Dhaka Univ. J. Sci. 58(1): 35-39 2010 (January)

Traffic Flow and Optimal Configuration of Bluetooth Scatternet

Mohammed Shafiul Alam Khan

Institute of Information Technology, Dhaka University, Dhaka-1000, Bangladesh

M. Lutfar Rahman

Department of Computer Science and Engineering, Dhaka University, Dhaka-1000, Bangladesh

shafiul@univdhaka.edu, lrahman@univdhaka.edu

Received on 18.07.2008. Accepted for Publication on 04.07.2009

Abstract

Bluetooth is a promising recent radio technology for ad-hoc networking which is based on connecting together piconets, to form a scatternet. Due to mobility of the nodes of a piconet, forming an optimal scatternet for a given traffic pattern may be not enough, rather a scatternet that best supports traffic flows as they vary in time is required. However, much of the research attention has spanned the formation of the scatternet from the isolated Bluetooth nodes, the design and standardization of routing in Bluetooth scatternet, medium access control protocols, and wireless channel allocation algorithm. Instead, in this paper, a polynomial time algorithm capable of dynamically configuring the formed scatternet topology to the existing traffic connections between scatternet nodes is proposed. The periodic configuration of the scatternet topology to the traffic connections enables the routing algorithms to identify shorter paths between communicating scatternet nodes. This will allow higher network throughput, lower packet delivery delay, and reduced communication overhead. Simulation results illustrate the performance of the configured scatternet.

Keywords: Ad-hoc networking, Bluetooth, Scatternet configuration, New Protocol

I. Introduction

Bluetooth is a short-range, low power wireless communication technology with ad hoc networking capabilities intended to replace the cable connecting portable and/or fixed electronic devices. At the present time, the utilization of intelligent handheld devices became part of the everyday experience. It is predicted that 4th generation wireless system will extensively rely on the unlicensed operations provided by ad hoc communication. The Bluetooth technology, operating in the unlicensed ISM (Industrial-Scientific-Medical) band, as described in the specification of Bluetooth system version 2.0 [1], is expected to be one of the most promising enabling technologies for ad hoc networks and pervasive computing.

According to the Bluetooth specification version 2.0, when two Bluetooth nodes, which are within each other communication range, want to setup a communication link, one of them must assume the role the master of the communication while the other becomes its slave. Thus a single hop communication network is formed which is known as piconet and may include several slaves, no more than 7, of which can be actively communicating with the master at the same time [2]. The specifications allow each node to assume multiple roles. A node can be a master in one piconet and a slave in one or more other piconets or a slave in multiple piconets. There is no restriction about the role of a node in a scatternet except that a node can not be master of more than one piconet at a time. Devices with multiple roles act as gateways to adjacent piconets thus forming a multi-hop ad hoc network called a scatternet. Although the specification defines how to organize Bluetooth enabled devices in a network but the problem of efficient scatternet formation is not addressed in the latest version of the specification [1]. As a result of investigation, it is found that there is a strong relationship between the path length connecting communicating devices and the overall throughput and power consumption of the scatternet. Intuitively, if a data packet has to cross many links to reach its destination it will occupy a large amount from the limited capacity of the links on its route, which then cannot be used by other traffic flows. Therefore, the lower the number of links crossed by a packet, the higher the unallocated amount of bandwidth. In this paper this issue is addressed. The main objective of this paper is to propose an algorithm that configures the topology of Bluetooth scatternet based on the traffic flows in the network, such that the new topology allows a higher overall throughput in the scatternet.

The remainder of this paper is structured as follows. In Section 2 we present an overview of related work available in the literature. Section 3 introduces the notation used in this paper to represent scatternet and formulate the addressed problem specifically. In Section 4 the proposed scatternet configuration algorithm is discussed. Finally in Section 5 we present numerical results. Section 6 concludes the paper.

II. Related Work

The area of ad hoc networking has gathered significant research interests in recent years. Many studies have concentrated on the routing issues of ad hoc networks [3,4]. An ad hoc network can be modeled as a graph such that the in-range nodes are adjacent. For example, simulation-based studies [5,6] of ad hoc routing protocols have been conducted with a link-layer model based on or similar to the IEEE 802.11b standard. An ad hoc network based on Bluetooth, however, brings new challenges. There are specific Bluetooth constraints not present in other wireless networks. So the research related to Bluetooth network gives a new dimension in ad hoc network. Physical layer properties of the Bluetooth radio have been studied thoroughly in [7]. Several other researches [8] were conducted on the scheduling discipline used by a piconet master in polling the slaves.

The main aim of New protocol is to build up a connected scatternet in which each piconet has no more than seven slaves [9]. The formed scatternet contains the following properties: small number of piconets for minimizing inter-piconet interference and small degrees for the devices to avoid networks bottlenecks. In addition, according to the simulations, the diameter of the scatternet, which corresponds to the maximum routing distance between nodes, is about *O (log n)* [9].

Casba, Roberto and Carla configure the scatternet based on traffic by reducing the number of hops between communication peers [12]. There approach consists of modifying the nodes links and roles, and in searching randomly for such a scatternet topology that minimize the distance (measured in hops weighted with the corresponding traffic intensity) between every source-destination pair. The main drawbacks of the optimization procedure is the random reconfiguration (i.e., moves of nodes), therefore its long execution time. Moreover the procedure was centralized by nature. But the approach dynamically adapts a Bluetooth scatternet topology to changing traffic condition. According to simulation, they showed that slave reconfiguration increase the number of piconets, but this is compensated by master reconfiguration, thus obtain an overall hop reduction between all communication peers.

III. Problem Formulation

(a) Scatternet Representation

Let N be the set of nodes in the scatternet, M the set of all masters, and S the set of all slaves. Notice that pure masters are not elements of S, and $S \cap M \neq \phi$ if there are master & bridge nodes in the scatternet. We denote with C the set of traffic connections in the scatternet. $R = \{r_{ij}^{sd}\}$, the routing matrix, stores the path between each source-destination pair $(s, d) \in C$; we have $r_{ij}^{sd} = 1$ if connection (s, d) is routed on arc (i, j) or $r_{ij}^{sd} = 0$ otherwise. $T = \{t_{sd}\}$ is the traffic matrix with $t_{sd} \in [0, 1]$ indicating the intensity of the data flow on the connection (s, d). $t_{sd} = 0$ means that there is no traffic flow between the nodes s and d. $H = \{h_{sd}\}$, the hop matrix, contains the minimum number of hops between any connection $(s, d) \in C$. This also specify the route of traffic intended for destination d form source s. The link matrix $L = \{l_{ij}\}$ is defined as $l_{ij} = 1$ if i is master of j or $l_{ij} = 0$ otherwise. The link matrix indicates the master-slave connections in the scatternet. Link matrix properties are explained below and summarized in Table 1.

1) A *pure master* has on its row one entry equal to 1 for each of its slaves and has all 0 entries on its column (e.g., node 2 in Figure 1).

2) A *pure slave* has one entry equal to 1 on its column corresponding to its master (e.g., node 6 in Figure 1).

3) A *slave & bridge* has on its column exactly one entry equal to 1 for each of its masters (e.g., node 4 in Figure 1).

4) A *master & bridge* node has one entry equal to 1 for each of its slaves on its row and for each of its masters on its column (e.g., node 1 in Figure 1).

Table. 1. Link matrix properties based on nodes' role

Role of node k	Property
Master	$\sum_{j=1}^{N} l_{kj} \geq 1$
Pure Slave	$\sum_{i=1}^{N} l_{ik} = 1$
Slave & Bridge	$\sum_{i=1}^{N} l_{ik} \geq 2$
Master & Bridge	$\sum_{j=1}^{N} l_{kj} \geq 1$ and $\sum_{i=1}^{N} l_{ik} \geq 1$
Free node	$\sum_{j=1}^{N} l_{kj} = 0$ and $\sum_{i=1}^{N} l_{ik} = 0$

5) A *free node* – node not belonging to any piconet – has all 0s on both, its row and column.

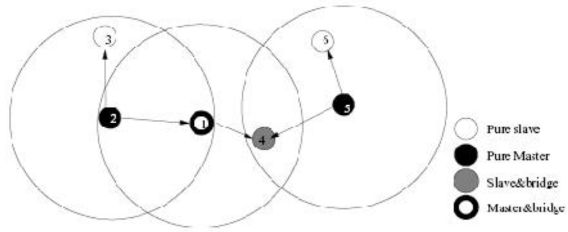

Fig. 1. Example scatternet with three piconets

The link matrix is a square matrix with as many rows and columns as nodes in the scatternet. The researcher assigns to each row and column the scatternet node with the corresponding identifier (identifiers from 1 to $|N|$). Using the link matrix scatternet representation, link creation or deletion is as simple as switching the appropriate position of the matrix from 0 to 1 and vice-versa. Role modifications happen automatically, as the number of 1s changes on the rows and columns of the node. The above stated properties are very rewarding when modeling scatternet reconfigurations.

(b) Problem Configuration

Using the notations from the previous section now it is possible to define function F as

$$F = \sum_{(s,d) \in C} t_{sd} h_{sd}$$

F is the sum of the number of hops weighted with the traffic intensity between all source-destination pairs in the scatternet. The objective is to solve the following minimization problem,

P : Min F

Subject to the following constraints [10]

- A piconet must contain one master and up to 7 slaves $\sum_{j=1}^{N} l_{kj} \leq 7, \forall k \in M$
- If i is master of j, then j can not be master of i $l_{ij} + l_{ji} \leq 1, \forall i, j \in N; i \neq j$
- Traffic between source s and destination d can be routed through edge (i, j) only if i and j communicate, i.e. either i is assigned to j, or j is assigned to i $r_{ij}^{sd} \leq l_{ij} + l_{ji} \quad \forall (s, d) \in C, \forall i, j \in N$

- All traffic between two nodes is routed through a minimum length path, with no loops. The selected path may not necessarily be the same in both directions, if more than one minimum length path exists. $h_{sd} = h_{ds} \ \forall (s, d) \in C$ $\quad \sum_{i,j \in N}^{N} r_{ij}^{sd} = h_{sd} \ \forall (s, d) \in C$
- Standard constraints used for routing should also be considered

IV. Proposed Configuration Procedure

The configuration procedure runs on a selected node, possibly with strong computational power, capable of collecting all the necessary data about the scatternet nodes. It coordinates various kind of modifications performed on the scatternet topology, aimed at reducing the number of hops between communicating nodes. The node which executes the configuration procedure collects relevant information about all of the scatternet nodes, such as the identity and role of the nodes and their neighbors, masters and communication peers, and feed it into the configuration algorithm. The configuration algorithm is presented in Figure 2 and Figure 3. Before going details about the configuration algorithm a set of terms used in the algorithm are disclosed.

Node: one piconet consisting of at most 7 slaves and a master.

Link: Link represents the communication path between any two Bluetooth devices. If there is a bridge between two piconets then there is a link presents between these two nodes which represent the two piconets respectively.

Link Weight: The sum of all traffic that passes through the link due to the traffic flows of the scatternet. Link weight represents the cumulative intensity of traffic on the specified link.

At the beginning of the configuration procedure several initializations are performed. First the initial state of the link matrix L is saved (line 2 in Figure 2). In line 3, the list of piconets is acquired as the configuration procedure will treat each piconet as a node to take the configuration decision. To identify the link used to flow the traffic it is need to know the routing information.

```
1.  CONFIGURE SCATTERNET
2.  L_init = L
3.  piconetlist = list of piconets
4.  Generate_Route_Matrix()
5.  for all i, j
6.      Link_Weight_ij = 0
7.  end
8.  for all (s, d) ∈ C
9.      for all r_ij^sd = 1
10.         Link_Weight_ij = Link_Weight_ij + t_sd
11.     end
12. end
13. Sort Link_Weight in descending order
14. if (Such link present)
15. then
16.     for each link from the sorted Link_Weight
17.         Identify two piconets P1, P2 connected by such link
18.         Optimize_Me(P1, P2)
19.     end
20. else
21.     Do nothing
22. END CONFIGURE_SCATTERNET
```

Fig. 2. Pseudo-code of the configuration algorithm

In line 4, Generate_Route_Matrix function is called which filled the routing table. The details description of the function is shown in line 1 to 5 in Figure 3. In this case DSDV (destination sequence distance vector) routing is implemented (line 6 to 16 in Figure 3). In line 5 to 12 of Figure 2, the amount of total traffic carried out in each link of the scatternet is calculated. The configuration procedure solely depends on the traffic present in the network. Therefore the link weight of the link greatly affects the configuration procedure. After calculating the link weight the link is sorted in descending order which involve traffic between two piconets (line 13 in Figure 2). The highest weighted link is picked first and the piconets involved in this link are identified. These two piconets is then passed to the main optimization module to configure the scatternet such that *F,* the optimization function's value reduces. All these are done through line 14 to 21 in Figure 2.

The optimization module (line 17 to 28 in Figure 3) takes two piconets information as parameter and tries to optimize the communication between these two piconets. If it is successful to optimize then it will have immense effect on the overall scatternet performance. The module searches whether the bridge between these two piconets can be converted to the master of any piconet and in that case changes the topology of the selected piconet. This is done through changing the link matrix representing the scatternet topology.

```
1.  Generate_Route_Matrix ()
2.  for each piconet p1 in piconetlist
3.      Calculate_SingleSource_AllDestination (p1)
4.  end
5.  End Generate_Hop_Matrix

6.  Calculate_SingleSource_AllDestination (node p1)
7.  Set all piconets visited property to false
8.  ExpandingList = empty
9.  NextExpandingPiconet = p1
10. Add p1 to ExpandingList
11. while (ExpandingList != empty)
12.     do
13.         Add adjacent piconets of NextExpandingPiconet in ExpandingList
14.         Update R
15. end
16. End Calculate_SingleSource_AllDestination

17. Optimize_Me (node temp, node picked_node)
18. if (master of temp and picked_node is a bridge to other piconet)
19. then return
20. else if (master of temp node is not a bridge)
21. then  Release the master of temp node
22.         Assign the bridge between node temp and picked_node as master of temp
23.         Assign the released master as slave of temp
24. else
25.   Release the master of node picked_node
26.   Assign the bridge between node temp and
      picked_node as master of picked_node
27.  Assign the released master as slave of picked_node
28. End Optimize_Me
```

Fig. 3. Pseudo-code of the sub function of configuration algorithm

V. Simulation and Performance Evaluation

(a) Simulation Environment

The configuration algorithm is implemented in C #; additional library is used to represent the scatternet visually. Netron graph library v2.2 is used for this purpose [11]. The simulation is performed by constructing scatternet of varying nodes. Nodes can join in any preferred piconet of the scatternet and there can be present varying traffic between any two nodes of the scatternet. The traffic intensity on these connections is in between 0.1 and 0.8.

(b) Simulation Results

(1) Traffic Connection and Number of Hops

Initially an optimal scatternet generated from isolated Bluetooth nodes. A set of different traffic connection (seventy different communication peers) were initiated in the scatternet. The number of hops between these communication peers was calculated. The proposed configuration procedure was executed to form the configured scatternet. The number of hops between the same communication peers was calculated again. The result is showed in Figure 4.

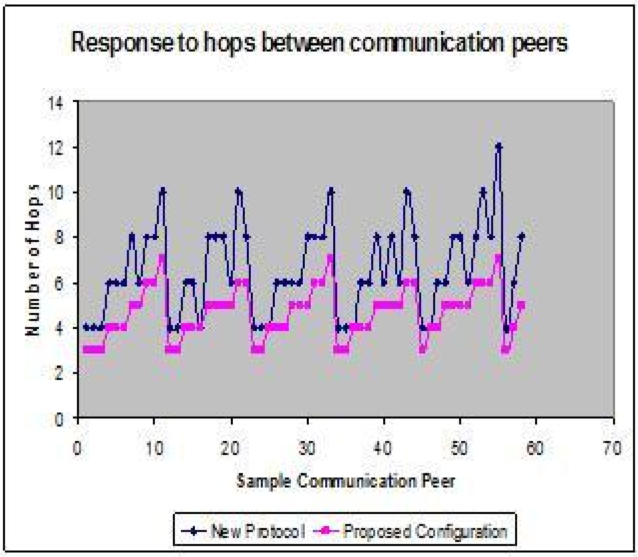

Fig. 4. Response to hops

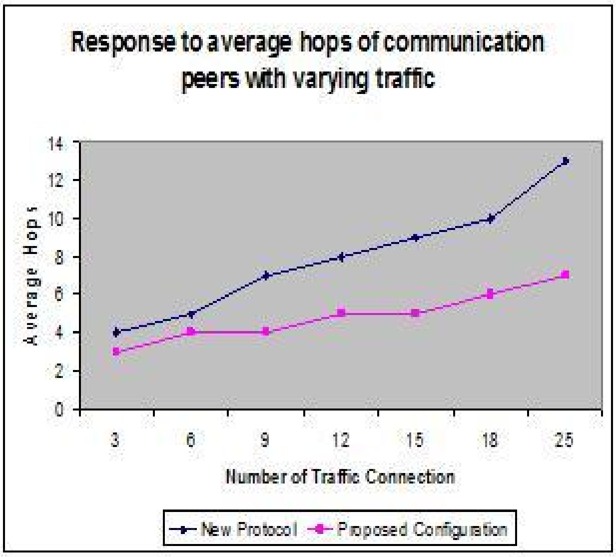

Fig. 5. Average hops with varying traffic

Figure 5 shows the response to average hops of communication peers with increasing number of traffic connections in the scatternet.

(2) Traffic Connection and Network Diameter

Network diameter is the maximum number of hops between any two nodes in the network. If the network diameter of a scatternet can be reduced then it will facilitate more traffic with reduced cost. The simulation result in Figure 6 shows the response to the scatternet diameter with the increase of traffic connection for a fixed node scatternet. In Figure 7 the response of the network diameter with increase number of nodes is plotted. Therefore, the configuration procedure performs better when the formed scatternet is relatively large.

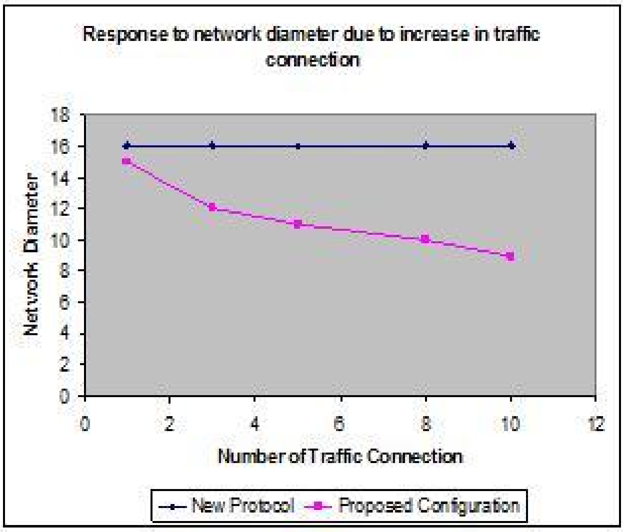

Fig. 6. Diameter with increase traffic connection

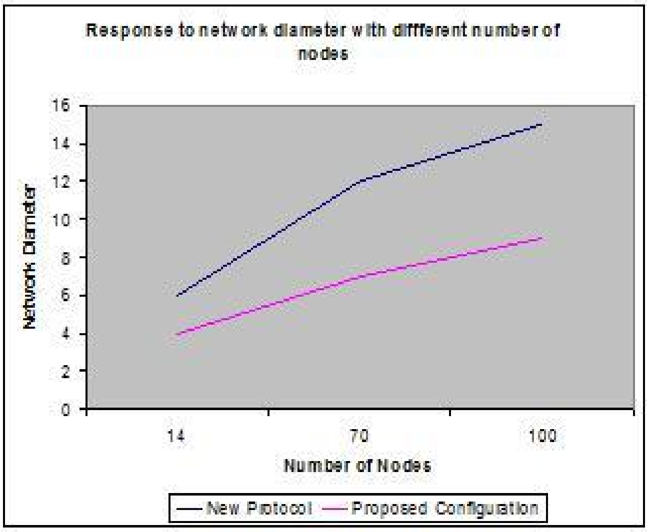

Fig. 7. Diameter with different number of nodes

(3) Traffic Connection and Minimization Function Value

The objective of the scatternet configuration is to minimize the value of predefined minimization function (F) as mentioned in Section 3.2. In Figure 8, the simulation outcome of the effect on F for the proposed configuration procedure with dynamic traffic connection is presented. The upshot of the simulation shows that with large traffic the value F reduce more with the proposed configuration algorithm.

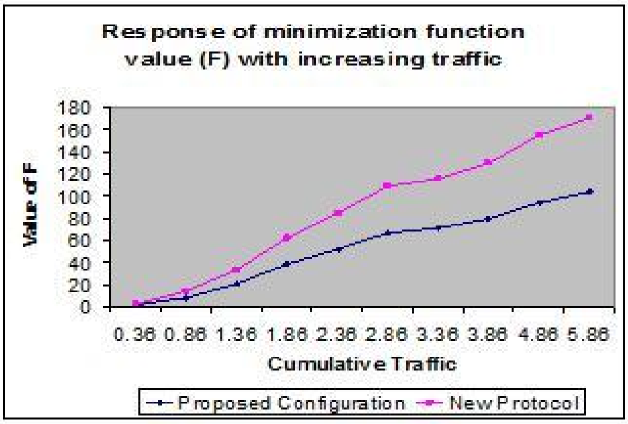

Fig. 8. Minimization function value (*F*)

	Proposed Configuration	Configuration Described in [12]
Number of Piconets	Does not change	Increase by 15.7%
Number of Slave	Increase	Decrease
Number of Bridge	Does not change	Increase
Algorithmic Complexity	Polynomial	Exponential

Fig. 9. Overall performance Comparison

(4) Overall Performance

The art of the optimization module is in its reduced algorithmic complexity. The optimization module's complexity is linear. If there are *n* nodes in the scatternet then maximum number of piconets in the scatternet will be bounded by *n/k* according to the adopted scatternet formation algorithm, where *k* is the average number of slaves in each piconet. Therefore, if each piconet pair in the scatternet involve in traffic then total number of traffic link will be at most a fraction of n^2. As the links involved between two piconets are considered when calling optimization module, overall complexity of the scatternet configuration procedure becomes polynomial. Simulation result shows significant improvements in proposed configuration procedure over scatternet configuration described in [12]. The improvement is depicted in Figure 9.

VI. Conclusion and Future Scope

The paper presents the work on Bluetooth scatternet configuration. The issue of scatternet configuration based on active traffic of the scatternet through the reduction of path length between the communicating nodes is discussed extensively. In this purpose, a polynomial time algorithm was developed that configures the scatternet by means of reconfiguring the roles and links of the nodes so as to minimize the number of hops between communicating nodes in the scatternet. The simulation results of the configuration algorithm shows notable throughput gain over the initial formed scatternet. Moreover, the configuration algorithm does not increase the number of piconets in the scatternet which will lead low inter piconet interference. However simulation results vary with the initial structure of the formed scatternet, but it never degrades the performance of the scatternet. The centralized nature of the configuration procedure is its main weakness. Although, the proposed polynomial time configuration procedure improve the overall performance but it can be further improved through initiation of a decentralized solution.

----

1. Bluetooth SIG. *Bluetooth Specification.* V2.0. October 15, 2004.
2. Michael Miller. Discovering Bluetooth. SYBEX Inc., 2001, ISBN: 0-7821-2972-2.
3. Perkins C.E.. 2000, Ad Hoc Networking. Addison-Wesley, Reading, MA.
4. Toh. C.K. 2001, Ad Hoc Mobile Wireless Networks: Protocols and Systems. Prentice Hall, New York.
5. Broch J. D.A. Maltz, D.B. Johnson, Y.C. Hu, and J. Jetcheva. 1998, A performance comparison of multi-hop wireless ad hoc network routing protocols, Mobile Computing and Networking.
6. Das S. R., R. Castañeda, and J. Yan. 2000, Simulation-based performance evaluation of routing protocols for mobile ad hoc networks, Mobile Networks and Applications.
7. Zurbes S., W. Stahl, K. Matheus, and J. Haartsen. 2000, Radio Network Performance of Bluetooth, IEEE International Conference on Communications, 2000, 1563-1567 vol.3, New Orleans, USA, 18-22 June.
8. Johansson N., U. K¨orner, and P. Johansson. Performance Evaluation of Scheduling Algorithms for Bluetooth, Kluwer International Federation For Information Processing Series, Kluwer Academic Publishers, Norwell, MA, USA, 2000, 139-150 **159**, ISBN: 0-7923-8677-9.
9. Ching Law, Amar K. Mehta, and Kai-Yeung Siu. A new Bluetooth Scatternet Formation Protocol, Mobile Networks and Applications, 8, Issue 5, 485–498, ISSN: 1383-469X, Kluwer Academic Publishers, Hingham, MA, USA, October 2003.
10. Marco Ajmone Marsan, Carla-Fabiana Chiasserini, Antonio Nucci, Giuliana Carello and Luigi DE Giovanni. Optimizing the topology of Bluetooth wireless personal area networks, In INFOCOM, New York, USA, June 2002.
11. Netron graph library v2.2. http://Netron.Sourceforge.net. (September, 2007).
12. Csaba Kiss Kalló, Carla-Fabiana Chiasserini, Roberto Battiti, and Marco Ajmone Marsan. Reducing the Number of Hops between Communication Peers in a Bluetooth Scatternet, *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC),* 5, 1, Atlanta, Georgia, USA, 21-25 March 2004.