A Note on the Study of Bluetooth Networks' Distributed Algorithms

Ahmed Jedda, Nejib Zaguia, Guy-Vincent Jourdan
University of Ottawa, Canada
ajedd077@uottawa.ca, zaguia@site.uottawa.ca, gvj@site.uottawa.ca

Abstract

Bluetooth has provided many features that enable wireless ad-hoc networks, but it has also introduced many problems. The root cause of these problems lies in its communication mechanisms. We argue in this paper that the models that have been used to study distributed algorithms on Bluetooth networks do not adequately model these networks in most cases, and were often oversimplified. This is mainly due to how the many restrictions that the Bluetooth specifications impose on such networks are taken into account, and the lack of "shared knowledge" of these restrictions among the researchers of this field. We give some examples to back our argument. We give also some suggestions and proposals to overcome these issues.

1. Introduction

Bluetooth is one of the many radio technologies that were proposed to enable wireless ad hoc networks. The technology is promising in many ways. First, it uses a communication mechanism that is resistant to noise and robust against interference. Second, Bluetooth devices are efficient with respect to power consumption. Third, the cost of Bluetooth devices is relatively low. Fourth, and more importantly, Bluetooth offers ad hoc networking capabilities. These points, among others, make the Bluetooth technology very promising for real ad hoc networks.

But while the Bluetooth technology has many benefits and solved many problems in wireless ad hoc networks, it has also raised many other problems. In our opinion, these problems are mainly caused by two factors. The first is the Frequency Hopping Spread Spectrum (FHSS) communication mechanism that is used by Bluetooth, which allows a pair of neighboring nodes to communicate with each other without imposing a significant interference on other their neighbors. The second factor lies in the introduction of the INQUIRY, INQUIRY SCAN, PAGE and PAGE SCAN states, which are used by Bluetooth devices for the device discovery and link establishment procedures. These two factors make Bluetooth networks models different from traditional radio networks. The FHSS technique and the introduction of the Bluetooth states introduce new restrictions that should be taken into account when designing distributed algorithms for Bluetooth networks. If these restrictions are not addressed properly when designing distributed algorithms, then a significant penalty may occur, especially in term of their execution time.

In this paper, we provide some of the results we obtained while investigating Bluetooth Scatternet Formation (BSF) algorithms. The BSF problem can be briefly defined as the problem of forming wireless networks of Bluetooth devices in an efficient manner. It is the most studied network problem (if not the sole) in the Bluetooth literature that is solved by a distributed algorithm. A relatively large number of algorithms were proposed to solve this problem. We studied these algorithms and we focused on how they dealt with the problems introduced by the FHSS technique and the Bluetooth states (or simply, the problems introduced by the device discovery and link establishment procedures of the Bluetooth specifications.) During our investigation, we observed three main points, which are:

1. The theoretical models used to study BSF algorithms do not adequately model Bluetooth networks in most cases, and were often oversimplified.
2. The existence of many features in the Bluetooth specifications that have a large impact on BSF algorithms, and thus should not be ignored.
3. The absence of a common model/framework for the problems introduced by the specification of the device discovery and link establishment procedures that would be shared by the researchers of the field.

We elaborate on each of these points in more details later. We consider that these three points represent a weakness that affect negatively the progress of research in BSF algorithms, and the distributed algorithms of Bluetooth networks in general. Solutions should be provided to overcome this weakness. Because of the distinctive principle of work behind Bluetooth and because of the problems introduced by the device discovery and link establishment procedures, we believe that the establishment of a "general computational model" is one of the essential solutions. This model should define clearly what can be done (and at what cost) and what cannot be done by a distributed algorithm in a Bluetooth network. Also, this model would be used as a basis to design distributed algorithms over Bluetooth networks, to analyze them and to compare them.

Section 2 of this paper talks about the features that we believe make Bluetooth algorithmically different from other traditional distributed systems. Section 3 describes briefly the BSF problem. Section 4 and 5 give examples that show that the problems introduced by the device discovery and link establishment procedures were not always properly addressed in BSF algorithms. Section 6 and 7 talk about some observations (or weaknesses) we noted during our study. Section 8 concludes and gives some suggestions and proposals.

2. What distinguishes Bluetooth?

Bluetooth uses the Frequency Hopping Spread Spectrum (FHSS) technique as its communication mechanism. This technique divides the frequency medium into a number of non-overlapping frequency's ranges (or hops.) A transmitter and a receiver use a pseudo-random sequence of frequency hops, which is known to both, in order to exchange messages through these hops, using one at each time slot. With the FHSS technique, two neighboring devices can communicate with each other using separate channels without considerably disturbing their neighbors, and thus there is no shared medium as in broadcast radio networks such as the IEEE 802.11 and Ethernet-like networks. One problem in the FHSS technique lies in the necessity of setting up an agreement among the communicators on the utilized pseudo-random sequence of hops.

Some complexities on the procedures of device discovery and link establishment are introduced by the FHSS technique. If a Bluetooth device wants to discover a neighbor, it goes to the INQUIRY state and keeps alternating rapidly within a range of frequencies announcing its existence by broadcasting small-sized special packets called ID packets. Despite that name, ID packets do not contain the identity of the sender. On the other side, a device that wants to be discovered goes into the INQUIRY SCAN state and alternates at a slower rate within the same range of frequencies scanning for neighboring inquirers. If an inquirer and a scanner meet on the same frequency, both exchange some packets that are sufficient for them to establish a link. After that, if the two devices want to establish a link, then one of them goes into the PAGE state and the other must go to the PAGE SCAN state. The devices, then, exchange some messages until a link is established between them.

We see that the FHSS technique and the Bluetooth states just mentioned above (i.e. the INQUIRY, INQUIRY SCAN, PAGE and PAGE SCAN states) impose restrictions on the Bluetooth networks' distributed algorithms. If these restrictions are not addressed properly by these algorithms, then a significant penalty in terms of execution time may occur. We give next some examples that show that BSF algorithms did not always address properly these restrictions. But before that, we discuss briefly the BSF problem.

3. The Bluetooth Scatternet formation problem

According to the specifications of Bluetooth, if two devices want to communicate, then both must belong to the same *piconet* or *scatternet*. A piconet is a star network with one master and up to seven slaves. Piconets may have up to 255 slaves, however, at most seven of them can be active at one time; all the others would be *parked* (i.e. not active.) A scatternet is multiple interconnected piconets. To interconnect piconets, we use bridge nodes which are nodes that belong to two or more different piconets. A node can be a master to only one piconet, but it can be a slave in an unlimited number of piconets.

The BSF problem is the problem of assigning a role to each node in the network (i.e. master or slave, bridge or non-bridge) in a way that generates scatternets that match some performance criteria. For instance, the scatternet should contain all the network's nodes (i.e. be connected.) The scatternet, also, should avoid having piconets with more than 7 slaves (i.e. be degree constrained.) Moreover, it is desirable that the communication load is divided fairly on the nodes of the scatternet. These criteria, and many others, make the BSF a challenging and unique problem.

For any scatternet to exist, its nodes must first discover each other. Only basic recommendations in the Bluetooth specifications were given to solve this problem. Briefly, if two nodes want to discover each other, one must go to the INQUIRY state while the other goes to the INQUIRY SCAN state. In reality, BSF algorithms use more complicated device discovery techniques and mechanisms. A detailed survey on these techniques can be found in [11]. We categorized these techniques into *static* and *dynamic* techniques. We also categorized the BSF algorithms into *static* and *dynamic* techniques, depending on the device discovery technique they use.

In *static* BSF algorithms, the nodes execute in an external phase a device discovery technique that allows each of them to know the identities of a subset of its neighbors. BlueMIS [1] and BlueStars [2] are examples of static BSF algorithms. A widely used static discovery technique was introduced by Salonidis *et al.* in [3]. The idea of this technique is to make all the nodes initially alternate randomly between the INQUIRY and INQUIRY SCAN states. If two nodes discover each other, a piconet is constructed to exchange some data and it is destroyed after that. For simplicity, we called this technique ALTERNATE. In *dynamic* BSF algorithms, the nodes implicitly discover their neighbors while executing the formation algorithm, making it possible for the nodes to join the scatternet on-the-fly. Examples of dynamic BSF algorithms are TSF [4] and SHAPER [5].

4. Communication models of Bluetooth networks

Different computational models were used to study BSF algorithms. In this section, we briefly describe these models. We give examples that show that there are some restrictions imposed by the Bluetooth specifications that were not always properly addressed in BSF algorithms. This shows that the models used to study BSF algorithms did not adequately model Bluetooth networks in most cases.

Modeling static BSF algorithms

In static BSF algorithms, the network is initially represented with a vicinity graph. The vertices are the nodes and a link is set between two nodes if they are in the radio range of each other. The vicinity graph is usually assumed to be a Unit Disk Graph (UDG) and it is always assumed that all nodes have the same radio range. The nodes execute a static discovery technique, which generates another graph. In the generated graph, a link is set between two nodes iff they discovered each other. It is not necessarily that each node discovers all its neighbors – even in error-free environments. Thus, the generated graph can be assumed to be a UDG, a subgraph of a UDG (such as the case of ALTERNATE,) or a degree-bounded subgraph of a UDG (see [6].)

One of the problems in static BSF algorithms is that they assume sometimes that there are discovery techniques that can generate a UDG in a reasonable time. The correctness of the solutions of some of these algorithms depends mainly on this assumption. However, we found after a detailed survey and experiments that such a technique does not yet exist in the literature. It seems that the only way for generating UDG's, by a static discovery technique, would be by using a radio technology other than Bluetooth.

After executing the discovery technique, static BSF algorithms model the network as a traditional wired network (in most cases.) This is because there is no shared communication medium in these networks, as it is the case with other broadcast radio networks. Communication between neighboring devices is done using PAGE messages. In other words, each time two nodes need to communicate with each other, they establish a temporary piconet, exchange some messages, and destroy the piconet then. No INQUIRY messages are needed because the two communicating nodes know already the identities of each other because of the external discovery phase. The transmission time of PAGE messages is usually assumed to be negligible (in the order of milliseconds.) However, this happens only under the condition that one of the nodes is in the PAGE state while the other is in the PAGE SCAN state at the same time. Few researchers considered this condition when designing their BSF algorithms (see for example [2] and [7].) In many other algorithms, this condition was not given the necessary attention, making *randomization* the only method to implement the transmission of a PAGE message. A node resides in the PAGE SCAN state most of the time. If the node wants to send a message then it *randomly* alternates between the PAGE and PAGE SCAN state so that it does not miss any message sent to it from its neighbors.

This randomization approach may have negative effect on the execution time of a BSF algorithm. In an unpublished work, we implemented BlueMIS [1] with three different implementations. One of them followed the randomized approach; the other two were similar to the approach followed by BlueStars [2] and BlueMesh [7]. It was found that the randomized implementation work very poorly (in term of execution time) compared to the other implementations. It was found also that BlueMIS, using this implementation, does not terminate sometimes. Furthermore, in [8], Bluenet [9], which is a static BSF algorithm that use this randomized approach, was found to have a relatively poor performance when compared with other BSF algorithms that do not follow the same approach of implementation, mainly because of this randomized approach.

Modeling dynamic BSF algorithms

Dynamic BSF algorithms construct components of nodes and keep continuously merging them until there are no more nodes to add to the scatternet. A component can be an isolated node, a piconet or a scatternet. Nodes in a component schedule their time between intra-scatternet communication and discovering new neighbors. The way how a component discover its neighbors differ from one BSF algorithm to another. In [10] and [11], we gave a more detailed discussion about these methods, which we called dynamic device discovery techniques. Most dynamic BSF algorithms assume that the sole difference between the model they use and the traditional wired network model is the need to employ a dynamic discovery technique.

We believe that there is a problem that most (if not all) dynamic BSF algorithms did not focus on. A dynamic BSF algorithm deals with three types of messages; each has a different cost. These types are:

1. *Inquiry messages*: these are messages between two nodes that never discovered each other before. They are the most expensive and their cost depends on the discovery technique employed.
2. *Page messages*: these are messages between two nodes that already know the identities of each other. This is the same type of messages we find in static BSF algorithms.
3. *Intra-scatternet messages*: These are messages exchanged between nodes belonging to the same component. Their cost depends on the device discovery technique used and on the *inter-piconet scheduling algorithms*[1]. For instance, assume that that two nodes (*x* and *y*) need to exchange messages. There is only one path between them. Because a node in the INQUIRY or PAGE state cannot do any communication task, if one of the nodes in this path was inquiring for or paging some of its neighbours, then the communication between *x* and *y* will be blocked until all the nodes of the path are free to foreword the messages of *x* or *y*. To avoid this issue, the scatternet's nodes must run an efficient inter-piconet scheduling algorithm. It should be noted that the scheduling and formation algorithms were studied separately most of the times.

We note that in both dynamic and static BSF algorithms communication models, there are many types of messages each with a different cost. The cost of such messages depends on the mechanisms that the BSF algorithm uses. Considering this fact, BSF algorithm designers need to avoid mechanisms that may cause costly messages. This can help also in the analysis and comparisons of the different BSF algorithms, as it is possible sometimes to compute the number of messages of each type. The existence of different type of messages (with different costs) can complicate more the analysis of BSF algorithms.

5. Bluetooth vs. IEEE 802.11 networks

In this section, we want to show that distributed algorithms running over Bluetooth networks perform very poorly in term of their execution time when compared with distributed algorithms running over IEEE 802.11 networks, which are the most used in the literature of wireless ad hoc networks. This is caused mainly because of the improper addressing of the new restrictions imposed by the device discovery and link establishment procedures of Bluetooth.

For the comparison to be reasonable, we need to study two algorithms, each having the same principle of work but each work on a different type of networks. We compare the results of two independent studies. The first study [12] examined an on-demand BSF algorithm, while the second [13] examined the performance of two reactive routing algorithms for wireless ad hoc networks; AODV and DSR (running over IEEE 802.11 networks.) In all these algorithms, the nodes communicate with their neighbors, mainly, by means of messages broadcasting. Comparing the results of the two papers, it is found that broadcasting in IEEE 802.11 networks can be executed in about 28 times faster than in Bluetooth networks. The reason behind this phenomenon is that Bluetooth does not have a shared medium for communication as it is the case in IEEE 802.11 networks. However, it should be noted that Bluetooth performed much better compared to IEEE 802.11 in term of other performance metrics, such as power consumption for instance (see [12].)

This example shows how different are Bluetooth networks from traditional wireless ad-hoc networks. It also shows the criticality of the problems introduced by the device discovery and link establishment procedures of Bluetooth. We believe that more care should be given to these problems in order to improve the execution time of BSF algorithms. The understanding of these problems has proven to be useful for BSF algorithms. This can be seen clearly in static BSF algorithms, which have their performance greatly improved with time.

[1] A bridge node schedules its time between the piconets it belongs to by running an inter-piconet scheduling algorithm. Solutions to this scheduling problem were left open to the researchers.

6. Minor changes, substantial impact

We showed in [10] how minor modifications on the specifications of Bluetooth ver1.1 can substantially improve the execution time of static BSF algorithms. In reality, these modifications were standardized in Bluetooth ver1.2 and are still available in all the later Bluetooth versions. However and more interestingly, we showed in the same paper that the same modifications do not have the same large impact on dynamic BSF algorithms.

In a work published in [11], we showed how a minor change in the implementation of the specifications can substantially degrade the performance of static BSF algorithms. To understand this change, we must get into some details of the INQUIRY procedures. In order for an inquirer node to discover its neighbors, it needs to broadcast small-sized packets. The inquirer alternate within a sequence of frequencies to broadcast its packets. This sequence is derived from the inquirer's clock value. The inquirer uses only 32 frequency hops. These frequencies are divided into two sets; called Train A and Train B. The inquirer uses a different train each 2.56 seconds (by default.) We implemented the ALTERNATE technique, which was described in Section 1, in two different ways. Both implementations are legal according to the specifications. In the first implementation, which we called the restricted case, the inquirer starts with Train A each time it goes to the INQUIRY state. In the second implementation, which we called the non-restricted case, we kept the counter that controls the switching of the trains always running (i.e. the inquirer start the INQUIRY with whatever train the counter points to.) We found that ALTERNATE does not discover more than 65% of the links of the networks on average (after 40 seconds of simulation) – even in error-free environments – if the restricted case implementation was used, compared to 90-99% in the non-restricted case. If the performance of the ALTERNATE was as it is in the restricted case, then most static BSF algorithms would not perform as expected. The discovery of at most 65% of the network's links would also generate more disconnected networks, as our experiments have shown.

We want to show by these sets of experiments that there are many features in the Bluetooth specifications that have a large impact on Bluetooth networks' distributed algorithms which cannot be considered always as "details".

7. The lack of "shared knowledge"

We found during our work some discovery techniques that require modifications on the specifications of Bluetooth. However, some of these techniques have their equivalents in the literature that do the same task, with no negative side-effects, and require no modifications on the specifications. These techniques actually were used by some of the major BSF algorithms. The details of these techniques and their equivalents can be found in [11]. This example, most of the examples given above, and many others show that there is, to some extent, a lack of shared knowledge among the researchers of the field of the problems introduced by the device discovery and link establishment procedures. We believe that this has a negative impact on the advance of this research field. We believe that publicizing the problems of device discovery and link establishment is one of the solutions to overcome this issue.

We want to note herein also that many researchers examined thoroughly the device discovery procedures, from a point of view other than that of BSF algorithms. Some of them proposed solutions to weaknesses in the specifications. However and because of the use of non-uniform performance metrics, comparing the solutions of these researchers is a very hard task without actually implementing them and then comparing them, and therefore, their results and analysis could not be exploited perfectly.

8. Conclusion and remarks

In this paper, we presented some points we observed during our study of BSF algorithms (see Section 1.) In our opinion, these points are considered as weaknesses that affect negatively the progress of research in BSF algorithms, and the distributed algorithms of Bluetooth networks in general. These weaknesses show us that the problems that were introduced by the device discovery and link establishment procedures of Bluetooth were not properly addressed in most BSF algorithms. We presented few examples that back our argument. To overcome these weakness points, we believe that a "general computational model" should be established. This model should 1) define clearly the different restrictions imposed by the Bluetooth specifications on the distributed algorithms, and 2) it should make such restrictions as a "shared knowledge" among the researchers of the field, and whence it can be used as a starting point to design distributed algorithms in Bluetooth wireless networks – including BSF algorithms.

Such a computational model, if available, would simplify the theoretical analysis and comparisons of the different BSF algorithms existing in the literature, and to keep simulation experiments as a last step for comparisons. At this point, we want to make clear that simulation experiments were the only mean to compare BSF algorithms. However, comparing networks algorithms using simulation experiments is a question of debate in the research community of wireless ad hoc networks. A number of researchers believe that there are issues of credibility in the simulations of wireless ad hoc networks. Some of the researchers who have this opinion, such as [14] and others, suggested to give more care to the simulations and showed some of the pitfalls that the researchers fall in while conducting their experiments. These researchers usually come from the simulation and modeling research area. Other researchers had a different opinion for the solution of these credibility issues. For instance, Stojmenovic advocated in [15] the use of simple simulation models matching assumptions and metrics in the problem statement to provide a basic “proof of concept”. These simple models would be to study the performance of the algorithms and to compare them with truly competing solutions. This argument was based on the fact that “theoretical proofs of performance are difficult (often probably impossible) to derive” [15]. However, it was shown in [16] that even straightforward distributed algorithms, such as the flooding algorithm, may have “significant divergences” in its performance when it is executed under different simulators, even if these simulators were “popular”. It is worth-mentioning that there have been some attempts in the literature to avoid simulations as much as possible by providing “concise enough theoretical models of computations”, even if they were “harsh” models (see [17] and references therein.) In our opinion, simulation experiments should be the last step taken to compare the different network algorithms (including BSF algorithms.) However, although BSF algorithms seem to be very hard to be analyzed mathematically, such a general computational model can be used to compare these algorithms – at least informally.

All our work considered BSF algorithms, since the BSF problem is the most studied network problem (if not the sole) in the literature. However, we believe that other classical problems of theoretical distributed computing, such as election, broadcasting or tree formation, should also be studied with considering the restrictions imposed by Bluetooth networks. Note that some BSF algorithms use some ideas of such classical algorithms or use some of these classical algorithms as “building blocks”. Note also that some of these classical problems were studied under different distributed computational models, such as the asynchronous and synchronous wired networks, broadcast radio networks, self-stabilizing model and the mobile agents’ model, but it was never studied with the same intensity under the Bluetooth networks model.

9. References

[1] N. Zaguia, Y. Daadaa, I. Stojmenovic, “Simplified bluetooth scatternet formation using maximal independent sets,” Integ. Comp.-Aided Eng. 15, 3, 2008, 229-239.

[2] C. Petrioli, S. Basagni, M. Chlamtac, "Configuring BlueStars: multihop scatternet formation for Bluetooth networks," Comp., IEEE Trans., 52, 6, 2003, 779-790.

[3] T. Salonidis, P. Bhagwat, L. Tassiulas, R. LaMaire, “Distributed topology construction of Bluetooth personal area networks,” Proc. IEEE INFOCOM, 2001.

[4] F. Cuomo, T. Melodia, I.F. Akyildiz, "Distributed self-healing and variable topology optimization algorithms for QoS provisioning in scatternets," IEEE Selected Areas in Comm. 22, 7, 2004, 1220-1236.

[5] G. Tan, A. Miu, J. Guttag, and H. Balakrishnan. "An Efficient Scatternet Formation Algorithm for Dynamic Environments," IASTED CCN, 2002.

[6] D. Dubhashi, O. Häggström, G. Mambrini, A. Panconesi, C. Petrioli. “Blue Pleiades, a new solution for device discovery and scatternet formation in multi-hop bluetooth networks.” Wire. Nets. 13, 1, 107-125.

[7] C. Petrioli, S. Basagni, and I. Chlamtac, “BlueMesh: degree-constrained multi-hop scatternet formation for Bluetooth networks.” Mob. Nets. App. 9, 1, 2004, 33-47.

[8] S. Basagni, R. Bruno, G. Mambrini, and C. Petrioli, “Comparative performance evaluation of scatternet formation protocols for networks of bluetooth devices”. Wireless Netorks. 10, 2, 2004, 197-213.

[9] Z. Wang, R.J. Thomas, J. Haas, “Performance comparison of Bluetooth scatternet formation protocols for multi-hop networks.”, Wireless Network.

[10] A. Jedda, N. Zaguia, G-V. Jourdan, “Analyzing the discovery phase in Bluetooth Scatternet Formation Algorithms”, IEEE SAHNS 2009 in conjunction with ICDCS 2009 Montreal, Canada.

[11] A. Jedda, “The device discovery and Bluetooth Scatternet Formation Algorithms”, M.Sc. Thesis, University of Ottawa.

[12] Y. Liu, M.J. Lee, T.N. Saadawi, “A Bluetooth scatternet route structure for multi-hop ad hoc networks,” IEEE J. Selected Areas in Communications, 21, 2, 2003, 229-239

[13] C.E. Perkins, E.M. Royer, S.R. Das, M.K. Marina, "Performance comparison of two on-demand routing protocols for ad hoc networks," Pers. Comm., IEEE , 8, 1, 2001, 16-28.

[14] K. Pawlikowski, H.-D.J. Jeong, J.-S.R. Lee "On credibility of simulation studies of telecommunication networks," Comm. Mag., IEEE, 40, 1, 2002, 132-139.

[15] I. Stojmenovic, “Simulations in wireless sensor and ad hoc networks: Matching and advancing models, metrics and solutions,” IEEE Comm. Mag., 46, 12, 2008

[16] D. Cavin, Y. Sasson, A. Schiper “On the accuracy of MANET simulators.” Proc. POMC '02. ACM, 38-43.

[17] T. Moscibroda, R. Wattenhofer “Maximal Independent Sets in Radio Networks.” Proc. 24th ACM (PODC'05).