# Throughput, Energy and Path Length Tradeoffs in Bluetooth Scatternets

UCLA CSD Technical Report TR040031
09-28-2004

Csaba Kiss Kalló
Mauro Brunato
Dipartimento di Informatica e Telecomunicazioni
Università di Trento, Trento, Italy
Email: {kkcsaba,brunato}@dit.unitn.it

Sewook Jung
Ling-Jyh Chen
Mario Gerla
Department of Computer Science
University of California, Los Angeles
Email: {sewookj,cclljj,gerla}@cs.ucla.edu

***Abstract*— In this work we determine an analytical relationship between the average path length of traffic connections of a Bluetooth scatternet and the overall throughput and power consumption of the network.**

**Results obtained implementing this analytical relationship to different scatternet topologies are presented and discussed. By reducing the hop count in a scatternet we can achieve better performance in terms of throughput and power consumption. Therefore, the issue of minimizing the hop count in the presence of mobility, changing traffic flows and varying interference receives an important role. In our analysis we also show the impact of the link quality on the overall throughput.**

**The obtained results motivate the importance of heuristics aimed at reducing the average communication path length in a scatternet [1].**

## I. Introduction

Bluetooth is a short-range wireless network technology that supports ad hoc networking. In Bluetooth a maximum of 8 active nodes are organized in a star-shaped cluster, called *piconet*. The cluster head is called *master* while the other nodes are its *slaves*. If multiple piconets are interconnected through so-called *bridge* nodes they form a scatternet. Bridges are nodes participating in more than one piconet on a time sharing basis. We call *slave&bridges* those nodes that have only slave role in all the piconets they participate in, while nodes having both, slave and master roles in different piconets are *master&bridges*.

The latest Bluetooth Specification 1.2 [2] introduces the concept of scatternet formation but it does not define it in detail. Although numerous research papers were published addressing scatternet formation and optimization issues [3], [4], [5], [6], [7], [8], [9], [10], there are still aspects that can be improved in order to achieve high performance. This issue is discussed in an earlier paper [1] that addresses the problem of dynamically adapting the scatternet topology to the current traffic flows. In that optimization work we aim to correct the suboptimal traffic paths that are formed when nodes change their communication peers or migrate across the scatternet. Our algorithms update the topology of the scatternet making it possible for the routing algorithms to identify shorter paths between the communicating peers. This, in turn, results in higher aggregate throughput and reduced power consumption. In that work we devised an algorithm suite for reducing the hop count between all communication peers in the scatternet. Now we demonstrate analytically that hop reduction indeed has positive impact on the throughput and power consumption of the scatternet. Our goal is to determine an analytic relation between the number of hops connecting communication peers in Bluetooth scatternets and the overall throughput and power consumption of the network.

Bluetooth scatternets with dynamic traffic connections can be found in several application scenarios. Beside the well-known conference-room scenario, we can foresee the use of scatternets in *interfering industrial environments* with machinery that autonomously or semi-autonomously accomplishes its tasks. Components of such an automated environment are static and *mobile* robots, sensors of various type and human supervisors. All these components need to be networked for exchanging the data necessary for accomplishing their tasks. Raw input data required for the tasks, sensor data, progress reports and a whole series of control data are all examples for information that need to be exchanged among the components. Also, each node may have multiple communication peers sustaining *random data traffic sessions* with them, sequentially and/or in parallel.

A data network supporting such a scenario needs to be adaptive for achieving high performance in terms of throughput, power consumption and packet delivery delay. Factors that influence networking predictably in such a scenario, and that in principle can reduce the aggregate system performance, are mobility, interference and random communication sessions. Bluetooth scatternets are a good candidate for supporting such an ad hoc networking scenario since the technology is robust to interference, given its communication mode based on frequency hopping.

## II. Overview

To determine an analytic relation between the hop count and throughput in a scatternet we need to take into account two fundamental issues: the *Bluetooth packet types* and *link scheduling*.

Bluetooth data communication happens through Asynchronous Connectionless Links (ACL) using time slots of $625\mu s$. Data packets may use 1, 3 or 5 slots and they may be Forward Error Coded (FEC). FEC packets are DM1, DM3 and DM5 while the non-error coded ones are DH1, DH3 and DH5 (with the digits indicating the number of slots used). The useful maximum payload of these packets is 136, 968 and 1816 bits for DM and 216, 1464 and 2712 bits for DH packets, respectively. Packets with bigger payloads can achieve higher throughput in error-free environments (i.e. with high link quality). However, if a bit gets corrupted, the whole packet will have to be retransmitted. Therefore, when retransmissions happen often, smaller packets are more efficient. DM packets have smaller payloads than their DH counterparts, but their content is error checked, in contrast with DH packets.

*Link scheduling* refers to the allocation of time slots in a bridge node to its piconets. A bridge node can be present in one piconet at a time. Therefore, it has to switch continuously between its piconets for being reachable by each of its masters and to relay traffic efficiently. Since we aim at analyzing the scatternet throughput, link scheduling is indispensable for us. Therefore, in the next section we present an analytical model of the link scheduling algorithm that we used in our work.

Link scheduling is a complex task and since it is not the target of our work, we try to keep it as simple as possible. However, we still obtain a quite complicated analytic interpretation of our link scheduling algorithm, described in Section III.

Taking advantage of the packet types and link scheduling model, in Section IV we present how the overall scatternet throughput and power consumption can be calculated, while in Section V we evaluate our model. In our evaluation we aim at proving that hop reduction can improve scatternet performance, showing also several performance aspects regarding packet types and link quality.

## III. Sharing the Communication Capacity

In our approach each piconet is assigned an overall traffic capacity of 1. (Hence, the traffic rate of a pure master is equal to 1, too.) This capacity is divided among the slaves of that master according to the expressions (1)–(10), making distinction between the piconet of a pure master and a master&bridge, respectively. For a pure master we simply have:

$$p(pm) = 1, \quad (1)$$

where with $p(pm)$ we denoted the communication capacity allocated to the piconet of pure master $pm$.

Since master&bridge nodes have to switch among different piconets, we assume that each piconet switching takes two slots, $625\mu s$ each. We denote the communication capacity of a node wasted for one piconet switching by $\sigma$. On average, a node spends in each of its piconets about $40ms$, as proposed in [11]. The capacity dedicated to the piconets of a master&bridge node $mb$ is obtained using (2).

$$p(mb) = \frac{1}{NrM(mb)+1} - \sigma, \quad (2)$$

where $p(mb)$ is the communication capacity allocated to the masters of a master&bridge $mb$ as well as its own piconet; $NrM(mb) + 1$ is the total number of $mb$'s piconets. Note that the fact that $mb$ is a master&bridge implies that (2) is applicable only when $NrM(mb) \geq 1$.

Next we define a scheme for sharing the available capacity among the nodes of a piconet. A simple scheme is to allocate the same amount of bandwidth to each slave of a piconet. The problem with this simplistic approach is that it allocates the same amount of bandwidth also for bridge nodes that can dedicate less of their communication capacity to a particular master since they have to be present also in other piconets. Therefore, bandwidth will be allocated to nodes that can not take advantage of it.

To solve the above problem, we define the *available communication capacity*, $\beta$, as the capacity that a node can allocate from its whole communication capacity for a piconet P, and the *allocated bandwidth*, $\gamma$, as the portion of P's communication capacity that can be dedicated to the node. Then, we denote by $\alpha$ the *availability factor of a node with respect to a piconet* (hereinafter simply availability factor), i.e., the ratio of the allocated bandwidth to the available communication capacity of the node: $\alpha = \gamma/\beta$.

Taking advantage of the availability factor definition, we observe the following properties of a node. A node is said to be *underloaded* with respect to a particular piconet if it can dedicate more bandwidth to that piconet than the amount of bandwidth that the piconet can allocate to the node, i.e., $\alpha < 1$. Clearly, if a node is not underloaded then $\alpha \geq 1$. Therefore in the following we define a node as *overloaded* if either $\alpha = 1$ or $\alpha > 1$.

By using the above notions, next we present the availability factor computation for nodes with different role. We notice that in the formulas defined for this calculus it is not necessary to explicitly consider the slots used for switching, since at this phase it is only important that these slots are busy and is insignificant to emphasize that they are not used for communication.

- *Pure slave ($ps$)*: $\gamma$ is the fraction of bandwidth allocated by the piconet master to $ps$ out of the piconet's total capacity of $1$. $\beta$ is equal to $1$ since $ps$ dedicates its whole bandwidth to its master. Thus, for pure slaves $\alpha \leq 1$ for any possible value of $\gamma$ and $\beta$. Therefore we can state that pure slaves are always underloaded. (Notice that $\alpha = 1$ only when the scatternet consists of merely one piconet made of 2 nodes: a pure master and its pure slave – this case is insignificant from the viewpoint of scatternets.)
- *Pure master ($pm$)*: since the pure master manages the entire piconet bandwidth, $\gamma = 1$. Similarly, a pure master dedicates all of its bandwidth to its piconet, therefore $\beta = 1$ as well. Thus, for pure masters we always have $\alpha = 1$ i.e., these nodes are always overloaded.

- *Slave&bridge (sb)*: a $sb$ is independently allocated a certain bandwidth $\gamma$ in each piconet it belongs to. On the other hand, a $sb$ shares its own capacity among all of its masters. Initially each slave&bridge shares its capacity uniformly among its masters. Therefore, each piconet is allocated a portion of $\beta = 1/NrM(sb)$ from the total capacity of 1 of the $sb$. Thus the availability factor is $\alpha = \gamma \cdot NrM(s)$, where $\gamma \in (0, 1]$ decreases with the increasing number of nodes in the reference piconet. In this case, $\alpha$ may be either smaller or greater than 1. Therefore, slave&bridges may be both, underloaded or overloaded.
  For example, if in a piconet P the slave&bridge $sb$ is allocated a bandwidth $\gamma = 0.2$ (which may be the case when there are 5 slaves in P) and $sb$ is a slave in three piconets, i.e., $NrM(sb) = 3$, then $\alpha = 0.2/0.3 = 0.66$. In this case $sb$ is underloaded with respect to P because it can dedicate more bandwidth to P than P can allocate for $sb$. However, if there were only 2 slaves in P, we would have $\gamma = 0.5$ and hence $\alpha = 0.5/0.3 = 1.66$. In this latter case $sb$ is overloaded because P dedicates more bandwidth to it than it can handle.
- *Master&bridge (mb)*: a $mb$ shares its bandwidth uniformly among its masters and its piconet, hence it dedicates to its own piconet a portion of $\beta = 1/(NrM(mb)+1)$. On the other hand, $mb$ manages the whole bandwidth available for its piconet, thus $\gamma = 1/(NrM(mb) + 1)$. Therefore, the load factor of a master&bridge with respect to its own piconet is $\alpha = 1$.
  Notice that the availability factor of an $mb$ with respect to the piconets of its masters can be calculated similar to the slave&bridge case. The $mb$ is allocated a certain $\gamma$ by each piconet master while it is able to dedicate at most $\beta = 1/(NrM(mb) + 1)$ of its own bandwidth to each of its masters. Thus the expression of the availability factor is $\alpha = \gamma \cdot (NrM(mb) + 1)$

Thus we can write that the number of underloaded nodes in the piconet of any master $m$ ($NrUN(m)$) is the sum of the pure slaves ($NrPS(m)$) and the underloaded bridges ($NrUB(m)$) (3). Then we can calculate the number of overloaded slaves ($NrOS(m)$) through (4), where $NrS(m)$ is the number of slaves of $m$.

$$NrUN(m) = NrPS(m) + NrUB(m) \tag{3}$$

$$NrOS(m) = NrS(m) - NrUN(m) \tag{4}$$

Once we have computed the number of underloaded and overloaded nodes in a piconet, we can define the link capacities ($l$) in each piconet as follows.

For overloaded links ($\alpha \geq 1$) from masters to slave&bridges we have:

$$l^o_{sb} = \frac{1}{NrM(sb)} - \sigma \tag{5}$$

For overloaded links ($\alpha \geq 1$) from masters to master&bridges we have the same expression as in (2), since master&bridges allocate the same communication for both their masters and piconet:

$$l^o_{mb} = p(mb) = \frac{1}{NrM(mb) + 1} - \sigma \tag{6}$$

The capacity of a pure master or master&bridge $m$ that is not used by overloaded links is uniformly shared among the underloaded links in its piconet, similar to the max-min fair technique [12], [13]. For each such master $m$, the obtained coefficients are stored in a vector $\rho^m = \{\rho^m_i | i = \overline{0, NrUN(m)}\}$. The fraction of the unallocated capacity that is not used by the links is stored in $\rho^m_0$. Note that if the unallocated capacity can be fully redistributed among the links then $\rho^m_0 = 0$. Equation (7) captures the redistributed capacity of an underloaded link, connecting any type of master $m$ to any type of slave $s$.

$$l^u_s(m) = (p(m) - \sum_{i=1}^{NrOS(m)} l^o_i) \cdot \rho^m_s, \tag{7}$$

where $\sum_{i=1}^{NrOS(m)} l^o_i$ gives the total bandwidth allocated for all overloaded slaves of master $m$. Notice that $p(m)$ should be expressed as in (1) or (2) for pure masters or master&bridges, respectively. In (7) we subtract from the total communication capacity of the piconet the bandwidth allocated for the overloaded nodes (obtaining the total unallocated capacity of $m$), then we multiply it by the fraction corresponding to the underloaded link connecting master $m$ to its slave $s$.

Before terminating the capacity allocation, each node compares its own communication capacity of 1 to the total amount of bandwidth received from other nodes. If the received bandwidth is smaller than 1 then the node has some unallocated capacity. Each node having unallocated capacity tries to allocate it to its neighbors. For each node $n$, these redistributed capacity fractions (i.e. $\delta^n_i$) are stored in the vector $\delta^n = \{\delta^n_i | i = \overline{0, NrN(n)}\}$ where $NrN(n)$ is the number of neighbors (denoted by the index $i$) of node $n$ with unallocated capacities. After several iterations of this latter phase all nodes will have allocated as much as possible from their capacities (stored in the vectors $\delta^n$). The corresponding updated formulas for (5)–(7) are (8)–(10), respectively.

$$l^o_{sb} = \tfrac{1}{NrM(sb)} - \sigma + \delta^n_{sb} \tag{8}$$

$$l^o_{mb} = p(mb) = \tfrac{1}{NrM(mb)+1} - \sigma + \delta^n_{mb} \tag{9}$$

$$l^u_{ps}(m) = (p(m) - \textstyle\sum_{i=1}^{NrOS(m)} l^o_i) \cdot \rho^m_{ps} + \delta^n_{ps} \tag{10}$$

## IV. Throughput and Power Estimation

For our calculus we consider as input variable the total *number of hops* between all communication peers in the scatternet. The outputs of interest are the overall scatternet *throughput* and *power consumption*.

Let $\mathcal{N}$ be the set of nodes, $\mathcal{L}$ the *set of all radio links*, $\mathcal{C}$ the *set of all traffic connections* in the scatternet and $h_{sd}$ the minimum hop count between an $(s, d) \in \mathcal{C}$ source-destination communication pair.

Based on the results in Section III, we can calculate the maximum usable bandwidth, $c_{ij}$, of a radio link $(i, j) \in \mathcal{L}$, as follows:

$$c_{ij} = \begin{cases} l^o_{sb}, & \alpha_{ij} \geq 1, \; i \text{ is master, } j \text{ is slave\&bridge,} \\ l^o_{mb}, & \alpha_{ij} \geq 1, \; i \text{ is master, } j \text{ is master\&bridge,} \\ l^u_{ps}(m), & \alpha_{ij} < 1, \; i = m \text{ is master, } j \text{ any slave} \end{cases}$$

where $\alpha_{ij}$ is the availability factor of node $j$ with respect to piconet $i$. The indices $i$ and $j$ are interchangeable.

The maximum bandwidth fraction of a link ($c_{ij}$) is shared by the traffic connections crossing that specific link as shown in (11). In (11) we denoted by $(s, d) \supset (i, j)$ all the connections $(s, d)$ crossing link $(i, j)$.

$$c_{ij} = \sum_{(s,d) \supset (i,j)} f^{sd}_{ij} \tag{11}$$

We use the max-min fair bandwidth allocation algorithm to compute the portion $f^{sd}_{ij}$ that is allocated to each particular connection $(s, d)$ from the available bandwidth on a link $(i, j)$. Let us denote by $F_{ij} = \{f^{sd}_{ij} | (s, d) \in \mathcal{C}\}$ the vector of bandwidth portions allocated to each connection $(s, d)$ on a link $(i, j)$. We can then express the throughput of an $(s, d) \in \mathcal{C}$ traffic connection as in (12).

$$\theta^{sd} = C \cdot \min_{(i,j) \in (s,d)} (f^{sd}_{ij} \cdot q_{ij}) \tag{12}$$

where $C$ is the maximum capacity of a Bluetooth radio link, specific for each DH and DM packet type, $min_{(i,j) \in (s,d)}(f^{sd}_{ij} \cdot q_{ij})$ denotes the smallest usable bandwidth portion on the links of a connection $(s, d)$ (i.e. the bottleneck), while $q_{ij}$ is the packet success rate (PSR) of the link $(i, j)$. PSR can be obtained from the packet error rate (PER), as in (13), while PER, denoted by $r$, can be calculated as a function of the bit error rate (BER), using the formulas (14) and (15), for DH and DM packet types, respectively [14].

$$q = 1 - r \tag{13}$$

$$r = 1 - (1 - b)^s \tag{14}$$

$$r = 1 - ((1 - b)^{15} + 15b(1 - b)^{14})^{s/15} \tag{15}$$

where $s$ is the size of the packet in bits and $b$ is the BER.

The BER can be obtained from the link quality (LQ) value with some vendor-specific formula. However, [2] states that link quality values should be normalized to the range [0,255]. In our calculus we use the CSR (Cambridge Silicon Radio) model, given in (16).

$$\begin{aligned} BER &= (255 - LQ)/40000, & 215 \leq LQ \leq 255 \\ BER &= 32 \cdot (255 - LQ)/40000, & 105 < LQ \leq 215 \\ BER &= 256 \cdot (255 - LQ)/40000, & 0 \leq LQ \leq 105 \end{aligned} \tag{16}$$

Finally, the aggregate throughput over all traffic connections (i.e. the throughput of the scatternet) can be calculated as:

$$\theta_a = \sum_{(s,d) \in \mathcal{C}} \theta^{sd} = C \cdot \sum_{(s,d) \in \mathcal{C}} \min_{(i,j) \in (s,d)} (f^{sd}_{ij} \cdot q_{ij}) \tag{17}$$

Having obtained the expression of the scatternet throughput, we now have to demonstrate the relation between $\theta_a$ and the hop count ($h$) of the scatternet. Notice that $h$ can be calculated as the sum of bandwidth portion vector elements (i.e. connections) on all links:

$$h = \sum_{(i,j) \in \mathcal{L}} |F_{ij}| \tag{18}$$

In (18) each unitary hop count reduction implies the decrease by one of exactly one bandwidth portion vector's ($F_{ij}$) number of elements . This, on turn, implies that one bandwidth portion of the involved link is released. If the link capacity was not fully utilized before the hop reduction then the network throughput will remain unchanged. (However, the power consumption will decrease, as we will see later in this section.) Secondly, if the link capacity was fully utilized then after the hop reduction the bandwidth used by the old connection will be distributed among the remaining ones. In other words, the bandwidth portions $f^{sd}_{ij}$ will increase on the involved link. This implies that all connections having their bottleneck on the link in question will be allocated new bandwidth, i.e. the minimum $f^{sd}_{ij}$ value will grow. It can be seen in (17) that this growth has direct positive impact on the aggregate throughput $\theta_a$. This clearly shows why lower scatternet hop counts can produce higher network throughput.

Let us now take a closer look at the power consumption. We assume that the power consumption when transmitting and receiving data at the full capacity of a radio link is $P_t$ and $P_r$, respectively. Data is transmitted and received by all nodes along a path, excepting the source from one reception and the destination from one transmission. Therefore, all data bits are transmitted and received as many times as the number of hops along the path. Thus, the power consumption of an $(s, d) \in \mathcal{C}$ traffic connection can be expressed as

$$P^{sd} = (P_t + P_r) \cdot h_{sd} \cdot \min_{(i,j) \in (s,d)} (f^{sd}_{ij}) \tag{19}$$

Notice that the factor $min_{(i,j) \in (s,d)}(f^{sd}_{ij})$ in (19) adapts the power consumption to the bandwidth of the bottleneck link along the path.

The aggregate power consumption through all connections, $P_a$, is then given by:

$$P_a = \sum_{(s,d) \in \mathcal{C}} P^{sd} = (P_t + P_r) \cdot \sum_{(s,d) \in \mathcal{C}} h_{sd} \cdot \min_{(i,j) \in (s,d)} (f^{sd}_{ij}) \tag{20}$$

The dependence of the power consumption on the hop count is easy to see in this case since $h_{sd}$ appears explicitly in the expressions (19) and (20).

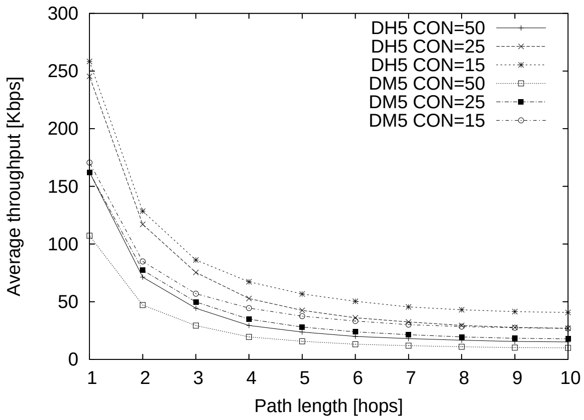


Fig. 1. Throughput versus connection length

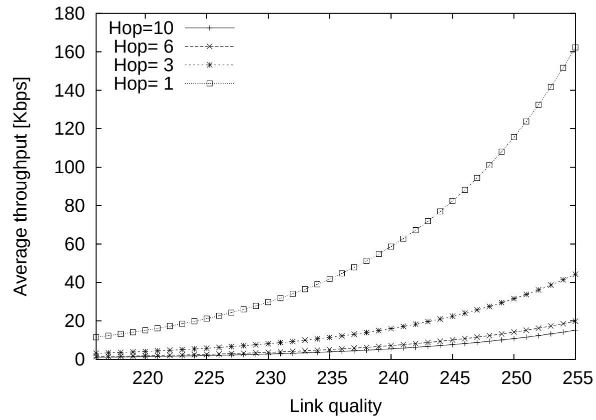


Fig. 2. Throughput versus link quality

## V. Evaluation

For evaluating our throughput and power consumption calculus, we implemented our model in C++. We performed experiments with 50 scatternets, each made of 100 randomly positioned nodes with communication range of 10m, on a 66x66m$^2$ area. On all these scatternets we generate 15 to 50 random bidirectional traffic connections. The number and length of the connections are fixed for each particular experiment. We perform experiments varying the length of connections from 1 to 10 hops as well as modifying the link quality value in the range of $[215, 255]$. The lower bound of 215 corresponds to the maximum bit error rate of 0.1% allowed by the Bluetooth Specification at the distance of 10m with no obstacles. Finally, during our experimentation we set $P_r = 150mW$ and $P_t = 170mW$. The experimental results shown in Fig. 1–3 are averaged over the 50 different scatternets.

In our first experiment (Fig. 1) we calculate the average throughput on 15, 25 and 50 bidirectional traffic connections. In this figure we show one of the main objectives of our work, i.e. the throughput decreases with the number of hops. The results show the maximum achievable average throughputs, since we use the two biggest packet types, (i.e. DH5 and DM5) and the link quality is set to 255 (i.e. no packet loss). As we expected, the highest average throughput per connection is achieved with 30 connections, with the DH5 packets, since in this case more bandwidth can be allocated to each connection. The curves then follow each other in the order of number of connections and the packet size.

In our second experiment (Fig. 2) we show the dependence of the throughput on the link quality. In this experiment the number of bidirectional connections is fixed to 50 and we use DH5 packets only. On the other hand, the connection length is different on each curve. In the figure, shows the expected result: the throughput increases with the link quality. Again, shorter connections are less affected by the link quality, while the longer ones have a very low throughput.

In our third experiment (Fig. 3) we tested the average power consumption on 15, 25 and 50 bidirectional connections. The packet type in this case has no importance since power is consumed at the same extent by both, useful payload bits and error coding bits.

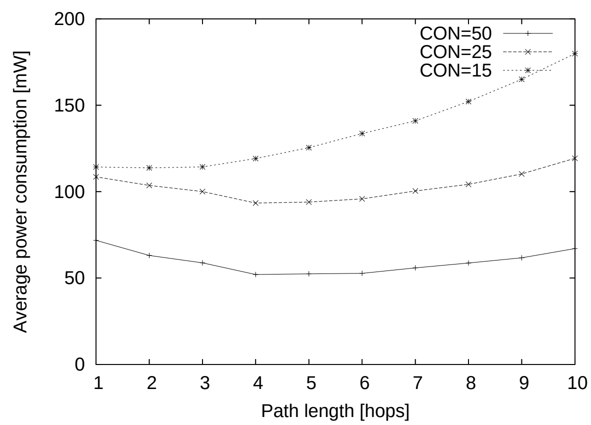


Fig. 3. Energy efficiency versus connection length

We can observe in the figure that initially the power consumption decreases, then it starts increasing again. This is explained by the fact that when the connections are short the throughput is high, therefore a higher amount of power is consumed. In other words, power consumption is high because more traffic is transmitted and not because it is less efficiently used. However, after increasing the number of hops and the throughput goes down, the real tendency of the power consumption shows up. It can also be seen that the highest amount of power is consumed when we have 15 connections, since in this case the throughput is higher. This, on turn, makes the power consumption increase faster, as it can be also seen in the figure.

Finally, the power consumption does not depend on the link quality since power is consumed at the same extent for transmitting new packets or repeating the old corrupted ones.

## VI. CONCLUSION

In this work we presented a method for analytically evaluating the throughput and power consumption in a scatternet based on the average number of hops connecting communication peers. For our approach we also modeled a link scheduling algorithm, necessary for calculating the throughput in the scatternet. Our results show that with a lower number of hops separating communication peers in a scatternet, it is possible to obtain a much higher throughput while power is used more efficiently. Further, using a link quality representation, we demonstrated the dependence of the throughput on the packet loss.

For the future we propose to improve our link scheduling scheme and evaluate our model also through simulations.

## ACKNOWLEDGEMENT

This work has been partially funded by the MICRO/STMicroelectronics award number 03-032a and by the Autonomous Province of Trento through the WILMA project.

## REFERENCES


[1] C. Kiss Kalló, C.-F. Chiasserini, R. Battiti, and M. Ajmone Marsan, "Reducing the number of hops between communication peers in a Bluetooth scatternet," in *IEEE Wireless Communications and Networking Conference (WCNC'04)*, Atlanta, GA, USA, March 2004.

[2] *Bluetooth Specification 1.2*, May 2003. [Online]. Available: http://www.bloetooth.com

[3] T. Salonidis, P. Bhagwat, L. Tassiulas, and R. LaMaire, "Distributed topology construction of Bluetooth personal area networks," in *IEEE INFOCOM*, Anchorage, April 2001.

[4] G. V. Zaruba, S. Basagni, and I. Chlamtac, "Bluetrees - scatternet formation to enable Bluetooth-based ad hoc networks," in *ICC 2001*, vol. 99, 2001, pp. 273–7.

[5] C. Law and K. Y. Siu, "A Bluetooth scatternet formation algorithm," in *IEEE Symposium on Ad Hoc Wireless Networks 2001*, San Antonio, TX, USA, November 2001.

[6] S. Basagni and C. Petrioli, "A scatternet formation protocol for ad hoc networks of Bluetooth devices," in *IEEE Vehicular Technology Conference (VTC)*, 2002, pp. 424–8.

[7] M. Ajmone Marsan, C. F. Chiasserini, A. Nucci, G. Carello, and L. De Giovanni, "Optimizing the topology of Bluetooth wireless personal area networks," in *INFOCOM*, New York, USA, June 2002.

[8] C. F. Chiasserini, M. Ajmone Marsan, E. Baralis, and P. Garza, "Towards feasible distributed topology formation algorithms for Bluetooth-based WPANs," in *36th Annual Hawaii International Conference on System Sciences*, Big Island, Hawai, January 2003.

[9] Z. Wang, R. Thomas, and Z. Haas, "Bluenet – a new scatternet formation scheme," in *35th Annual Hawaii International Conference on System Sciences*, Big Island, Hawaii, 2002.

[10] S. Baatz, M. Frank, C. Kohl, P. Martini, and C. Scholz, "Adaptive scatternet support for Bluetooth using sniff mode," in *LNC*, Tampa, Florida, 2001.

[11] A. Rácz, G. Miklós, F. Kubinszky, and A. Valkó, "A pseudo random coordinated scheduling algorithm for Bluetooth scatternets," in *ACM Symposium on Mobile AD Hoc Networking and Computing*, vol. 99, Long-Beach, CA, USA, 2001, pp. 1–100.

[12] Q. Ma, P. Steenkiste, and H. Zhang, "Routing high-bandwidth traffic in max-min fair share networks," in *SIGCOMM*, Stanford, CA, USA, August 1996, pp. 206–217.

[13] J. Jaffe, "Bottleneck flow control," *IEEE Transactions on Communications*, vol. 29, no. 7, pp. 954–962, July 1981.

[14] L.-J. Chen, R. Kapoor, M. Y. Sanadidi, and M. Gerla, "Enhancing bluetooth TCP throughput via link layer packet adaptation," in *ICC*, Paris, France, June 2004.