Device Discovery in Bluetooth Networks: A Scatternet Perspective

Stefano Basagni
Department of Computer Science
The University of Texas at Dallas
E-mail: basagni@utdallas.edu

Raffaele Bruno
C.N.U.C.E.
C.N.R. Pisa, Italy
E-mail: Raffaele.Bruno@guest.cnuce.cnr.it

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

Abstract

The paper concerns device discovery in multi-hop networks of Bluetooth devices. We start from the observation that forming a Bluetooth *scatternet* (i.e., a multi-hop wireless topology) requires each pair of neighboring nodes to have a "symmetric" knowledge of each other, i.e., if node u knows node v then node v knows node u. We then investigate the use of the Bluetooth procedures for device discovery (*inquiry* procedures) in order to guarantee the needed symmetric knowledge for scatternet formation. Through the use of simulation results we observed that despite the long time required for all nodes to become aware of the presence of their neighbors, the Bluetooth topologies obtained by using the devices discovered after just 6 seconds are connected, their average degree (average number of neighbors of each node) and average route length are consistently close to the corresponding values that we would obtained if all the neighbors of a device were discovered.

Keywords: Bluetooth networks, Device discovery, Scatternet formation.

1 Introduction

Among the new technologies for wireless communication in the unlicensed ISM band (2.4GHz), the *Bluetooth* technology (from now on simply BT) [1] is expected to be the fastest growing. Bluetooth is a low cost, low power, short-range radio technology originally introduced as a cable-replacement technology to connect small devices such as headsets and cell phones, portable computers and printers, etc.

Although introduced for obviating to the need for connectivity via physical wires, the BT technology, as described in the the Specification of the Bluetooth System Version 1.1 [1], comes with several features that makes it one of the most promising technology for enabling *multi-hop wireless networks*. These networks, often termed *ad hoc networks*, are characterized by the absence of any fixed infrastructure. Every ad hoc node must thus be willing to relay packets to destinations that are not in their transmission range.

The way the BT technology is used to form multi-hop topologies like those of ad hoc networks is via forming a *scatternet*. If two BT devices[1] are into each others communication range, i.e., they are *neighbors*, 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*, and may include more than one slave. There is basically no limit on the maximum number of slaves connected to one master, although the number of active slaves at one time cannot exceed 7.

A scatternet is finally formed by joining piconets. The inter-piconet connection is enabled

---

[1] The terms node and device are used interchangeably in this paper.

by the possibility for a BT device to have multiple roles: a node can be a master in one piconet and a slave in one or more other piconets. The devices with multiple roles will act as gateways to adjacent piconets.

In this paper we are concerned about the very first operation of a BT device: The need of discovering neighboring devices. In particular, we are interested in investigating the use of the procedures as described in the BT specification so that each BT device is provided with enough information to set up piconets with neighboring devices and eventually a scatternet.

The BT specification provides the *inquiry procedures* for device discovery. By using these procedures, because of the radio nature of the BT technology, a node is enabled to discover some neighboring devices *without* having these devices know about itself. However, algorithms proposed for scatternet formation, all assume a symmetric" neighbor knowledge" at each node, i.e., they assume that if a BT device u is aware of the presence of a neighboring device v then device v is also aware of the presence of device u.

Examples of protocols for scatternet formation that require this kind of neighbor knowledge can be found in [2], [3] and [4].

The first paper do not consider device discovery. In [3] and in [4] a method for device discovery is proposed which is tailored explicitly for the scatternet formation protocols described in the papers. In [3] a node chooses randomly whether to go in inquiry or in inquiry scan mode. In [4] each device alternates between inquiry and inquiry scan modes. In both cases a leader is elected that finally decides about the final layout of the scatternet.

Both solutions are proven correct provided that each nodes is in the transmission range of every other node ("single-hop" topology). This crucial assumptions allows the device discovery phase to be fast and simple: There is no need for two neighboring devices to discover each other if this does not serve the purpose of the scatternet formation protocols.

In this paper we aim to investigate the use of the BT inquiry procedures for the most general case of multi-hop topologies. The radio vicinity of all nodes is not required. We consider the solution proposed in [4] where each BT device alternates between inquiry and inquiry scan and we measure the effectiveness of this method in providing symmetric neighbor knowledge at each node. Through the use of simulation in networks of up to 60 BT devices, we observed that when no nodes are allowed to exit the device discovery phase because of the radio vicinity of all nodes, the duration of the device discovery phase that (statistically) ensures the knowledge of over the 90% of neighbors is above 18 seconds.

At the same time, we also observed that after only 6 seconds the percentage of neighbors discovered is large enough to obtain connected topologies, i.e., connected scatternets. Moreover, the degree in the "BT network," i.e., the average number of discovered neighbors, is reduced with respect to the original average number of neighbors of each node, thus lowering the number of slaves that a master have to manage.

Finally, we have obtained results related to the average length of routes (shortest paths) in the topology obtained by the interconnection of the sole discovered devices. As for the average degree, whenever we obtain a BT connected topology, the difference between the average route length in the BT network and that of the "original topology" tends rapidly to 0 after a device discovery duration of 6 seconds.

The rest of the paper is organized as follows. Section 2 describe those parts of the BT technology that we used in this papers. Section 3 describe the use of the inquiry and and page procedures that allows at each node the symmetric knowledge of some of its neighbors. In Section 4 we describe the experimental results obtained by simulation and, finally, Section 5 concludes the paper.

2 The Bluetooth System: Basics

In this section we briefly describe those parts of the Bluetooth technology that are used in the rest of this paper. This section is not intended to provide a detailed description of the Bluetooth system, for which the reader is referred to [1].

Bluetooth operates in the 2.4GHz, unlicensed ISM band, a pretty crowded one. Frequency hopping spread spectrum is adopted to reduce interferences.

When two BT nodes are connected (i.e., they exchange data over an established 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*, 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. 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 applied for intra-piconet communications. Transmissions from a master to one of its slaves and viceversa are always coupled: To a master-to-slave transmission always follows a transmission from the polled slave to the 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. Piconets can be interconnected through gateways into a multi-hop ad hoc network called a *scatternet*.

A BT device behavior is described by the state diagram depicted in Figure 1.

A BT device can leave its default state (the Standby state, in which the unit is in low power mode) to scan for inquiry or page messages or to inquiry or page itself. The intervals with which these activities are carried out are discussed in Section 4, Experimental Results.

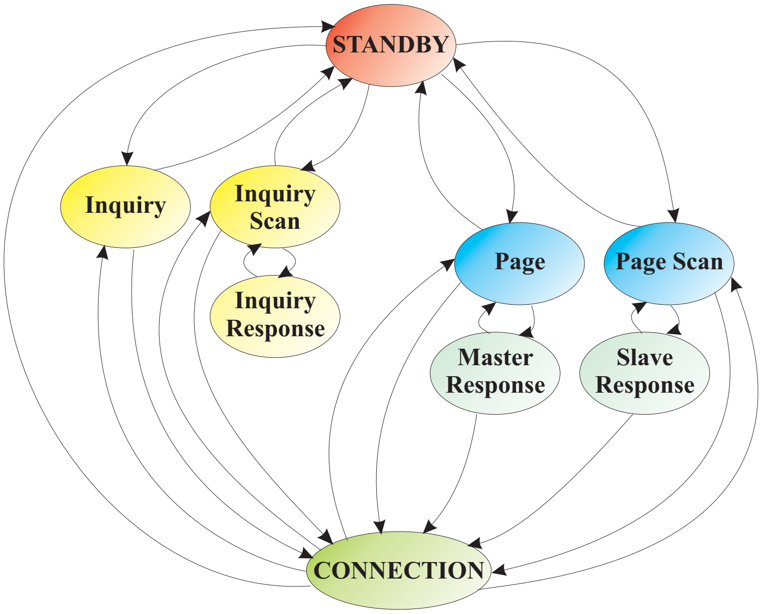

Figure 1: State diagram of a BT device.

The inquiry procedures, which comprise the two "matching" procedures of inquiry and inquiry scan, are used for device discovery. A BT device that want to discover another BT device enters the *inquiry* substate. In this substate, it continuously transmits the *inquiry packet*[2] at different hop frequencies. The inquiry hop sequence is always derived from the general inquiry access code. A device that allows itself to be discovered enters the *inquiry scan* substate to respond to inquiry messages. In particular, the scanning device listens for the inquiry access code long enough to completely scan for 16 inquiry frequencies. These 16 frequencies are chosen among the 32 frequencies of the unique inquiry frequency hopping sequence and constitute the 10ms "A train." After having scanned this train for at least $\mathrm{N}_{\text{inquiry}} = 256$ times, a new train, termed B, can be scanned, which comprises the remaining 16 frequencies. The inquiry response consists of the device in inquiry scan that transmits, after a backoff period necessary to avoid collisions with possible responses from other scan-

---

[2] The inquiry packet is a packet that do not contain any information about the source, but only a general inquiry access code, GIAC.

ning devices, the *frequency hopping sequence*, FHS, packet with its own unique BT address and its BT clock.

Notice that for each pair of neighboring devices u and v for which u discovered v the knowledge gained at each of the two nodes is "asymmetric." The node u (the inquirer) knows device v's access code (obtained from v's BT address) and BT clock. Device v knows nothing about device u.

Once gained information about a neighboring device with which it wants to set up a connection, a device that is going to be a master enter the *page* substate. The master tries to "capture" the slave by repeatedly transmitting the slave device access code (that it obtained by performing the inquiry procedures) in different hop channels. As for the inquiry procedure, the allowed frequencies are 32, split into two 10ms trains A and B with 16 frequencies each. Upon receiving a page from the master, a slave acknowledges it. Once received this acknowledgment the master enters the master response substate and transmits a FHS packet (ID and clock) to the slave which is in turn acknowledged back from the slave to the master. At this time, the two devices have symmetric knowledge of each other, namely master u knows slave v's ID and clock, and slave v knows master u's ID and clock.

The following section describes in details the use of the inquiry and page procedures for obtaining the information needed for scatternet formation.

3 Device Discovery in Bluetooth Networks

The inquiry procedure described in the specification indicates 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 these information to each other. 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 viceversa.

To overcome these drawbacks and attain mutual knowledge for each pair of nodes, we use a mechanism similar to that introduced in [4]. 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 (left unspecified in the BT specification).

The following procedure describes the operations performed at each device as it enters the topology discovery phase of the protocol. In the description of the procedure τ denotes the duration of the device discovery phase, function **rand**() returns either one of its argument with probability 0.5 and the two procedures inquiry and inquiryScan perform the BT inquiry and inquiry scan operation as described in the previous section.

```
Discovery()
begin
    $T_{\text{disc}} = \tau$;
    mode = rand(inquiry, inquiry scan);
    while ($T_{\text{disc}} > 0$)
      begin
        if (mode == inquiry) then
          begin
            compute $T_{\text{w inquiry}}$;
            inquiry(min($T_{\text{w inquiry}}, T_{\text{disc}}$));
            switch to w inquiry scan mode;
          end
        else
          begin
            compute $T_{\text{w inquiry scan}}$;
            inquiryScan(min($T_{\text{w inquiry scan}}, T_{\text{disc}}$));
            switch to inquiry mode;
          end
      end
    exit;
end
```

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 ($T_{\text{w inquiry}}$ or $T_{\text{w inquiry scan}}$). 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 ($T_{\text{w inquiry}}$ and $T_{\text{w inquiry scan}}$, 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. To allow each pair of neighboring devices to achieve a mutual knowledge of each others' ID and clock, 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. The master already knows ID and clock of the slave (through the inquiry phase). Setting up a piconet now ensure that the master send to the slave its FHS (i.e., its ID and clock) to the slave (this is accomplished through the slave and the master going into the slave response and master response substates, respectively). We notice that the temporary piconet set up time is extremely short, given that the two participating devices are already in the proper opposite paging modes (they do not have to find each other: the device in inquiry mode goes in paging mode right away, and the device in inquiry scan mode goes in paging scan mode immediately after inquiry response). Furthermore, the information to be exchanged is extremely short: The ID and clock of each device are included in the FHS packet, which is transmitted in one slot. As soon as this packet has been successfully transmitted 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 (i.e., the values of $T_{\text{w inquiry}}$ and $T_{\text{w inquiry scan}}$), we have high probability that any pair of neighboring devices will be in opposite modes for a sufficiently long time, thus allowing the devices to discover each other.

3.1 Topology discovery correctness

As a result of the device discovery phase each device v has the list of the IDs and synchronization information of all the devices that 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 (i.e., for each pair of neighbors u and v, u knows v's ID and viceversa). This symmetric knowledge of neighboring nodes is the basis for the correct functioning of protocols for scatternet formation, e.g., [2], [4] and [3].

Proposition 1 *Each device v terminates the execution of the device discovery phase. If device u has discovered a neighboring device v, it knows its ID and other synchronization information. Also, if device u discovered device v, then v discovered u.*

Proof. The proof of the first part of the claim is straightforward as each device exits phase one after timer T_{disc} has expired. The second part of the claim can be derived from the protocol operations: whenever a device discovers a neighbors, the two devices create a temporary piconet. The two neighbors then exchange required information 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 Experimental results

We have simulated the BT device discovery methods described in the previous section by using the VINT project Network Simulator ("ns2") [5]. Our original plan was to use BlueHoc [6], the IBM open-source extension to ns2 that implements the baseband and link layer of BT as described in the BT specification [1]. We soon realized that the BT services offered by BlueHoc were too limited for implementing device discovery procedures that are suitable for scatternet formation. Therefore, we have extended BlueHoc to provide the following features:

- The possibility for a node of dynamically choose its role. In BlueHoc a device is assigned either the role of master or the role of slave a priori.
- The possibility for a node of alternating between inquiry and inquiry scan. In BlueHoc a master is only allowed to go (once) in inquiry mode, and a slave is allowed to go (once) in inquiry scan mode.
- Packet collision detection for taking packet losses into account. (This feature is not available in BlueHoc.)
- Determination if two nodes are neighbors based on their transmission radius and on their distance. (This feature is not available in BlueHoc.)

Listed next are the parameters used in our experiment. They are all fully compliant with the BT specification, Version 1.1 [1].

- N_{inquiry}: the number of times that the A "inquiry frequency train" has to be repeated. The specification states that this parameter should be at least 256. In our simulations we used this value.

- $\mathrm{N}_{\mathrm{page}}$: the number of times that the A "page frequency train" has to be repeated. According to the specification, the continuous mode (R0) dictates only that $\mathrm{N}_{\mathrm{page}} \geq 1$ (see [1]). We chose a value of 128, although, for the way we use paging in setting up temporary piconets, we only use the A train twice, and thus this parameter does not affect the efficiency of the paging procedure.

- pageTO: The number of BT clock ticks for the paging phase. We have chosen 64 (two successive trains).

- pagerespTO: The number of BT clock ticks for the master/slave response that follows the paging/paging scan phase. This is a value that the BT specification mandates to be 16 clock ticks.

- $\mathrm{T}_{\mathrm{w\ inquiry}}$ and $\mathrm{T}_{\mathrm{w\ inquiry\ scan}}$: The time that each device spends on inquiry and inquiry scan mode, respectively. We selected these parameters randomly and uniformly in the range $[t_{\mathrm{train}}, t_{in}]$ seconds, where t_{train} is the duration time of a single frequency train, and $t_{in} = 2$ (see also [4]). We have conducted experiments with $t_{in} = 4$ and $t_{in} = 6$ without observing significant variations with respect to the results reported below.

- The backoff period used by a device in inquiry response is mandated by the BT specification to be 2048 clock ticks.

All the simulations in the present section were run on a number of generated topologies large enough to achieve a confidence level of 95% with a precision within 5%.

4.1 Device discovery in multi-hop networks

In what follows we term *original topology* the topology that we would obtain if each device could set up a bidirectional connection with all the devices in its transmission range (its neighbors). The term *BT topology*, instead, indicates the topology obtained by (bidirectionally) connecting only those neighbors that a device was able to discover in a predefined time T_{disc}.

Our first set of experiments concerns the simulation of the device discovery procedure described above in networks of up to 60 BT devices. These networks are multi-hop in the precise sense that the radio vicinity of *all* devices is not required (as it is in the single-hop networks considered in [4] and [3]). The devices are scattered randomly and uniformly in a square area whose side L was chosen large enough to produce connected topologies with high probability. All experiments have been conducted on connected topologies.

Figure 2 shows the percentage of discovered neighbors in at most 20 seconds in networks of 20 to 60 BT devices.

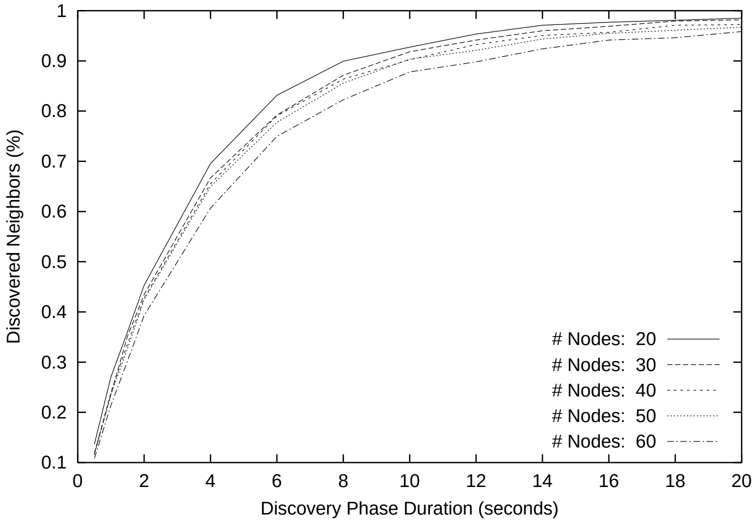

Figure 2: Percentage of discovered neighboring devices.

We observe that the curves are very similar, given the similar average degree (i.e., the average number of neighbors of each node), as listed in Table 1.

We notice that it is not possible, even in 20 seconds, to discover all neighbors.

However, Figure 3, shows that despite the number of device discovered is less than the

Table 1: Area dimension, average degree and average shortest path length

Number of BT devices	20	30	40	50	60
L	24	29	34	38	40
Avg. degree	6.982	7.851	8.058	8.378	9.213
Avg. shortest paths	1.882	2.264	2.666	2.966	3.065

number of possible neighbors in the original topology, when the original topology is connected, then the BT topology is connected as well, i.e., the possibility of obtaining a connected scatternet is not compromised.

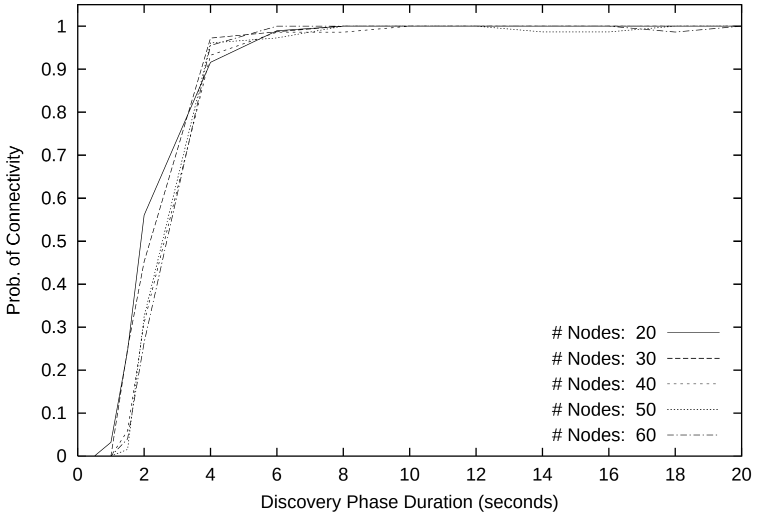

Figure 3: Connected BT topologies.

After 6 seconds the percentage of device discovered allows us to obtain a connected BT topology. Thus, the lower number of discovered devices could actually turn into “a blessing,” since connectivity is preserved and each node that will be a master has potentially less slaves to manage. The reduced degree is depicted in Figure 4.

At around 6 seconds the average degree of the BT topologies is always less than 7, i.e., always less than the maximum number of active slaves that a master can handle. We observe also

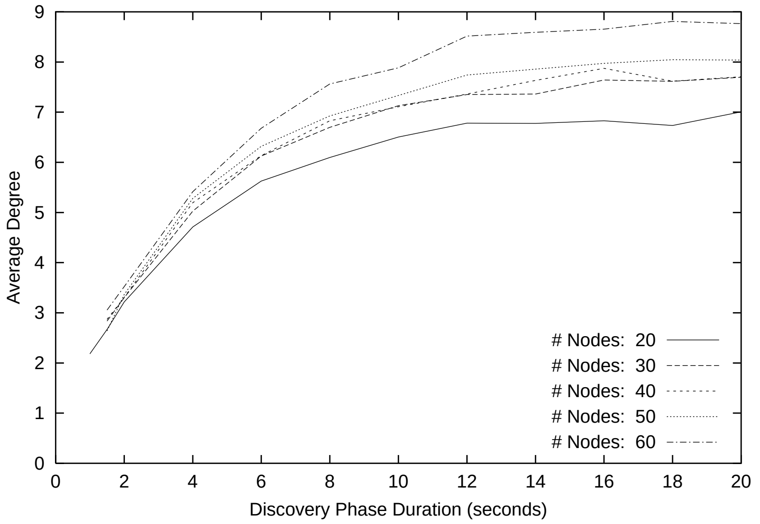

Figure 4: Average degree for BT networks.

that the longer the time of the discovery phase, the closer the “BT degree” becomes to the original degree (Table 1).

Finally, we computed the average shortest path length for both original topologies and their corresponding BT topology.

The average shortest path length for the original topology is listed in Table 1. Figure 5 shows that after 6 seconds the duration of the discovery phase T_{disc} does not sensibly affect the average length of the shortest paths in the BT topology. As noticed for the BT degree, the average length of the “BT shortest paths” converge to the corresponding value for the original topology (Table 1).

5 Conclusions

In this paper we have considered the problem of device discovery in multi-hop networks of Bluetooth devices. Solution to this problem is crucial for scatternet formation protocols where symmetric knowledge of neighboring nodes is the basis protocol correctness and performance. By exploiting the inquiry and paging procedures provided by the Bluetooth

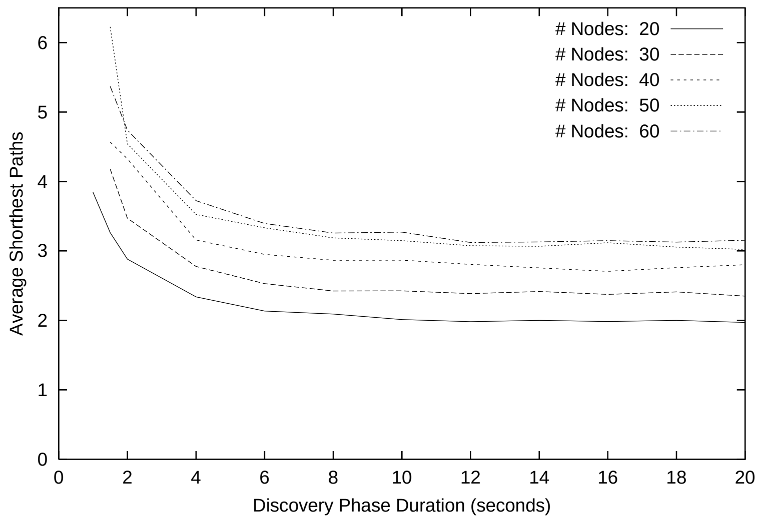

Figure 5: Average shortest path length for BT networks.

technology, we have defined a mechanism by which each pair of neighboring nodes try to gain knowledge of each other. The efficiency of our mechanism has been evaluated by the use of simulations. We considered the most general scenario of multi-hop topologies, i.e., topologies where the devices are not required to be in the radio vicinity of each other. The experiments we performed show the (statistical) impossibility for a device to gather the complete knowledge of its neighborhood in a short time. However, the topology formed by the sole devices discovered through our device discovery mechanism in just a few seconds is connected and so is the scatternet obtainable by this topology. Furthermore, we have observed that the Bluetooth average degree as well as the average length of shortest paths over the Bluetooth topology quickly converge to their corresponding values obtained on the original topology.

References

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

[2] 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.

[3] C. Law and K.-Y. Siu. A Bluetooth scatternet formation algorithm. In *Proceedings of IEEE Globecom 2001*, San Antonio, TX, 25–29 November 2001.

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

[5] The VINT Project. *The ns Manual.* http://www.isi.edu/nsnam/ns/, 2001.

[6] IBM. *BlueHoc: Bluetooth Ad Hoc Network Simulator, Version 1.0.* http://www-124.ibm.com/developerworks/projects/bluehoc, June 2001.