Configuring BlueStars: Multihop Scatternet Formation for Bluetooth Networks

Chiara Petrioli, *Member*, *IEEE*, Stefano Basagni, *Member*, *IEEE*, and Imrich Chlamtac, *Fellow*, *IEEE*

Abstract—This paper describes a new protocol for the establishment of multihop ad hoc networks based on Bluetooth devices. The protocol proceeds in three phases: device discovery, partitioning of the network into *Bluetooth piconets*, and interconnection of the piconets into a *connected scatternet*. The protocol has the following desirable properties: It is executed at each node with no prior knowledge of the network topology, thus being fully distributed. The selection of the *Bluetooth masters* is driven by the suitability of a node to be the "best fit" for serving as a master. The generated scatternet is a connected mesh with multiple paths between any pair of nodes, thus achieving robustness. Differently from existing solutions, no extra hardware is required to run the protocol at each node and there is no need for a designated node to start the scatternet formation process. Simulation results are provided which evaluate the impact of the Bluetooth device discovery phase on the performance of the protocol.

Index Terms—Bluetooth technology, scatternet formation, ad hoc networks.

1 Introduction

It is widely anticipated that fourth-generation wireless systems will extensively rely on the unlicensed operations provided by *ad hoc communications* [1]. Allowing spontaneous deployment and self-planning/management, ad hoc networking will play an important role in delivering all kinds of wireless services from the Internet to the very hands of the mobile user.

The Bluetooth (BT) technology, as described in the Specifications of the Bluetooth System Version 1.1 [2], is expected to be one of the most promising enabling technology for ad hoc networks. Originally introduced as short-range cable replacement, the BT specifications define ways for which each BT device[1] can set up multiple connections with neighboring devices so that communication can be established in a multihop fashion. In this sense, Bluetooth devices spread in a geographic area can provide the missing wireless extension to the various heterogeneous network infrastructures, allowing a more pervasive wireless Internet access.

One of the fundamental problems that needs to be addressed to turn this vision into reality is the design of solutions for self-organizing Bluetooth devices into connected multihop ad hoc networks.

According to the specifications, when two BT nodes that are into each other's communication range want to set up a communication link, one of them must assume the role of *master* of the communication while the other becomes its *slave*. This simple "one-hop" network is called a *piconet* (Fig. 1) and may include several slaves, no more than seven of which can be actively communicating with the master at the same time. Given its star-like topology, in the following we will also refer to a piconet as a *BlueStar*. If a master has more than seven slaves, some slaves have to be "parked." To communicate with a parked slave, a master has to "unpark" it, while possibly parking another slave.

The specifications allow each node to assume multiple roles. A node can be a master in one piconet and a slave in one or more other piconets (Fig. 2) or a slave in multiple piconets (Fig. 3). Devices with multiple roles act as *gateways* to adjacent piconets, thus creating a multihop ad hoc network called a *scatternet*.

The BT specifications describe methods for device discovery and for the participation of a node to multiple piconets. However, solutions for scatternet formation are not provided, thus leaving room for research in this area.

A first broader classification of the solutions proposed so far in the literature distinguishes between scatternet formation protocols that require the radio vicinity of *all* nodes (*single-hop* topologies) and protocols that work in the more general *multihop* scenario. All the solutions are distributed in the sense that the protocols are executed at each node with the sole knowledge of its immediate neighbors (nodes in its transmission range).

Solutions of the first kind are presented in [3], [4], and [5]. The solution proposed in [3] is based on a leader election process to collect topology information. Then, a centralized algorithm is run at the leader to assign the roles to the network nodes. In order to achieve desirable scatternet properties, the centralized scheme executed by the leader requires that the number of network nodes is ≤ 36. The scatternet formation protocols presented in [4] and [5] run over single-hop topologies with no limitations on the number of nodes. However, the resulting scatternet is a tree, which limits efficiency and robustness.

1. From now on, the terms node and device will be used interchangeably.

- *C. Petrioli is with the Dipartimento di Informatica, Università degli Studi di Roma "La Sapienza," Rome, Italy.E-mail: petrioli@dsi.uniroma1.it.*
- *S. Basagni is with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115. E-mail: basagni@ece.neu.edu.*
- *I. Chlamtac is with the Department of Computer Science, University of Texas at Dallas, PO Box 830688, EC 33, Richardson, TX 75083. E-mail: chlamtac@utdallas.edu.*

Manuscript received 5 Nov. 2001; revised 9 Aug. 2002; accepted 30 Oct. 2002. For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number 117662.

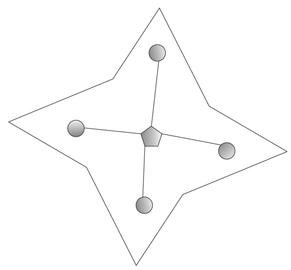

Fig. 1. A BlueStar with one master (pentagon shape) and four slaves.

Among the solutions that apply to the more general case of multihop topologies, the scatternet formation protocol described in [6] requires that the protocol be initiated by a designated node (the *blueroot*) and generates a tree-like scatternet. The blueroot starts the formation procedure by acquiring as slaves its one hop neighbors. These, in turn, start paging their own neighbors (those nodes that are two hops from the root) and so on, in a "wave expansion" fashion, till the whole tree is constructed. To the best of the authors' knowledge, the only solutions for scatternet formation in multihop BT networks that produce topologies different from a tree are those presented in [7], [8], and [9]. The main aim of the protocol proposed in [7] and [8] is to build up a connected scatternet in which each piconet has no more than seven slaves (i.e., the maximum number of active slaves that each master may have at the same time). To this purpose, degree reduction techniques are initially applied to the network topology graph so as to reduce the number of wireless links at each node to less than seven. A scatternet formation protocol (which is left unspecified) is then executed on the resulting topology. These techniques require each node to be equipped with additional hardware that provides to the node its current (geographic) location (e.g., a GPS receiver). Beyond being potentially expensive, this solution is not feasible when such extra hardware is not available. The scatternet formation scheme proposed in [9], *BlueNet*, produces a scatternet whose piconets have a bounded number of slaves. After an (unspecified) device discovery phase, some of the nodes enter the page state randomly (they will be masters) trying to invite a bounded number of slaves to join their piconet. During successive phases, nodes that have not selected their role during the first phase try to connect to some already formed piconet. Finally, the master of each piconet instructs its slaves to set outgoing links to neighboring piconets to form a scatternet. The connectivity of the resulting scatternet is not guaranteed

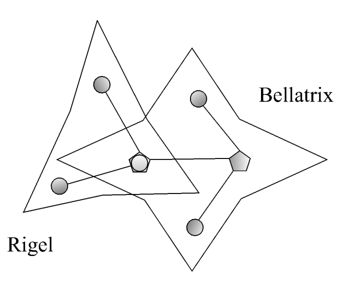

Fig. 2. A device is master in piconet Rigel and slave in piconet Bellatrix.

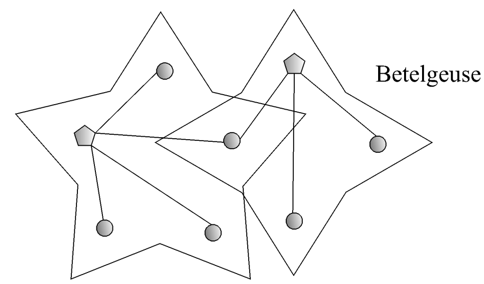

Fig. 3. A device is a slave in piconet Mintaka and a slave also in piconet Betelgeuse.

(i.e., not all the BlueNets are connected, even when the initial topologies are).

The scatternet formation protocol presented in this paper overcomes the limitations of previous solutions. Specifically, our distributed solution has the following features: It works for general multihop BT networks. The generated scatternet is a mesh with multiple paths between any pair of nodes. The selection of the BT masters is driven by the suitability of a node to be the "best fit" for serving as a master. The generated scatternet is connected whenever the initial network topology graph is connected. Finally, no extra hardware is required.

The protocol proceeds in three phases: topology discovery, BlueStars (i.e., piconet) formation, and the configuration of the BlueStars into a *BlueConstellation*, i.e., the connected scatternet. In the following, we give a brief description of each phase:

1. The first phase concerns the *discovery of neighboring devices*. Our protocol relies on the mutual, "symmetric" knowledge of neighboring devices, which means that if node v knows node u, u must also know v. The mechanisms provided by the BT specifications for device discovery (*inquiry* procedures) do not lead to the needed symmetric neighbor knowledge. Therefore, we adopt a specifications compliant mechanism by which, by alternating between *inquiry* and *inquiry scan* modes and by establishing temporary piconets, each device discovers neighboring devices and, at the same time, it makes them aware of its presence.
2. The second phase takes care of piconet formation: One master and some slaves set up communication links to form a *BlueStar* (*BlueStars formation* phase). Based on a locally and dynamically computed weight (a number that expresses how suitable that node is for becoming a master), each node decides whether it is going to be a master or a slave. This phase starts at some dynamically selected nodes and terminates with the formation of disjoint piconets, each with one master and possibly multiple slaves.
3. The final phase concerns the selection of *gateway devices* to connect multiple piconets so that the resulting *BlueConstellation* is connected. By using the information gathered during the BlueStars formation phase, each master selects some of its slaves to be gateways to neighboring piconets. The

selection of the gateways is performed so that the obtained scatternet is connected.

The correctness of each single phase of the protocol is shown, hence proving the whole scatternet formation procedure correct.

Simulation results are provided which identify the features of the generated scatternets, as well as the impact of the device discovery phase on the protocol performance.

The rest of the paper is organized as follows: In the next section, we describe the features and procedures of the BT technology that are needed for the description of the protocol. Sections 3, 4, and 5 describe the details of the three phases of the protocol, which include, for each phase, correctness proofs and implementation issues according to the BT specifications. Simulation results are presented in Section 6. Section 7 concludes the paper and discusses further research directions.

2 The Bluetooth System: Basics

In this section, we briefly describe the procedures of the Bluetooth technology that are used in the three phases of our protocol. This section is not intended to provide a detailed description of the Bluetooth system, for which the reader is referred to [2].

Bluetooth operates in the 2.4GHz, unlicensed ISM band. A frequency hopping spread spectrum is adopted to reduce interferences.

In order to establish a connection between two BT nodes, one of them assumes the role of *master* of the communication and the other one becomes its *slave*. This simple "one hop" network is called a *piconet* (Fig. 1) and may include many slaves, no more than seven of which can be active at the same time. All devices in a piconet share the same channel (i.e., a frequency hopping sequence) which is derived from the unique ID and Bluetooth clock of the master.

Communication to and from a device is always performed through the master of the piconet to which it belongs. In particular, a Time-Division Duplex (TDD) scheme is employed for intrapiconet communications: Transmissions occur in pairs of $625\mu s$ slots, the first of which is for master-slave communication and the second for the communication from the polled slave to the master.

A BT device can timeshare among different piconets. In particular, a device can be either the master of one piconet and a slave in other piconets or a slave in multiple piconets. A node with multiple roles acts as a *gateway* between the piconets to which it belongs. Piconets can be interconnected through gateways into a multihop ad hoc network called a *scatternet*.

Piconet formation is performed in two steps: First, devices must become aware of their neighboring nodes (device discovery); then, information must be exchanged to set up a link between a candidate slave and a candidate master (link establishment). According to the current BT specifications, the former step is accomplished by means of the *inquiry* and *inquiry scan* procedures, while the latter requires the *page* and *page scan* procedures.

For device discovery to happen, two neighboring devices have to be in "opposite" modes, namely, one must be the inquirer, the discovering device, and the other device has to be willing to be discovered. These modes are implemented in BT by having the inquirer in inquiry mode and the other device in inquiry scan mode. The inquirer transmits inquiry ID packets asking neighboring devices to identify themselves and to provide synchronization information needed for link establishment at a later time. To minimize the device discovery time, the BT specifications state that ID packets must be very small (i.e., they include only the General Inquiry Access Code, GIAC, and nothing else) and that they must be transmitted over the frequencies of a predefined inquiry/inquiry scan frequency hopping sequence, changing frequencies at a high rate (twice a slot). A device in inquiry scan hops among different frequencies at a very low rate (one frequency every $1.28s$), thus increasing the probability of an handshake on the same frequency of the inquirer. As soon as an ID packet is received at a device in inquiry scan mode, the device computes a backoff interval and starts listening again. Only when an ID packet is received after the backoff phase will the unit in inquiry scan mode send an FHS (Frequency Hop Synchronization) packet containing its identity and synchronization information (its BT clock).

The described inquiry procedures lead to an asymmetric knowledge of two neighboring devices: The inquirer identity is not known at the device that received an inquiry ID packet. After successful reply from the device in inquiry scan mode, instead, the inquirer knows the identity and the clock of the neighbor that just replied. This enables the inquirer v to estimate the frequency hopping sequence used by its neighbor and thus to invite it to join its piconet as a slave. This invitation is accomplished by means of the paging procedures.

In order for two neighboring devices u and v to establish a link, one must be in page mode, e.g., node v, and the other in page scan mode (node u). By definition, the device that goes in page mode is the master. Node v transmits a page ID packet on u's frequencies, containing u's address. When u, which is in page scan, receives such a packet, it immediately acknowledges it. At this point, v transmits to u a FHS packet that bears all the required information for u to synchronize on v's own frequency hopping sequence. Finally, the two devices exchange all the information for setting up a link and a piconet is formed with v as the master and u as its slave.

It may happen that device u, which is in page scan, is already the master of another piconet and that it could host v as one of its slaves. In this case, once a piconet has been established between v and u, with v as the master, the slave u can request a *switch* of role. This situation is explicitly addressed by the BT specifications and it is implemented via exchanging a specific Link Manager Protocol (LMP) packet that instructs the two devices to switch to the frequency hopping sequence of the new master.

To save the energy of BT devices, "low power operation modes" have been included in the specifications which allow BT nodes to "go to sleep" when they are not actively involved in communication. We use this feature to let a master "release" a slave so that the slave can perform protocol related operations in another piconet. Among the several modes provided in the specifications for low power operations, in our protocol, we take advantage of the *park mode*. A slave that has been put in park mode by its master cannot be actively involved in communication with that master. However, parked slaves periodically wake up in predefined beacon slots to listen to their master communication. Unparking of (possibly multiple) devices is achieved by transmitting an LMP unpark Protocol Data Unit (PDU) in the beacon slot. This packet carries the ID of the devices to be unparked and their new active slave addresses. Parked slaves can trigger an unpark LMP PDU by sending explicit requests during preallocated slots (access window). Similarly, active devices can ask to be parked (or they can be parked by their master) by exchanging an LMP park packet with their master.

3 Scatternet Formation I: Topology Discovery

In this section, we describe how each pair of neighboring devices obtain information about the ID and weight of each other by using the procedures of the inquiry and page substates.

The inquiry procedures described in the BT specifications indicate how a device in inquiry mode can trigger a peer device in inquiry scan mode to send its ID and the synchronization information needed for link establishment (see Section 2). However, no indication is given on how to guarantee that neighboring devices are in opposite inquiry modes, which is the needed condition for them to communicate. Furthermore, the inquiry message broadcast by the source does not contain any information about the source itself. Thus, once two neighboring devices complete an inquiry handshake, only the source knows the identity of the device in inquiry scan mode, not vice versa.

To overcome these drawbacks and attain mutual knowledge of ID and weights, we use a mechanism similar to that introduced in [3]. Each device is allowed to alternate between inquiry mode and inquiry scan mode, remaining in each mode for a time selected randomly and uniformly in a predefined time range (details can be found in [10]). The operations while in each of the two modes are those as described in the specifications. When two nodes in opposite inquiry modes handshake, they set up a temporary piconet that lasts only the time necessary to exchange their ID and weight (thus achieving the required mutual knowledge).

The following procedure describes the operations performed at each device v as it enters the topology discovery phase of the protocol.

```
DISCOVERY(v)
T_disc ← ℓ_td
while T_disc > 0
do if RAND(0, 1) < 0.5
      then INQUIRYMODE
         COMPUTE(T_inq)
         INQUIRY(min(T_inq, T_disc))
         INQUIRYSCANMODE
         COMPUTE(T_scan)
         INQUIRYSCAN(min(T_scan, T_disc))
      else INQUIRYSCANMODE
         COMPUTE(T_scan)
         INQUIRYSCAN(min(T_scan, T_disc)),
         INQUIRYMODE
         COMPUTE(T_inq)
         INQUIRY(min(T_inq, T_disc)),
EXIT
```

The generic device v that executes the discovery procedure, sets a timer T_{disc} to a predefined time length of the discovery phase ℓ_{td}. This timer is decremented at each clock tick (namely, T_{disc} keeps track of the remaining time till the end of this phase).

Device v then randomly enters either inquiry or inquiry scan mode and computes the length of the selected phase (T_{inq} or T_{scan}). While in a given mode, device v performs the inquiry procedures as described by the BT specifications. The procedures that implement the inquiry mode (Procedure INQUIRY) or the inquiry scan mode (Procedure INQUIRYSCAN) are executed for the computed time (T_{inq} and T_{scan}, respectively), not to exceed T_{disc}. Upon completion of an inquiry (inquiry scan) phase, if $T_{\text{disc}} > 0$, a device switches to the inquiry scan (inquiry) mode. As mentioned, to allow each pair of neighboring devices to achieve a mutual knowledge of each others ID and weight, our scheme requires that, whenever a device in inquiry (inquiry scan) mode receives (sends) an FHS packet, a temporary piconet is set up by means of a page phase and devices exchange their ID and weight, together with the synchronization information required for further communication. As soon as this information has been successfully communicated, the piconet is disrupted.

The effectiveness of the described mechanism in providing the needed mutual knowledge to pairs of neighboring devices relies on the idea that, by alternating between inquiry and inquiry scan mode and randomly selecting the length of each inquiry (inquiry scan) phase, we have high probability that any pair of neighboring devices will be in opposite mode for a sufficiently long time, thus allowing the devices to discover each other.

Simulation results presented in [10] evaluate the effectiveness of the presented mechanism for device discovery and provide insights on parameter tuning.

3.1 Topology Discovery Correctness

As a result of the device discovery phase, each device v has the list of the IDs, weights, and synchronization information of all the devices it was able to discover within T_{disc}. Only statistical guarantees can be provided on a device being able to become aware of all its neighbors. Given that all devices enter the topology discovery phase in a given time interval (t_0, t_1), $t_1 < T_{\text{disc}}$, the greater the value of T_{disc}, the higher the probability for a device to discover all its neighbors, the longer the discovery phase duration.

In the following, we prove the termination of the topology discovery phase and that, whenever all packets are successfully received, all devices have a symmetric view of its neighbors.[2] This symmetric knowledge of neighboring nodes is the basis for the correctness of the following phases of our protocol.

Proposition 1. *Each device v terminates the execution of the topology discovery phase. If device v has discovered a neighbor u, it knows its ID and weight and so does u of v.*

Proof. The proof of the first part of the claim is straightforward as each device exits this phase after timer T_{disc} has expired. The second part of the claim can be derived from the protocol operations: Whenever a device discovers a neighbor, the two devices create a temporary piconet. The two neighbors then exchange ID and weight and the piconet is torn down. The construction of the piconet and the information exchanged during its brief lifetime guarantee the symmetric knowledge of the piconet nodes. □

4 Scatternet Formation II: BlueStars Formation

In this section, we describe a distributed protocol for grouping the BT devices into piconets. Given that each piconet is formed by one master and a limited number of slaves that form a star-like topology (Fig. 1), we call this phase of the protocol the BlueStars formation phase.

Based on the information gathered in the previous phase, namely, the ID, the weight, and synchronization information of the discovered neighbors, each device performs the protocol locally. The rule followed by each device is the following: A device v decides whether it is going to be a master or a slave depending on the decision made by the neighbors with bigger weight (v's "bigger neighbors"). In particular, v becomes the slave of the first master among its bigger neighbors that has paged it and invited it to join its piconet. In case no bigger neighbors invited v, v itself becomes a master. Once a device has decided its role, it communicates it to all its (smaller) neighbors so that they can also make their own decision.

Let us call the nodes that have the biggest weight in their neighborhood *init devices*. If two nodes have the same weight, the tie can be broken arbitrarily, provided that a total ordering among all network devices is obtained. (Here, ties are broken by using the device's unique ID.) Given the definition of weight (a real number ≥ 0) and the corresponding total ordering, there is always at least an init node. Init nodes are the devices that initiate the BlueStars formation phase. They will be masters. These are the only devices that go in page mode immediately after the device discovery phase. All the other devices go in page scan mode. The init devices are depicted as triangles in Fig. 4. A line between two devices indicates that they are in the transmission range of each other.

The generic device v stays in page scan mode till it has received a page from all its bigger neighbors. As mentioned, in this phase, if at least a bigger neighbor is a master, then v joins the piconet of the first master that pages it. Otherwise, v itself will be a master. In any case, once all the bigger neighbors have paged it, v switches to page mode and pages one by one those neighbors (if any) that are unaware of its role.

2. In case packets are lost or corrupted during transmissions, by the end of the device discovery phase, some devices may have an "asymmetric view" of their neighborhood. This does not affect the correctness of the following two phases of the protocol, provided that timeouts are introduced in those phases to reestablish the symmetry.

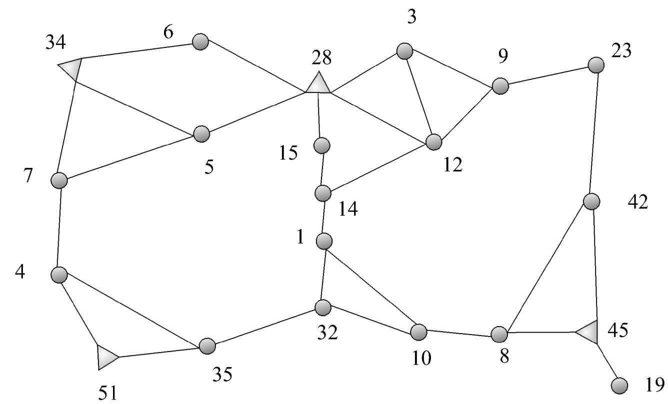

Fig. 4. The init devices.

The protocol operations in this phase are described by the following procedures. The first procedure is executed by every device v as soon as the topology discovery phase is over.

```
INITOPERATIONS(v)
if (INIT(v))
    then PAGEMODE
        for each smaller u
            do PAGE(u, v, 'master', v)
              EXIT
    else PAGESCANMODE
```

The procedure INIT(v) determines whether v is an init node or not. Only the init devices go to page mode and start paging their smaller neighbors one by one. The parameters of the page are the identity of the paged device u, the identity of the paging device v, the role of the latter (either "master" or "slave"), and, in the case where the paging device v is a slave, the identity of the device to which it is affiliating. (In case v is a master, this parameter is irrelevant and can be set to v itself.) All non-init devices go to page scan mode.

The following procedure is triggered at a non-init device v by the reception of a page:

```
ONRECEIVINGPAGE(v, u, role of u, t)
RECORDROLE(u)
if (weight[v] < weight[u])
    then if (role[u] == 'master')
        then if (role[v] == 'none')
            then master[v] = u
                role[v] = 'slave'
            else inform u on master[v]
          if (some bigger neighbor z has to page)
            then WAIT PAGE(v, z, r, w)
            else PAGEMODE
                if (for all bigger u: role[u] == 'slave')
                  then role[v] = 'master'
                    for each smaller z
```

```
14                  do PAGE(z, v, 'master', v)
15                for each bigger slave w
16                  do PAGE(w, v, 'master', v)
17                EXIT
18            else for each slave neighbor z
19                  do k = master[v]
20                    PAGE(z, v, 'slave', k)
21                    PAGESCANMODE
22    else if (some smaller neighbor z has to page)
23          then WAIT PAGE(v, z, r, w)
24          else EXIT
```

The procedure of recording the role of a device u (line 1) includes all the information of synchronization, addressing, etc., that enable v to establish a communication with u at a later time, if needed. In addition, if node u is a slave, the identity of u's master is also recorded.

Upon receiving a page from a device u, device v starts checking if this is a page from a bigger neighbor or from a smaller one. (Pages from smaller neighbors are needed for gathering information used later in the gateway selection procedure.) In case the page is from a bigger neighbor u, v checks if u is a master. If so, and v is not part of any piconet yet, it joins device u's piconet. If, instead, device v has already joined a piconet, it informs device u about this, also communicating the ID of its master. Device v then proceeds to check if all its bigger neighbors have paged it. If this is not the case, it keeps waiting for another page (exiting the execution of the procedure).

When successfully paged by all its bigger neighbors, device v knows whether it has already joined the piconet of a bigger master or not. In the first case, device v is the slave of the bigger master that paged it first. In the latter case, device v itself is going to be a master. In any case, device v goes to page mode and communicates its decision first to all its smaller neighbors and then also to its bigger neighbors that are slaves.

At this point, a master v exits the execution of this phase of the protocol. If device v is a slave, it returns to page mode and waits for pages from all its smaller neighbors. Indeed, some of a slave's smaller neighbors may not have decided their role at the time they are paged by the bigger slave. As soon as a device makes a decision on its role, it therefore pages its bigger neighbors and communicates whether it is a master or a slave, along with its master ID (if it is a slave). This exchange of information is necessary to implement the following phase of gateway selection for obtaining a connected scatternet (see Section 5).

Notice that the outermost else is executed only by a slave node since, once it has paged all its neighbors, a master has a complete knowledge of its neighbors' role and of the ID of their masters and thus it can quit the execution of this phase of the protocol.

The functioning of the BlueStars formation phase is illustrated by the following example:

Consider the BT network depicted in Fig. 4, where a link between two devices indicates that the two nodes have discovered each other during the topology discovery phase. Beside each node is indicated its weight (for the sake of readability, we have omitted the devices' unique ID, assuming all devices have different weights; this allows

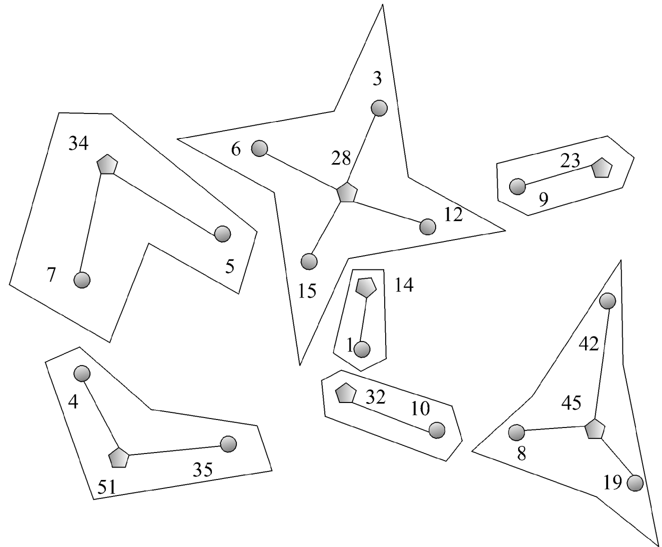

Fig. 5. Outcome of the BlueStars formation phase executed on the BT network of Fig. 4.

us to identify each device with its weight). At the beginning of the BlueStars formation phase, all devices execute the procedure INITOPERATIONS. Given that they are the devices with the bigger weight in their neighborhood, only devices 51, 45, 34, and 28 are init devices (depicted as triangles in the figure). They go to page mode and start paging their neighboring devices. All the other nodes go in page scan mode. Device 51 will successfully page devices 4 and 35, which will become slaves in the resulting "piconet 51" (we follow the BT use of identifying a piconet with its master). Piconet 45 is formed by its master, device 45, and all its neighbors: devices 8, 19, and 42. Master 34 successfully pages devices 5 and 7, which become the two slaves of its piconet. Device 6, a neighbor of master 34, has joined piconet 28, given that master 28 paged it before 34 did. Device 3, 12, and 15 also join piconet 28. At this point, the four init devices quit the execution of this phase of the protocol. In piconet 45, slave 42 has been paged by all its bigger neighbors. It switches to page mode and starts paging its smaller neighbors, namely, devices 8 and 23. Upon receiving a page from device 42 stating that it is a slave of master 45, device 23 decides to be a master itself (all its bigger neighbors have communicated that they are slaves) and pages its smaller neighbor 9 which joins piconet 23 as a slave. Similarly, device 14, "released" by device 15 which joined piconet 28, can now decide to be a master. It then pages nodes 1 and 12 (which already joined piconet 28), gaining node 1 as a slave in its piconet. Piconet 32 is formed similarly, after slave 35 communicated to device 32 that it joined master 51. Of all device 32's smaller neighbors (nodes 1 and 10), only device 10 will be its slave since device 1 already joined piconet 14. Of the 21 devices of the network, seven are masters (four of which are init devices) and all the other devices are slaves to one of those masters. The results of the BlueStars formation phase are displayed in Fig. 5 (masters are depicted as pentagons).

4.1 BlueStars Formation Correctness

All devices are always able to distinguish between a page sent by a device in phase one (where paging is used to set up a temporary piconet to achieve symmetric topology knowledge) and a page of the second phase based on the parameter role. In the topology discovery phase, the parameter role is always set to "none," whereas, in the BlueStars formation phase it is always either "master" or "slave" since devices send pages only after having decided about their role.

This said, the correctness of the BlueStars formation phase, is based on the following facts:

1. Each device terminates the execution of this phase of the protocol.
2. Upon termination each device v:
   a. Is either a master or a slave.
   b. Belongs to only one piconet.

We start by proving the following result.

Proposition 2. *Each device v decides which role to assume (either master or slave) and communicates it to its neighbors. Device v communicates its role only after all its bigger neighbors have communicated their role to it.*

Proof. We assume that there are no transmission errors, i.e., that all attempted pages are successful and lead to the exchange of all the information needed for the formation of a piconet.

Let B be the set of all Bluetooth devices, ($|B| = n$), and let $I \subseteq B$ be the set of the init devices. We enumerate the devices in B according to the following one-to-one correspondence $\mathrm{big} : B \rightarrow \{1, \dots n\}$, defined as follows:

$$\mathrm{big}(x) = \begin{cases} |I| - \mathrm{ord}_I(x) + 1 & \text{if } x \in I; \\ n - \mathrm{ord}_{B \setminus I}(x) + 1 & \text{if } x \in B \setminus I, \end{cases}$$

where $\mathrm{ord}_S : S \rightarrow \{1, \dots |S|\}$ is defined so that that $\mathrm{ord}_S(x) = i$ if x is the ith *order statistics* in the set S according to the devices' weight (as customary, ties are broken by using the devices' unique ID) [11]. Function big orders the devices in such a way that the init devices come first, sorted in decreasing order with respect to their weight, and all the other devices come after, sorted in decreasing order according to their weight.

We now proceed by induction on $\mathrm{big}(x) = k \leq n$, $x \in B$, i.e., the number of consecutive (with respect to big) devices.

The induction base case is comprised of the init devices, i.e., those devices i such that $\mathrm{big}(i) \leq |I|$. Since there is always at least an init device, the set I is never empty. The proof of the base case is based on the code of Procedure INITOPERATIONS: Having no bigger devices, an init node immediately decides its role, goes to page mode, and communicates its role to its neighbors (which are in page scan mode).

Let us now assume that all nodes x such that $\mathrm{big}(x) \leq h$, $h > |I|$, have decided their role and have communicated it after all their bigger neighbors did.

Consider the device y such that $\mathrm{big}(y) = h + 1$. By definition of big, all y's bigger neighbors z are such that $\mathrm{big}(z) < \mathrm{big}(y)$ (bigger neighbors come first).

By inductive hypothesis, every such neighbor z has decided its role and, as stated by the code of Procedure ONRECEIVINGPAGE, it has paged each of its smaller neighbors (which include y) to communicate its role and it is in page scan mode. This implies that device y has received a page from every bigger neighbor and that only one of the following cases holds: Either 1) all its bigger neighbors are slaves or 2) there is at least a master that paged y. In case 1), device y decides to be a master, goes to page mode, and communicates its role to all its smaller neighbors and to all its bigger neighbors that are slaves, as described in Procedure ONRECEIVINGPAGE. In case 2), device y, affiliated with the master that paged it first (i.e., it decides to be a slave), goes to page mode and communicates to all the smaller neighbors and to all its bigger neighbors that are slaves that it is a slave along with the ID of its master. The smaller devices are either in the topology discovery phase (which will be exited in finite time) or in page scan mode since they are not allowed to switch to page mode till every bigger neighbor pages them so that the role and master ID will be successfully communicated. The bigger neighbors that are slaves are in page scan mode by inductive hypothesis. □

A useful corollary of Proposition 2 is stated by the following result.

Corollary 1. *Every slave receives information about the ID, role, and the ID of the master of all its smaller neighbors.*

We are finally able to state the correctness of the BlueStars formation phase.

Proposition 3. *Each device terminates the execution of the BlueStars formation phase of the protocol having being assigned either the role of master or the role of slave in one single piconet.*

Proof. Each master device terminates the execution of the BlueStars formation phase as soon as it has made its decision and it has paged all its neighbors about it (Proposition 2). It is not possible for a master device to reenter the execution of this phase of the protocol and to assume another role.

As soon as a slave has communicated its decision, it is no longer able to execute the **then** branch of the "**if** (weight[v] < weight[u])" command (line 2 of Procedure ONRECEIVINGPAGE). This means that it cannot change its role. Termination, in this case, derives from Proposition 2 and Corollary 1. □

Another useful property of the BlueStars formation phase is stated by the following proposition.

Proposition 4. *Upon exiting the execution of the BlueStars formation phase, a device v knows the identity, the role, and, if slaves, the ID of the master of all its neighbors.*

Proof. Derives immediately from the code of the procedures that implement this phase of the protocol and from the previous propositions. □

4.2 Implementation According to the Bluetooth Technology

The protocol operations of this phase all rely on the standard Bluetooth page procedures. However, the page and page scan procedures used here assume the possibility of exchanging additional information, such as the device role and, for slaves, the ID of their masters. This information cannot be included in the FHS packet which is the packet exchanged in the standard page procedures.

Our proposal is to add an LMP protocol data unit (PDU), including fields to record the role of the sending device and the ID of its master, to easily exchange the information needed for scatternet formation while possibly avoiding a complete set up of the piconet.

Of course, whenever a slave joins a nontemporary piconet, a complete piconet set up has to be performed, after which the slave is put in park mode to allow it to proceed with the protocol operation (e.g., performing paging itself when needed).

5 Scatternet Formation III: Configuring BlueStars—The BlueConstellation

The purpose of the third phase of our protocol is to interconnect neighboring BlueStars by selecting interpiconet gateway devices so that the resulting scatternet, a *BlueConstellation*, is connected.[3] The main tasks accomplished by this phase of the protocol are thus gateway selection and gateway interconnection.

We start by introducing some definitions.

Two masters are said to be *neighboring masters* (*mNeighbors*, for short) if there is either a two-hop path between them, with the intermediate node being a slave of one of them (*gateway slave*), or there is a three-hop path going through two of their slaves (called *intermediate gateways*). For instance, in Fig. 4, masters 51 and 34 are mNeighbors since they can be interconnected via the two intermediate gateways 4 and 7.

A master is said to be an *init master*, or simply an *iMaster*, if it has the biggest weight among all its mNeighbors. Therefore, the set of masters that results from the BlueStars formation phase is partitioned into two sets, the iMasters and the non-iMasters devices. Referring again to Fig. 4, the iMasters are masters 51 and 45. The remaining 5 masters are non-iMasters.

The connectivity of the scatternet is guaranteed by the following result, first proven in [12].

Theorem 1 (Theorem 1 [12]). *Given the piconets resulting from the BlueStars formation phase, a BlueConstellation (a connected BT scatternet) is guaranteed to arise if each master establishes multihop connections to all its mNeighbors. These connections are all needed to ensure that the resulting scatternet is connected in the sense that, if any of them is missing, the scatternet may be not connected.*

Theorem 1 provides us with a criterion for selecting gateways that ensures the connectivity of the resulting scatternet: For every two masters, their slaves in the two and three-hop paths between them will be gateways. If there is more than one interconnection path between the same two masters (as between piconets 28 and 23 in Fig. 4), they might decide to keep only one gateway slave or one pair of intermediate gateways between them or to maintain multiple interconnections. (In the following, we assume the former rule for interconnecting masters.)

Gateway selection is performed locally at each master based on the information gathered by the end of the BlueStars formation, namely, 1) the ID and weight of all its own slaves, 2) the ID, weight, and the ID of the masters of its slaves' one-hop neighbors (gathered via its own slaves), and 3) the ID, weight, and the ID of the masters of its neighboring slaves that did not join its piconet. Pairs of neighboring masters adopt consistent rules of selection. For example, in the case of two-hop mNeighbors, the biggest gateway slave is chosen. In the case of three-hop mNeighbors, the pairs of intermediate gateways the sum of whose weight is the biggest.

5.1 The BlueConstellation: Establishment of a Connected Scatternet

Once the gateway selection has been performed, we are finally able to establish all the connections and the needed new piconets for obtaining a BlueConstellation, i.e., a connected scatternet.

This phase is initiated by all masters v by executing the following procedure.

```
MINITOPERATIONS(v)
1  if (MINIT(v))
2     then for each gateway u
3            do INSTRUCTPAGE(u)
4          PAGEMODE
5          for each selected gateway u: master[u] != v
6            do PAGE(u, v, 'master', v)
7          EXIT
8     else for each gateway to bigger mNeighbors u
9            do INSTRUCTPAGESCAN(u)
10         if (BIGGERMNSLAVES(v))
11           then PAGESCANMODE
12                for each gateway u to bigger mNeighbors t
13                  do WAIT PAGE(v, u, 'slave', t)
14         for each gateway u to smaller mNeighbors
15           do INSTRUCTPAGE(u)
16         if (SMALLERMNSLAVES(v))
17           then pAGEMODE
18                for each gateway u to smaller mNeighbors
19                  do PAGE(u, v, 'master', v)
```

Every master v starts by checking (via Procedure MINIT(v), line 1) whether it is an iMaster or not. If it is an iMaster, then, via Procedure INSTRUCTPAGE(u), it instructs each of its gateway slaves and intermediate gateways u to go into page mode and to page (if any):

- The mNeighbors for which u has been selected as gateway slave. In this case, as soon as u has become the master of an mNeighbor t, they perform a switch of roles (Section 2) so that u also becomes a slave in t's piconet. In this case, no new piconet is formed

3. In the following, we assume that there is the physical possibility of obtaining a connected scatternet.

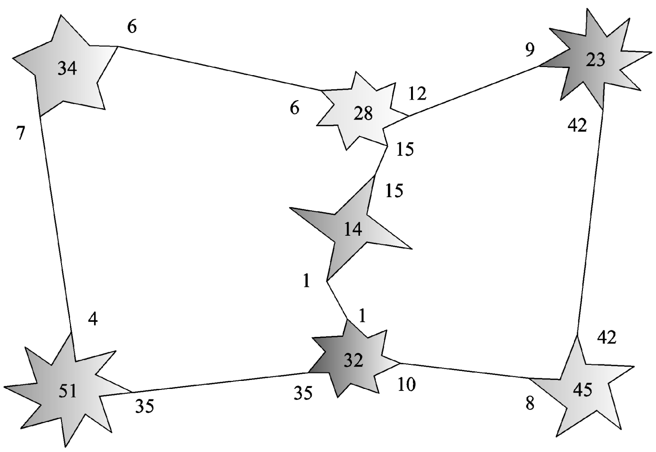

Fig. 6. The BlueConstellation obtained for the network of Fig. 5.

and the slave between v and t is now a slave in both their piconets, as desirable.

- Its peer intermediate gateways selected to interconnect v with its three-hop mNeighbors t. In this case, u becomes also a master of a piconet whose slave is also a slave to t, i.e., a new piconet is created to be the *trait d'union* between the two masters.

The iMaster v itself can then go into page mode (line 4 in the previous procedure) to recruit into its piconet some of those neighboring nodes (if any) that joined some other piconets so that these nodes can be the gateways to their original masters.

Notice that, given the knowledge that every master has about its "mNeighborhood," an iMaster v instructs each of its gateways u about exactly who to page and the resulting new piconet composition. If, for instance, u is gateway to multiple piconets, v knows exactly to which of the neighboring piconets u is going to be also a slave and if it has to be master of a piconet that can have, in turn, multiple slaves.

A node v that is not an iMaster uses Procedure INSTRUCTPAGESCAN to instruct all its slaves u that are gateways to bigger mNeighbors to go to page scan mode and to wait for the specified nodes to page them (line 9). Then, v checks if there are gateway slaves of bigger mNeighbors to whom it has to interconnect (the check is performed via the Boolean function BIGGERMNSLAVES). If this is the case, v waits to be paged by them. After the links to bigger mNeighbors have been so established, v starts to set up links to smaller mNeighbors. To this purpose, it acts as if it were an iMaster.

When the gateways of a non-iMaster device v have set up proper connections toward bigger mNeighbors, they will go into page mode and page those of v's two-hop mNeighbors and those of the slaves of v's three-hop mNeighbors with which they have been requested by v to establish a connection.

The "Orion-like" BlueConstellation resulting from this phase when executed on the BlueStars system of Fig. 5 is depicted in Fig. 6. The name of each BlueStar is the name of the corresponding master. The two IDs that label each link indicate the devices that are acting as gateways. The two IDs are the same in the case of gateways slaves, while they are different when the two piconets are joined by intermediate gateways (through a new piconet).

5.2 BlueConstellation Establishment Correctness

The correctness of the BlueConstellation establishment is stated by the following result.

Proposition 5. *Each node terminates the BlueConstellation establishment phase having being assigned the role of master of at most one piconet and the role of slave in multiple piconets. The resulting scatternet is connected.*

Proof. The proof follows the lines of the proof of Proposition 3, where now the nodes ordered by the function big are the masters and the neighborhood relation is the mNeighborhood defined at the beginning of this section. The connectivity is guaranteed by Theorem 1. □

5.3 Implementation According to the Bluetooth Technology

The mechanism described above can be easily implemented by means of the BT standard procedures for parking and unparking devices and those for link establishment. In particular, upon completion of the second phase of the protocol, a slave asks its master to be unparked (by transmitting a request during the slave access window). The master will then proceed to activate (unpark) different groups of slaves and collect from them all the information required for configuring the BlueConstellation. Based on this information, the master will then make a decision on which links to establish to connect with its mNeighbors and will unpark the gateways in groups of seven to inform them of the piconets to which they are gateway. Each gateway will then run the distributed procedure for interconnecting neighboring piconets described above.

6 Simulation Results

In this section, we investigate the impact of the device discovery phase on the protocol performance. (A more thorough performance evaluation can be found in [10].)

Our methodology has been the following: First, we have generated a large number of topologies, called network "visibility graphs," each made up of n Power Class 3 BT nodes scattered randomly and uniformly in a square of size L. A link between two nodes is added if and only if their Euclidean distance is less than their transmission range $r = 10$m. In the simulated scenarios, the number of BT nodes n has been varied in the range 30, 50, 70, 90, and 110, while L has been set to 30m. (This allowed us to test our protocol on increasingly dense networks.)

We have then run the ns2-based BT simulator described in [10] on the generated visibility graphs. This simulator implements all the details of the device discovery phase described earlier. Starting from a visibility graph, the simulator produces a so-called "BT topology." We have run the simulator for different values of the device discovery phase duration T_{disc}. Here, we have chosen $T_{\text{disc}} = 10s$ and $20s$. The time spent (on average) in inquiry and inquiry scan modes has been set to be $1s$.

TABLE 1
Average Degrees in the Visibility and BT Topologies

Number of BT devices	30	50	70	90	110
Avg. degree, vis. graph	7.4	12.6	17.7	20.9	28.0
Avg. degree, $T_{\text{disc}} = 20s$	5.6	8.3	10.2	11.5	12.4
Avg. degree, $T_{\text{disc}} = 10s$	4.7	6.5	7.4	7.9	8.0

Finally, we have run the second and third phases of the protocol on the obtained BT topologies and evaluated the effects of the different choices of T_{disc} on the average number of piconets and on the number of slaves per piconet of the obtained scatternets. For this part of the protocol evaluation, we have used a simulator of BT-based ad hoc networks implemented in C++.

All experiments have been performed on a number of connected topologies that allow us to achieve a confidence level of 95 percent and a precision within 5 percent.

The impact of the device discovery phase is already evident from Table 1, which shows a significant decrease of the network nodal degree (average number of neighbors for each node) of the BT topologies over the visibility graphs. As T_{disc} increases, the degree increases to account for the increased number of neighbors discovered.

As the network density of the visibility graph increases, it gets more and more time consuming to discover all neighbors. However, a small number of neighbors need to be discovered for obtaining connected BT topologies. This is shown in Table 2: Both for $T_{\text{disc}} = 10s$ and $20s$, all the BT topologies are connected in case of moderately dense to dense visibility graphs ($n \geq 50$ nodes). For $n = 30$, which corresponds to a sparse scenario where less than 95 percent of the generated visibility graphs are connected, the discovery of a very high percentage of neighbors is required to maintain connectivity. However, the neighbors discovered in $10s$ are already enough to produce connected BT topologies 95 percent of the time.

Despite the comparable percentage of connected BT topologies at the two different values of T_{disc}, the different number of links in the BT topologies still noticeably affects the protocol performance. We have considered three metrics, namely, the average number of piconets generated by our protocol, the average number of slaves per piconet, and the 99th percentile of the number of slaves per piconet.

Fig. 7 shows the average number of piconets generated by the protocol during the two phases of piconet formation and interconnection and their combined results. We have considered BT topologies at $10s$ and $20s$. The two cases show similar trends. The number of piconets generated in both phases and, hence, in total, increases with the number

TABLE 2
Percentage of Connected BT Topologies

Number of BT devices	30	50	70	90	110
Conn. BT top., $T_{\text{disc}} = 20s$	98	100	100	100	100
Conn. BT top., $T_{\text{disc}} = 10s$	95	100	100	100	100

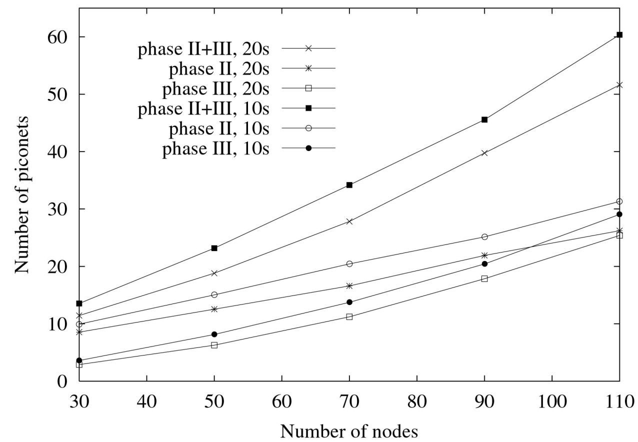

Fig. 7. Average number of piconets per connected topologies.

of nodes. The number of piconets generated in the second phase ranges from 28 percent to 33 percent (23 percent to 28 percent) of the nodes for $T_{\text{disc}} = 10s$ ($T_{\text{disc}} = 20s$). As the number of nodes (and, thus, the number of piconets) increases, a higher number of extra piconets is needed to interconnect neighboring piconets via intermediate gateways. The number of the needed extra piconets falls short of the number of piconets generated in the second phase for highly dense networks. While T_{disc} increases, we observe a general decrease in the number of generated piconets. This is due to the fact that the BT topologies are increasingly denser (more neighbors are discovered), which leads to a more efficient partitioning of the network into piconets (second phase) and to a higher probability that a node between two masters can be selected as a gateway slave.

The average number of slaves per piconet is depicted in Fig. 8, together with the 99th percentile of the number of slaves. In this case, the increased number of links obtained with a longer device discovery leads to a slight increase both in the average number of slaves per piconets and, more importantly, in the size of the "bigger" piconets. However, in all the simulated scenarios, the 99th percentile of the number of slaves per piconet remains below 14, ranging from 7 ($n = 30$, $T_{\text{disc}} = 10s$) to 14 ($n = 110$, $T_{\text{disc}} = 20s$).

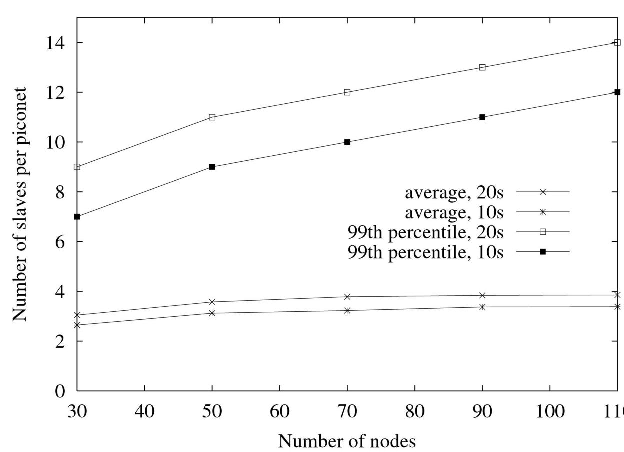

Fig. 8. Average number of slaves per piconet and 99th percentile.

7 Conclusions and Further Work

In this paper, we have presented a protocol for the establishment of a multihop wireless ad hoc network of Bluetooth devices. Our protocol ensures proper local topology discovery, allows devices to self-organize into piconets, and enables the interconnection of the formed piconets into a single connected scatternet. The three phases of the protocol have been described, taking into account the BT technology as described in the last released version of the specifications (version 1.1). The obtained scatternet has multiple paths between each pair of nodes (i.e., the resulting network topology is a mesh) and the protocol operation does not require all the BT devices to be in each other's transmission range. These properties improve upon solutions previously proposed for the problem of BT scatternet formation.

Among the many interesting aspects of the BT technology, especially among those related to its being an enabling technology for ad hoc networks, we are already investigating two main directions that were not taken into explicit consideration in this paper. The first concerns the impact of the device discovery phase on the performance of multihop scatternet formation protocols. Device discovery in Bluetooth is a time demanding operation due to three major factors: 1) the need to adopt (stochastic) mechanisms to have neighboring nodes in opposite inquiry modes, so they can discover each other, 2) the impossibility of identifying the inquirer, which demands the construction of a temporary piconet between neighbors that discovered each other already, and 3) the length of the backoff as stipulated in the BT specifications. Investigating solutions that mitigate the effects of these three factors and lead to a more effective device discovery is the object of current research. Preliminary results, reported in [10], show that, by having the possibility of setting the backoff interval length at one fourth of its current value (i.e., to 512 BT clock ticks) yields remarkable improvements over the results shown in this paper. More precisely, results similar to those shown in Table 2, are obtained by setting $T_{\rm disc} < 4s$.

Another interesting issue is the development of protocols which build connected scatternets whose piconets are guaranteed to have a bounded number of slaves. Modifications to the protocol presented here are being designed to take into account the constraint on the number of active slaves that a master can manage at a time (the "magic" number in this case is seven). This allows us to avoid the use of the parking/unparking mechanism for managing more than seven slaves. A scatternet formation protocol that achieves this goal is described in [13].

Acknowledgments

The authors wish to thank Raffaele Bruno for providing insightful comments. Chiara Petrioli was partially supported by the European Commission under grant IST-2001-34734-EYES. Stefano Basagni was partially supported by the European Commission under grant IST-2001-34734-EYES.

References

[1] W. Kellerer, H.-J. Vögel, and K.-E. Steinberg, "A Communication Gateway for Infratructure Independent 4G Wireless Access," *IEEE Comm. Magazine*, vol. 40, no. 3, pp. 126-131, Mar. 2002.

[2] *Specification of the Bluetooth System, Volume 1, Core, version 1.1*, http://www.bluetooth.com, 22 Feb. 2001.

[3] T. Salonidis, P. Bhagwat, L. Tassiulas, and R. LaMaire, "Distributed Topology Construction of Bluetooth Personal Area Networks," *Proc. IEEE Infocom 2001*, pp. 1577-1586, Apr. 2001.

[4] C. Law, A.K. Mehta, and K.-Y. Siu, "A New Bluetooth Scatternet Formation Protocol," *ACM/Kluwer J. Mobile Networks and Applications (MONET)*, special issue on mobile ad hoc networks, A. Campbell, M. Conti and S. Giordano, eds., vol. 8, no. 5, Oct. 2003, to appear.

[5] G. Tan, A. Miu, J. Guttag, and H. Balakrishnan, "An Efficient Scatternet Formation Algorithm for Dynamic Environment," *Proc. IASTED Comm. and Computer Networks (CCN)*, Nov. 2002.

[6] G. Záruba, S. Basagni, and I. Chlamtac, "Bluetrees—Scatternet Formation to Enable Bluetooth-Based Personal Area Networks," *Proc. IEEE Int'l Conf. Comm. (ICC 2001)*, June 2001.

[7] I. Stojmenovic, "Dominating Set Based Bluetooth Scatternet Formation with Localized Maintenance," *Proc. Workshop Advances in Parallel and Distributed Computational Models*, Apr. 2002.

[8] X. Li and I. Stojmenovic, "Partial Delaunay Triangulation and Degree-Limited Localized Bluetooth Scatternet Formation," *Proc. AD-HOC NetwOrks and Wireless (ADHOC-NOW)*, Sept. 2002.

[9] Z. Wang, R.J. Thomas, and Z. Haas, "BlueNet—A New Scatternet Formation Scheme," *Proc. 35th Hawaii Int'l Conf. System Science (HICSS-35)*, Jan. 2002.

[10] S. Basagni, R. Bruno, and C. Petrioli, "Performance Evaluation of a New Scatternet Formation Protocol for Multi-Hop Bluetooth Networks," *Proc. Fifth Int'l Symp. Personal Wireless Multimedia Comm. (WPMC 2002)*, Oct. 2002.

[11] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, *Introduction to Algorithms*, second ed. Cambridge, Mass. and New York: The MIT Press and McGraw-Hill, 2001.

[12] I. Chlamtac and A. Faragó, "A New Approach to the Design and Analysis of Peer-to-Peer Mobile Networks," *Wireless Networks*, vol. 5, no. 3, pp. 149-156, May 1999.

[13] C. Petrioli and S. Basagni, "Degree-Constrained Multihop Scatternet Formation for Bluetooth Networks," *Proc. IEEE Globecom 2002*, Nov. 2002.

Chiara Petrioli received the Laurea degree with honors in computer science in 1993 and the PhD degree in computer engineering in 1998, both from Rome University "La Sapienza," Italy. She is currently an assistant professor in the Computer Science Department at Rome University "La Sapienza." Her current work focuses on ad hoc and sensor networks, Bluetooth, energy-conserving protocols, and QoS in IP networks. Prior to Rome University, she was a research associate at the Politecnico di Milano and worked with the Italian Space Agency (ASI) and Alenia Spazio. She has published more than two dozen refereed technical papers. She is also a coauthor of book chapters. She is an area editor of the *ACM/Kluwer Wireless Networks* journal, the *Wiley Wireless Communications & Mobile Computing* journal, and of the *Elsevier Ad Hoc Networks* journal. She was also a guest editor of the ACM/Kluwer journal on *Special Topics in Mobile Networking and Applications* (*MONET*) special issue on energy conserving protocols. She has served on the organizing committees and technical program committees of several leading conferences in the area of networking and mobile computing including ACM MobiCom, ACM MobiHoc, and IEEE ICC. She was a Fulbright scholar and is a member of the ACM, IEEE, ACM SIGMOBILE, and IEEE Communications Society.

Stefano Basagni received the PhD degree in electrical engineering from the University of Texas at Dallas (December 2001) and the PhD degree in computer science from the University of Milano, Italy (May 1998). He received the BSc degree in computer science from the University of Pisa, Italy, in 1991. Since Winter 2002, he has been a member of the faculty of the Department of Electrical and Computer Engineering at Northeastern University, Boston. From August 2000 to January 2002, he was a professor of computer science in the Department of Computer Science of the Erik Jonsson School of Engineering and Computer Science, The University of Texas at Dallas. His current research interests concern research and implementation aspects of mobile networks and wireless communications systems, Bluetooth and sensor networking, definition and performance evaluation of network protocols, and theoretical and practical aspects of distributed algorithms. He has published more than two dozen referred technical papers. He is also a coauthor of book chapters. He served as a guest editor of the special issue of the ACM/Kluwer journal on *Special Topics in Mobile Networking and Applications* (*MONET*) on multipoint communication in wireless mobile networks as well as as a guest editor of the special issue on mobile ad hoc networks of the Wiley Interscience's *Wireless Communications & Mobile Networks* journal, for which he serves on the editorial board. He also served and serves on technical program committees, including the ACM/SIGMOBILE MobiCom TPC, and ACM/SIGMOBILE MobiHoc TPC, IEEE Globecom, and IEEE ICC. He is a member of the ACM (including the ACM SIGMOBILE) and of the IEEE (Computer and Communication Societies).

Imrich Chlamtac received the PhD degree in computer science from the University of Minnesota. Since 1997, he has been the Distinguished Chair in Telecommunications at the University of Texas at Dallas. He also holds the titles of the Sackler Professor at Tel Aviv University, Israel, The Bruno Kessler Honorary Professor at the University of Trento, Italy, where he is currently on sabbatical, and University Professor at the Technical University of Budapest, Hungary. He is a fellow of the IEEE and ACM, a Fulbright Scholar, and an IEEE Distinguished Lecturer. He is the winner of the 2001 ACM SIGMOBILE annual award and the IEEE ComSoc TCPC 2002 award for contributions to wireless and mobile networks, and of multiple best paper awards on wireless and optical networks. He has published close to 300 papers in refereed journals and conferences and is the coauthor of the first textbook on *Local Area Networks* (Lexington Books, 1981, 1982, 1984) and of *Mobile and Wireless Networks Protocols and Services* (John Wiley & Sons, 2000)—an *IEEE Network Magazine* 2000 Editor's Choice. He serves as the founding editor-in-chief of the *ACM/URSI/Kluwer Wireless Networks* (*WINET*), the *ACM/Kluwer Mobile Networks and Applications* (*MONET*) journals, and the *SPIE/Kluwer Optical Networks* (*ONM*) magazine.