

The Problem of Bluetooth Pollution and Accelerating Connectivity in Bluetooth Ad-Hoc Networks

Somil Asthana
Department of Computer Science
State University of New York
Buffalo, NY 14226
asthana@cse.buffalo.edu

Dimitris N. Kalofonos
Nokia Research Center
5 Wayside Road
Burlington, MA 01803
dimitris.kalofonos@nokia.com

Abstract

This paper investigates a real-world problem termed “Bluetooth (BTH) pollution”, which is expected to become commonplace as the mass deployment of BTH devices continues. BTH pollution is caused by the presence of a large number of non-cooperating BTH devices within the same radio coverage area, which beyond introducing radio interference, also affect each other’s basic BTH operations such as discovery, paging and connection setup. The overall result can be a significant slow-down in BTH networking operations, e.g. ad-hoc network formation and healing. We believe the effect of BTH pollution is largely ignored in the BTH ad-hoc networking literature. In this paper we provide an insight on the causes of this problem, as well as simulation and experimental results that illustrate it. Finally, we propose a solution to accelerate BTH ad-hoc networking based on the use of the Class of Device (CoD) information specified in the BTH standard.

1 Introduction

Because of the large number of BTH-enabled phones and other devices currently being rolled out, BTH is becoming a de-facto leading technology for ad-hoc Wireless Personal Area Networking (WPAN). The proliferation of BTH devices is expected to also give rise to a new problem, a phenomenon we refer to as “BTH pollution”. Beyond issues caused by radio interference (e.g. link-level packet collisions and

retransmissions), the coexistence of a large number of non-cooperating BTH devices within range of each other (e.g. in an office environment or a mall), affects basic BTH operations such as inquiry, paging and connection setup. The affected operations are used extensively in BTH ad-hoc networking protocols, therefore the effect of BTH pollution can be significant in cases ranging from simple point-to-point connection setup (simple piconets) to more complex multi-hop network formation (scatternets).

The BTH standard [5] provides the mechanisms necessary for discovery and connectivity establishment in ad-hoc environments. Devices discover their neighbors by using a resource intensive low-level BTH operation called inquiry, during which the devices try to discover other devices by sending ID packets. On the other hand, devices that want to be discovered perform a much more light-weight process called inquiry scanning, during which they look for ID packets and respond back with Frequency Hopping Synchronization (FHS) packets. The FHS payload contains the sender’s 48-bit BTH device address (BD_ADDR) and the 24-bit Class-of-Device (CoD) information, among other fields. After the discovery phase, two devices choosing to connect follow a shorter process called paging and the simplest BTH network, a piconet, is created; by connecting two or more piconets a multi-hop network can be created called scatternet. Using these low-level BTH operations makes it possible to discover further information from remote devices by establishing increasingly higher-level connections: BTH friendly names can be retrieved by establishing Asynchronous ConnectionLess (ACL)

connections, BTH Service Discovery Protocol (SDP) records can be retrieved by establishing L2CAP connections and IP-level discovery protocol records (e.g. UPnP Simple Service Discovery Protocol (SSDP) [1]) by establishing BTH Network Encapsulation Protocol (BNEP) connections.

In this paper we investigate the problem of BTH pollution, which has attracted only limited research interest to date. Existing work has viewed this problem mainly from the perspective of accelerating device and service discovery. However, the research literature on BTH ad-hoc networking (e.g. scatternet formation and healing protocols) largely ignores the problem, by assuming that only nodes willing to collaborate exist in a given coverage area. In this paper we provide simulation and experimental results to illustrate the effect of BTH pollution on (a) basic BTH operations necessary for connection setup and (b) multi-hop scatternet formation and healing protocols. Finally, we propose a solution to accelerate scatternet formation and healing based on the use of the Class of Device (CoD) information included in the inquiry responses.

The rest of this paper is organized as follows: Section 2 reviews related work and existing proposals to accelerate BTH device and service discovery; Section 3 investigates the various aspects of the BTH pollution problem and gives simulation and experimental results that illustrate it; Section 4 presents our proposal to accelerate connectivity in BTH ad-hoc networks and Section 5 presents performance results that show the improvement; finally, in Section 6 we present our conclusions.

2 Related Work

Most of the work to date investigates ways of accelerating BTH device and service discovery. Some proposed approaches require the existence of a separate out-of-band channel, provided by a second short-range wireless technology, e.g. Infrared or RFID [9], [2]. On the other hand, Zaruba [14] proposed a solution to accelerate BTH inquiry for Personal Area Network by probabilistically predicting the number of inquiring and scanning devices. This solution uses these numbers to calculate and set the random back-off value for a inquiry response device. However, in a dynamic environment it is difficult to predict the number of in-

quiring and scanning devices at a given time. This uncertainty may often lead to an inaccurate selection of the random back-off parameter. Also, this solution probably requires adding new BTH Host Controller Interface (HCI) command to set the random back-off parameter and the current version of BTH specification does not provides any command to set this parameter.

In contrast to the approach requiring a second channel or changing random back-off value, [6] proposed two schemes using the CoD information contained in inquiry responses. The first scheme uses a coordinator node with a unique CoD, which holds the service descriptions of all the BTH devices in proximity. Other devices then use the unique CoD to locate the coordinator and connect to it to retrieve all the service descriptions. The second scheme does not require any centralized coordinator. Each device hashes its service information into 22 bits of the CoD field. Devices searching for a specific service description then try to match this CoD information. Our approach is similar to the second scheme in [6] and we propose an extended scheme based on CoD to deal with the effect of BTH pollution in BTH ad-hoc networks.

On the other hand, research on BTH scatternet formation protocols usually assumes that only devices willing to participate in the scatternet are present in an area. This will not be true in crowded environments such as office areas and shopping malls. For example, in protocols such as [10], [12], [4] devices gather information about their neighbors by randomly switching between INQ and INQ_SCAN states. Other protocols (e.g. [13]) assume that devices have prior knowledge of their neighbors and need to periodically update this information. However, in realistic conditions, the above procedures will be substantially delayed by the presence of non-cooperating devices and that effect needs to be investigated.

3 The Problem of BTH-pollution

3.1 Problem Description

We use the term BTH pollution to describe the problem which arises when a large number of BTH devices coexist within radio range and their basic BTH operations interfere with each other, thus causing a slowdown in BTH networking. BTH pollution does not re-

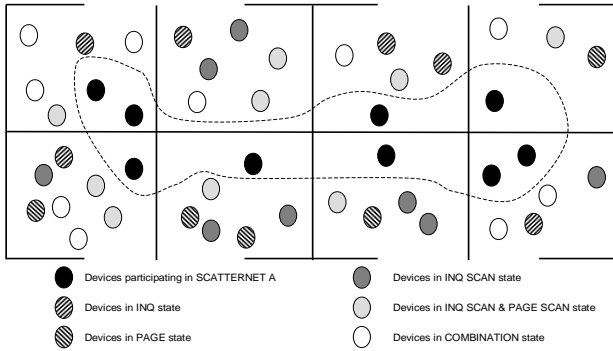


Figure 1. An example illustrating the problem of BTH pollution

fer to problems caused by radio interference, e.g. link-level packet collisions and retransmissions. It refers to the negative interaction of basic BTH operations such as inquiry, paging and connection setup in the presence of a large number of unknown, non-cooperating devices. These unknown devices do not collaborate within the context of a specific area-wide networking protocol; instead they perform their own operations independently. An example illustrating the problem is shown in Figure 1. From the point of view of devices willing to participate in Scatternet A, other unknown non-cooperating devices can be in any of the following BTH states:

- **INQUIRY:** The unknown device is performing BTH inquiry trying to discover its neighbors.
- **PAGING:** The unknown device is performing BTH paging trying to connect to some device.
- **INQ SCAN:** The unknown device is initialized to be only discoverable but not connectable.
- **INQ SCAN and PAGE SCAN:** The unknown device is initialized to be both discoverable and connectable.
- **COMBINATION:** The unknown device is randomly switching among the above states.

In the rest of this section we describe the effect of non-cooperating devices on the BTH operations of the devices of interest.

3.2 Impact on Connection Setup: Inquiry and Paging

One connection setup operation affected negatively by BTH pollution is the *Inquiry* operation. Besides the negative impact of a potentially large number of unrelated inquiry responses which would delay networking among devices willing to communicate, BTH pollution affects the Inquiry operation in a more fundamental way that reduces its effectiveness. A large number of unknown non-cooperating devices makes it more probable that an inquiring device will not find a desired device which performs inquiry scanning. This problem is illustrated in Table 1, where we used the Blueware ns-simulator [11] with the modifications proposed in [3], [7] to calculate the probability of finding a specific device in isolation and in the presence of 10 or 30 unknown devices. The unknown devices were assigned randomly to any of the five interfering states identified in Section 3.1. The reduction in probability is due to the adverse effect of BTH pollution on the inquiry scanning process. Once an inquiry scanning device successfully receives an ID packet from an inquiring device, it waits for a random back-off period (RAND) between 0 to 640 ms to avoid collisions with other scanning devices replying at the same time. After the RAND period the inquiry scanning device enters the inquiry response state and upon receiving an ID packet again, it responds with sending an FHS packet and then re-enters the inquiry scanning state. Since the ID packet consists of only the Inquiry Access Code (IAC), it is not possible to distinguish between ID packets sent by different inquiring devices. This can lead to the following problem in the presence of many non-cooperating inquiring devices. Due to BTH design, it is highly probable that after successfully receiving the first ID packet and waiting for RAND period, the scanning device enters inquiry response and receives a second ID packet from an unknown new inquiring device and not from the original one. Upon receiving the second ID packet the scanning device sends the FHS response to the new device, thus leaving the original device waiting for a longer time until it receives the FHS packet. This problem intensifies as the number of inquiring devices increases.

Another connection setup operation affected negatively by BTH pollution is *Paging* of devices that

Inquiry Time	Probability of finding a specific device		
	0 unknown devices	10 unknown devices	30 unknown devices
5.12 s	100 %	78 %	64 %
6.40 s	100 %	86 %	71 %
7.84 s	100 %	88 %	79 %
8.96 s	100 %	96 %	85 %
10.2 s	100 %	96 %	89 %

Table 1. Impact of BTH pollution on effectiveness of inquiry.

have responded to an inquiry, to retrieve their BTH friendly names or BTH Service Discovery Protocol (SDP) records. BTH friendly names and SDP are used extensively by applications to retrieve more information about discovered remote devices and their BTH services. In the presence of BTH pollution an inquiring device will get a large list of inquiry responses. In order to retrieve their BTH friendly names or SDP records, the inquiring device has to page all the devices in the list. This whole process of paging every device can consume a lot of time and energy. The problem is exacerbated because some devices are discoverable but not connectable and attempting to connect to them often leads to page timeouts. We observed this problem in a busy office environment and to investigate further we conducted the following experiment: a BTH 1.1 compliant device running the BlueZ BTH stack [8] inquired for 10.24 seconds and then tried to retrieve the BTH friendly names of all the discovered devices. The experiment was conducted 100 times and each inquiry discovered approximately 13 to 15 unknown devices. Figure 2 depicts the probability distribution of the delay to retrieve the BTH friendly names. Similar results are expected when retrieving SDP records.¹ We found that although approximately 55% of devices responded in less than 1.5 s, many of them took longer and about 15% of devices didn't respond to the BTH friendly name query at all, leading to a page timeout of about 20 s (last bin in Figure 2). These devices were initialized to be discoverable but not connectable by using the `HCI_Write_Scan_Enable` command in such a way that inquiry scanning was en-

¹In the case of retrieving SDP records there is an extra overhead of setting the necessary L2CAP connections.

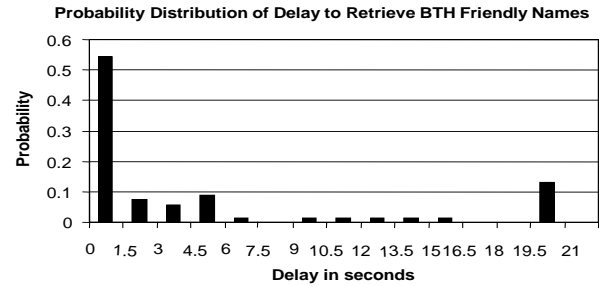


Figure 2. The probability distribution of the delay to retrieve BTH friendly names.

abled and page scanning was disabled. Usually this feature is used to save battery power and bandwidth consumed by periodic page scanning. According to the BTH specification such devices can still be paged because they are required to enter the page scan mode for a brief period immediately after responding to an inquiry. However, paging all the newly discovered devices simultaneously upon inquiry completion is not possible using a single radio. Therefore, by the time the inquiring device tries to page these devices, many have already left the page scan state and a page timeout occurs.

3.3 Impact on BTH Ad-Hoc Networking

The basic BTH operations of inquiry and paging to discover remote devices and retrieve more information about them are used extensively in BTH ad-hoc networking protocols. The presence of a large number of non-cooperating devices affects adversely these operations and consequently the networking protocols. However, perhaps the biggest problem is created by the potentially large number of unrelated inquiry responses, which delays networking among the devices actually willing to communicate. BTH pollution not only causes delays in network formation, but sometimes it may prevent these protocols from converging properly, especially when these protocols involve timeouts. Furthermore, if there are devices implementing more than one BTH ad-hoc network formation protocols in an area, more problems are caused as devices attempt to attach to the wrong devices implementing

different protocols. In general some high-level authentication mechanism may prevent the wrong devices to connect, but still extensive delays may be observed. Most scatternet protocols in the literature do not consider this potential problem and do not specify an authentication mechanism. Finally, there are cases that a new device may want to connect to a specific piconet, which is part of a large BTH ad-hoc network. In order to find a device in the specific piconet to attach, this new device may have to spend considerable amount of time in inquiring and paging unrelated devices.

4 Accelerating Connectivity in BTH Ad-Hoc Networks

4.1 Use of Dedicated Inquiry Access Codes (DIAC)

The BTH standard [5] has introduced the Dedicated Inquiry Access Codes (DIAC) to reduce the number of unrelated inquiry responses and therefore accelerate connectivity in BTH networks. However, in practice this mechanism has not been implemented adequately to address the problems that BTH pollution poses in BTH ad-hoc networking. According to the BTH specification every inquiring device sends out ID packets containing an Inquiry Access Code (IAC). In the vast majority of cases the General Inquiry Access Code (GIAC) is used. Besides the GIAC, 64 available DIACs have been reserved to limit the possible inquiry responses that a device needs to consider. However, only one of the 64 (the Limited IAC-LIAC) has been currently specified and is required by the foundation Generic Access Profile. Even then, according to the BTH standard the LIAC should be used by devices for only a limited period of time of up to one minute. Hardware implementations of most BTH devices available now and in the visible future implement only these two IACs (GIAC and LIAC). If all the above issues were addressed, this mechanism would help devices to weed out most unrelated inquiry responses by selecting among the 64 available DIAC by using the `HCI_Write_Current_IAC_LAP` command. Of course, the number of available codes is only 64, which would still lead to “DIAC collisions” in BTH polluted areas.

4.2 A Solution Based on the BTH Class of Device (CoD)

The BTH CoD is a 24-bit field in the FHS packets sent by devices when responding to Inquiries. The BTH standard has specified some limited categories and subcategories of devices (e.g. laptop, phone, desktop, access point, audio etc.) to be encoded in this field, to provide quick and rudimentary information about the discovered remote devices and accelerate connectivity. Since this information becomes available at the Inquiry phase, the process is relative fast (up to 10 seconds). The use of this field as currently specified provides limited assistance in deciding if the devices discovered are in the desired category.

We propose to address the BTH pollution problem by expanding the use of the 24 bits BTH CoD field. This proposal does not require any modification to the BTH standard, since the BTH CoD is a parameter that can be set by any application by using the `HCI_Write_Class_of_Device` command. A similar approach was proposed in [6] to accelerate service discovery with BTH; we extend that approach and apply it to accelerate connectivity in BTH ad-hoc networks. According to our scheme, devices in the desired category hash (encode using a hashing function such as MD5) any information they select to a number of agreed upon bits of the BTH CoD 24-bit field. We allow hashing of more than one level of information by grouping the available 24 bits into different groups, each group hashing a different level of information. Of course, since the overall available number of bits is 24, more groups means lower number of bits in each group and, therefore, only a limited number of groups (e.g. two or three) can be used in practice. A client application searching for devices in the desired category can then calculate the same hash and perform a regular inquiry. The application then looks for devices in the inquiry responses whose selected bits in the BTH CoD field match the hash value and attempts to connect to only those matching devices to retrieve further information (e.g. BTH friendly names) that is used in ad-hoc network formation protocols.

The use of the BTH CoD scheme is not supposed to guarantee that the devices that match the hash are indeed in the desired category. In fact, there may be conflicts with other devices encoding the same hash value

and the probability of such conflicts will grow higher the smaller the number of bits (among the available 24-bits) used to hash the information into. The scheme is intended to weed out devices that are not in the desired category with high probability, thus presenting the inquiring devices with a much shorter list of candidates to probe further. An authentication scheme using standard BTH security mechanisms should then be used to further prevent connecting to the wrong devices.

The main disadvantage of the proposed CoD scheme is that during the time that devices choose to use it, they will be interpreted as “unknown class” by devices that use the BTH SIG conventions to set and search the BTH CoD field. In practical terms, based on the most common usage of this field, this is not anticipated to be a significant problem. An inquiring device which is not implementing our scheme would interpret the inquiry responses of devices implementing it as “unknown class” and would have to proceed further to retrieve their BTH remote names or SDP records, which is a step that would most probably happen anyway. The proposed CoD scheme changes 2 bit Format Type to 11 instead of 00 assigned by BTH SIG convention as shown in Figure 3. Therefore our proposal does not require any changes in existing BTH specification.

4.3 Application to BTH Ad-Hoc Networking

Scatternets and piconets usually have some kind of application-level ID records, which may contain names, passwords and any other information used in identifying them. In our proposal, devices participating in a scatternet set n bits of their BTH CoD by hashing their scatternet ID record and the remaining $22-n$ bits by hashing their piconet ID record.² A device that seeks to join a piconet and a scatternet sets its own BTH CoD as described above by hashing the scatternet and piconet ID records and then performs an Inquiry. It then searches the BTH CoD in the Inquiry response list to find matches in the two groups of bits it is looking for. These devices that match are most likely in the desired category and the process continues by attempting to connect, authenticate with passwords, retrieve SDP or other Service Discovery records, or anything else that will allow the device to connect to the

²The remaining 2 bits are used to denote the CoD format type and should be different than 00.

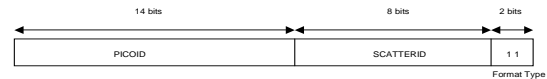


Figure 3. Using the CoD field to accelerate network formation.

desired scatternet and piconet. Devices not belonging in the desired scatternet and piconet will most probably be using BTH device classes that do not match the hashed information. But even if any of them do use the same bit patterns, this may only lead to unsuccessful attempts to connect and will increase the time to connect to the right BTH network. In general, even with a large number of devices in the area it is expected that the number of “false alarms” will be low and the connection process will be significantly accelerated.

As an example, we have implemented the CoD scheme as an optimization option of a BTH scatternet formation and healing protocol that we have proposed in [3], [7]. However, the CoD scheme is independent from specific protocols and could be applied to most scatternet formation protocols in the literature. According to [3], [7], a scatternet is formed by joining user piconets. Each user’s piconet has a name and a password and each scatternet has a scatternet name and password. Devices already connected in the scatternet set their BTH Device class as shown in Figure 3, where $PICOID = \langle md5sum(piconet\ name, piconet\ password) \rangle$ (14bits) and $SCATTERID = \langle md5sum(scatternet\ name, scatternet\ password) \rangle$ (8bits). If a piconet wants to join the scatternet, its master performs inquiry and looks for the responses whose last 8 bits match the desired SCATTER ID. It then proceeds to connect to any of the matching devices and uses the scatternet password and standard BTH security mechanisms to authenticate itself. If this fails it repeats this process with the next match until it succeeds. On the other hand, if a user brings a new device to join his/her piconet which may or may not be a part of the scatternet, the new device performs inquiry and looks for the responses whose first 14 bits match the desired PICOID.

The main benefits from the proposed CoD scheme in the area of BTH network formation are shorter inquiry and paging processes. When a device finds a matching CoD field it can stop inquiring by issuing a

HCI.INQUIRY.CANCEL command. Depending on the network formation protocol, the device may then attempt to page the matching remote device. The use of the CoD scheme eliminates with high probability the paging timeouts to devices that are discoverable and not connectable, which can cost long delays in network formation. Finally, it is worth mentioning that the use of the CoD scheme not only helps in accelerating network formation, but also in reducing the overall power consumption, since inquiry and paging are power intensive operations.

5 Performance Results

In Section 3.2 we presented experimental measurements and simulation results to investigate and expose the problem of BTH pollution. The experiments were conducted using BTH v1.1 compliant devices, with CSR chipsets (HCI Ver:1.1 (0x1), HCI Rev:0x72, LMP Ver:1.1 (0x1), LMP Subver: 0x72), running Linux kernel 2.4.18 with the BlueZ [8] BTH stack. For our simulation results we used the Blueware nsimulator [11] with the modifications proposed in [3], [7]. In all our simulation results, the unknown, non-cooperating devices are assigned randomly to any of the five interfering states identified in Section 3.1. In the rest of this section we present simulation results to demonstrate the benefits of using the proposed CoD scheme.

We first evaluated the delay in connection establishment between two devices in the vicinity of a number of unknown devices. The results are shown in Figure 4. When only the BTH friendly name is used (currently the most common way to select the desired remote device), the connection establishment delay increases almost linearly with the number of unknown devices. This is because after inquiring for 10.24 s the client device connects to each newly discovered device to retrieve its BTH friendly name until it finds a match with the desired BTH friendly name. On the other hand, if the BTH friendly name is hashed in the CoD, the connection establishment delay is almost constant and equal to the inquiry time plus the paging time. In the above two experiments the inquiry time was fixed to 10.24 s; if the inquiry were allowed to stop when the first CoD match occurred, the benefits in connection delay would be even higher as also shown in Figure 4.

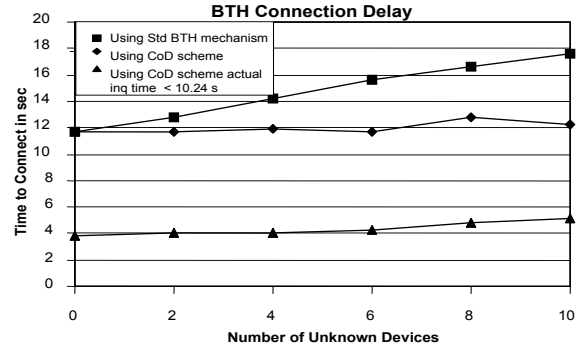


Figure 4. Connection setup improvement using the CoD scheme.

The connection delay in such case is less than 6 s and equal to the partial inquiry time plus the paging time.

Figure 5 shows the impact of BTH pollution in scatternet formation delay. We calculated the network formation delay both in isolation and in the presence 10, 20 and 30 unknown devices, with and without using the CoD scheme. Even though the results were obtained using the protocol proposed in [3], [7], the effect depicted applies to most scatternet formation protocols, since they all use inquiry and paging in similar ways to discover and connect to neighboring devices. As expected, the scatternet formation delay without using the CoD scheme is much higher in the presence of unknown devices. This happens because devices trying to form a scatternet spend time trying to connect to unrelated devices, only to realize that they are not the right ones. In some cases this happens after lengthy delays caused by page timeouts. On the other hand, using the CoD scheme filters out unrelated responses at the inquiry level and therefore the scatternet formation is minimally affected by BTH pollution and almost behave as if devices are operating in isolation. The CoD scheme results shown in Figure 5 were calculated for 10 unknown devices. We found similar results for 20 and 30 of unknown devices.

6 Conclusions

In this paper we exposed and described the problem of BTH pollution, a problem not adequately investigated in the literature, and we provided an insight on how it affects basic BTH connection setup operations

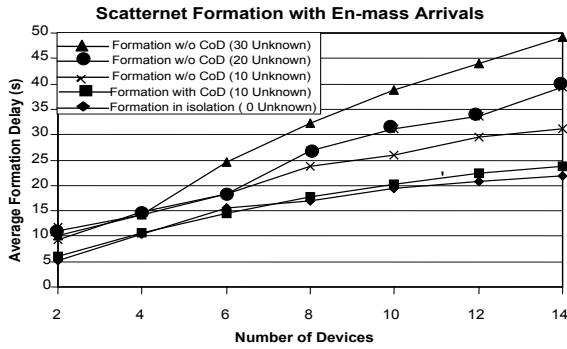


Figure 5. Impact on scatternet formation delay in the presence of 10, 20 and 30 unknown devices. Improvement using the CoD scheme for 10 unknown devices.

such as inquiry and paging to retrieve BTH friendly names and other remote device information. Furthermore, we examined the impact of BTH pollution on ad-hoc networking protocols. We proposed a solution for accelerating connectivity using the BTH CoD field and demonstrated that it can drastically reduce the adverse effects of BTH pollution. As part of our future research we will elaborate on hashing group codes in the CoD fields and formulate a mathematical model and study the BTH pollution problem.

References

- [1] *UPnP Device Architecture 1.0*. December 2003. <http://www.upnp.org/resources/documents/CleanUPnPDA101-20031202s.pdf/>.
- [2] M. S. A. Busboom, I. Herwono and G. Zavagli. Unambiguous device identification and fast connection setup in bluetooth. 0, February 2002.
- [3] S. Asthana and D. Kalofonos. Enabling secure ad-hoc group collaboration over bluetooth scatternets. August 2004.
- [4] F. Cuomo, G. D. Bacco, and T. Melodia. SHAPER: a self-healing algorithm producing multi-hop Bluetooth scatternets. December 2003.
- [5] B. S. I. Group. *Specification of the Bluetooth System, v1.1*. February 2001.
- [6] S. I., P. S., C. C., M. Haase, and D. Timmermann. Time and energy efficient service discovery in bluetooth. 1:418–422, April 2003.
- [7] D. N. Kalofonos and S. Asthana. A bth scatternet formation and healing protocol for group collaboration.

- [8] M. Krasnyanskiy and M. Holtmann. Bluez official linux bluetooth protocol stack. 2003. <http://www.bluez.org/>.
- [9] T. C. R. Woodings, D. Joos and C. Knutson. Rapid heterogeneous connection establishment: Accelerating bluetooth inquiry using irda. 3:804–811, March 2002.
- [10] T. Salonidis, P. Bhagwat, L. Tassiulas, and R. LaMaira. Distributed topology construction of bluetooth personal area networks. pages 1577–1586, April 2001.
- [11] G. Tan. Blueware: Bluetooth simulation for ns. <http://nms.lcs.mit.edu/projects/bluware>.
- [12] G. Tan, A. Miu, J. Guttag, and H. Balakrishnan. An efficient scatternet formation algorithm for dynamic environments. November 2002.
- [13] G. Zaruba, S. Basagni, and I. Chlamtac. Bluetrees - scatternet formation to enable Bluetooth-based ad hoc networks. June 2001.
- [14] G. Zaruba and I. Chlamtac. Accelerating bluetooth inquiry for personal area networks. 22:702–706, December 2003.