Device Discovery in Bluetooth Networks: A Scatternet Perspective

Stefano Basagni[1], Raffaele Bruno[2], and Chiara Petrioli[3]

[1] Northeastern University, Dept. of Electrical and Computer Engineering
basagni@ece.neu.edu
[2] CNUCE Institute, C.N.R. Pisa, Italy
bruno@guest.cnuce.cnr.it
[3] Università degli Studi di Roma, "La Sapienza," Dipart. di Scienze dell'Informazione
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 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 simulations we observed that despite the long time required for each node to become aware of the presence of all its neighbors, the Bluetooth topologies obtained by using the devices discovered after just 6 seconds are connected. The average number of neighbors of each node and the average route length are also consistently close to the values that we would obtain if all the neighbors of a device were discovered.
Keywords: Bluetooth networks, Device discovery, Scatternet formation.

1 Introduction

Bluetooth Technology (BT) [1] is emerging as one of the most promising enabling technologies for ad hoc networks.

When two BT devices 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*, 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.

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*.

Scatternet formation algorithms have been proposed in [2], [3], [4], [5]. These works have identified neighbor discovery (i.e., the process through which neighbors acquire a symmetric knowledge of each other) as the first and most time consuming operation to be performed by a BT device.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1087–1092, 2002.

A major problem is that the *inquiry procedures* provided in the BT specification for device discovery are time consuming and asymmetric. For two neighbor devices to handshake, they must be in "opposite" inquiry modes, namely one must be the inquirer, in *inquiry mode*, and the other device has to be willing to be discovered, i.e. it must be in *inquiry scan mode*. Also, the inquirer node is enabled to discover a neighboring device *without* having to identify itself to this device. In [4] and [5] "symmetric" methods for device discovery are proposed. In [5] each device alternates between inquiry and inquiry scan modes, randomly selecting the time to spend in each mode. In [4] time is divided into fixed length steps, and a node chooses randomly at each step whether to go in inquiry or in inquiry scan mode. When an inquirer node discovers one of its neighbors, a temporary piconet is created (by means of the *paging procedures*) so that the discovered neighbor can be made aware of the inquirer identity.

The Scatternet formation algorithms proposed in [4] and [5] rely on the assumption each nodes is in the transmission range of every other node ("single-hop" topology). This crucial assumption 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 (centralized) scatternet formation protocols. The only solutions proposed so far which address the more general and practical case in which the original topology can be multi-hop, ([2] and [3]) require each node to become aware of its one-hop neighborhood.

In this paper we investigate the effectiveness of the device discovery scheme proposed in [5], and adopted in [2], for the most general case of multi-hop topologies. Simulation results show that the time for each node to be made aware of over 90% of its neighbors is over 18 seconds in case of dense networks. However, we also observed that after only 6 seconds the percentage of neighbors discovered is large enough to obtain connected topologies, i.e., connected scatternets. Knowing a smaller number of neighbors has also the desirable effect of lowering the number of slaves that a master has to manage. We finally present numerical results about the average length of routes (shortest paths) in the topology obtained by considering all nodes and the sole links corresponding to the discovered devices.

The paper is organized as follows. Section 2 describe the use of the inquiry and page procedures that allows at each node the symmetric knowledge of some of its neighbors. In Section 3 we describe the experimental results obtained by simulations and, finally, Section 4 concludes the paper.

2 Device Discovery in Bluetooth Networks

For a detailed description of the Bluetooth system, the reader is referred to [1]. In the following we focus on the inquiry procedures 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*[4] at different

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

hop frequencies. The inquiry hop sequence is always derived from the general inquiry access code. 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 scanning 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.

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. 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 [5]. 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). Hereafter, we describe the operations performed at each device during the topology discovery phase. The generic device v that executes the discovery procedure, sets a timer $T_{\rm disc}$, which is decremented at each clock tick (namely, $T_{\rm disc}$ 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 ($T_{\rm w\ inquiry}$ or $T_{\rm w\ inquiry\ scan}$). While in a given mode, device v performs the inquiry procedures as described by the BT specification. The procedures that implement the inquiry mode or the inquiry scan mode are executed for the computed time ($T_{\rm w\ inquiry}$ and $T_{\rm w\ inquiry\ scan}$, respectively), not to exceed $T_{\rm disc}$. Upon completion of an inquiry (inquiry scan) phase, if $T_{\rm disc} > 0$, 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 $T_{\mathrm{w\ inquiry}}$ and $T_{\mathrm{w\ inquiry\ scan}}$), 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 Experimental Results

We have simulated the BT device discovery methods described in the previous section by using the VINT project Network Simulator ("ns2") [6] and BlueHoc [7], the IBM open-source extension to ns2 that implements the baseband and link layer of BT as described in the BT specification [1]. We have extended BlueHoc to provide: *i*) packet collision detection, *ii*) alternation between inquiry and inquiry scan, *iii*) determination if two nodes are neighbors based on their transmission radius and on their distance, and *iv*) dynamic selection of Master or Slave role at each node. We selected the $\mathrm{T_{w\ inquiry}}$ and $\mathrm{T_{w\ inquiry\ scan}}$ 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 [5]). We have conducted experiments with $t_{in} = 4$ and $t_{in} = 6$ without observing significant variations with respect to the results reported below. 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%.

3.1 Device Discovery in Multi-hop Networks

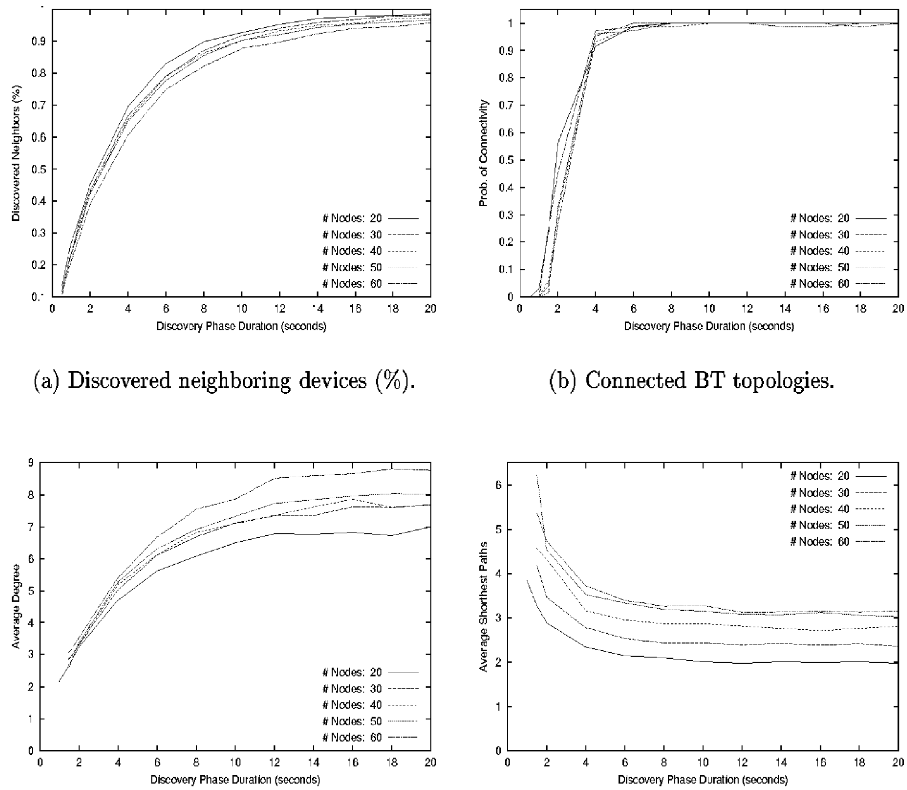
In what follows we term *original topology* the topology that we would obtain if each device could set up a bidirectional connection with all the devices in its transmission range (its neighbors). The term *BT topology*, instead, indicates the topology obtained by (bidirectionally) connecting only those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

Our 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 [5] and [4]). 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. The properties of average degree and average shortest paths are listed in Table 1.

Figure 1(a) shows the percentage of neighbors that each nodes locally discovers in at most 20 seconds in networks of 20 to 60 BT devices. The results are averaged over all the nodes in the network. We observe that the curves are very similar, given the similar average degree (i.e., the average number of neighbors

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



(a) Discovered neighboring devices (%).

(b) Connected BT topologies.

Fig. 1. Some characteristics of the discovered BT topologies.

of each node), as listed in Table 1. We notice that it is not possible for a node, even in 20 seconds, to discover all its neighbors. However, Figure 1(b), shows that despite the number of device discovered is less than the 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. 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 1(c). 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 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 1(d) 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).

4 Conclusions

In this paper we have considered the problem of neighbor discovery in multi-hop networks of Bluetooth devices. By means of extensive simulations we have shown that, despite the long time required for each node to become aware of the presence of all its neighbors, the Bluetooth topologies formed by devices discovered after just 6 seconds are connected, and do not result in significantly increased shortest paths between pairs of BT devices. Finally, we have shown that the length of the neighbor discovery phase is a powerful tuning knob to control the nodes degree, and therefore limit the number of slaves that a master has to manage.

References

1. http://www.bluetooth.com: Specification of the Bluetooth System, Volume 1, Core. Version 1.1 (2001).
2. Basagni, S., Petrioli, C.: Multihop scatternet formation for Bluetooth networks. To appear in: Prooceedings of the IEEE VTC Spring 2002, Birmingham, Alabama, May 6–9, 2002.
3. Zaruba, G., Basagni, S., Chlamtac, I.: Bluetrees—Scatternet formation to enable Bluetooth-based ad hoc networks. In: Prooceedings of the IEEE International Conference on Communications (ICC 2001). Volume 1., Helsinki, Finland, June 11–14 2001, pp. 273–277.
4. Law, C., Siu, K.Y.: A Bluetooth scatternet formation algorithm. In: Prooceedings of the IEEE Symposium on Ad Hoc Wireless Networks (SAWN 2001), San Antonio, Texas, 2001.
5. Salonidis, T., Bhagwat, P., Tassiulas, L., LaMarie, R.: Distributed topology construction of Bluetooth personal area networks. In: Proceedings of the IEEE INFOCOM 2001. Volume 3., Anchorage, Alaska, 22–26 April 2001, 1577–1586.
6. The VINT Project: The ns Manual. http://www.isi.edu/nsnam/ns/ (2001).
7. IBM: BlueHoc: Bluetooth Ad Hoc Network Simulator, Version 1.0. http://www-124.ibm.com/developerworks/projects/bluehoc (2001).