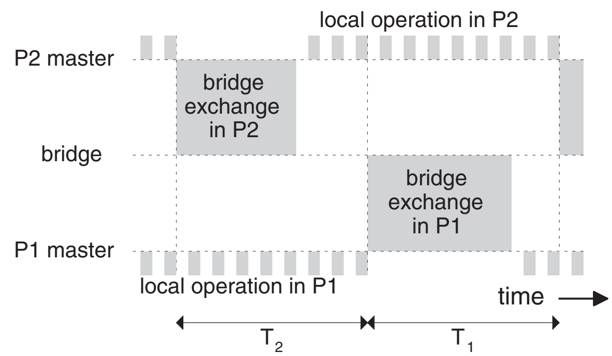


(b) Bridge operation.

Fig. 1. Bluetooth scatternet with an MS bridge.



(a) Scatternet topology.



(b) Bridge operation.

Fig. 2. Bluetooth scatternet with an SS bridge.

queues the packets for destinations in the other piconet. At the same time and in the same fashion the master of $P2$ services the slaves in its own piconet. When the interval T_1 is over, the bridge switches to piconet $P2$ as a slave, and the packets queued in the $P2$ master are sent to the bridge, where they are again queued for later delivery to their appropriate destinations. In the opposite direction, the packets queued in the bridge are sent to the master of $P2$, where they are queued for subsequent delivery. Once all the queued packets have been exchanged, both the master of $P2$ and the bridge return to servicing slaves in their respective piconets. Fig. 2(b) depicts schematically the operation of the scatternet with an MS bridge.

We will assume that the packet exchange between the master of piconet $P2$ and the bridge device is done exhaustively, i.e., it will last until all queued packets in both $P2$ master and the bridge device are exchanged. The rationale for this is simple: if the exchange were to be performed in some other way, with master of $P2$ polling the bridge and its own slaves during the exchange period, there can be no guarantee that all the packets to be routed through the bridge will eventually be exchanged. Moreover, since all communications in a Bluetooth piconet must be initiated by its master, slaves in either piconet cannot communicate at all during the exchange period. Therefore, the bridge exchange should be as short as possible, which in turn necessitates the use of exhaustive scheduling.

The exchange may be terminated when a NULL packet is received from a slave in response to a POLL packet from $P2$ master. In Bluetooth communications, a POLL packet means that the master has no data to send, while a NULL packet means that the polled slave (which, in this case, is the bridge device) has no data to send [3]. Alternatively, since the number of packets in either queue is known and will not change during the exchange (which is performed in the exhaustive mode), the bridge and $P2$ master can exchange this information before the actual packet exchange starts. Once all packets are sent, the bridge exchange may stop, and both the master of $P2$ and the bridge may return to local operation. Either way, the number of frames during a exchange is equal to the number of packets to be sent from the master of $P2$ to the bridge device or the number of packets to be sent from the bridge device to the master of $P2$, whichever is larger, plus one frame for coordination at the beginning or at the end of the exchange.

Note that the duration of the time interval T_1 between successive bridge exchanges must be known to both master of $P2$ and the bridge, so that they can switch to the exchange operation simultaneously. (Otherwise, the participant that switches earlier would have to wait for the one that switches later, and waste some time in doing so.) The Bluetooth MAC level provides mechanisms to support this: for example, the master of $P2$ may put the bridge device in HOLD mode for a specified time interval [3], [13], after which the bridge should join the piconet for an exchange. Of course, other arrangements are possible, including several recently proposed extensions to the Bluetooth specification itself [7], [17].

B. SS bridge operation

The other scatternet topology, shown in Fig. 2(a), uses the bridge device that acts as slave in both piconets and alternates between them: it spends some time (say, T_1) in $P1$, then switches to $P2$ for some time (say, T_2), then switches back to $P1$, and so on. Whenever the bridge joins a piconet, the master starts to exchange packets with the bridge. After finishing this exchange, the master returns to servicing regular slaves in its piconet. Local operation lasts until the bridge joins the piconet again; it includes the time interval in which the bridge is with the other p

0-7695-1874-5/03 $17.00 © 2002 IEEE

Note that this scenario, which is shown schematically in Fig. ??, is totally symmetric with respect to piconets $P1$ and $P2$.

We will assume that the bridge exchanges are performed exhaustively, as in the case of the scatternet with an MS bridge, so as to guarantee that all the packets to be routed through the bridge will eventually be exchanged. The end of the bridge exchanges may be sensed from a POLL-NULL packet exchange between the bridge and the corresponding master, or it may be timed in advance (i.e., the number of packets to be exchanged may be negotiated prior to actual exchange).

Note that the duration of the time intervals T_1 and T_2 must be agreed upon by the bridge and respective piconet master, so that the exchange operation may be started simultaneously, for reasons explained in the previous Subsection. We will assume that the time intervals T_1 and T_2 have fixed values, with $T_1 = T_2$ on account of the symmetry of the scatternet. (The impact of the actual values for T_1 and T_2 on scatternet performance will be analyzed in detail in subsequent discussion.) This operation may be accomplished through the use of SNIFF mode, in which the bridge is instructed to attempt communication with the master at regular time intervals [1], [3], but other arrangements are possible as well.

C. Additional idle times

Regardless of the particular topology of the scatternet, the bridge device will periodically switch from one piconet to another and back, and each such switch requires resynchronization of the bridge device. Namely, all transmissions in Bluetooth are synchronized, in frequency and phase, with the time slot clock of the master device, and all frames start on even-numbered slots [3]. Therefore, the bridge device has to adjust the frequency and phase of its own clock to the frequency and phase of the clock of the master in the piconet it joins, as shown in Fig. 3. The actual duration of the clock sync time required for synchronization—during which no transmission can take place—depends on the phase difference of corresponding clocks; it may take any value from 0 to $2T$ [1], but its mean value may easily be calculated to be equal to the Bluetooth time slot T.

An additional idle time may occur when the bridge device switches from one piconet to another because Bluetooth technology does not guarantee that both participants in a bridge exchange will actually be ready at the same time, even though an agreement about it has been made in advance. As Bluetooth packets may be one, three, or five packets long, one of the participants may join the exchange exactly at the appointed time, while the other one might be in the midst of a frame that has to be completed before the switch can take place, as shown in Fig. 3. The frame sync time may take any value from 0 to $8T$ either way, and its mean value depends on the mean packet length.

Fortunately, both of these delays are small compared to other delays in the scatternet, as will be seen in subsequent analyses. If the switches are not made too often, they will not have any noticeable effect on results.

D. Scheduling policy

Local communication within each piconet may be performed using any of several scheduling policies known [16]; in this paper we assume that the piconet master polls its slaves (excluding the bridge) using exhaustive service scheduling. Under this policy, the master stays with one slave as long as there are packets to send either way, and moves on to next slave only when both downlink and uplink queues for the current one are empty, as detected via a POLL-NULL frame. The next slave to be polled is determined through a simple round-robin, or cyclical, algorithm. The reason for choosing just one scheduling policy, and just this particular scheduling policy, is as follows.

It is known that optimal performance in a polling system may be obtained using scheduling policies such as Stochastically Largest Queue (SLQ) [12]. However, these policies require that "the lengths of all queues are known at all times" to the server (i.e., piconet master), and the next queue (i.e., master-slave channel) to be served is the one which has the highest sum of master and slave queues [4].

Unfortunately, this is not quite feasible in Bluetooth piconets, because the Bluetooth packet structure has no provisions for simple exchange of relevant information: packet headers do not have any fields to carry this information, nor are there any spare bits to be used to that effect [3]. As a consequence, if the information on queue status is to be exchanged between the master of a piconet and its slaves, it will have to be done at the expense of actual packet payload. This may not seem too high a price to pay – from the viewpoint of higher layers of the protocol stack, which by default incur some administrative overhead. However, at the Bluetooth MAC level, it may be more practical to focus on policies that do not require the master to have any knowledge of the length of slaves' queues [4].

Our choice is thus restricted to simple policies, such as limited service and exhaustive service scheduling, in which case exhaustive service is the scheduling policy of choice, since it has been shown to offer best performance for polling systems that are symmetrical (i.e., the same traffic is offered to all queues) and non-idling (i.e., the master does not repeatedly poll the slave with an empty queue) [11], [12]. Indeed previous experiments with Bluetooth piconets have demonstrated the superiority of exhaustive service over other scheduling policies [4], [5], [9], [10].

E. Performance indicators and basic assumptions

The operation of both scatternet topologies may be analyzed using a queueing model, under the assumption that only proper slaves generate and receive any traffic; piconet masters and bridge device just route packets between their corresponding source and destination devices. We will assume that each slave device maintains a packet queue, which will be referred to in subsequent discussions as the uplink queue. The master of each piconet maintains a number of downlink queues, one per each slave in its piconet. (The downlink queue for one slave and the corresponding uplink queue will be referred to as the slave channel.) At the same time, the master(s) that communicate with the bridge device maintain an output queue, in which packets with destinations in the other piconet are queued until they are sent through the bridge. The bridge device maintains a similar queue (in case of an MS bridge), or two queues in case of an SS bridge.

Obviously, packets will have to be queued at each intermediate device or node before being sent further on. The

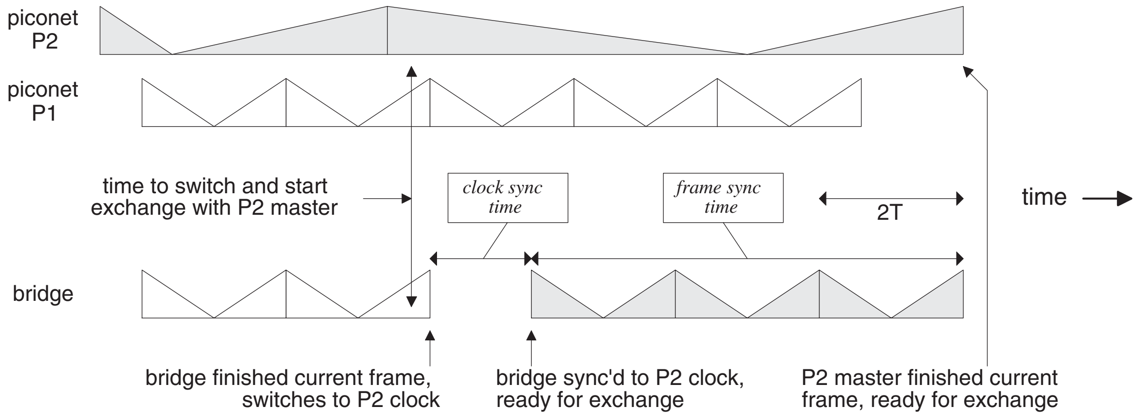


Fig. 3. Additional synchronization intervals when switching from one piconet to another.

the performance of the scatternet will be mainly determined by queueing delays in each of the queues the packets have to wait in, and we will use three composite delay variables as our main performance indicators. First is the access delay W_{ai} $(i = 1, 2)$, the time a data packet has to wait in the uplink queue of the source device before it is serviced. This parameter is important for both local and non-local traffic, as all packets, regardless of their destination, must wait in the corresponding uplink queue at the originating device. The other two are end-to-end delays: measures of the total time a packet has to spend in transit, i.e., from the moment it enters the uplink queue at the source, to the time it arrives at its destination device. We distinguish between end-to-end delay for local traffic, W_{ii} $(i = 1, 2)$, and the one for non-local traffic, W_{ij} $(i, j = 1, 2; i \neq j)$, as packets with non-local destinations have to pass through the bridge.

Packets are generated in bursts or batches, the arrivals of which follow a Poisson distribution with arrival rate of λ. (Modeling using single packets with Poisson arrivals has been shown to be inaccurate for real life data traffic [14].) This approach corresponds to the case where Bluetooth is used as the data link layer for IP traffic: since IP PDUs are longer than Bluetooth ones, a single IP PDU will have to be segmented into several Bluetooth packets – thus giving rise to a packet burst.

The length of the burst follows the probability distribution that may be described with a probability generating function (PGF) $G_b(x) = \sum_{k=0}^{\infty} b_k x^k$, where b_k is the probability that the burst will contain exactly k packets [6]. We will also need the mean value of the burst length $\overline{B} = G'_b(1)$, while its second factorial moment is defined as $\overline{B^{(2)}} = E[B(B-1)] = G''_b(1)$. In this work we will use geometric distribution of packet burst length as a first approximation, since the exact characteristics of Bluetooth data traffic are unknown at this time. Other, possibly more realistic distributions may easily be incorporated into our analysis framework, provided their first and second moments are known.

All packets within a single burst have the same destination, and all destinations are equally probable. All packet-generating slaves within a piconet exhibit the same value for traffic locality P_l, i.e., the probability that both the source and destination of the burst will be in the same piconet. Local traffic will be handled within either piconet, whereas the traffic with non-local destinations will have to be forwarded through the bridge device.

The probabilities of packet length being one, three, and five slots—the only lengths allowed by the Bluetooth specification [3]—are p_1, p_3, and p_5, respectively; of course, $p_1 + p_3 + p_5 = 1$. The corresponding PGF is $G_p(x) = p_1 x + p_3 x^3 + p_5 x^5$. First and second moments of the packet length distribution are equal to $\overline{L} = G'_p(1)$ and $\overline{L^2} = G''_p(1) + G'_p(1)$, respectively.

The queueing theoretic analysis presented in this paper will make use of the equivalent Laplace-Stieltjes transform (LST) of the probability distributions. The LST may be obtained from the corresponding PGF by substituting the variable x with e^{-s} [16]; for example, the LST of the packet burst length PDF is $G^*_b(s) = \sum_{k=0}^{\infty} b_k e^{-ks}$, and the LST of the packet length PDF is $G^*_p(s) = p_1 e^{-s} + p_3 e^{-3s} + p_5 e^{-5s}$.

III. Performance of the MS bridge

As noted above, we will assume that only proper slaves generate and consume data packets. The number of packet-generating slaves in the scatternet with an MS bridge is $m_1 - 1$ and $m_1 - 2$ for piconets $P1$ and $P2$. Then, the burst arrival rates for inter-piconet traffic will be $\lambda_{b12} = (m_1 - 1)\lambda(1 - P_l)$ and $\lambda_{b21} = (m_2 - 2)\lambda(1 - P_l)$, for traffic flows from $P1$ to $P2$, and from $P2$ to $P1$, respectively.

The PGF for the length of the exchange time between the bridge and the master of piconet $P2$ can be written in the form

$$G_r(x) = e^{(\lambda_{b12}+\lambda_{b21})T_1(G_b(G_p(x))-1)} \cdot e^{|\lambda_{b21}-\lambda_{b12}|T_1(G_b(x)-1)} \cdot x^2 \tag{1}$$

and the mean value of the exchange time is $\overline{T_r} = G'_r(1)$.

In order to determine the mean access delay, we need the mean duration of piconet service cycle: the time interval for a piconet master to service all of its slaves once. Since our model is symmetrical with respect to the two piconets, as well as to the slaves within either piconet, we may consider just one master-slave channel in one piconet, say $P1$. As noted above, this channel may be modeled as a pair of queues, for which the burst arrival rates will be $\lambda_{u1} = \lambda$ for the slave (uplink) queue, and $\lambda_{d1} = \lambda P_l + \lambda_{b21}/(m_1 - 1)$ for the corresponding downlink queue at the master.

In local piconet operation, a number of packet pairs (downlink, followed by an uplink) may be exchanged betw master and a single slave during a single visit to that slave. The actual number of packets exchanged is equal to the number of packets in the downlink queue or the number of packets in the corresponding uplink queue, whichever is larger, plus one (as explained in Sec. II-A.) Let us denote the length of the cycle time in piconet $P1$ with X_{c1} and its PGF as $G_{X_{c1}}(x)$. Also, let X_{ms1} denote the single channel service time (the time to empty both uplink and downlink queues for one particular slave channel) with the PGF $G_{X_{ms1}}(x)$. Then, we have the following relationship between the $G_{X_{ms1}}(x)$ and $G_{X_{c1}}(x)$:

$$G_{X_{ms1}}(x) = e^{(\lambda_{u1}+\lambda_{d1})G_{X_{c1}}(x)(G_b(G_p(x))-1)} \cdot e^{|\lambda_{u1}-\lambda_{d1}|G_{X_{c1}}(x)(G_b(x)-1)} \cdot x^2 \quad (2)$$

The piconet cycle time is the sum of the random variables which correspond to the channel service times and the bridge exchange time. The PGF of the bridge exchange component of the piconet cycle time is

$$G_{rc}(x) = \frac{\overline{X_{c1}}}{T_1} G_r(x) + \left(1 - \frac{\overline{X_{c1}}}{T_1}\right) \quad (3)$$

where $\overline{X_{c1}}$ denotes average piconet cycle, i.e.:

$$\overline{X_{c1}} = G'_{X_{c1}}(1) \quad (4)$$

Finally, the PGF for the piconet cycle time will be

$$G_{X_{c1}}(x) = G^{m_1-1}_{X_{ms1}}(x) G_{rc}(x) \quad (5)$$

By solving the system of equations 2 to 5, we can obtain a closed form expression for the PGF of the cycle time.

In queueing theory, the concept of vacation may be applied to a system in which multiple queues are serviced by a single server [16]. When the server services a client according to the chosen service discipline, it then goes on to service other clients, and thus becomes unavailable to that particular client. From the viewpoint of that client, the server takes a *vacation*, which lasts until the next visit to the client. (If the client queue is empty at the time of next server visit, the server will immediately start a new vacation.) The vacation time is, then, the time while the server is busy servicing other client queues. The duration of the vacation period V_1 may be described with the following PGF:

$$G_{V_1}(x) = x^2 G^{m_1-2}_{X_{ms1}}(x) G_{rc}(x) e^{\lambda_{d1} G_{X_{c1}}(x)(G_b(G_p(x))-1)} \quad (6)$$

and its first and second moments are $\overline{V_1} = G'_{V_1}(1)$ and $\overline{V_1^2} = G''_{V_1}(1) + G'_{V_1}(1)$, respectively.

In order to calculate access delay, we need to determine the time it takes to service a single packet – the packet service time [16]. In an isolated piconet, packet service time will be equal to the packet length. In a scatternet, however, this time will be extended due to the fact that the master does not service its local slaves all the time – after every T_1 spent in local service, a bridge exchange takes place. The PGF for the packet service time is:

$$G_{pe}(x) = G_p(x) G_r(x) \frac{\overline{L}}{T_1} + G_p(x)\left(1 - \frac{\overline{L}}{T_1}\right) \quad (7)$$

and its first and second moments are $\overline{L_e} = G'_{pe}(1)$ and $\overline{L_e^2} = G''_{pe}(1) + G'_{pe}(1)$, respectively.

By substituting e^{-s} in place of x in the previously derived PGFs we obtain the corresponding LST for the access delay in piconet $P1$ as:

$$W^*_{a1}(s) = \frac{1 - V^*_1(s)}{s\overline{V_1}} \cdot \frac{1 - G_b(G^*_{pe}(s))}{\overline{B}\,(1 - G^*_{pe}(s))} \cdot \frac{s(1 - (m_1 - 1)(\lambda_{u1} + \lambda_{d1})\overline{B}\,\overline{L_e})}{s - (m_1 - 1)(\lambda_{u1} + \lambda_{d1})(1 - G_b(G^*_{pe}(s)))} \quad (8)$$

The mean access delay can then be obtained as:

$$\overline{W_{a1}} = -(W^*_{a1})'(0) = \frac{(m_1 - 1)(\lambda_{u1} + \lambda_{d1})\overline{B}^2\,\overline{L_e^2} + \overline{B^{(2)}}\,\overline{L_e}}{2\overline{B}(1 - (m_1 - 1)(\lambda_{u1} + \lambda_{d1})\overline{B}\,\overline{L_e})} + \frac{\overline{V_1^2}}{2\overline{V_1}} \quad (9)$$

Note that the burstiness of the traffic is essentially preserved under exhaustive service scheduling, because the entire burst from the slave is transferred without interruption to the corresponding downlink queue. (Occasionally, a burst might be interrupted due to the bridge exchange, but even in this case the transfer could be resumed after the exchange, and such burst will actually reappear in the corresponding downlink queue.) Therefore, the queueing delay at the master is equal to the access delay at the slave, $W^*_{m1}(s) = W^*_{a1}(s)$, which in turn may be obtained from eqn. 8.

The calculation of end-to-end delay is slightly more involved, as two distinct cases may be observed. If both the source and destination nodes of a packet are in the same piconet, the total delay is the sum of two components, the access delay at the slave and the queueing delay at the master. The LST of this delay is $W^*_{11}(s) = W^*_{a1}(s) W^*_{m1}(s)$, and the mean value of the end-to-end delay for local traffic is equal to

$$\overline{W_{11}} = \overline{W_{a1}} + \overline{W_{m1}} \quad (10)$$

An analogous expression holds for piconet $P2$.

On the other hand, the packet may go from the source in one piconet to the destination in the other one. Such packets will have to pass through the bridge, which incurs an additional queueing delay. Since the packets in the bridge are served exhaustively, the corresponding LST for the bridge queueing delay (for packets going from $P1$ to $P2$) is

$$W^*_{b12}(s) = \frac{1 - T^*_1(s)}{s\overline{T_1}} \cdot \frac{s(1 - \lambda_{b12}\overline{B}\,\overline{L})}{s - \lambda_{b12} + \lambda_{b12} G_b(G^*_p(s))} \cdot \frac{1 - G_b(G^*_p(s))}{\overline{B}(1 - G^*_p(s))} \quad (11)$$

where $T^*_1(s) = \overline{T_1}/s$. The LST for the distribution of end-to-end delay of non-local traffic going from $P1$ to $P2$ is, then, $W^*_{12}(s) = W^*_{a1}(s) W^*_{b12}(s) W^*_{m2}(s)$.

The mean bridge queueing delay for packets going from $P1$ to $P2$ is:

$$\overline{W_{b12}} = \frac{\lambda_{b12}\overline{B}\,\overline{L^2}}{2(1 - \lambda_{b12}\overline{B}\,\overline{L})} + \frac{\overline{B^{(2)}}\,\overline{L}}{2\overline{B}(1 - \lambda_{b12}\overline{B}\,\overline{L})} + \frac{\overline{T_1^2}}{2\overline{T_1}} \quad (12)$$

where $\overline{L^2} = G''_p(1) + G'_p(1)$ denotes the second moment of packet length distribution. Also, since T_1 is constant, the second component of the previous expression is only $T_1/2$.

By the same token, packets going from $P2$ to $P1$ will experience the queueing delay which may be obtained as:

$$\overline{W_{b21}} = \frac{\lambda_{b21}\overline{B}\,\overline{L^2}}{2(1-\lambda_{b21}\overline{B}\,\overline{L})} + \frac{\overline{B^{(2)}}\,\overline{L}}{2\overline{B}(1-\lambda_{b21}\overline{B}\,\overline{L})} + \frac{\overline{T_1^2}}{2\overline{T_1}} \quad (13)$$

Overall, the mean end-to-end delay time will be:

$$\overline{W_{12}} = \overline{W_{a1}} + \overline{W_{b12}} + \overline{W_{m2}} \quad (14)$$

$$\overline{W_{21}} = \overline{W_{a2}} + \overline{W_{b21}} + \overline{W_{m1}} \quad (15)$$

for packets going from $P1$ to $P2$ and from $P2$ to $P1$, respectively.

IV. Performance of the scatternet with an SS bridge

Since this particular scatternet topology is symmetrical with respect to the piconets it contains, it will suffice to consider packet exchange between the bridge and one of piconet masters only. In order to determine the duration of the exchange, we will assume that the scatternet operates in a steady state. This means that all the packets queued in between the end of one exchange and the start of the subsequent one, will be serviced. In other words, all packets waiting in the piconet master will be transferred to the bridge, while all those from the bridge will be sent to the master and distributed in the corresponding downlink queues. The number of frames exchanged between a master and the bridge in a single session is equal to the number of packets sent to the bridge or the number of packets received from it, whichever is larger, plus an extra frame for coordination (Sec. II-A). Then, given that one cycle of bridge exchanges is equal to $T_1 + T_2$, we could write the PGF for the length of the exchange time as

$$G_{r1}(x) = e^{(\lambda_{b12}+\lambda_{b21})(T_1+T_2)(G_b(G_p(x))-1)} \cdot e^{|\lambda_{b21}-\lambda_{b12}|(T_1+T_2)(G_b(x)-1)} \cdot x^2 \quad (16)$$

where $\lambda_{b12} = (m_1-2)\lambda(1-P_l)$ and $\lambda_{b21} = (m_2-2)\lambda(1-P_l)$ stand for burst arrival rates for inter-piconet traffic from $P1$ to $P2$, and from $P2$ to $P1$, respectively. The mean value of the exchange time may, then, be obtained as $\overline{T_r} = G'_{r1}(1)$. (Note that the last equation is essentially the same as eqn. 1, except that the time interval between bridge exchanges T_1 has been replaced with $T_1 = T_2$.)

The burst arrival rate in the slave (uplink) queue will be $\lambda_{u1} = \lambda$, and the burst arrival rate for the master (downlink) queue of this channel will be $\lambda_{d1} = \lambda P_l + \lambda_{b21}/(m_1-2)$.

Let us again denote the length of the cycle time in piconet $P1$ with X_{c1} and the time to serve one channel as X_{ms1}. Their PGFs will be denoted as $G_{X_{c1}}(x)$ and $G_{X_{ms1}}(x)$ respectively and the relationship between them is the same as (2):

$$G_{X_{ms1}}(x) = e^{(\lambda_{u1}+\lambda_{d1})G_{X_{c1}}(x)(G_b(G_p(x))-1)} \cdot e^{|\lambda_{u1}-\lambda_{d1}|G_{X_{c1}}(x)(G_b(x)-1)} \cdot x^2 \quad (17)$$

Piconet cycle time is again the sum of channel service times and the bridge exchange time. The PGF which models the bridge exchange time in a single piconet cycle is equal to:

$$G_{rc1}(x) = \frac{\overline{X_{c1}}}{T_1+T_2-\overline{T_r}}G_{r1}(x) + (1 - \frac{\overline{X_{c1}}}{T_1+T_2-\overline{T_r}}) \quad (18)$$

where $\overline{X_{c1}}$ denotes average piconet cycle time, given with the expression:

$$\overline{X_{c1}} = G'_{X_{c1}}(1). \quad (19)$$

Then the PGF for the piconet cycle time is equal to:

$$G_{X_{c1}}(x) = G_{X_{ms1}}^{m_1-2}(x)G_{rc1}(x). \quad (20)$$

By solving the system of equations 17 to 20, we can obtain closed form expression for the PGF of the cycle time.

Then, the PGF for the server vacation time can be determined as follows:

$$G_{V_1}(x) = x^2 G_{X_{ms1}}^{m_1-3}(x)G_{rc1}(x) \, e^{\lambda_{d1}G_{X_{c1}}(x)(G_b(G_p(x))-1)} \quad (21)$$

and its first and second moments are $\overline{V_1} = G'_{V_1}(1)$ and $\overline{V_1^2} = G''_{V_1}(1) + G'_{V_1}(1)$, respectively.

The packet service time may be determined as follows: in an isolated piconet, it will depend only on the packet length; in a scatternet, it will be extended due to the fact that local piconet operation is periodically (i.e., once every T_1+T_2) interrupted due to the bridge exchange. The corresponding PGF then becomes:

$$G_{pe}(x) = G_p(x)G_{r1}(x)\frac{\overline{L}}{T_1+T_2-\overline{T_r}} + G_p(x)(1-\frac{\overline{L}}{T_1+T_2-\overline{T_r}}) \quad (22)$$

and its first and second moments are $\overline{L_e} = G'_{pe}(1)$ and $\overline{L_e^2} = G''_{pe}(1) + G'_{pe}(1)$, respectively.

By substituting e^{-s} in place of x in the previously derived PGFs, the corresponding LST for the access delay in piconet $P1$ can be obtained as:

$$W^*_{a1}(s) = \frac{1-V_1^*(s)}{s\overline{V_1}} \cdot \frac{1-G_b(G^*_{pe}(s))}{\overline{B}\,(1-G^*_{pe}(s))} \cdot \frac{s(1-(m_1-2)(\lambda_{u1}+\lambda_{d1})\overline{B}\,\overline{L_e})}{s-(m_1-2)(\lambda_{u1}+\lambda_{d1})(1-G_b(G^*_{pe}(s)))} \quad (23)$$

Mean access delay can be obtained from expr. 23 as $\overline{W_{a1}} = -(W^*_{a1})'(0)$, which evaluates to:

$$\overline{W_{a1}} = \frac{(m_1-2)(\lambda_{u1}+\lambda_{d1})\overline{B}\,\overline{L_e^2}}{2(1-(m_1-2)(\lambda_{u1}+\lambda_{d1})\,\overline{B}\,\overline{L_e})} + \frac{\overline{B^{(2)}}\,\overline{L_e}}{2\overline{B}(1-(m_1-2)(\lambda_{u1}+\lambda_{d1})\overline{B}\,\overline{L_e})} + \frac{\overline{V_1^2}}{2\overline{V_1}} \quad (24)$$

The LST for the delay at the master has the same form as for the access delay, $W^*_{a1}(s) = W^*_{m1}(s)$. The LSTs for both and master delay in the piconet $P2$ can be calculated from the expressions similar to 23.

As in the case of the scatternet with an MS bridge, local and non-local end-to-end delays have to be considered separately. The local traffic will experience two delays only, the access delay and the queueing delay in the downlink queue. The mean end-to-end delay for local traffic will be $\overline{W_{11}} = \overline{W_{a1}} + \overline{W_{m1}}$ and $\overline{W_{22}} = \overline{W_{a2}} + \overline{W_{m2}}$, for piconets $P1$ and $P2$, respectively.

On the other hand, non-local traffic will have as much as four hops to make and four delays to experience: access delay, delay in the master bridging queue, delay in the bridge queue towards the master of the other piconet, and finally the delay in that master's downlink queue. The first and last component have already been calculated. The LST for the waiting time at the bridge queue of $P1$ master is

$$W^*_{m1b}(s) = \frac{1 - V^*_T(s)}{s\overline{V_T}} \cdot \frac{s(1 - \lambda_{b12}\overline{B}\,\overline{L})}{s - \lambda_{b12} + \lambda_{b12}G_b(G^*_p(s))} \cdot \frac{1 - G_b(G^*_p(s))}{\overline{B}(1 - G^*_p(s))} \quad (25)$$

where $V^*_T(s) = T^*_2(s)T^*_1(s)/G^*_{r1}(s)$ and $\overline{V_T} = T_2 + T_1 - \overline{T_r}$. By the same token, the LST for the delay in the bridge-to-master-of-$P2$ queue is

$$W^*_{a1b}(s) = \frac{1 - V^*_T(s)}{s\overline{V_T}} \cdot \frac{s(1 - \lambda_{b12}\overline{B}\,\overline{L})}{s - \lambda_{b12} + \lambda_{b12}G_b(G^*_p(s))} \cdot \frac{1 - G_b(G^*_p(s))}{\overline{B}(1 - G^*_p(s))} \quad (26)$$

Therefore, the mean values of corresponding component delays are

$$\overline{W_{m1b}} = \frac{\lambda_{b12}\overline{B}^2\,\overline{L^2} + \overline{B^{(2)}}\,\overline{L}}{2\overline{B}(1 - \lambda_{b12}\overline{B}\,\overline{L})} + \frac{\overline{V^2_T}}{2\overline{V_T}} \quad (27)$$

$$\overline{W_{a1b}} = \frac{\lambda_{b12}\overline{B}^2\,\overline{L^2} + \overline{B^{(2)}}\,\overline{L}}{2\overline{B}(1 - \lambda_{b12}\overline{B}\,\overline{L})} + \frac{\overline{V^2_T}}{2\overline{V_T}} \quad (28)$$

where $\overline{L^2} = G''_p(1) + G'_p(1)$ denotes the second moment of packet length distribution.

With these components, the overall LST for the distribution of end-to-end delay of non-local traffic going from $P1$ to $P2$ may be written as $W^*_{12}(s) = W^*_{a1}(s)W^*_{m1b}(s)W^*_{a1b}(s)W^*_{m2}(s)$, and its mean value is:

$$\overline{W_{12}} = \overline{W_{a1}} + \overline{W_{m1b}} + \overline{W_{a1b}} + \overline{W_{m2}} \quad (29)$$

Analogous expressions for the packets flowing in the opposite direction may be obtained with ease, due to the inherent symmetry of the scatternet.

V. Analysis and discussion of results

As before, we have calculated and plotted analytical solutions for different delay variables, as well as the values obtained through simulation. The following parameter values were used, unless otherwise specified: each piconet had six active slaves, i.e., for MS bridge $m_1 = 7$, $m_2 = 8$ and for SS bridge $m_1 = m_2 = 8$. Burst arrival rate per slave, if fixed, was $\lambda = 0.0015$; mean burst size was $\overline{B} = 10$ (giving the packet arrival rate of $\lambda\overline{B} = 0.015$ per slave); mean packet length is $\overline{L} = 3$, with $p_1 = p_2 = p_3 = 1/3$; traffic locality was $P_l = 0.9$; and times between bridge exchanges, if fixed, were $T_1 = T_2 = 100T$, where $T = 0.625\mu s$ is the time slot of the Bluetooth clock.

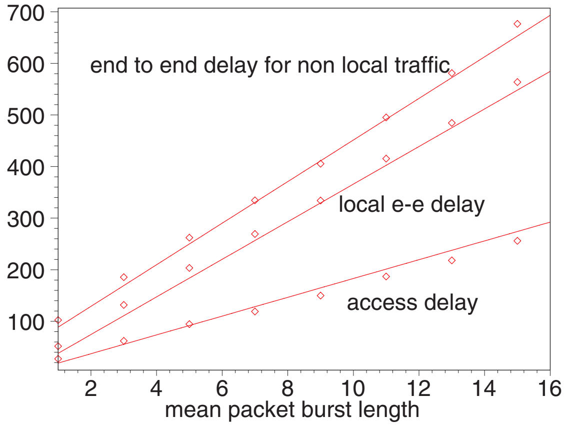
In order to assess the impact of traffic burstiness, we have calculated and plotted the delay times as functions of mean burst size $\overline{B}$, as shown in Fig. 4. In both diagrams, values obtained for MS are shown on the left, and those obtained for the SS are shown on the right. (Analytical solutions are shown as continuous lines, while diamonds stand for results obtained through simulation.) We observe that all delays are nearly linear functions of mean burst size. Also, local end-to-end delay (as well as access delay) is lower for the SS bridge, while non-local end-to end delay is smaller for the MS bridge. Note that the agreement between analytical and simulation results is quite good for both topologies.

The dependency of mean access delay on burst arrival rate λ and time between bridge exchanges T_1 (we assume that $T_2 = T_1$ for the SS bridge) is shown in Fig. 5. Again, the results for the MS bridge are shown on the left, while those for the SS bridge are on the right; analytical solutions are shown in the upper row, and simulation results in the lower row. We observe that for small values of T_1, SS bridge has lower access delay than the MS bridge, while for large values of T_1 SS and MS bridges have similar delays. Mathematically, this can be explained by comparing expressions 18 and 3; the difference follows from the fact that master devices in the SS bridge topology have more time slots to service its slaves than their MS bridge counterparts: $T_1 + T_2 - \overline{T_r}$ compared to only T_1.

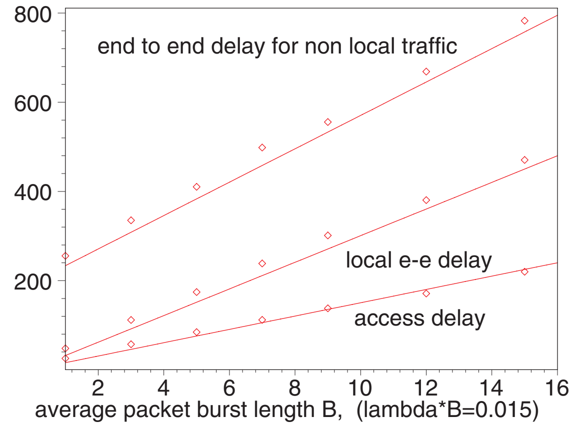
Again, the agreement between analytical solutions and simulation results is very good.

Next two sets of diagrams (Fig. 6) show the dependency of mean non-local end-to-end delay on burst arrival rate λ and time between bridge exchanges T_1 (we assume that $T_2 = T_1$). We do not show the local end-to-end delay, since it would provide little extra information except for the fact that it is lower than the its non-local counterpart by about T_1 and $T_1 + T_2$ respectively. Note that the end-to-end delay for non-local traffic for SS bridge increases with T_1, the time between bridge exchanges, esp. at lower burst arrival rates, unlike its MS bridge counterpart where the rate of increase is barely noticeable. This is due to the fact that bridge exchanges (or, rather, the time instants in which they start) are spaced exactly T_1 apart, hence the non-local end-to-end delay virtually contains $2T_1$ as an additive component. As before, the agreement between analytical solutions and simulation results is very good. This provides further proof of the validity of our queueing theoretic approach.

Fig. 7 shows the dependency of mean non-local end-to-end delay on traffic locality P_l and time between bridge exchanges T_1. From the previous analyses and figures it is obvious that absolute value of non-local end-to-end delay is smaller for the MS bridge. However, less obvious fact is that relative growth of this delay when T_1 decreases is larger for MS bridge than for the SS bridge: about 25% compared to 17%, roughly.

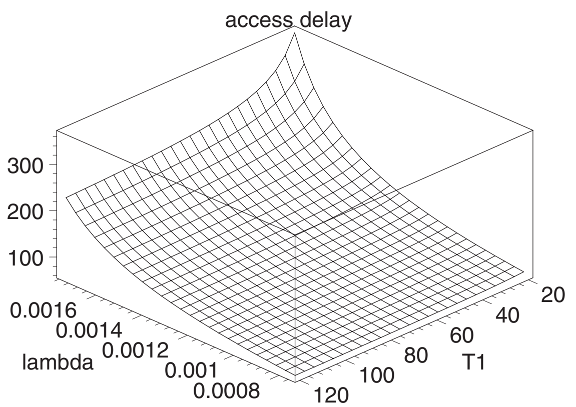


(a) MS bridge.

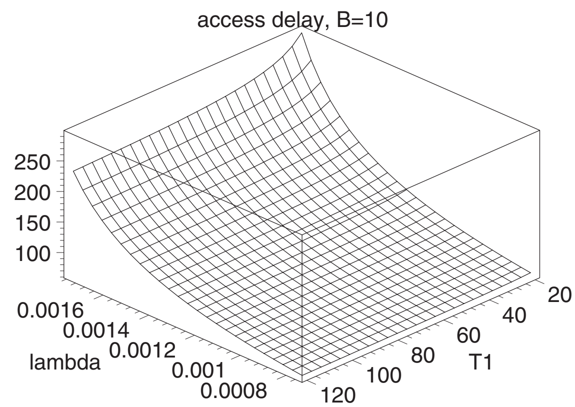


(b) SS bridge

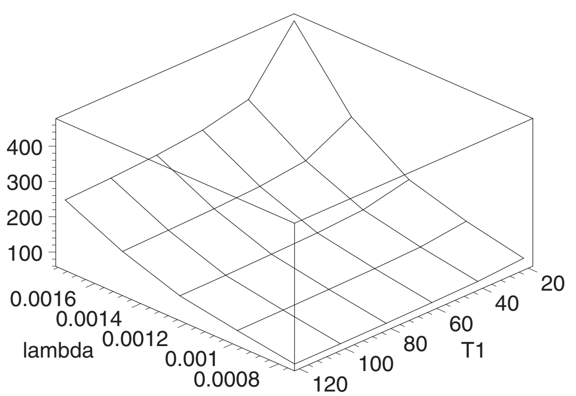
Fig. 4. Delays as functions of mean packet burst length, under constant aggregate packet arrival rate for MS and SS bridge respectively.



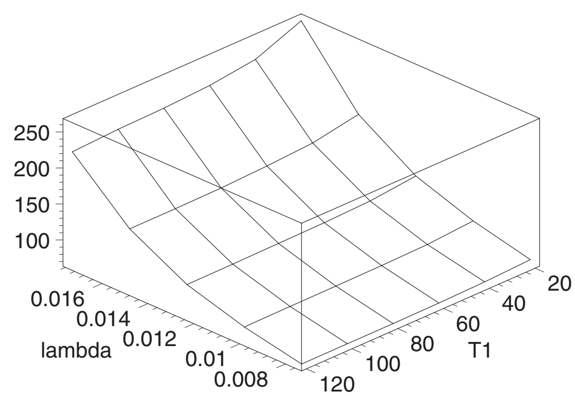
(a) MS bridge, analytical solutions.



(b) SS bridge, analytical solutions.



(c) MS bridge, simulation results.



(d) SS bridge, simulation results.

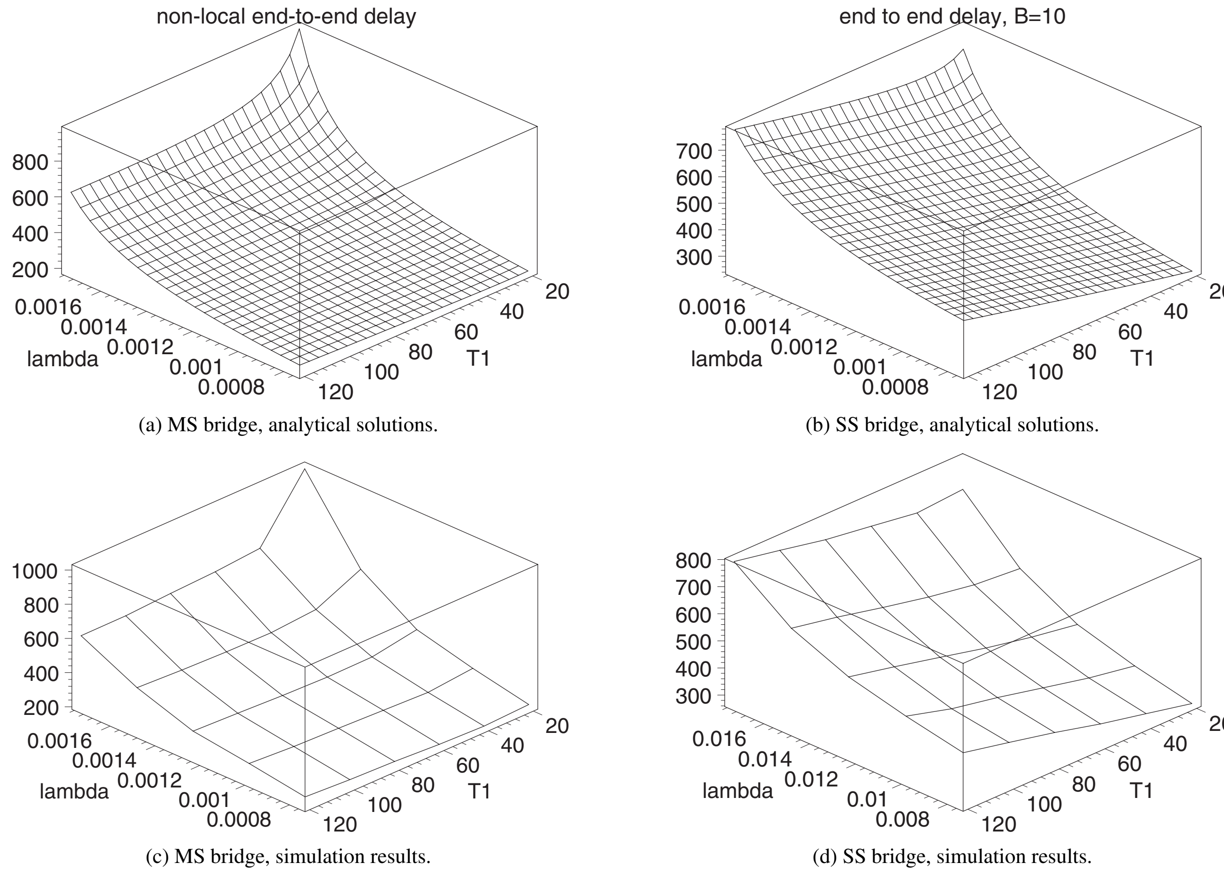
Fig. 5. Mean access delay as a function of burst arrival rate λ and time between bridge exchanges T_1 for MS and SS bridge respectively.

We have also calculated the squared coefficient of variation of end-to-end delay for non-local traffic, defined as $\mathrm{Var}(W_{12})/E[W_{12}]^2$; the dependency of this variable on burst arrival rate and time interval between bridge exchanges is shown in Fig. 8. The squared coefficient of variation has lower values in the configuration with an SS bridge, meaning that the non-local traffic will exhibit lower packet arrival jitter. Again, this can be attributed to the longer time that piconet masters can devote to servicing their slaves in this topology.

VI. Summary and Directions For Further Research

Our analysis of the performance of Bluetooth scatternet topologies under different scheduling policies may be summarized as follows.

Performance of Bluetooth networks, expressed in terms of mean access delay and mean end-to-end delay, shows monotonic behavior with respect to the main traffic parameters such as packet or burst arrival rate, traffic locality, and mean burst length. The dependence is by no means linear, though, and sharp increases of delay times may be experienced at high burst



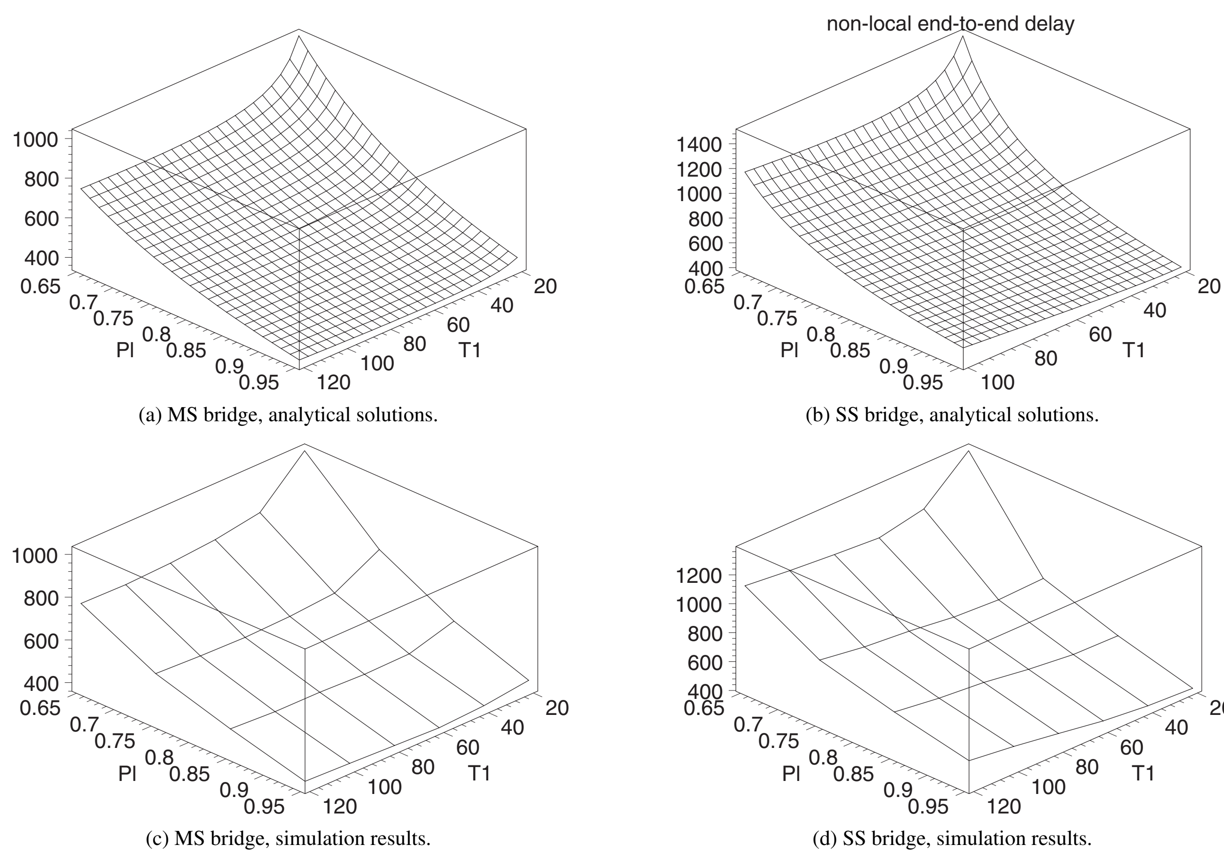
(a) MS bridge, analytical solutions.

(b) SS bridge, analytical solutions.

(c) MS bridge, simulation results.

(d) SS bridge, simulation results.

Fig. 6. SS bridge: mean end-to-end delay for non-local traffic as a function of burst arrival rate λ and time between bridge switches T_1.



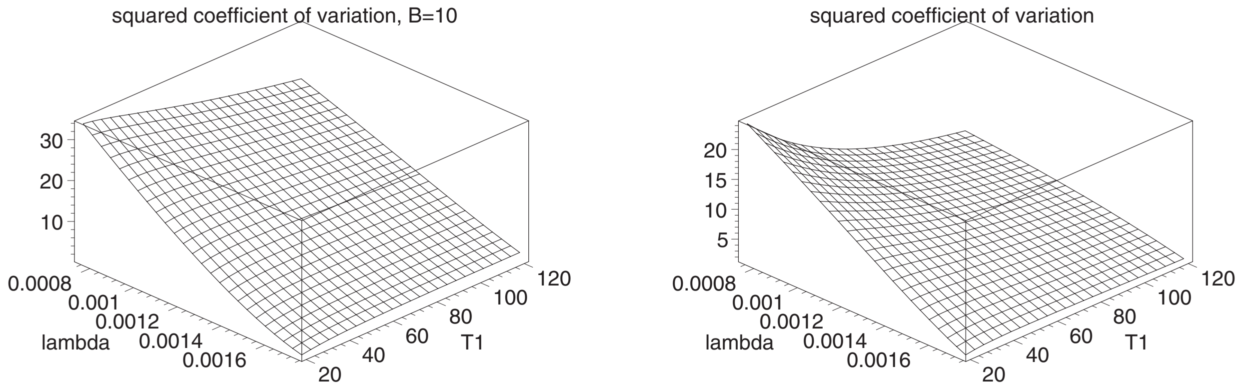
(a) MS bridge, analytical solutions.

(b) SS bridge, analytical solutions.

(c) MS bridge, simulation results.

(d) SS bridge, simulation results.

Fig. 7. Mean end-to-end delay for non-local traffic as a function of traffic locality P_l and time between bridge switches T_1 for MS and SS bridge respe



(a) MS bridge. (b) SS bridge.

Fig. 8. Squared coefficient of variation of end-to-end delay for non-local traffic as a function of burst arrival rate and exchange interval T_1.

rates, under low probability of local traffic, with short intervals between bridge exchanges, or any combination thereof.

Such behaviour should come as no surprise, given that the delays in the network are actually queueing delays, as indicated in a number of earlier papers (for example, in [4], [5], [8], [9]). Regardless of the topology and/or scheduling policy, delays are not too sensitive on T_1, the time interval between bridge exchanges. There is a caveat, though: this time interval should not be too small, and values in the range over 40 to $60T$ should give satisfactory results. In general, the access delay and local end-to-end delay slowly decrease when T_1 increases, while the non-local end-to-end delay increases at a slightly higher rate than the other two. (This rate is much higher in the topology with an SS bridge, for reasons explained in Sec. V, but it tends to diminish at higher burst arrival rates.) Still, the gradients are quite low and the exact value of T_1 does not seem to be critical for scatternet performance, although this issue probably deserves more attention in future research.

With regard to scatternet topology, we may conclude that: mean access delay is lower in the scatternet topology with an SS bridge, due to the fact that both piconet masters can spend more time servicing the slaves in their respective piconets. The same holds for mean end-to-end delay for local (i.e., intra-piconet) traffic.

On the other hand, mean end-to-end delay for non-local traffic is lower in the scatternet topology with an MS bridge, due to the lower number of hops that the packets have to pass: three, compared to four in the case of the topology with an SS bridge.

The scatternet with an MS bridge is more sensitive to the combination of (1) high packet burst arrival rate or low traffic locality, and (2) too short time interval between bridge exchanges. (Here 'more sensitive' should be taken to mean that the relative increase in delays is higher, or more significant, in the region where these conditions apply.) Again, these are extreme conditions that are not likely to be encountered in practice.

The differences between the two topologies are not high, however, and other considerations might be given priority when deciding on the preferred topology of a scatternet.

References

[1] S. Baatz, M. Frank, C. Kühl, P. Martini, and S. Christoph. Adaptive scatternet support for Bluetooth using sniff mode. In *Proceedings of the 26th Annual Conference on Local Computer Networks LCN 2001*, Tampa, FL, Nov. 2001.

[2] D. P. Bertsekas and R. Gallager. *Data Networks*. Prentice-Hall, Englewood Cliffs, NJ, 2nd edition, 1991.

[3] Bluetooth SIG. *Specification of the Bluetooth System*. Version 1.1, Feb. 2001.

[4] A. Capone, R. Kapoor, and M. Gerla. Efficient polling schemes for Bluetooth picocells. In *Proceedings of IEEE International Conference on Communications ICC 2001*, volume 7, pages 1990–1994, Helsinki, Finland, June 2001.

[5] 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 *Proceedings Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies IEEE INFOCOM 2001.*, volume 1, pages 591–600, Anchorage, AK, Apr. 2001.

[6] G. R. Grimmett and D. R. Stirzaker. *Probability and Random Processes*. Oxford University Press, Oxford, UK, 2nd edition, 1992.

[7] N. Johansson, F. Alriksson, and U. Jönsson. JUMP mode – a dynamic window-based scheduling framework for Bluetooth scatternets. In *Proceedings 2001 ACM International Symposium on Mobile ad hoc networking & computing*, pages 204–211, Long Beach, CA, Oct. 2001.

[8] N. Johansson, U. Körner, and P. Johansson. Performance evaluation of scheduling algorithms for Bluetooth. In D. H. K. Tsang and P. J. Kuhn, editors, *Proceedings of BC'99 IFIP TC 6 Fifth International Conference on Broadband Communications*, pages 139–150, Hong Kong, Nov. 1999.

[9] M. Kalia, D. Bansal, and R. Shorey. Data scheduling and SAR for Bluetooth MAC. In *Proceedings VTC2000-Spring IEEE 51st Vehicular Technology Conference*, volume 2, pages 716–720, Tokyo, Japan, May 2000.

[10] A. Kumar, L. Ramachandran, and R. Shorey. Performance of network formation and scheduling algorithms in the Bluetooth wireless ad-hoc network. *Journal of High Speed Networks*, 10:59–76, Oct. 2001.

[11] H. Levy, M. Sidi, and O. J. Boxma. Dominance relations in polling systems. *Queueing Systems Theory and Applications*, 6(2):155–171, 1990.

[12] Z. Liu, P. Nain, and D. Towsley. On optimal polling policies. *Queueing Systems Theory and Applications*, 11(1–2):59–83, 1992.

[13] B. A. Miller and C. Bisdikian. *Bluetooth Revealed: The Insider's Guide to an Open Specification for Global Wireless Communications*. Prentice-Hall, Upper Saddle River, NJ, 2000.

[14] V. Paxson and S. Floyd. Wide area traffic: the failure of Poisson modeling. *ACM/IEEE Transactions on Networking*, 3(3):226–244, June 1995.

[15] H. Takagi. Queuing analysis of polling models. *ACM Computing Surveys*, 20(1):5–28, Mar. 1988.

[16] H. Takagi. *Queueing Analysis*, volume 1: Vacation and Priority Systems. North-Holland, Amsterdam, The Netherlands, 1991.

[17] W. Zhang, H. Zhu, and G. Hao. Improving Bluetooth network performance through a time-slot leasing approach. In *Proceedings of IEEE Wireless Communications and Networks Conference WCNC 2002*, volume 2, pages 592–596, Orlando, FL, Mar. 2002.