Multihop Scatternet Formation for Bluetooth Networks

Stefano Basagni
Dept. of Elec. and Computer Engineering
Northeastern University
E-mail: basagni@ece.neu.edu

Chiara Petrioli
Dipartimento di Scienze dell'Informazione
Università degli Studi di Roma, "La Sapienza"
E-mail: petrioli@dsi.uniroma1.it

Abstract- **This paper describes a new protocol for the establishment of multihop ad hoc networks based on Bluetooth devices. The proposed solution is specification compatible, and achieves the following desirable properties, only a few of which are available in previous solutions. The protocol 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 topology (a *scatternet*, according to the Bluetooth terminology) is a connected mesh with multiple paths between any pair of nodes, thus achieving robustness. Differently from existing protocols, the proposed solution does not assume any designated device to start the scatternet formation process, and it is multihop in the precise sense that there is no requirement for each node to be in the transmission range of all the other nodes (one-hop networks).**

I. Introduction

Bluetooth Technology (BT) [1] is emerging as one of the most promising enabling technologies for ad hoc networks. It operates in the 2.4GHz, unlicensed ISM band, and adopts frequency hopping spread spectrum to reduce interferences.

When two BT nodes come into each others communication range, in order to set up a communication link, one of them assumes the role of *master* of the communication and the other becomes its *slave*. This simple "one hop" network is called a *piconet*, referred in the following as a *BlueStar*, and may include many slaves, no more than 7 of which can be active (i.e., actively communicating with the master) at the same time. 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.

All active 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 slave device is always performed through its master.

A BT device can timeshare among different piconets. In particular, a device can be the master of one piconet and a slave in other piconets, or it can be a slave in multiple piconets. Devices with multiple roles will act as gateways to adjacent piconets, resulting in a multihop ad hoc network called a *scatternet*.

Although describing methods for device discovery and for the participation of a node to multiple piconets, the BT specification does not indicate any methods for scatternet formation. The solutions proposed in the literature so far ([2], [3], and [4]), either assume the radio vicinity of all devices ([2] and [4]), or require a designated device to start the scatternet formation process, [3]. Furthermore, the resulting scatternet topology is a tree, which limits the efficiency and robustness of the resulting scatternet.

In this paper we present BlueStars, a new scatternet formation protocol for multi-hop Bluetooth networks, that overcomes the drawbacks of previous solutions in that it is fully distributed, does not require each node to be in the transmission range of each other node and generates a scatternet whose topology is a mesh rather than a tree.

The protocol proceeds in three phases:

1. The first phase, *topology discovery*, concerns the *discovery of neighboring devices*. By the end of this phase, neighboring devices acquire a "symmetric" knowledge of each other.

2. The second phase takes care of BlueStar (piconet) formation. By the end of this phase, the whole network is covered by disjoint piconets.

3. The final phase concerns the selection of *gateway devices* to connect multiple BlueStars so that the resulting *BlueConstellation* is connected.

Lack of space prevents us to present here the correctness of the proposed algorithm. The interested reader is referred to [5].

The rest of the paper is organized as follows. Section II, III and IV describe the three phases of the BlueStars scatternet formation protocol. Conclusions are drawn in Section V.

II. Topology Discovery

The first phase of the protocol, the topology discovery phase, allows each device to become aware of its one hop neighbors' ID and weight. According to the BT specification version 1.1, discovery of unknown devices is performed by means of the *inquiry 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. These ID packets contain just a generic access code, which does not identify the sender. A device in inquiry scan hops among the sequence of inquiry frequencies at a very low rate (once every 2048 slots), 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 the unit in inquiry scan mode will send an

FHS (Frequency Hop Synchronization) packet containing its identity and synchronization information (its BT clock).

We notice that the described inquiry procedures lead to an asymmetric knowledge of two neighboring devices: Once two neighboring devices complete an inquiry handshake, only the inquirer knows the identity of the device in inquiry scan mode, not viceversa. Also, 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.

To overcome these drawbacks and attain mutual knowledge of ID and weights, we use a mechanism similar to that introduced in [2]. Each device is allowed to alternate between inquiry mode and inquiry scan mode. The time spent by each device in a given mode is uniformly distributed in a predefined time range. The operations while in each of the two modes are those as described in the specification.

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

```
Discovery() {
    Set T_disc to the length of the topology discovery phase;
    Enter randomly INQUIRY or inquiry SCAN mode;
    While (T_disc > 0) {
        if (mode == 'INQUIRY') {
            compute the length of the next phase T_inq;
            execute inquiry(min(T_inq, T_disc));
            switch to 'SCAN' mode; }
        else {
            compute the length of the next phase T_scan;
            execute inquiryScan(min(T_scan, T_disc));
            switch to 'INQUIRY' mode; } }
    exit the execution of this phase of the protocol;
}
```

The generic device v that executes the discovery procedure, sets a timer T_{disc}, which 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 next phase (T_{inq} or T_{scan}). While in a given mode, device v performs the inquiry procedures as described by the BT specification. 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 $T_{\mathrm{disc}} > 0$, a device switches to the inquiry scan (inquiry) mode. 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 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.

III. 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, we call this phase of the protocol 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 *init devices* all the devices that have the biggest weight in their neighborhood (see Fig. 1). If two nodes have the same weight, the tie can be broken by using the devices unique ID. Init devices are the devices that initiate the BlueStars formation phase. They will be masters. As soon as the topology discovery phase is over, they go to page mode and start paging their smaller neighbors one by one. All the other devices go in paging scan mode.

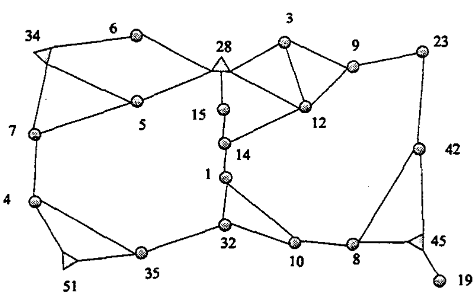

Figure 1 The BT network at the beginning of the BlueStars formation phase. A line between two devices indicates that they have discovered each other. The init devices are depicted as triangles.

The protocol operations in this phase are described by the initOperations() procedure described below.

```
initOperations() {
    if (for each neighbor u: myWeight > uWeight) {
        myRole = 'master';
        go to page mode;
        send page(v, master, v) to all smaller neighbors;
        exit the execution of this phase of the protocol; }
    else
        go to page scan mode;
}
```

The following procedure is triggered at a non-init device v by the reception of a page. The parameter of the page are the identity of the paging device u, its role (either 'master' or 'slave') and, in the case the paging device u is a slave, the identity of the device to which it is affiliating. (In case u is a master this parameter is irrelevant and can be set to u itself.)

```
onReceivingPage(deviceId u, string role, deviceId t) {
record that u has paged;
record role(u);
if (role(u) == 'slave')
    master(u) = t;
if (myWeight < uWeight) {
    if (role(u) == 'master')
        if (myRole == 'none') {
            join u's piconet;
            myMaster = u;
            myRole = 'slave'; }
        else
            inform u about myMaster's ID;
    if (some bigger neighbor has to page yet)
        exit and wait for the following page;
    else {
        switch to page mode;
        if (all bigger devices are slaves) {
            myRole = 'master';
            send page(v, master, v) to each neighbors
            (smaller neighbors first);
            exit the execution of this phase of the protocol; }
        else {
            send page(v, slave, myMaster) to each neighbors;
            switch to page scan mode; } } }
else
    if (all neighbors have paged)
        exit the execution of this phase of the protocol;
    else
        exit and wait for the next page;
}
```

The procedure of recording the role of a device u includes all the information of synchronization, addressing, etc., that enable v to establish a communication with u at a later time, if needed.

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. In the former case, it checks if the sender of the page 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 name 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 to all its smaller neighbors.

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 scan mode and waits for pages from all its smaller neighbors of which it still does not know the role. 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 slaves 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 IV).

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 master 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. 1, where a link between two devices indicates that the two nodes have discovered each other during the topology discovery phase. Aside 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 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 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, 7 are masters, (4 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. 2 (masters are depicted as pentagons).

A. Implementation in the Bluetooth Technology

The protocol operations of this phase all rely on the standard Bluetooth paging procedures. However, the paging and paging scan procedure described above assume the possibility of exchanging additional information, namely, a device role and for slaves, the ID of their masters. These information cannot be included in the FHS packet which is the packet exchanged in the standard paging

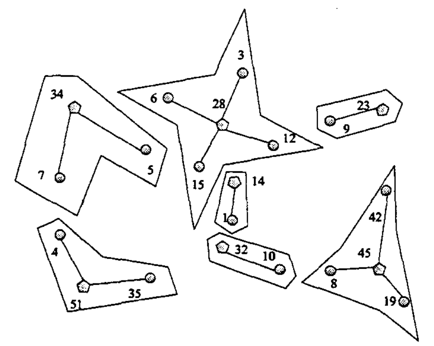

Figure 2 The BlueStars formation phase on the BT network of Fig. 1.

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 non-temporary 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).

IV. CONFIGURING BLUESTARS

The purpose of the third phase of our protocol is to interconnect neighboring BlueStars by selecting inter-piconet gateway devices so that the resulting scatternet, a *BlueConstellation*, is connected whenever physically possible. The main task accomplished by this phase of the protocol is gateway selection and interconnection.

Two masters are said to be *neighboring masters* (*mNeighbors*, for short) if they are at most three hops away, i.e., if the shortest path between them is either a two-hops path (there is only one slave between the two masters) or a three-hops path (there are two slaves). For instance, in Fig. 1, masters 51 and 34 are mNeighbors since there is a shortest path of at most three hops between them.

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. 1, the iMasters are masters 51 and 45. The remaining 5 masters are non-iMasters.

The connectivity of the scatternet is guaranteed by a result, first proven in [6], that states that 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.

This result provides us with a criterion for selecting gateways that ensures the connectivity of the resulting scatternet: all and only the slaves in the two and three-hops paths between two masters will be gateways. If there is more than one gateway device between the same two masters (as between masters 28 and 23 in Fig. 1) they might decide to keep only one gateway between them, or to maintain multiple gateways between them.

Upon completion of the previous phase of the protocol a master v is aware of all its mNeighbors. It directly knows all its neighboring slaves which in turn are aware of (and can communicate to the master v) the ID of their master and of the master of their one-hop slave neighbors.

A. Establishment of a connected scatternet

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() {
if (for each mNeighbor u: myWeight > uWeight) {
    myRole = 'iMaster';
    instruct all gateway slaves about which neighbors to page;
    go to page mode;
    page all the slaves which belong to a different piconet
        and have been selected as interconnecting devices;
    exit the execution of this phase of the protocol; }
else {
    tell all gateway slaves to bigger mNeighbors
        to go to paging scan mode;
    if (there are bigger mNeighbors' slaves in my neighborhood
        which will interconnect the two piconets)
            go to page scan mode;
    tell all gateway to smaller mNeighbors to go to paging mode
        when the links to bigger mNeighbors are established;
    if (there are smaller mNeighbors' slaves in my neighborhood
        which will interconnect the two piconets)
        go to page mode when the links to bigger mNeighbors are
up; }
}
```

Every master v starts by checking whether it is an iMaster or not. If it is an iMaster, then it instructs each of its gateway slaves to go into page mode and to page (if any):

- Its two-hop mNeighbors. In this case, as soon as v's slave has become the master of an mNeighbor u, they perform a switch of roles, as described in the BT specification, so that v's slave become also a slave in u's piconet. In this case, no new piconet is formed and the slave in between u and v is now a slave in both their piconets, as desirable.
- The slaves of its three-hop mNeighbors (that are two-hops away from v). In this case v's slave becomes also a master of a piconet whose slaves are also slaves to the three-hop mNeighbors, 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 paging mode to recruit into its piconet some of those neighboring slaves (if any) that joined some other piconets, so that these slaves can be the gateway to their original masters.

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

When the gateway slaves of a non-iMaster device v have set up proper connections toward bigger mNeighbors, they will go into page mode and page those of its two-hop mNeighbors and of the the slaves of its 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. 2 is depicted in Fig. 3. 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 (here we assume that if there is more that one gateway between neighboring piconets, only one is chosen). The two IDs are the same when there is only a slave between the two masters. They are different when the two piconets are joined by a new piconet.

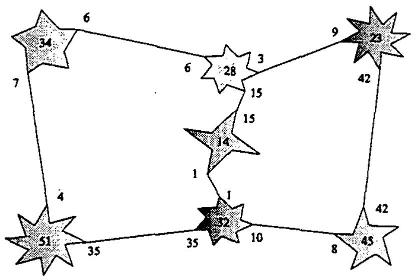

Figure 3 The BlueConstellation obtained for the network of Fig. 2.

B. *Implementation in 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. The master will then proceed activating (unparking) different groups of slaves, and collecting 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 in the previous section, at the end of which it will issue to the master a request for being unparked in order to communicate the list of links successfully established.

V. Conclusions and Further Work

In this paper we have presented a protocol for the establishment of a multihop wireless ad hoc network based on Bluetooth (BT) technology. Starting from a number of scattered BT devices our protocol ensures proper local topology discovery, allows devices to self-organize themselves into piconets and finally enables the interconnection of the formed piconets into a single connected scatternet (provided, of course, that network connectivity is physically achievable).

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 performance of the topology discovery phase. The BT specification really lacks in guidance about an optimal use of the inquiry procedure to produce symmetric knowledge among pairs of devices. Experiments that demonstrate the effectiveness of our proposed solution, taking into account the very details of the BT baseband, including the implementation of timing and the spread spectrum frequency hopping system, have been presented in [7]. Finally, the protocol is 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 7). Although the specification define in details that when a device has discovered more than seven neighboring devices and these all becomes slaves in its piconet it can park those that exceeds 7, it is reasonable to expect that the park/unpark operation will detrimentally affect the scatternet performance. Therefore, variations of the proposed solution are in order to take this constraint into account directly.

References

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

[2] T. Salonidis, P. Bhagwat, L. Tassiulas, and R. LaMaire, "Distributed topology construction of Bluetooth personal area networks," in *Proceedings of the IEEE Infocom 2001*, pp. 1577–1586, April 22–26 2001.

[3] G. Záruba, S. Basagni, and I. Chlamtac, "Bluetrees—scatternet formation to enable Bluetooth-based personal area networks," in *Proceedings of the IEEE International Conference on Communications, ICC2001*, (Helsinki, Finland), June 11–14 2001.

[4] C. Law, A. K. Mehta, and K. Y. Siu, "Performance of a new bluetooth scatternet formation protocol," in *Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc 2001*, Long Beach, CA, 4–5 October 2001.

[5] S. Basagni, I. Chlamtac and C. Petrioli, "Configuring BlueStars: Multihop scatternet formation for Bluetooth networks," TR-02-2001, Università di Roma "La Sapienza," August 2001.

[6] I. Chlamtac and A. Faragó, "A new approach to the design and analysis of peer-to-peer mobile networks," *Wireless Networks*, vol. 5, pp. 149–156, May 1999.

[7] S. Basagni, R. Bruno and C. Petrioli, "Device discovery in Bluetooth networks, a scatternet perspective," to appear in *Proceedings of Networking 2002*, Pisa, Italy, 19–24 May 2002.