On Efficient Configurations for Bluetooth Scatternets ⋆

Daniele Miorandi

CREATE-NET
Via Solteri 38
38100 – Trento (Italy)

Simone Merlin, Arianna Trainito, Andrea Zanella

Department of Information Engineering
University of Padova, Via Gradenigo 6/B
35131 – Padova (Italy)

Abstract

In this paper we investigate the relationship between *capacity* and *network topology* for Bluetooth scatternets. We start by considering the intrinsic capacity limits of a scatternet structure, and we show that capacity may be approached in the presence of very local traffic only and under specific conditions on the scatternet structure. A worst case analysis is presented, addressing the performance of various network configurations in the presence of a uniform end-to-end traffic matrix. In particular, some efficient configurations, based on Platonic solids, are introduced and analyzed. Finally, we focus on the *closed loop* configuration, for which algorithms able to improve scatternet performance by exploiting traffic locality are presented and discussed.

Key words: Bluetooth, scatternet, efficient configurations, network capacity, graph partitioning

⋆ Part of this work has been presented at PWC03, Venice (Italy). This work was partially supported by MIUR within the framework of the PRIMO project FIRB RBNE018RFY.

Email addresses: daniele.miorandi@create-net.it (Daniele Miorandi), {simone.merlin,arianna.trainito,andrea.zanella}@dei.unipd.it (Simone Merlin, Arianna Trainito, Andrea Zanella).

1 Introduction

Originally born as a wireless replacement for cables connecting electronic devices, Bluetooth [1] has been gaining a lot of consideration and attention by the scientific community in the last few years. The development of this technology is now focused on the area of the so-called Wireless Personal Area Networks (WPANs), where Bluetooth is expected to play a major role in the short and mid-term future. The commercial success of WPANs is intimately linked to their ability to support advanced digital services, like audio and video streaming, web browsing, etc. [2]. In such a scenario, the performance aspects of the radio technologies involved appear of primary importance.

Bluetooth has been designed to work in a scattered ad hoc environment, where multiple independent overlapping networks, called piconets, may coexist and be interconnected to form a multi-hop network, called *scatternet*. Recently, much attention has been devoted, by both academic and industrial world, to issues concerning scatternets formation and management. In particular, scatternet formation algorithms have been widely investigated, and many solutions have been proposed to build up a scatternet starting from disconnected units (see reference [3]). The focus is now moving to the characterization and design of efficient scatternet topologies [4–9], since it is clear that the way piconets are interconnected to form scatternets may dramatically impact the network performance. To the best of authors' knowledge, however, the literature still lacks of a thorough investigation of the optimal scatternet configurations.

The optimality of a scatternet configuration depends on the performance indexes considered. Some typical performance metrics are the number of piconets, the number of gateway devices, the number of roles per node etc. When data connections are considered, throughput, average and maximum traffic delay are taken as metrics of interest.

In this paper, we focus on the *network capacity*. This metric represents the supremum of the aggregated traffic that nodes can inject into the network without overflowing. In general, we say that a network configuration is *stable* if the total traffic offered to the network does not exceed the network capacity. In stable configurations, the average packet delay is almost surely finite and, provided that the dimensions of the buffers are adequately chosen, packets are never dropped because of overflows. Conversely, in unstable configurations some traffic connections will either experiment always increasing average packet delays or packet losses due to buffer overflows.

In this paper, we investigate the relationship linking the scatternet configuration, i.e., the way piconets are interconnected to form the scatternet, and the network capacity. First we propose a mathematical formalization of the notion of network capacity and show that it can be achieved in the presence of one-hop traffic patterns only. Then, we discuss some topological conditions that are required to approach the network capacity, in the presence of local traffic only. This will entail a formal justification of the inefficiency of tree topologies, and will lead to the characteriza-

tion of *closed-loop* as *capacity-efficient* (i.e., able to approach the capacity bound for any number of devices). Successively, we drop the assumption of traffic locality, in order to investigate the performance achieved by some specific scatternet topologies in the presence of a uniform traffic matrix, providing a sort of worst-case analysis. It will be shown that some efficient solid configurations, based on Platonic solids, are able to outperform standard planar configurations in the presence of a limited number of devices. This analysis will lead to the conclusion that *closed-loop* configurations possess some desirable properties, from both the point of view of network performance and ease of protocols implementation. A general framework will be presented for the use of graph partitioning algorithms to arrange the nodes of a scatternet into a purposeful configuration. The application will focus on closed-loop configurations, for which heuristics for the choice of master and gateway units will be introduced and discussed. The generalization of the proposed techniques to account for spatial constraints (due to the fact that not all the nodes may be in mutual communication range) will also be discussed.
This work represents an attempt to provide mathematical insight into the relationship linking scatternet topology and performance. Our approach can be applied, for instance, to the design of networks of static sensors or domestic appliances, where the end-to-end traffic matrix among the nodes may be known *a priori*. At this phase of the work, however, many aspects related to the network creation and management have not been considered.
The paper is organized as follows. Section 2 outlines the primary characteristics of the Bluetooth technology and presents some related works. Section 3 describes the system model used in the remainder of the paper. Section 4 provides the analysis of the limiting performance, in terms of capacity, achievable by Bluetooth scatternets. In Section 5, system performance is evaluated in the case of a uniform traffic matrix. Optimization issues, with a focus on closed-loop configurations, are addressed in Section 6, where also an extension accounting for spatial constraints is addressed. Section 7 concludes the paper with some remarks on the limit of the work and directions for future work.

2 Bluetooth Scatternets: Background and Related Works

Bluetooth operates in the $2.4\,\mathrm{GHz}$ ISM unlicensed band, providing a raw bit rate of $1\,\mathrm{Mb/s}$ by using a binary Gaussian-shaped FSK modulation. In order to reduce interference with other devices operating in the ISM band, Bluetooth makes use of a frequency hopping (FH) spread spectrum technique, spanning 79 RF carriers, $1\,\mathrm{MHz}$ wide each. Two up to eight active Bluetooth units may be connected in a small network, called *piconet*. In each piconet, a unit acts as *master*, controlling the channel access by means of a simple polling scheme, while the other units act as *slaves*. Nodes actively exchanging data with master are given an *Active Member Address* and are said to be *active*. The standard encompasses a *parking state*.

Each piconet may host up to 256 nodes in the parking state; such nodes maintain the synchronization with the master node but do not participate to the piconet data exchange.
Time is divided into consecutive slots of $625\,\mu$s each, which are used for downlink (master-to-slave) and uplink (slave-to-master) transmissions, alternatively, in a time division duplex (TDD) fashion. Each time-slot is associated to an entry in the frequency hopping sequence, resulting in a nominal hop rate of $1600\,\mathrm{hop/s}$.
Different piconets are associated to independent FH channels. This allows more piconets to share the same physical space and spectrum without excessively increasing the mutual interference. Piconets may communicate by sharing a device on a time division basis, forming what is commonly referred to as a *scatternet*. The shared device, usually named *gateway*, spends a given amount of time in each piconet it belongs to, forwarding the inter-piconet traffic. The time a gateway spends in a piconet is called *permanence time*. To switch among piconets the gateway need a synchronization time called *switchover time*
The scatternet configuration is determined by:
• the assignment of the nodes to the various piconets;
• the assignment of the master and slave roles in each piconet;
• the identification of the shared units;
• the way piconets are interconnected.
The Bluetooth standard, while encompassing the notion of scatternet, does not specify how a scatternet has to be formed. A classification of feasible Bluetooth scatternet topologies is given in reference [9], where six possible configurations are considered:
• Single Piconet Model (SPM): only one piconet is present, with devices alternating between activity and parking mode to limit the number of active devices;
• Slave/Slave Mesh (SSM): the shared devices are slaves in all the piconets they belong to, and the resulting graph structure is a mesh, with multiple paths between any two nodes;
• Master/Slave Mesh (MSM): the shared devices are master in one of the piconets they belong to, resulting in a mesh configuration;
• Tree Hierarchy (TH): nodes are organized according to a tree structure, with Master/Slave gateways;
• Master/Slave Ring (MSR): piconets are interconnected in a logical ring, with Master/Slave gateways;
• Slave/Slave Ring (SSR): nodes form a ring, the shared units being Slave/Slave gateways.
Any of these configurations present advantages and drawbacks. For example, tree structures are inefficient from the point of view of bandwidth utilization, but they allow for an easy implementation of both scatternet formation and routing protocols. On the other hand, mesh topologies require the deployment of complex network management algorithms, but result in extreme robustness against link failures, and can exploit multipath routing in order to achieve load balancing. Ring structures may offer an interesting compromise, resulting in a robust structure over which routing is easy to perform.

One distinction of particular importance is the one between Slave/Slave configurations (where bridges can only be slave units) and Master/Slave ones (where bridges are master in a piconet and slave in the others). Although no studies have been conducted in order to rank these two approaches in terms of overall system efficiency [7], in reference [10] it is proved that S/S configurations lead to lower delays for the interpiconet traffic. Each topology is further characterized by the way in which interconnections are realized. This issue turns out to be critical in terms of network performance. In particular, there is a general consensus on the beneficial impact of a small gateways degree and small number of gateways between neighboring piconets [9]. Indeed, limiting the gateway degree may be beneficial since it limits the time wasted for switching from a FH channel to another one. Also, the assumption of a single gateway between any two piconets, while limiting the network resilience against link failures, aims at enhancing network performance by avoiding the introduction of unnecessary links.
Most of the works in the literature assume that all the devices are in range, so that centralized protocols can be used to build up the scatternet [3]. While this assumption is hard to justify in real-world deployments, it is usually taken to simplify the analysis of scatternet formation algorithms. In this case, an important issue is the interference among the (possibly) large number of overlapping piconets [3]. Due to the FH mechanism encompassed at the baseband layer, performance degrades only when the number of overlapping piconets is fairly large, around 50 [9,11].

3 System Model

In the remainder of the paper we assume that all the nodes are in range, except for the Section 6.3, where more realistic spatial constraints are considered. Neglecting the spatial constraints present in a realistic scenario on the "physical topology" [12] allows us to maximize the degrees of freedom for the choice of the "logical topology" to realize. For the sake of simplicity, we assume that only one-slot long packets are used (namely, $DH1$ packets) and that our network operates under ideal channel conditions. While such assumptions may look restrictive, our interest is on the analysis and design of efficient scatternet configurations, rather than on an accurate evaluation of the network performance. As such, we believe that the outcomes of our work can be used as design guidelines also in the case of networks acting in a lossy environment and using multi-slot packets. Furthermore, we limit our analysis to the simplest polling scheme, i.e., Pure Round Robin (PRR). With PRR, all slaves get polled cyclically by the master, whether they have data to transmit or not. Although many efficient polling schemes have been proposed in the last few years [13–15], at this time available devices basically support only PRR. This is essentially due to the necessity of keeping the complexity of the firmware as low as possible, in order to reduce manufacturing costs and power consumption. PRR

does not require complex logic to be embedded on the chip and, thus, it represents the most attractive choice for low-cost, energy-conserving products.
We introduce a probability space $\{\Omega, \mathcal{F}, P\}$, over which all the stochastic processes of interest are defined. In particular, arrivals to the systems are described by means of a marked point process $\{Z_n\}_{n\in\mathbb{Z}} = \{T_n, \xi_n\}_{n\in\mathbb{Z}}$, where T_n represents the arrival epoch of the n-th packet and ξ_n is a mark of the form (i, j), representing a packet generated from node i to node j, where i and j can be any couple of nodes of the scatternet. The process is assumed to be stationary ergodic with respect to the measure induced by P. We denote by $\delta_{i,j}$ the intensity of such arrival process on the link (i, j), in packets/slot. For a given configuration and routing protocol, we can construct an equivalent arrival process $\{\tilde{Z}_n\}$ which represents the packet arrival process at the baseband entities, thus accounting for multihop communications. We denote by $\lambda_{i,j}$ the intensity of such modified arrival process at node i to node j, in packets/slot. Notice that, in this case, i and j have to be members of a common piconet, either i or j being the piconet master. The switchover time and the permanence time are assumed to be independent and identically distributed (i.i.d.) random variables, with mean ε and ψ, respectively.
Let us denote by N the total number of nodes (which will be considered fixed) in the scatternet and by M the number of resulting piconets. [1] We denote by $\Delta = [\{\delta_{i,j}\}]$ the $N \times N$ *end-to-end traffic matrix*. Note that, the end-to-end traffic matrix Δ refers to the traffic produced at the transport layer, and, hence, it does not depend on the specific network topology.
Given the end-to-end traffic matrix, network topology and routing tables, [2] we can unequivocally determine the *effective traffic matrix* $\Lambda = [\{\lambda_{i,j}\}]$, which gives the actual data traffic flowing in the physical network links. Since direct communication can occur between master and slave (or gateway) units only, the (i, j)-th entry $\lambda_{i,j}$ of the effective matrix Λ can be greater than zero only if i and j belong to a common piconet, one of them being the piconet master. For the generic piconet $i \in \{1, 2, \ldots, M\}$, let n_i be the total number of nodes (master, slaves and gateways) the piconet consists of. Moreover, let ϑ_i denote the number of units that act as gateways in the i-th piconet. Clearly, it has to be $\vartheta_i \leq n_i$ and $n_i \leq 8$, since a piconet cannot have more than 8 active units. It may be worth noting that each gateway is counted in the ϑ_i of all the piconets it belongs to. The total number of gateways in the scatternet is denoted by Θ. Thus, the following relation holds:

$$N = \sum_{i=1}^{M} (n_i - \vartheta_i) + \Theta \; . \tag{1}$$

For the reader convenience, the notation used throughout the paper is summarized in Table 1.

[1] We have as many piconets as nodes playing the role of master.
[2] In the following we will consider balanced routing algorithms, in the sense that, if node A wants to communicate with node B and there are P shortest paths, the traffic $A \rightarrow B$ is equally divided among the P routes.

$\delta_{i,j}$	end-to-end traffic intensity (pkts/s)
$\lambda_{i,j}$	one-hop traffic intensity (pkts/s)
Δ	end-to-end traffic matrix
Λ	one-hop traffic matrix
n_i	number of node in the i-th piconet
N	total number of node
M	number of piconets
θ_i	number of gateways in the i-th piconet
Θ	total number of gateways
ε	mean permanence time
ψ	mean switchover time

Table 1
Notation

In the following, we will focus our analysis on symmetric scatternet configurations. We suppose gateways to be equally shared among the piconets they belong to, so that a gateway connected to γ_i piconets spends $\frac{1}{\gamma_i}\%$ of the time in each piconet it belongs to. Further, we assume that, when possible, all piconets have the same number $(n-1)$ of slaves, and the same number θ of gateways.

4 Network Capacity

The ability of a given scatternet configuration to carry information flows can be measured in terms of *network capacity*, defined as the supremum of the aggregated traffic that can be injected into the network preserving the network stability. A Bluetooth network can be modeled as a system of interconnected queues, and stability can then be defined in terms of the resulting network of queues.
As a first step, network capacity will be evaluated without considering the inter-piconet interference [16]. The effect of interference, however, will be considered in the successive sections.
Given the scatternet topology, we can define the set Ξ of all the end-to-end traffic matrices Δ that preserve the network stability for that specific topology. Let us denote by L the packet payload length (in bit) and by T the Bluetooth slot, $T = 0.625\,\mathrm{ms}$. Then, for a traffic matrix $\Delta = [\{\delta_{i,j}\}]$, we define the *offered load*, $G(\Delta)$, expressed in bit/s, as:

$$G(\Delta) = \frac{L}{T} \sum_{i=1}^{N} \sum_{j=1}^{N} \delta_{i,j} \,. \tag{2}$$

Note that G defines the effective aggregate user traffic offered to the network, since the traffic carried over multiple hops is counted just once. Given these preliminary definitions, we can define the ***network capacity*** $\mathcal{C}$ as:

$$\mathcal{C} = \sup_{\Delta \in \Xi} \{G(\Delta)\} \,. \tag{3}$$

Analogously, for a given network topology, we can define the set Ξ^* of all stable effective traffic matrices $\Lambda = [\{\lambda_{i,j}\}]$. Hence, we define the *aggregated throughput* $S(\Lambda)$ of the network as:

$$S(\Lambda) = \frac{L}{T} \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_{i,j} \,; \tag{4}$$

where also the traffic forwarded in a multi-hop fashion is taken into account. The parameter S gives the total traffic flowing in the physical links of the network.
Finally, we define the ***transport capacity*** $\mathcal{T}$ as:

$$\mathcal{T} = \sup_{\Lambda \in \Xi^*} \{S(\Lambda)\}. \tag{5}$$

Trivially, under stability conditions, the offered load G cannot exceed the aggregated throughput S and the two quantities are equal if all the traffic goes through a single hop. Taking the supremum of both G and S and using the notation defined above, we get the following lemma, whose proof is reported in reference [17].

Lemma 1 *For a given topology, the network capacity $\mathcal{C}$ equals the transport capacity $\mathcal{T}$: $\mathcal{C} = \mathcal{T}$.*

In short, the lemma states that the optimal network utilization is reached when communication occurs between nodes that are directly connected and all traffic gets through a single hop. Therefore, in order to compute the network capacity, we will consider single-hop traffic only (thus working with the transport capacity), and we will study the stability of the resulting network of queues.
Our aim is to find out the capacity for most interesting/common scatternet topologies. As a first step, we report some easy results related to the use of gateways, which will be used to obtain more general results on scatternet capacity.

4.1 Piconets Interconnections

In principle, the application of a queueing theoretical framework to complex scatternet structures is not feasible. However, some fundamental results can be found by considering three basic structures: an isolated piconets, a piconet with a Slave/Slave gateway and a piconet with a Master/Slave gateway. Successively, such results will be used to deduce the scatternet capacity.
A network of queues is said to be stable if any queue admits a stationary regime which is P-almost surely finite [18]. This leads, for any (i, j), to a condition of

the form $\lambda_{i,j} < \mu_{i,j}$, given that the conditions of the Baccelli-Foss saturation rule [18,19] are satisfied, which is usually the case in all configurations of interest. The capacity of link (i,j) will then be defined as $\mathcal{L}_{i,j} = \frac{\mu_{i,j}L}{T}$ (expressed in bit/s), where L denotes the baseband packet payload length (in bits) and T is the Bluetooth slot duration. For a given topology, the network capacity will then be defined as $\mathcal{C} = \sum_{i,j} \mathcal{L}_{i,j}$.

4.1.1 Single Piconet Model

We can analyze, for a piconet of n nodes, a system of $2(n-1)$ single-server queues with vacations, each one corresponding to a master-to-slave or slave-to-master link. We recall that we assume the use of 1-limited pure round robin as polling strategy. Under our assumptions, the equivalent service time of a packet in any of such queues is deterministic and equals a cycle time, of duration $2(n-1)T$. A necessary and sufficient condition for stability is thus given by:

$$\lambda_{i,j} 2(n-1) < 1, \quad i \neq j. \tag{6}$$

In this case, all links have the same capacity, which, expressed in bit/s is given by $\mathcal{L}_i = \frac{L}{2(n-1)T}$. Since the number of feasible links (i,j) is $2(n-1)$ we get the following proposition.

Proposition 2 *The capacity of a single isolated piconet is (in bit/s):*

$$\mathcal{C} = \sum_{i,j} \mathcal{L}_{i,j} = \frac{L}{T}.$$

4.1.2 Slave/Slave Gateway

Let us consider a piconet of n nodes with one device shared with other $(\gamma - 1)$ piconets (γ is thus the degree of the gateway node), so that the number of effective slaves is $(n-1)$ for a part of the time, and $(n-2)$ for the rest. Let us again consider an equivalent system of $2(n-1)$ queues, each one corresponding to a link. We first consider the link g between the master and the gateway. This link is active for a fraction $\frac{\psi}{\gamma(\psi+\varepsilon)}$ of the time. Its capacity is given by:

$$\mathcal{L}_g = \frac{L}{2(n-1)T\gamma} \cdot \frac{\psi}{(\psi+\varepsilon)}. \tag{7}$$

Next, we assume that the piconet consists of $n > 2$ nodes, so there is at least a pure slave. We focus on the link s between such (non-shared) slave and the master. In this case, the link is always active. What is varying is the equivalent service time of a packet, which equals $2(n-1)T$ for a fraction $\frac{\psi}{\gamma(\psi+\varepsilon)}$ of the time and $2(n-2)T$

otherwise. The link capacity is thus given by:

$$\mathcal{L}_s = \frac{L\psi}{2(n-1)T\gamma(\psi+\varepsilon)} + \frac{L[\gamma\varepsilon+(\gamma-1)\psi]}{2(n-2)T\gamma(\psi+\varepsilon)}. \tag{8}$$

Summing over the various links, we get the following proposition.

Proposition 3 *The capacity of a piconet composed by a master, a slave/slave gateway and at least one pure slave is*

$$\mathcal{C} = 2\mathcal{L}_g + 2(n-2)\mathcal{L}_s = \frac{L}{T}. \tag{9}$$

If $n = 2$ (the piconet is composed only by a master and a gateway) we get $C = 2\mathcal{L}_g = \frac{L}{T\gamma}\frac{\psi}{(\psi+\varepsilon)}$, so that the presence of a pure slave leads to an increased capacity, that is also independent on the permanence and switchover times. The proposition generalizes in the presence of multiple shared slave devices, under the condition that at least one slave is present at any time. If the piconet consists of n devices and each slave is shared among γ piconets, this condition may be satisfied if the number of shared slaves is strictly less than $(n-2)$. If all the slaves are shared devices, on the other hand, the capacity decreases for any $\varepsilon > 0$. In the case $\varepsilon = 0$ (i.e., no switchover times), indeed, we can, at least in principle, find a synchronization among the gateway devices such that at least one is active in the piconet at any time, getting:

Proposition 4 *Given a piconet with n devices and k slaves which act as gateway, with $k < n-2$, the piconet capacity satisfies:*

$$\mathcal{C} = \frac{L}{T},$$

regardless of the switchover times and permanence times.

4.1.3 Master/Slave Gateway

We now consider a piconet, composed of n nodes, among which the master node is the gateway. This is typically the case in tree structure, one of the configurations mostly studied in the literature [3]. In this case, we assume that the master behaves as slave in $(\gamma-1)$ other piconets. In this case, the capacity of any master-slave link is given by:

$$\mathcal{L}_{i,j} = \frac{L}{2(n-1)T\gamma}\frac{\psi}{\psi+\varepsilon}. \tag{10}$$

The piconet capacity is thus given by:

$$\mathcal{C} = \frac{L\psi}{T\gamma(\psi+\varepsilon)}. \tag{11}$$

Taking the limit $\frac{\varepsilon}{\psi} \to 0$, we get the following proposition:

Proposition 5 *Given a piconet with n devices, whose master act as gateway device, the piconet capacity satisfies the following inequality:*

$$\mathcal{C} \leq \frac{L}{2T}.$$

Note that, in this case, the capacity can reach at most one half of the capacity of Propositions 3 and 5, thus indicating that master/slave gateways are harmful from the capacity point of view.

4.2 Bounds on Scatternet Capacity

Now we use previous results to define some bounds on the scatternet capacity.

Lemma 6 *Given M isolated piconets, the network capacity is equal to $\mathcal{C} = \frac{ML}{T}$.*

In the more general case of interconnected piconets, exploiting the results of the previous section, we get the following result on the capacity.

Proposition 7 *A scatternet consisting of M piconets can achieve a network capacity of at most $\frac{ML}{T}$. This happens if the following conditions are satisfied:*

(1) the shared units are slaves in all the piconets they belong to;
(2) in each piconet, there is at least one slave present at any time.

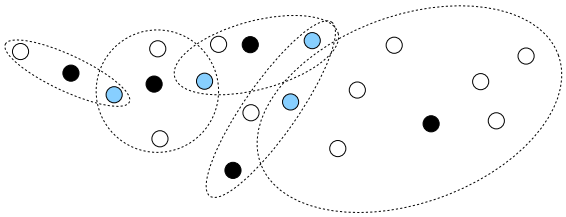


Fig. 1. Chain configuration.

It is not difficult to find structures that satisfy such conditions. An example is given by the simple chain configuration depicted in Figure 1. Under the conditions of Proposition 7, the capacity turns out to be linearly increasing with the number of piconets, suggesting that, in order to get a larger network capacity it is beneficial to construct small piconets. [3] Therefore, it may be interesting to determine the maximum capacity that can be achieved by a scatternet of N nodes. A partial answer is given by the following proposition, whose proof can be found in reference [17]:

[3] We recall that, in this section, interpiconet interference is not considered.

Proposition 8 *The network capacity of a scatternet consisting of N interconnected nodes cannot exceed is $\left\lfloor \frac{N}{2} \right\rfloor \frac{L}{T}$.*

It is noteworthy to stress that the bound may be achieved only if $\varepsilon = 0$.

4.3 Capacity of Some Scatternet Configurations

General results obtained above are now applied to determine the capacity of some scatternets with specific configurations, such as closed loop, tree, star shaped and platonic solid. In the following analysis, the inter-piconet interference will be neglected. However, the impact of inter-piconet interference will be considered within the successive sections, where some possible scatternet configurations are analyzed in case of uniform end-to-end traffic matrices.

4.3.1 Closed-Loop Configuration

The upper bound defined by Proposition 8, can be achieved by specific network topologies, such as the closed-loop configuration depicted in Figure 2. In this case, each piconet consists of $n = 3$ nodes (except for at most one piconet, with $n_M = 4$ if N is odd) and $\vartheta = 2$ gateways. The total number of piconets and gateways are, then, equal to $M = \left\lfloor \frac{N}{2} \right\rfloor$ and $\Theta = \left\lfloor \frac{N}{2} \right\rfloor$. The regularity of the structure enables us to find a scheduling of the gateways such that there is always at least one slave in each piconet under the hypothesis of null switchover time ($\varepsilon = 0$). Thus, conditions of Proposition 7 are satisfied and, consequently, the limiting capacity of the network is equal to $\frac{ML}{T} = \left\lfloor \frac{N}{2} \right\rfloor \cdot \frac{L}{T}$.
We thus say that the closed-loop configuration is capacity-efficient, since it achieves the bound for any value of N.

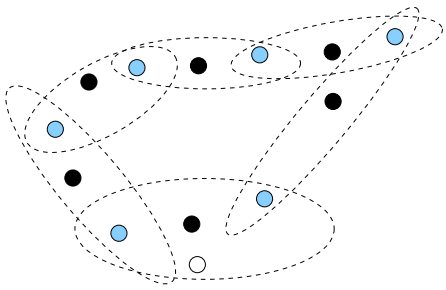


Fig. 2. Closed-loop configuration.

Note that, the above results may provide useful insight into efficient configuration design. Another interesting observation concerns the ring architecture considered in reference [20], where all nodes act as master/slave relays. It is easy to prove that also this configuration (which can be, in a certain sense, considered a degeneration of the closed-loop architecture) is capacity-efficient. (In fact, we conjecture that

these are the only two capacity-efficient architectures). The ring topology, however, presents scalability problems and performance rapidly worsens when traffic has to be forwarded in a multi-hop fashion.

For the closed-loop configuration, we may also get some insight into the effect of non-null switchover times. Indeed, using results from the previous sections, we get the following propositions, whose trivial proofs are not reported for the sake of conciseness:

Proposition 9 *Given a closed-loop structure with N nodes and $M = \left\lfloor \frac{N}{2} \right\rfloor$ piconets, the network capacity is given by:*

$$\mathcal{C} = \left\lfloor \frac{N}{2} \right\rfloor \frac{\psi L}{T(\psi + \varepsilon)}.$$

In order to get a bound which is independent on ψ and ε, we need to enlarge the piconet size to $n = 4$, getting:

Proposition 10 *Given a closed-loop structure with N nodes, with $M = \lfloor \frac{N}{3} \rfloor$ piconets, the network capacity is given by:*

$$\mathcal{C} = \left\lfloor \frac{N}{3} \right\rfloor \frac{L}{T}.$$

regardless of both ψ and ε.

The two propositions above tell us that, in the presence of a small ratio $\frac{\varepsilon}{\psi}$, it is convenient to make piconets with $n = 3$, whereas, in the presence of a larger value, it is more convenient to make larger piconets, ensuring that at least one slave is present in the piconet at any time. In particular, the threshold value can be easily seen to be equal to:

$$\left(\frac{\varepsilon}{\psi}\right)^* = \frac{\lfloor \frac{N}{2} \rfloor}{\lfloor \frac{N}{3} \rfloor} - 1 \approx \frac{1}{2},$$

the approximation holding for large N.
In the following we will consider, for the sake of clarity, the case $\varepsilon = 0$ only. While this may be considered a limit of our analysis, we think that, in most applications, the persistence time will be much larger than the switchover time, thus making our results sound.

4.3.2 Tree

Now, let us consider, as a case study, tree structures and in particular binary trees (the reasoning can be easily extended to k-trees, but it turns out that their capacity is monotone decreasing in k). We get (see reference [17] for the proof):

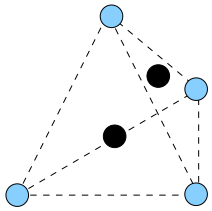


Fig. 3. Tetrahedral configuration (only the nodes of two piconets are shown).

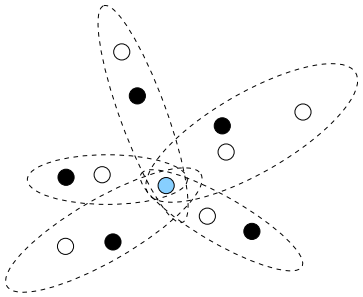


Fig. 4. Single gateway configuration.

Proposition 11 *Given N nodes, a binary tree configuration achieves a network capacity* $\mathcal{C} = \left(\left\lfloor \frac{N}{2} \right\rfloor + 1\right) \frac{L}{2T}$.

4.3.3 Other configurations

For a fixed N, it is easy to find different configurations which achieve the capacity bound. For example, when $N = 8$, the limiting capacity may be achieved by a tetrahedral structure, as that shown in Figure 3. Another interesting configuration is the single-gateway configuration, characterized by a star topology, as depicted in Figure 4. For this configuration, we get the following result [17].

Proposition 12 *Given N nodes, a single-gateway configuration may achieve a network capacity* $\mathcal{C} = \left\lfloor \frac{N-1}{2} \right\rfloor \frac{L}{T}$.

The Proposition above tells us that single-gateway configurations are not far from the optimal performance; however, as we will see in the next section, their performance rapidly decreases as soon as we consider non-local traffic.

5 Efficient Configurations for Bluetooth Scatternets

In this section we provide a sort of worst case analysis, considering network capacity in the presence uniform end-to-end traffic matrices, where each node generates, on average, the same amount of traffic to every other node in the network. In this situation, any reordering of the nodes would not increase the traffic locality. Consequently, network performance for uniform traffic matrices will depend on the scatternet topology only and not on single node positions.
To gain insight and to keep the analysis simple, we focus on symmetrical configurations only. We first introduce some stability issues, which are at the basis of the successive capacity evaluation. Then, we introduce a model for inter-piconet interference [16] and present some numerical results.

5.1 Stability conditions

The traffic matrix Δ has entries defined as $\delta_{i,j} = \delta$ if $i \neq j$ and 0 otherwise. In such case, it is easy to check that the bottleneck of the network is represented by the links involving shared units. Further, under our assumptions on network symmetry, it suffices to consider a single master-to-gateway link in order to derive a bound for the offered uniform traffic, δ, that preserves the network stability.
Let us consider a master-to-gateway link. Its capacity $\mathcal{L}$ depends on the scatternet topology only. We can express the traffic offered to this link as a function of the traffic matrix Δ and, indeed, as a function, $g(\delta)$, of the offered end-to-end traffic. In order to satisfy the stability condition it has to be $g(\delta) < \mathcal{L}$ which translates to the bound $\delta < \zeta$, with $\zeta = g^{-1}(\mathcal{L})$. Given such bound, the offered traffic $G(\Delta)$ will satisfy the relationship $G(\Delta) < N(N-1)\zeta$. Considering the set $\mathcal{U}$ of all the uniform end-to-end traffic matrices Δ which preserve the network stability, we define the *uniform capacity* $\tilde{\mathcal{C}}$ as the supremum of the offered traffic that can be reached within $\mathcal{U}$, i.e., (in bit/s)

$$\tilde{\mathcal{C}} = \sup_{\Delta \in \mathcal{U}} G(\Delta) = \frac{LN(N-1)\zeta}{T} \ . \tag{12}$$

Now, it remains to explain how to compute ζ. Notice that, given a specific topology, the problem is to compute the function $g(\cdot)$, and the link capacity $\mathcal{L}$.
Due to our symmetry assumption, we find that the number of slaves present in each piconet draws a periodical pattern. Let us denote by T_c the cycle time. [4] Applying the Baccelli-Foss saturation rule [19], together with the renewal reward theorem, we get the following stability condition:

$$\lambda_{i,j} < E\left[\frac{1}{T_c}\int_0^{T_c} \frac{b_{i,j}(t)}{2\left[n_{i,j}(t)-1\right]}dt\right], \tag{13}$$

where $n_{i,j}(t)$ represents the number of nodes in the considered piconet at a given time instant t and $b_{i,j}(t)$ is either 1 or 0 according to the active or idle state of link (i,j) at time t. Note that, since cycles are i.i.d., the expectation can be taken with respect to the measure induced by P.
In the following, we will take the switchover times to be P-almost surely zero, and the permanence time to be deterministic. Under such assumption, we can subdivide the cycle time into Q periods during which the number of nodes in each piconet is fixed, getting the following expression for the stability condition:

$$\lambda_{i,j} \ < \ \frac{1}{2T_C}\sum_{m=1}^{Q} \frac{b_m}{n_m - 1}(T_m - T_{m-1}), \tag{14}$$

[4] Notice that, T_c is deterministic only if both the switchover and permanence times are deterministic

Topology	Number of piconets
single-gateway	$M = \lfloor \frac{N-1}{n-1} \rfloor$
closed-loop	$M = \lfloor \frac{N}{n-1} \rfloor$
tetrahedron	$M = 4$

Table 2
Relationship between N, n and M for different scatternet topologies.

where n_m and b_m are the values assumed by $n_{i,j}(t)$ and $b_{i,j}(t)$, respectively, for $t \in (T_{m-1}, T_m)$.
For the sake of simplicity, in the following we assume that all the piconets have the same number of units, i.e., $n_i = n$, $i = 1, \ldots, M$. Table 2 shows the relationship that ties N, n and M for some scatternet topologies that will be considered in the following.

5.1.1 Planar Configurations

Let us, first, evaluate the uniform capacity for two simple planar structures, namely the star-shaped and the closed-loop topologies.
The star topology relies on a single gateway to forward traffic among the M piconets. The average aggregate traffic offered to the gateway by the nodes of each piconet is equal to $g(\delta) = (n-1)(N-n+1)\delta$ packets per slot. Since the gateway spends a fraction $1/M$ of its time in each piconet, getting a fraction $1/[2(n-1)]$ of the piconet capacity during its permanence time, we have $\mathcal{L} = \frac{1}{2M(n-1)}$. We now can deduce ζ and the limiting capacity (expressed in bit/s) turns out to be

$$\tilde{\mathcal{C}} = \frac{L}{T} \cdot \frac{N(N-1)}{2M(n-1)^2(N-n+1)} \approx \frac{NL}{2T(n-1)(N-n+1)}\,, \tag{15}$$

where we used $M \approx \frac{N-1}{n-1}$. Please note that, for large N, $\tilde{\mathcal{C}} \sim \frac{L}{2T(n-1)}$. It is clear that, in such a situation, the gateway represents a bottleneck that limits the performance of the network. In order to overcome this weakness, a closed-loop structure, as depicted in Figure 2, can be considered. In this case, routing should be taken into account. We assume that the routing algorithm preserves somehow the symmetry of the structure (i.e., each gateway handles equal load), and that gateways are synchronized, so that a single gateway at a time is active in each piconet. Under these assumptions, each gateway gets a traffic rate of approximately $g(\delta) = \delta \cdot N^2/8$ packet/slot from each of the two piconets it belongs to. The capacity of a gateway-to-master link is $\mathcal{L} = \frac{L}{4T(n-2)}$. The uniform capacity (in bit/s) is thus approximately equal to: [5]

$$\tilde{\mathcal{C}} \approx \frac{8LN(N-1)}{4T(n-2)N^2} = \frac{2L(N-1)}{NT(n-2)}\,. \tag{16}$$

[5] For the sake of conciseness, we omit the derivation of the exact expression of the uniform capacity that, however, can be easily obtained distinguishing the cases of M even and odd.

In the case of large N, we have $\tilde{\mathcal{C}} \sim \frac{2L}{T(n-2)}$. We can thus conclude that the closed-loop topology scales better than the single-gateway topology in the presence of a uniform traffic matrix.

5.1.2 Solid Configurations

In order to achieve better performance, we might add another dimension to the network, so that the resulting structure may be represented by a *solid* structure. In this case, each piconet can be represented as a regular polygon having the gateways as vertices. To have a "fair" structure, the resulting polyhedron has to be regular. To avoid problems with the routing schemes employed, the resulting polyhedron has also to be convex. However, it is a well-known result (which dates back to the ancient Greeks) that there are just five regular convex polyhedra, namely tetrahedron, cube, octahedron, dodecahedron and icosahedron. These are usually referred to as the Platonic solids.
Denoting by F and V the number of facets and vertices of the polyhedron, respectively, and by E the number of edges of the building polygon, we have that the minimal and maximal number of Bluetooth nodes such a structure may handle are equal to $N_{min} = F + V$ and $N_{max} = (8 - E) \cdot F + V$. Focusing on the tetrahedron, the uniform capacity can be shown to be [17]:

$$\tilde{\mathcal{C}} = \frac{N(N-1)L}{6T(n-3)(n-2)^2} \,. \tag{17}$$

5.2 On the Impact of Inter-Piconet Interference and Lossy Channels

The mutual interference among overlapping piconets may have a strong effect on scatternet performance. Let $P_s(M)$ be the probability that two stations in piconet A successfully exchange a packet when $(M-1)$ other piconets are co-located in the scatternet. Therefore, the effective uniform network capacity can be expressed as:

$$\mathcal{C}' = \tilde{\mathcal{C}} \cdot P_s(M)\,; \tag{18}$$

where $\tilde{C}$ is given by (15), (16) and (17). We evaluate inter-piconet interference in the worst case, i.e., with always-transmitting unsynchronized piconets. Thus, according to the work in reference [16], we have $P_s(M) = a^{M-1}$, where the parameter a is equal to $a = 0.9852$. (Please note that the value of a depends on the peculiar packet type used, so that in the case of multislot packets we can easily obtain bounds for the network capacity.)

In Figure 5 and Figure 6 the uniform capacity $\mathcal{C}'$ is plotted versus the number of nodes N, for "fat" piconets ($n = 8$) and "slim" piconets ($n = 3$), respectively. (Clearly, the parameter n has been varied only for the single-gateway and closed-

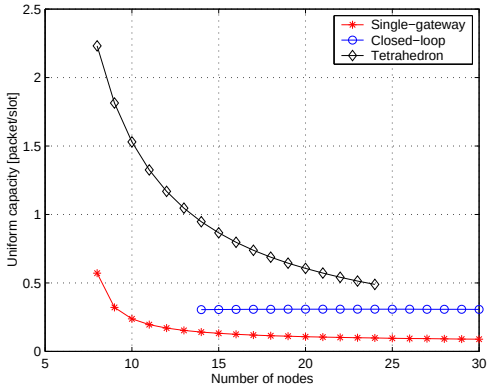


Fig. 5. Uniform capacity C' versus number of nodes N with "fat" piconets ($n = 8$).

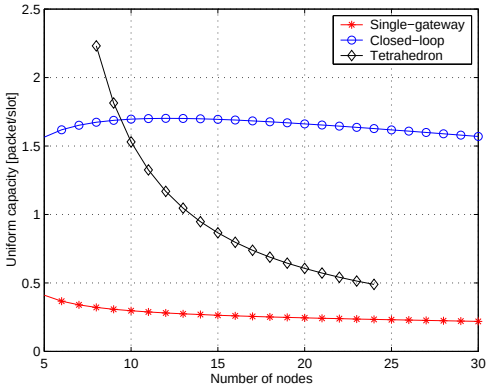


Fig. 6. Uniform capacity C' versus number of nodes N with "slim" piconets ($n = 3$).

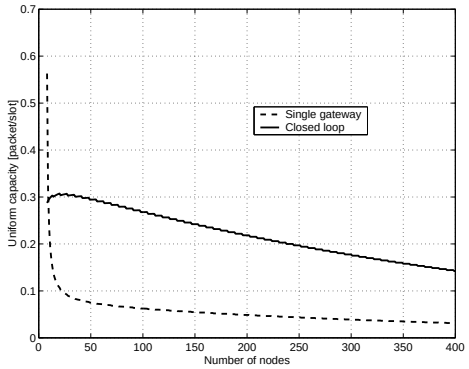


Fig. 7. Uniform capacity C' for high number of nodes N, for $n = 8$.

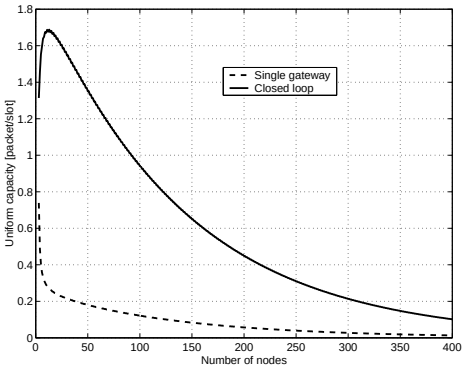


Fig. 8. Uniform capacity C' for high number of nodes N, for $n = 3$.

loop configurations, since for the tetrahedral configuration, the number of nodes per piconet is unequivocally determined by N.) As expected, the single-gateway topology always performs worse than the others, due to the bottleneck introduced by the unique gateway. The tetrahedron topology achieves the highest capacity, but can host only a very small number of nodes N. On the contrary, the closed-loop topology achieves good performance for a wide range of N values, provided that the piconets are "slim", i.e., the number n of nodes per piconet is small.

In Figure 7 and Figure 8, the uniform capacity for single-gateway and closed-loop topologies is depicted for a high number of nodes N for $n = 8$ and $n = 3$, respectively. As expected, the overall capacity tends toward zero as N grows, because of the increasing interference.
Given the number of nodes N and the scatternet topology, it may be interesting to evaluate the uniform capacity for different number of piconets M. In Figure 9 the number of piconets M that maximizes the uniform network capacity is depicted at the varying of the total number N of nodes in the scatternet, for the single-gateway and closed-loop topologies. Figure 10 shows the maximum capacity achieved by choosing the optimal M for each N. We can note that building up "slim" piconets seems to be preferable, with respect to the overall throughput, for N smaller than

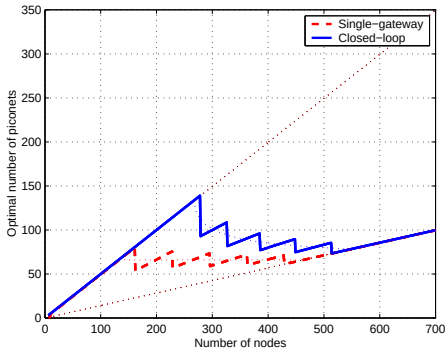


Fig. 9. Optimal number of piconets M for the closed-loop and single-gateway topologies.

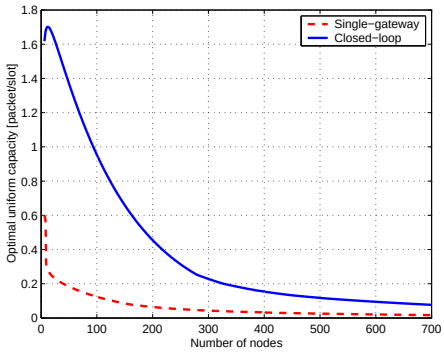


Fig. 10. Maximum capacity versus the number of nodes N, for the closed-loop and single-gateway topologies.

≈ 250. Then, the behavior is rather variable until N is bigger than ≈ 550, after which "fat" piconets, i.e., piconets consisting of $n = 8$ nodes, represent the best choice because of the strong influence of interference. Note that, the "comb"-like trend that can be observed in Figure 9 for values of N between 250 and 550 is due to the assumption that all the piconets contain the same integer number n of nodes.

6 Optimization Issues

In this section, we aim at presenting some heuristics for optimizing the network efficiency exploiting traffic locality. The focus will be on closed-loop configurations, due to the remarkable properties shown by such topology in terms of efficiency under a wide range of traffic conditions. Nonetheless, most solutions presented henceforth apply to *any* scatternet configuration, and result thus of general interest. Throughout the section, we assume the traffic matrix to be known *a priori* or, in any case, estimated at runtime. Once we assume the scatternet topology to have a closed-loop shape, the following issues need to be addressed.

(i) The partition of the nodes among the various piconets.
(ii) The arrangement of the piconets along the logical ring.
(iii) The choice of the shared units.
(iv) The choice of the master nodes.

These issues have not been previously analyzed, in the field of Bluetooth networks, apart for the last question, for which some criteria have been proposed and discussed in reference [21]. Moreover, it is worth remarking that the issues arising at points (i), (iii) and (iv) are common to any chosen scatternet configuration, so that our results may be applied also to different topologies.
As previously specified in Section 2, some other works have already appeared in the literature proposing SSR configurations for Bluetooth scatternets. Indeed, apart

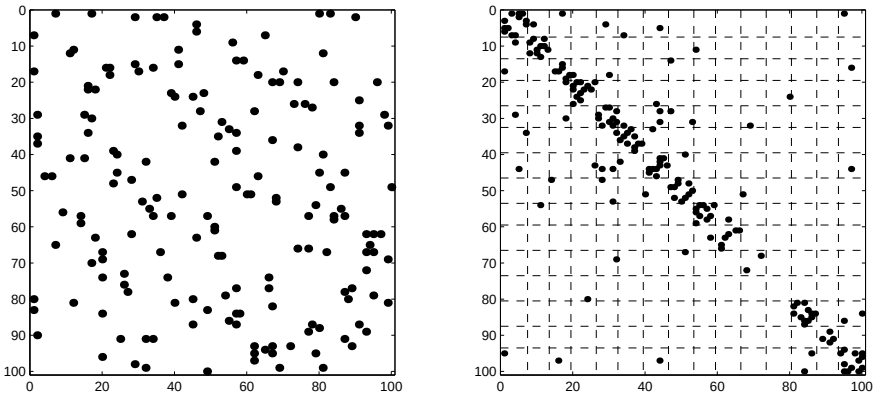
from the remarkable performance in terms of network capacity, loop configurations present other desirable properties. First, the existence of two distinct paths between any couple of nodes makes the network resilient with respect to link (or node) failures. Second, routing is easy to implement. Further, the lack of a "center" of the network helps in avoiding the formation of bottlenecks, spreading the traffic load on the whole network (in this sense, the adoption of balanced load routing could be foreseen to be beneficial) and enhancing the performance in terms of packet delay. On the other hand, ring structures are, in principle, not easy to build up. In reference [22] a centralized protocol for scatternet formation is presented, which works under the assumption that all nodes are in range. In the same paper, recovery mechanisms to overcome single and multiple path breaks are proposed.

6.1 Scatternet Partition Algorithms

Given the end-to-end traffic matrix Δ, the problem number (i) may be rephrased in terms of *communicating groups* [4]. In a realistic scenario, indeed, it is highly unlikely that a device wants to exchange data with all the others, but nodes will instead tend to form communication clusters. It is, then, clear that a desirable feature, from the network performance point of view, is that of obtaining an efficient mapping of such clusters into Bluetooth piconets, in order to lower the traffic that has to be forwarded in a multihop fashion.
Therefore, our aim is to find a reordering of the $N \times N$ traffic matrix Δ to approach a block-diagonal form, the blocks being of dimension $n \leq 8$, each block corresponding to a single piconet.
The problem can be defined in terms of the corresponding communication graph. From the matrix Δ, indeed, we can build a graph $\mathcal{G} = (V, E)$, whose vertices represent the nodes and in which an edge between node i and node j corresponds to a non-zero entry in the (i, j)-th position of the matrix Δ. The label of that edge corresponds to the value assumed by $\delta_{i,j}$. The problem is, then, to divide the vertices into p sets in such a way that the sum of the weights of edges crossing between sets is minimized. [6] Even for the simplest case, $p = 2$, this problem (that in such case is referred to as *equicut*) is NP-complete. Therefore we resort to heuristics to find a suboptimal solution, but in a reasonable time. The two classical algorithms used for such a problem are the spectral algorithm [23] and the one originally proposed by Kernighan and Lin [24]. In particular, we will focus on the latter one with the linear time modification provided by Fiduccia and Mattheyses [25]. The efficiency of the algorithm can be further enhanced by using a multilevel approach [26], as we will do in the following. All the numerical results presented have been obtained by means of the `CHACO` software package [27].
We considered a scatternet of $N = 100$ nodes. The matrix Δ was generated as a

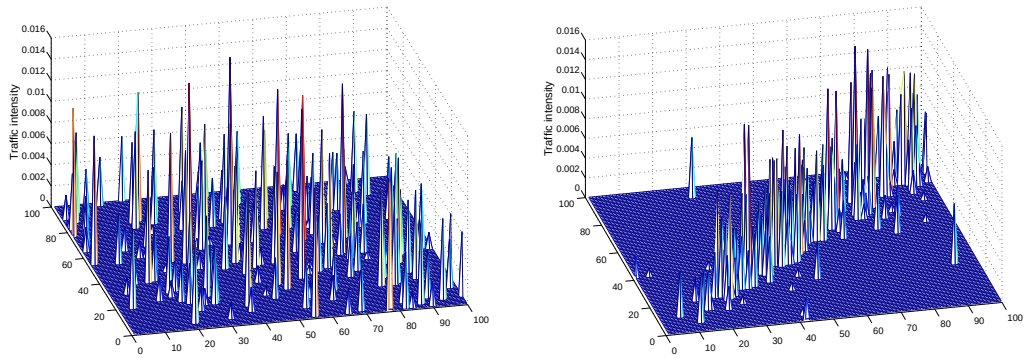
[6] In reality, a smarter criterion would aim at having similar load in the various piconets. This is an issue we are currently investigating.

symmetric matrix with 200 random non-zero entries. The skeleton of the initial matrix Δ is reported in the left-hand side of Figure 11, the right-hand side showing the matrix obtained by partitioning Δ into 15 sets. The same matrices are visualized in Figure 12, where it can be noticed that not only most of the non-zero entries are moved closer to the main diagonal, but also the largest entries are grouped along the diagonal. Roughly speaking, the distance of an entry from the main diagonal is proportional to the number of hops that traffic will have to undergo, so that having non-zero entries grouped along the main diagonal is beneficial in that it minimizes the number of traffic flows forwarded in a multihop fashion. For similar reasons, the largest entries of the traffic matrix should be moved as close as possible to the main diagonal, in order to limit the actual amount of traffic to be forwarded by gateway units.



(a) Skeleton of Δ (b) Skeleton of Δ after partition

Fig. 11. Skeleton of the original and partitioned traffic matrices, results obtained with the multilevel Fiduccia-Mattheyses algorithm, number of piconets $M = 15$, graph coarsened down to 50 vertices.

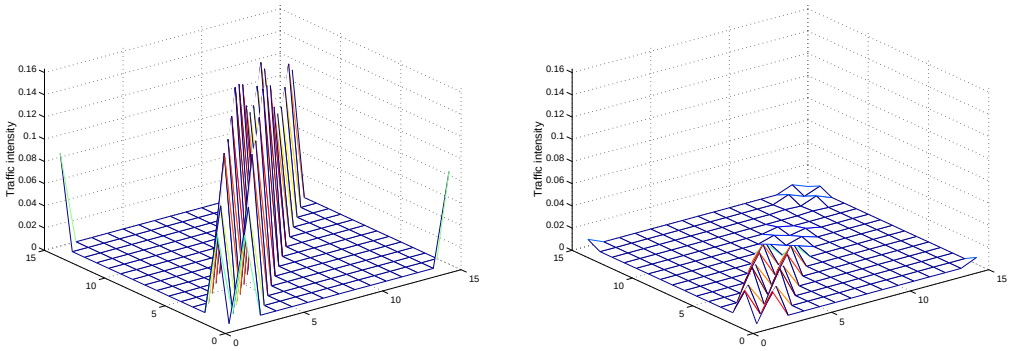


(a) Original matrix Δ (b) Matrix Δ after partition

Fig. 12. Original and partitioned traffic matrices, results obtained with the multilevel Fiduccia-Mattheyses algorithm, number of piconets $M = 15$, graph coarsened down to 50 vertices.

In order to check the effect on the load sustained by the gateways, we can consider

the inter-piconet traffic. Namely, for a given matrix Δ (original or after partition), we can define an $M \times M$ matrix Σ, whose (i, j)-th entry, $\sigma_{i,j}$, denotes the traffic carried from the i-th to the j-th piconet. Clearly, considering ring structures, traffic from the i-th piconet will go only to the $(i + 1)$-th or $(i - 1)$-th piconet. Results are reported in Figure 13, where the matrix Σ is plotted before and after partition. It can be seen that, for the particular traffic matrix considered, the burden on the gateways is reduced by approximately a factor 5.

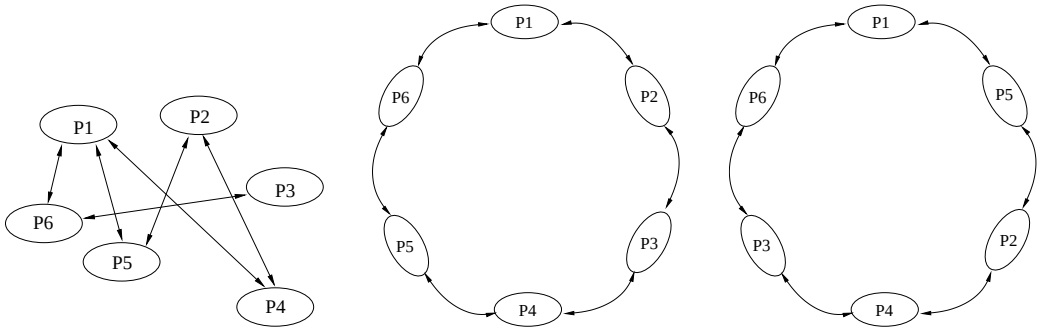


(a) Original matrix Σ (b) Matrix Σ after partition

Fig. 13. Cluster-to-cluster traffic matrix Σ before and after the partition of the matrix Δ.

One further improvement can be done considering the issue described at point (ii). Indeed, it is clear that a smart ordering of the way in which piconets are placed along the logical ring may have a notable impact on network performance. For example, consider the situation illustrated in Figure 14. In the first case, corresponding to a random ordering of the piconets, part of the inter-piconet traffic has to pass through $2, 3$ and even 4 gateways, leading to poor network performance. On the other hand, with the smart configuration reported on the right, only the traffic P1 $\longleftrightarrow$ P4 goes through more than 1 gateway.
The problem can be rephrased in terms of finding an efficient block permutation of the partitioned traffic matrix, in order to move the entries which are out of the clusters closer to the main diagonal. This problem has a well-known equivalent in both circuit design and assignment of tasks in a multiprocessor computer. A technique has been developed, called *terminal propagation* which allows to integrate this task directly into the graph partition algorithm [28]. In this case, the first two issues (i) and (ii) are jointly solved, leading to performance which are clearly better than those which could have been obtained by separately solving the two problems in sequence. Referring the interested reader to the specific bibliography, we will just stress that the incorporation of such metric into the multilevel Fiduccia-Mattheyses algorithm is straightforward and does not modify the algorithm complexity.

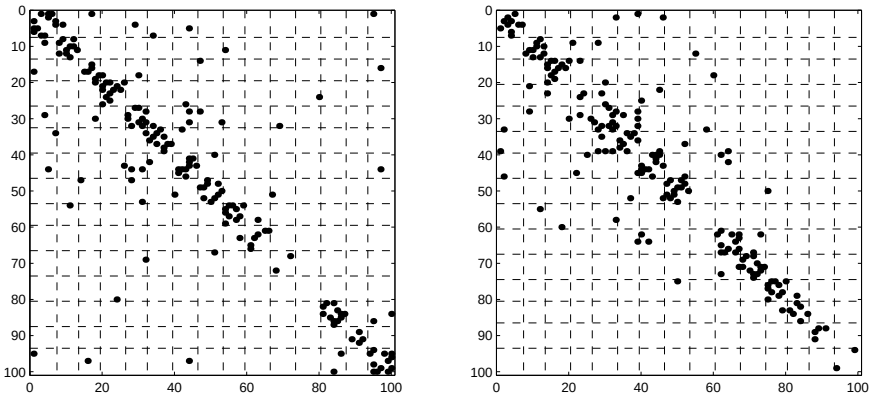
In order to visualize the beneficial effect of terminal propagation, we reported in Figure 15 the partition obtained without and with terminal propagation. Further, we plotted in Figure 16 the cluster-to-cluster matrix Σ. From the latter, it may be noticed that the application of the terminal propagation technique leads to a



(a) Traffic flows (b) Random placement (c) Smart placement

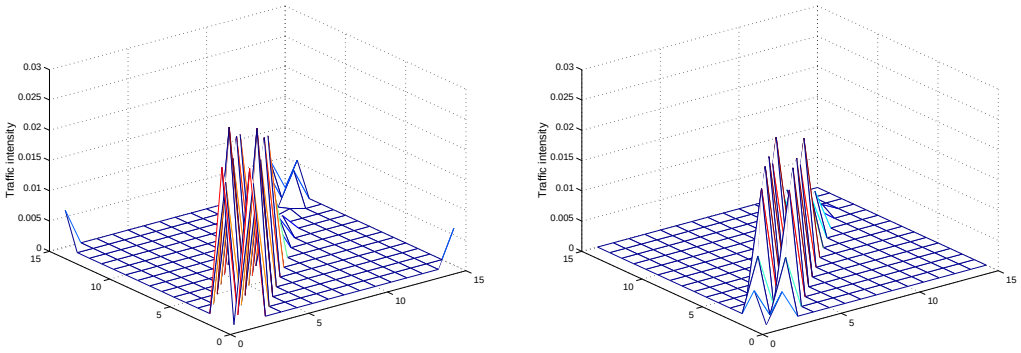
Fig. 14. Example: random and smart placement of the piconets over the logical ring.

lowering of the inter-piconet traffic of around 40%.



(a) Without terminal propagation (b) With terminal propagation

Fig. 15. Skeleton of the partitioned traffic matrix with and without terminal propagation.



(a) Without terminal propagation (b) With terminal propagation

Fig. 16. Cluster-to-cluster traffic matrix with and without terminal propagation.

Now, we turn our attention to issues (iii) and (iv), regarding the choice of the gateway and master devices. For the latter problem, we resort to the heuristic discussed in reference [21], which could be summarized as follows: "Given a piconet, the master is the node which generates/receives the largest amount of traffic". As far as the choice of the gateway units is concerned, many heuristics can be developed. It is worth remarking that, even after the application of our algorithms for a smart solution of the problems (i) and (ii), the gateways may still be considered to represent the bottlenecks of the network. Hence, it seems reasonable to try to maximize the bandwidth available to the multihop traffic on such bottlenecks. However, the incorporation of such a metric into the partition algorithm does not seem straightforward. Here, we propose two heuristics, whose effectiveness is under investigation.

The first simple heuristic considers a set of N nodes, and fixes the number M of piconets (this can be done according to the results presented in Section 5). Then, the M slaves that generate/receive the smallest amount of traffic, are extracted from the traffic matrix, obtaining a graph $\tilde{\mathcal{G}}$. We apply the partition algorithm to $\tilde{\mathcal{G}}$ generating the M piconets. The last step consists in the assignment of the gateways to the piconets. To this end we may proceed as follows. Consider the cluster-to-cluster matrix $\tilde{\Sigma}$.[7] Then, order the cluster-to-cluster links according to the amount of traffic that needs to pass through them, in a decreasing order. Accordingly, rank the gateways according to the amount of traffic they generate/receive, in an increasing order. Finally, assign the first gateway to the first cluster-to-cluster link, and proceed accordingly.

A second simple heuristic for choosing a gateway node can be based on the selection of the node that is less loaded by local traffic. In order to exploit the traffic locality, we first partition the graph, and set the size of each partition (piconet) to be less than or equal to 7. Now, for each partition we set as gateway the node which generates/receives the minimum amount of local traffic. Since a reordering of the nodes inside a block does not modify the optimality of the partition itself, we can assign, without any loss of generality, the lowest index inside a block to the node that has been chosen as gateway to the "previous" piconet. In Figure 17 the resulting matrix structures is shown with the solid squares on the diagonal grouping the nodes of a cluster, and the dashed line showing the resulting piconets. The shaded areas cover the entries used in the computation of local traffic load.

While such heuristics are not optimal, we believe that the solutions proposed to the problems (i)-(iv) lead to a substantial performance improvement with respect to random configurations. We are currently investigating, by means of numerical simulations, the impact of such heuristics on the network performance.

[7] Please note that this is not the usual matrix Σ, since we are not considering the traffic generated/received by the gateway units.

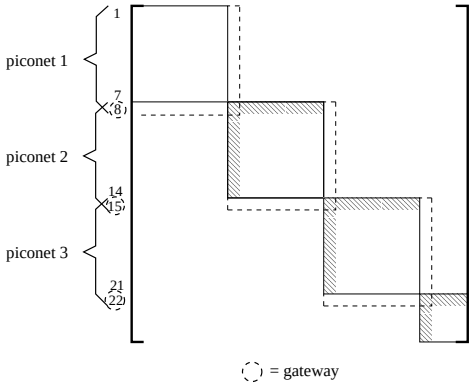


Fig. 17. Δ matrix partitioned for gateways selection.

6.3 Dropping the All-Nodes-in-Range Assumption: an Algorithmic Perspective

Throughout the previous sections, we have always assumed that all nodes were within mutual communication range. While this assumption is a widely used one in the study of scatternet formation/management algorithms [3], it cannot be granted *a priori* for any realistic Bluetooth deployment. In this subsection, we briefly discuss the problems in extending our approach to such case, and overview a possible approach for the extension of the proposed techniques (in particular, the use of graph partitioning algorithms) to case where *most* nodes are in range. [8]

Let us denote by L the adjacency matrix, assumed to be symmetric. The entries of L are binary digits, $L_{i,j} = 1$ indicating that node j is within transmission range of node i. [9] The problem is now to find a mechanism for grouping nodes that belong to the same *logical* cluster (i.e., they transmit a lot among them) as well as to the same *physical* cluster (i.e., they are into mutual communication range). The simplest approach would be to simply perform a entry-by-entry multiplication between L and Δ and perform the graph partitioning on the resulting matrix. We found this approach to behave poorly in most cases (basically, it "weights" too much the spatial constraints with respect to the traffic features). Therefore, we resorted to a different approach. For each node i, let us construct from L the minimum spanning tree rooted in i, and then map all the traffic generated from i on such tree. We repeat the procedure for all i and obtain N matrices; we sum them and perform the graph partitioning on the resulting matrix B. In such way, we are accounting for the spatial constraints (since if $L_{i,j} = 0$ then $B_{i,j} = 0$) while at the same time "weighting" somehow the multihop traffic.

Now we have individuated the clusters, but we cannot ensure them to correspond

[8] This comes from the observation that, if the configuration is mainly driven by the connectivity constraints, the number of degrees of freedom available to system designers are very limited, which restricts the impact of the proposed procedure.

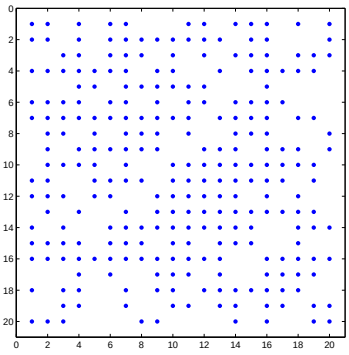
[9] This simplistic model does not account for the fact that different degrees of connectivity may translate into different coding rates/packet types, with consequent different throughput on the various links [29].

to feasible piconets, because of the spatial constraints (in principle, they could even not be connected; the problem is more evident in the case of rather sparse adjacency matrices). In this case, we propose to split the non-complete clusters in feasible piconets (see reference [17]); in this procedure, we may need to also decide, for some piconets, on the ID of the master node. (It is worth remarking that one effect of the introduction of connectivity constraints is the impossibility of solving issues $i) - iv)$ separately, introducing the need to perform a joint optimization.) The next step concerns the ordering of the piconets along the logical ring, which has to be jointly accomplished with the choice of the gateway units, in order to satisfy the connectivity constraints. Our approach consists in considering the clusters as "super-nodes", derive the associated multigraph from matrix B and solve the Traveling Salesman Problem (TSP) on it. In this way, we will find an ordering of the piconets along the logical ring that tries to minimize the interpiconet traffic (note that, in principle, we cannot ensure the existence of an Hamiltonian circuit on such multigraph, and hence we could not be able to construct a closed-loop configuration in all situations; however, numerical results suggest that this approach works well if the number of spatial constraints is rather limited). Each link selected for the TSP provides a couple of nodes; the one generating/receiving the smallest amount of traffic is chosen as gateway (see reference [17] for details).

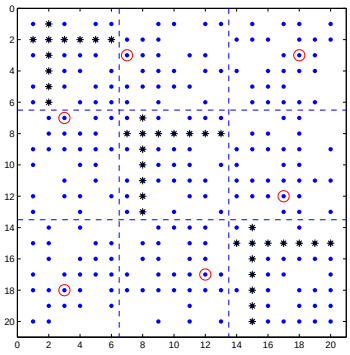
In order to show the performance improvement obtainable with such approach, we considered a bunch of 20 nodes; the adjacency matrix is randomly generated, with an average density of 0.6 ones. The traffic matrix is generated using the same parameters used in the all-nodes-in-range case. The results are plotted in Fig. 18, where we reported, respectively: (a) the initial adjacency matrix; (b) the adjacency matrix re-arranged indicating the obtained piconets, the master units (with a $*$) and the gateway nodes (with a $\circ$); (c) the cluster-to-cluster traffic matrix obtained with a random choice of the clusters (basically, all the feasible topologies are considered and one among them is chosen randomly); (d) the cluster-to-cluster matrix of the obtained configuration. As it may be seen, there is still a significant performance improvement with respect to the random case, but, as expected, the enhancement is less evident than in the all-nodes-in-range case.

7 Conclusions

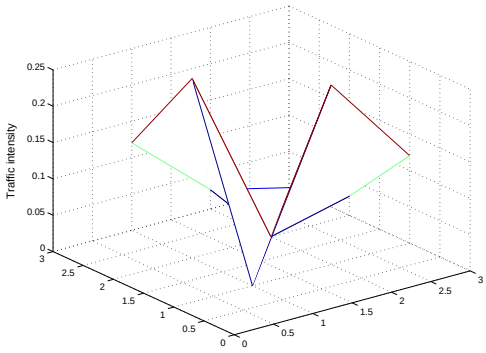
In this paper we have analyzed the impact of the network configuration on the performance of a Bluetooth scatternet.
We have discussed how Bluetooth nodes should be organized to build up a scatternet, where the efficiency of the resulting configuration is measured in terms of network capacity. First, we have presented a theoretical study of the intrinsic capacity limits of a scatternet, which may be approached in the presence of local traffic only. Hence, we have performed a worst-case analysis, by considering uniform end-to-end traffic matrices. We have showed the impact of mutual interference on network



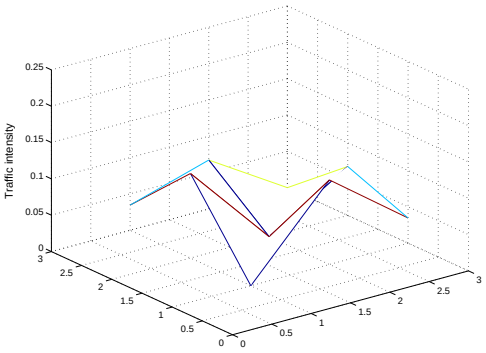
(a) Initial adjacency matrix



(b) Re-arranged adjacency matrix



(c) Cluster-to-cluster matrix after random partitioning



(d) Cluster-to-cluster matrix after "smart" partitioning

Fig. 18. Results in the case of not all the nodes in range, $N = 20$, average density of ones in the adjacency matrix equal to 0.6.

capacity, and we have studied how it affects the optimal configuration choice. We have introduced some regular configurations based upon Platonic solids, which are able to achieve high performance for a low number of nodes.

Closed-loop configurations have shown to possess some interesting features, since they are able to achieve limiting network capacity while at the same time performing well in the presence of a uniform traffic pattern. They are further characterized by ease of protocols implementation and robustness with respect to link failures. As such, we may conclude that they represent a valuable choice as scatternet configuration.

We have then presented some optimization results on closed-loop configurations in the presence of a generic traffic matrix. The proposed procedure, based on graph-theoretical tools, aims at exploiting the traffic locality in order to enhance the network performance. Heuristics for the choice of both master and gateway units have been presented and discussed. Most of the results presented in this part can be applied also to different topology, their scope being not limited to closed-loop configurations.

Some insights on the adaptation of the optimization approaches to the case where not all nodes are in range are also discussed. Some promising directions for future

work seem to be a more in-depth investigation of the case where not all the nodes are in range, as well as the design of a scatternet formation algorithm able to build up an efficient scatternet structure.

References

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

[2] J. Karaoguz, "High–rate wireless personal area networks," *IEEE Comm. Mag.*, pp. 96–102, Dec 2001.

[3] R. M. Whitaker, L. Hodge, and I. Chlamtac, "Bluetooth scatternet formation: a survey," *Ad Hoc Networks*, vol. 3, pp. 403–450, Jul. 2005.

[4] M. Kalia, S. Garg, and R. Shorey, "Scatternet structure and inter-piconet communication in the Bluetooth system," in *Proc. IEEE National Conference on Communications*, New Delhi, 2000.

[5] S. Baatz, C. Bieschke, M. Frank, P. Martini, C. Scholz, and C. Kuehl, "Building efficient Bluetooth scatternet topologies from 1- factor," in *Proc. IASTED WOC*, Banff, Canada, 2002.

[6] S. Baatz, M. Frank, C. Kuhl, P. Martini, and C. Scholz, "Bluetooth scatternets: An enhanced adaptive scheduling scheme," in *Proc. of IEEE INFOCOM*, New York, 2002.

[7] T. Melodia and F. Cuomo, "Ad hoc networking with Bluetooth: key metrics and distributed protocols for scattenet formation," *Ad Hoc Networks*, no. 2, pp. 185–202, 2004.

[8] C. F. Chiasserini, M. A. Marsan, E. Baralis, and P. Garza, "Towards feasible topology formation algorithms for Bluetooth-based WPANs," in *Proc. Hawaii Int. Conf. on System Science*, Big Island, Hawaii, Jan. 2003, pp. 313–322.

[9] K. E. Persson, D. Manivannan, and M. Singhal, "Bluetooth scatternet: criteria, models and classification," *Ad Hoc Networks*, 2004, In press, available at: www.sciencedirect.com.

[10] J. Misic and V. B. Misic, "Bridges of Bluetooth county: topologies, scheduling, and performance," *IEEE Journal on Sel. Areas in Communications*, vol. 21, no. 2, pp. 240–258, Feb. 2003.

[11] S. Pupolin and A. Zanella, "On the impact of fading and direct–piconet interference on Bluetooth performance," in *Proc. of WPMC'02*, Honolulu, Hawaii, Oct. 27–30, 2002.

[12] R. Guérin, J. Rank, S. Sarkar, and E. Vergetis, "Forming connected topologies in Bluetooth ad-hoc networks – an algorithmic perspective," in *Proc. of ITC 18*, Berlin, DE, 2003.

[13] A. Capone, M. Gerla, and R. Kapoor, "Efficient polling schemes for Bluetooth picocells," in *Proc. IEEE ICC*, Helsinki, 2001.

[14] M. Kalia, D. Bansal, and R. Shorey, "Data scheduling and SAR for Bluetooth MAC," in *Proc. IEEE VTC (Spring)*, Tokyo, 2000.

[15] R. Kapoor, A. Zanella, and M. Gerla, "A fair and traffic dependent polling scheme for Bluetooth," in *Proc. of the joint IEEE conferences ICWLHN 2002 and ICN 2002*, Atlanta, USA, 2002.

[16] 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.

[17] D. Miorandi, S. Merlin, A. Trainito, and A. Zanella, "Efficient configurations for bluetooth scatternets," CREATE-NET, Tech. Rep., 2005. [Online]. Available: http://www.create-net.it/∼dmiorandi

[18] F. Baccelli and P. Bremaud, *Elements of Queueing Theory*. Berlin: Springer–Verlag, 1994.

[19] F. Baccelli and S. Foss, "On the saturation rule for the stability of queues," *J.Appl.Prob.*, vol. 32, no. 2, pp. 494–507, 1995.

[20] C. C. Foo and K. Chua, "Bluerings - Bluetooth scatternets with ring structures," in *Proc. IASTED WOC*, Banff, Canada, 2002.

[21] D. Miorandi and A. Zanella, "On the optimal topologies of Bluetooth piconets: roles swapping algorithms," in *Proc. of MedHocNet*, Chia, Italy, 2002.

[22] C. Lin, Y. Tseng, C. Chang, and C. Tu, "Formation routing and maintenance protocols for the BlueRing scatternet of Bluetooth," in *Proc. Hawaii Int. Conf. on System Science*, Big Island, Hawaii, Jan. 2003, pp. 313–322.

[23] A. Pothen, H. Simon, and K. Liou, "Partitioning sparse matrices with eigenvectors of graphs," *SIAM J. Matrix Anal.*, vol. 11, pp. 430–452, 1990.

[24] B. Kerninghan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *Bell. Sys. Tech. J.*, vol. 29, pp. 291–307, 1970.

[25] C. M. Fiduccia and R. M. Mattheyses, "A linear time heuristic for improving network partitions," in *Proc. of IEEE Design Automation Conference*, 1982.

[26] B. Hendrickson and R. Leland, "A multilevel algorithm for partitioning graphs," in *Proc. of ACM Supercomputing*, Barcelona, Spain, 1995.

[27] ——, "The Chaco user's guide, version 2.0." [Online]. Available: http://www.cs.sandia.gov/∼bahendr/chaco.html

[28] B. Hendrickson, R. Leland, and R. V. Driessche, "Enhancing data locality by using terminal propagation," in *Proc. of Hawaii Conf. System Sciences*, 1996.

[29] I. Stojmenovic, A. Nayak, J. Kuruvila, F. Ovalle-Martinez, and E. Villanueva-Pena, "Physical layer impact on the design and performance of routing and broadcasting protocols in ad hoc and sensor networks," *Computer Communications*, vol. 28, pp. 1138–1154, 2005.