Mobile Information Systems 7 (2011) 93–106
DOI 10.3233/MIS-2011-0112
IOS Press

BGN: A novel scatternet formation algorithm for bluetooth-based sensor networks

Yongjun Li[a], Hu Chen[a], Rong Xie[a] and James Z. Wang[b,*]
[a]*School of Computer Science & Engineering, South China University of Technology, Guangzhou, 510640, China*
[b]*School of Computing, Clemson University, Clemson, SC 29634, USA*

Abstract. To address the unreliability and inefficiency of the existing algorithms for Bluetooth scatternet formation, this paper proposes a new efficient algorithm, Bluetooth Growing Network (BGN) algorithm, which constructs Bluetooth scatternet by adding reserve links among its branch nodes so that the resulting scatternet can maintain a high degree of connectivity while avoiding exccessive redundant links. Extensive simulation results demonstrate that the scatternet constructed by BGN algorithm is efficient in terms of the number of bridge nodes, average data transmission distance, network reliablity and overall network traffic.

Keywords: Wireless sensor network, BGT, BGN, scatternet

1. Introduction

In recent years, advances in wireless technology and mobile applications have made mobile devices an essential part of people's daily life. Many mobile applications, such as mobile databases and navigation [1–5], have been widely adopted. Recent advances in wireless ad-hoc and sensor networks further expand the mobile application arena because mobile devices can form personal area networks (PANs) without using the base stations and wired networks. More and more mobile applications [6–9], such as healthcare, fitness, gaming, etc. are now based on wireless ad-hoc and sensor networks. Bluetooth is a wireless protocol utilizing short-range communication technologies to facilitate data transmission over short distances between fixed and/or mobile devices. Because of its excellent wireless properties, such as low cost, low power consumption and good anti-jamming performance, and its advantages in network facilitation, Bluetooth has become an attractive communication protocol for wireless ad-hoc and sensor networks.

To construct a Bluetooth network using Bluetooth protocol, devices are first grouped into piconets, each of which consists of no more than eight devices due to a 3-bit address space. These piconets then interconnect with each other by one member which is also another piconet's member. The device participating in two piconets forms a "bridge" between these two piconets and relays data between their members. All interconnected piconets form a scatternet to support communication among devices in the network.

[*] Corresponding author. E-mail: jzwang@cs.clemson.edu.

A tough challenge for a scatternet formation algorithm is to find an optimal network configuration in terms of the number of bridges and the connectivity degree of each bridge to achieve the best network performance without affecting the reliability and connectivity of the network. In general, the number of bridges and the connectivity degree of each bridge affect the performance and survival time of the Bluetooth-based sensor networks. With more bridges, the network is usually more reliable and the average data transmission distance is shorter. However, the bridge node is responsible of forwarding packets between two or more piconets in a time-shared manner. Thus, the bridge nodes usually have shorter survival time because they consume more energy due to the extra communication responsibilities. Therefore, too many bridge nodes often result in the decline of the network overall survival time.

To address the weaknesses of the existing algorithms, this paper proposes a novel algorithm, Bluetooth Growing Network (BGN) algorithm. This algorithm constructs the scatternet from a tree structure by adding reserve links among its branch nodes. It has the merits of both the mesh-based and the tree-based Scatternet formation algorithms. The resulting scatternet can maintain a high degree of connectivity while avoiding excessive redundant links. Our simulation studies demonstrate that this algorithm outperforms existing algorithms in terms of the number of bridges, reliability, and average transmission distance.

The rest of this paper is organized as follows. We first discuss related work in Section 2. Then, in Section 3, we define the Bluetooth Growing Tree (BGT) and how to construct the BGT, and propose the BGN scatternet formation algorithm. In Section 4, we compare the performance of our BGN algorithm with the tree-based and mesh-based algorithms through extensive simulation studies. Finally, we give concluding remarks and discuss the future work in Section 5.

2. Related work

Recently, researchers proposed many scatternet formation algorithms which can be roughly divided into three categories: tree-based algorithms [10–13], ring-based algorithms [14–16] and mesh-based algorithms [17–22].

The scatternet constructed by a tree-based algorithm has a single root node and descendant tree nodes. A bottleneck can easily form at the trunk node of the tree, although such scatternet has a logarithmic average path length and its data routing algorithm is very simple. Zaruba et al. [10] proposed a tree-based algorithm (BlueTree Algorithm), which utilizes tree structure characteristics (N devices only need N-1 links to interconnect with each other) to minimize the number of communication links. However, Not only such a tree structure has poor reliability, but also bridges can easily become the communication bottleneck. Rashid Bin Muhammad [11] proposed a method for range assignment problem on the Steiner tree based topology, but the method should throw away nodes that are too far from neighbor nodes which lead to too many dense nodes for the connectivity. Tang et al. [12] proposed a tree-based heuristic algorithm for mobile and pervasive computing based on special network processors. Saginbekov et al. [13] also proposed a tree-based shortest path algorithm, trying to realize the energy saving through minimizing the energy consumption during communication.

Ring-based algorithms try to alleviate the bottleneck problems in tree-based algorithms while maintaining simple scatternet routing properties. However, they suffer from network partition problems and significantly longer average path length. Huang et al. [16] presented a ring structure called Solidring and a routing algorithm to lower transmission time delay. But the assumption of devices can communicate directly with one another having strict restriction.

Thomas et al. [17] proposed an algorithm (BlueNet Algorithm) based on a mesh topology to construct a smooth network architecture to enhance network reliability. But due to the large number of piconets and redundant communication links, excessive interference among piconets often leads to low communication performance of the self-organized network. Song et al. [18] adopt the well-known de Bruijn graph [19] to form the backbone of Bluetooth scatternet, in which for every pair of nodes it can find a path with at most O(log n) hops without any routing table and the congestion of every node is at most $\mathrm{O}(\frac{\log n}{n})$. Chang et al. [20] proposed a traffic-aware restructure protocol for partially restructuring a piconet or a pair of two neighboring piconets by applying role switch mechanism to reduce the route path length, however, the frequent role switch will result in more transmission time delay. Zaguia et al. [21] introduced a new protocol based on maximal independent sets (BlueMis) attempt to simplify the BlueMesh procedure [22], but the number of slaves for each node is not limited, and in some cases, e.g. complete graph, one node may be selected as slave to all other nodes.

3. The BGN algorithm

In this section, after defining the Bluetooth Growing Tree and discussing the basic ideas on how to construct the scatternet using BGT, we propose a distributed implement of our BGN algorithm.

3.1. Definitions

Definition 1 (**Piconet**)**:** A piconet is a network consisting of a master node and at most seven slave nodes dominated by this master node. A piconet P_i can be represented as:

$$P_i = \{(M_i, S_{i,j})|1 \leqslant j \leqslant 7\},$$

where M_i is the master node and $S_{i,j}$ is the slave nodes dominated by M_i.

Definition 2 (**Scatternet**)**:** A scatternet is formed by two or more independent, asynchronous piconets overlapped in time and space. Formally, a scatternet can be presented as:

$$S = \{\cup P_i\}, i \in [1, n], \quad n \geqslant 2,$$

where $\forall k, j, k \neq j \wedge \exists P_k \xrightarrow{path} P_j$, i.e., given any two piconets P_k and P_j in S, there exists a path from P_k to P_j.

Definition 3 (**Bridge**)**:** A node in a scatternet S is a bridge node if and only if it is shared by two or more piconets. It receives data from one piconet and forwards the data to another piconet. A bridge node can be slave nodes in two separate piconets (slave/slave bridge), or a master node in one piconet and a slave node in another piconet (master/slave bridge). A set of bridge nodes in a scatternet S can be presented as: $B = \{t_k|t_k \in P_i \cap P_j\}$, where $i \neq j$ and $P_i, P_j \in S$.

As shown in Fig. 1, piconet P1, P2 and P3 are partially overlapped in time and space, where M_2 is the master node in piconet P2 and it controls the slave node S_1 and bridge nodes B_{21} and B_{23}.

3.2. Growing tree based bluetooth scatternet

In this subsection, we introduce the concept of Bluetooth Growing Tree and discuss its construction process.

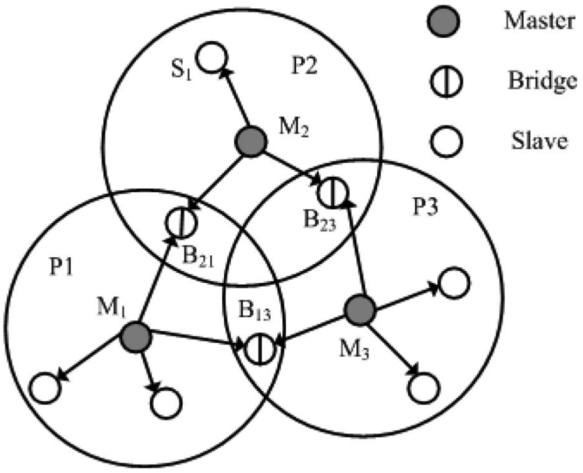


Fig. 1. The topology of a Bluetooth scatternet.

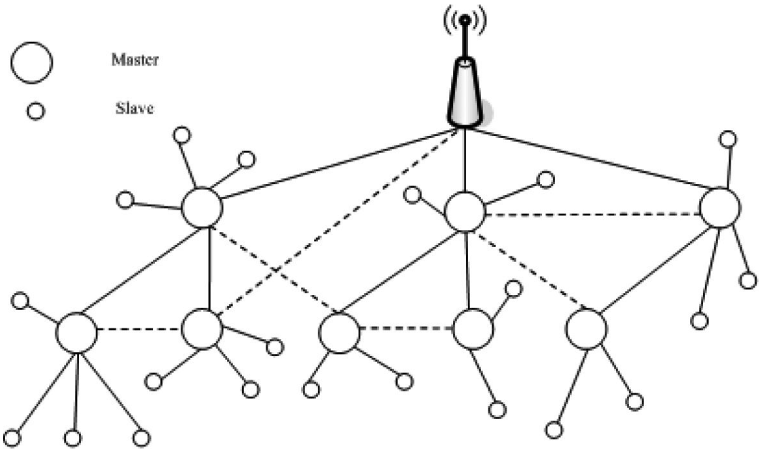


Fig. 2. The structure of a Bluetooth growing tree.

3.2.1. Bluetooth growing tree construction

The BGT construction process starts with a random selection of a seed node. This seed node first selects up to 7 of its neighboring nodes to become its children. These child nodes then serve as seed nodes to find their neighboring nodes which are not yet selected by other nodes to be their children respectively. Once a node is selected to be a child by another one, it becomes a seed node and can compete with other nodes to select available nodes as its children. This process continues until all nodes are connected. Because this construction process essentially grows a tree through constantly bifurcating, we call the resulting tree as a Bluetooth Growing Tree (BGT), as shown in Fig. 2.

In Fig. 2, the topology formed by solid lines and their associated nodes is a BGT, in which all leaf nodes are slave nodes in their associated Piconet. A branch node plays two roles. It is a slave node in its parent Piconet, which is formed by its parent node during the growing tree process. On the other hand, it is a master node of the Piconet formed by this branch node when it grows the tree to its direct neighbors.

3.2.2. Scatternet formation through BGT

Using the Bluetooth protocol, one device may directly connect with up to seven neighboring devices. Therefore, it is possible that some branch nodes may only used a portion of their available wireless capacities during the BGT construction. Therefore, it is possible to add more links between the branch nodes as shown by the dotted lines in Fig. 2.

Adding links between the branch nodes makes the BGT a mesh-like scatternet, which is much more reliable than the BGT. In addition, edges between the branch nodes can help the traffic routing in the scatternet. Since this scatternet is constructed through a Bluetooth Growing Tree, we call it a Bluetooth Growing Network (BGN).

After adding the links between branch nodes, the scatternet grows into a mesh structure from the original BGT structure. Although, the number of backbone nodes remains the same, the reliability and throughput of the scatternet has significantly improved over the original BGT structure.

Definition 4: Given graph $G(V, E)$ and its Growing Tree $T(V', E')(V' \subseteq V$,$E' \subseteq E)$, the BGN is defined as:

$$\begin{aligned}&BGN = \{T \cup e(u, v) | e(u, v) \in E - E', u, v \in M,\\&\deg ree(u) \leqslant 7, \deg ree(v) \leqslant 7\}\end{aligned}$$

where M is the set of branch nodes of T, and *degree*(u), *degree*(v) are the numbers of connections associated with the branch nodes u and v respectively.

Definition 5: Candidate Edge: Given a Growing Tree T, the set of candidate edges for a branch node u is:

$$ce_u = \{e_{u,v} | length(e_{u,v}) \leqslant R, u \in M, v \in M\},$$

where R is the communication radius of the node, and M is the set of branch nodes.

3.2.3. Link capacity reservation

There are some restrictions for BGT growing into BGN. Based on Definition 5, as long as the wireless distance between two branch nodes is less than or equal to R, there exists a candidate edge. However, the maximum link capacity of each node is 7. If any branch node has reached its link capacity, no candidate edge can be added between this node and any other branch nodes. Since the BGT construction process is a competing process, it is highly possible a branch node has connected with 7 child nodes during the BGT growing process. This situation happens more often when the node density is high, limiting the number of candidate edges that can be added to the BGN.

To allow a BGT to grow into a BGN, we can limit the number of links of a branch node during the BGT growing process. That is, we reserve some link capacity for a branch node to add links to other branch ones. A smaller link capacity reservation for a branch node means it has less capacity to add links to other branch nodes and the resulting BGN is less reliable. On the contrary, a larger link capacity reservation for a branch node means it has more capacity to add links to other branch nodes. Having more links among the branch nodes will lead to more bridge nodes and affect the survival time of the BGN.

3.3. BGN algorithm

The BGN algorithm consists of two phases: (1) constructing the BGT, and (2) adding candidate edges between branch nodes.

The Bluetooth growing tree construction starts from a root node, called blueroot. This root node pages its neighbors one by one and becomes the master of the nodes that respond the paging messages. If a node is paged and is not yet a part of any piconet, it responds the paging message and becomes the slave of the paging node; otherwise, it may not respond to the page or may respond the paging with an "already taken" message to stop further paging messages sent to it. Once a node becomes a slave in a piconet, it starts paging all its neighbors one by one. This procedure is recursively repeated until all nodes join a piconet [23].

After completing the Bluetooth growing tree construction, each branch node runs the following BGN procedures to add candidate edges [24,25]. These procedures are essentially a competition process. The delay time for the branch node i during the competition is defined as:

$$t = \frac{1 + |B_i|}{2T} \tag{1}$$

where T is competition cyclic time and B$_i$ is the number of piconets connected of nodes i.

```
Procedure for all nodes during time period $T_q$
s.1 if the number of slave $\geqslant$ 7 then exit;
s.2 $t$ ←(1+Bi)/2$T$
s.3 schedule a delay in $t$ second
s.4 if receive any QUERY(ALL,sender) msg in t second then
          cancel the scheduler
          parent←sender
          role←loser
      else
          broadcast Query(ALL,myaddr) msg
          role←winner;
s.5 wait until $T_j$ period arrives
Procedure for nodes where role = winner during time period $T_j$
s.1 if receive JOIN(parent, sender) msg && sender not in $B_i$
          add a link to the sender
          $B_i \leftarrow B_i \cup$ sender
          Send Accept(sender,myaddr) msg
s.2 if receive none of JOIN msg then
          exit
      else
          wait until next $T_q$ period arrive
Procedure for nodes where role = loser during time period $T_j$
s.1 $t$ ←rand(0,$T_j$/2)
s.2 delay $t$ second
s.3 send JOIN(parent,myaddr) msg
s.4 if receive ACCEPT msg from parent then
          add a link to parent
          $B_i \leftarrow B_i \cup$ sender
s.5 wait until next $T_q$ period arrives
```

As shown in the above procedures, the BGN algorithm is a competition-based algorithm. It contains a query period T_q and a join period T_j in each competition cycle. Usually we have $T_q = T_j$. A branch node competes for sending the QUERY message in T_q and the JOIN message in T_j.

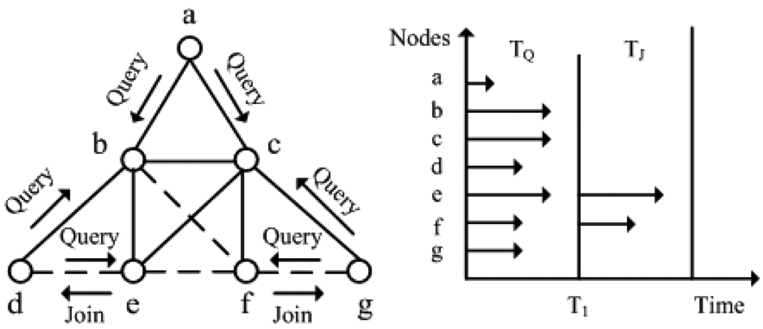


Fig. 3. An example of competition-based candidate edge adding.

During the query period T_q, each branch node computes its delay time t using Formula Eq. (1). If a branch node does not receive any QUERY message from its neighboring branch nodes in its delay time t, it is the winner in competition and will broadcast QUERY messages to its neighboring nodes.

Nodes receiving QUERY messages from other nodes are losers and will not broadcast QUERY messages to other nodes. Instead, these losers will send JOIN messages to the winner during period T_j. The winner will send an ACCEPT message to the node that sends a JOIN message to it the first time and is not subordinate to it. It will ignore other JOIN requests. A node receiving an ACCEPT message will add itself to the Piconet mastered by the node that sends the ACCEPT message, forming a link between two branch nodes. Each branch node repeats this procedure until the note reaches its link capacity or no JOIN message is received from its neighboring nodes.

Figure 3 shows an example of adding candidate edges among branch nodes. In Fig. 3, a solid line means that the connection is already established, while the dotted line represents a candidate edge. As shown in the right part of Fig. 3, each branch node computes its delay time according to Formula 1 during period T_q. Node a has the shortest delay time, and nodes b, c, e all have the longest delay time. During period T_q, only nodes a, d, f and g succeed in competition. Other nodes receive QUERY messages broadcasted by nodes a, d, f and g and will not send QUERY messages. During period T_j, nodes that received QUERY messages will check whether they have connected to source nodes, if not they will send JOIN message to source nodes. In Fig. 3, after competition, candidate edge (d, e) and (f, g) will be added to the BGN.

4. Performance evaluation

In this section, we use simulation to evaluate the performance of our BGN algorithm. We compare the scatternets constructed by our algorithm with those constructed by Bluetree [10] and BlueNet [13] algorithms respectively in terms of the number of candidate edges, the number of backbone nodes, the average transmission distance, network reliability and throughput. We conduct the simulation using the NS-2 network simulator. We assume there are N nodes randomly distributed in a 100 m $\times$ 100 m square area and each node has communication radius R $=$ 40 m. We vary N from 70 to 250 and observe the characteristics of the scatternets constructed by three different algorithms. For each fixed N, we repeat the simulation for 100 times and report the average values of these 100 simulation runs.

4.1. The optimal link capacity reservation

We first find the optimal link capacity reservation value θ and investigate its impact to the performance of our proposed BGN algorithm.

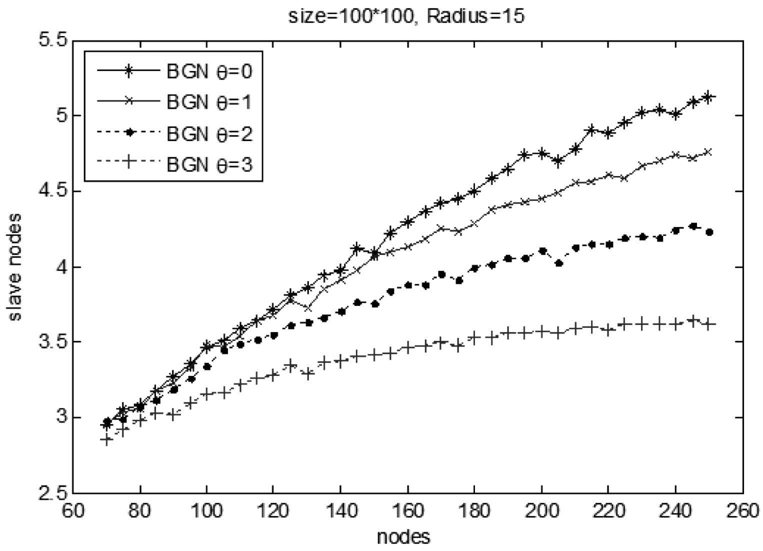


Fig. 4. The number of average slave nodes of piconet of Bluetooth growing tree.

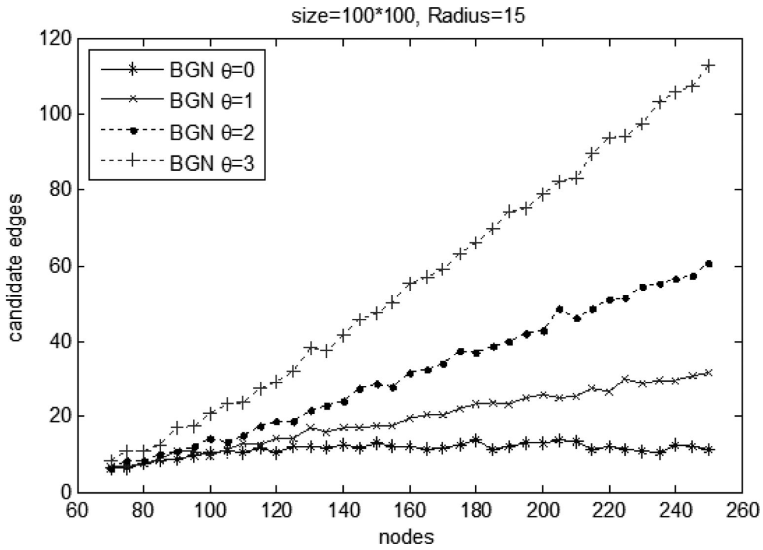


Fig. 5. The number of candidate edges of Bluetooth growing tree.

Figure 4 depicts the average number of slave nodes per piconet in the BGT under different θ values. Although the average number of slave nodes increases with the node density increases, this number does not exceed 5.5 when total number of network nodes in a 100 m $\times$ 100 m area reaches 250. This means there is still available wireless capacity in branch nodes to add candidate edges. In general, the greater the θ value, the more available wireless links are reserved for candidate edges.

Figure 5 shows the number of candidate edges in the BGN under different θ values. When the node density is low (N $\leqslant$ 120), the available candidate edges are limited because the physical distances between branch nodes are high. Therefore, the numbers of candidate edges added to the BGN are very close when $\theta = 0$, 1 and 2. But as the node density increases, the physical distances between the branch

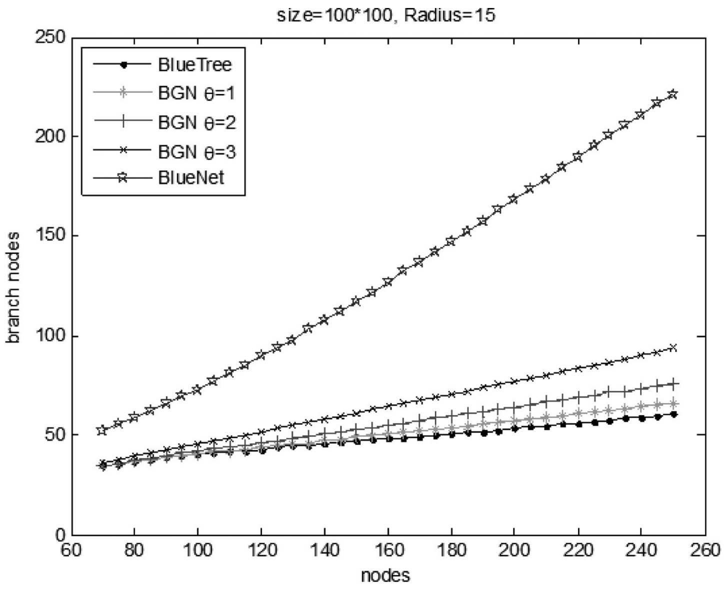


Fig. 6. The number of branch nodes of BlueTree, BlueNet and BGN.

nodes decrease, resulting more available candidate edges among the branch nodes. Therefore, with a higher link capacity reservation, more candidate edges can be added into the BGN. For example, when $N = 250$, given a link capacity reservation value $\theta = 3$, 112 candidate edges can be added. Meanwhile, when $\theta = 2$ and $\theta = 1$, only 60 and 38 candidate edges can be added respectively. Thus, when the node density is high, the link capacity reservation value θ has a significant impact on the number of candidate edges that can be added to the BGN. If we don't reserve any link capacity, i.e., $\theta = 0$, almost all branch nodes become "full" during the BGT growing process, only about 10 candidate edges can be added to the BGN in our simulation. Therefore, it is very important to reserve the link capacity for branch nodes to add candidate edges among them.

Now we compare the numbers of branch nodes in the Bluetooth Scatternets constructed by BGN, BlueNet and BlueTree algorithms respectively.

As shown in Fig. 6, the number of branch nodes in the Scatternet constructed by BlueNet algorithm is the largest, while the BlueTree algorithm requires the least number of branch nodes when constructing the Scatternet. The number of branch nodes in the Scatternet constructed by our BGN algorithm is very close to that constructed by BlueTree algorithm. For instance, when $N = 200$, BlueTree algorithm requires 39 branch nodes as shown in Fig. 6. Since it is a tree structure, there are only 38 edges among those branch nodes. When $\theta = 1$, the number of branch nodes in the Scatternets constructed by our BGN algorithm is only 42. Therefore, the number of edges in the BGT is 41 since BGT is also a tree structure. However, the number of candidate edges in the constructed scatternet is 19. It means the total number of edges in the scatternet constructed by BGN algorithm can be 62% more than that constructed by BlueTree algorithm with only a slight increase on the number of branch nodes. Therefore, BGN algorithm constructs a much better scatternet than BlueTree algorithm does in terms of the number of links in the backbone network. Thus, the resulting scatternet has better network throughput and reliability.

The results in Figs 5 and 6 indicate that setting the link capacity reservation value θ at 1 or 2 for BGN algorithm is better choices.

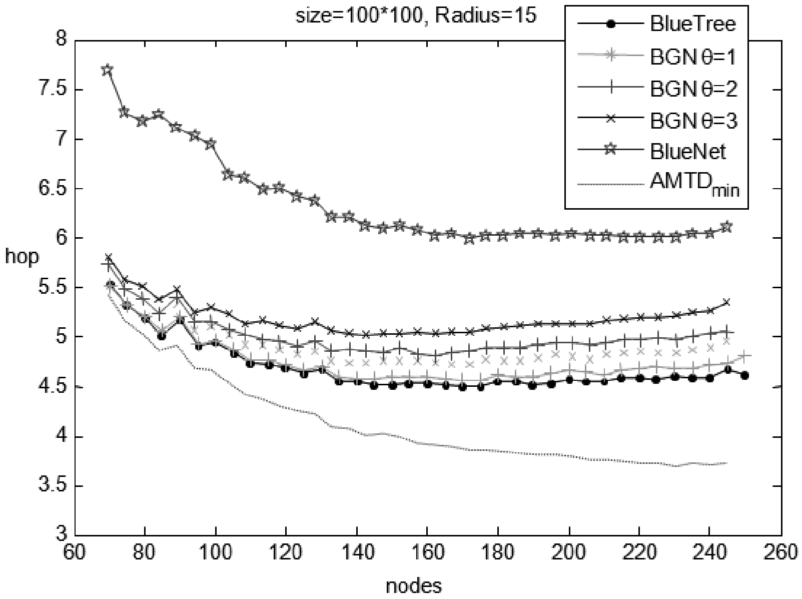


Fig. 7. Average minimum transmission distance under various node densities.

4.2. Average transmission distance

Let min_hop(i, 0) represent the minimal number of hops from node i to the base station. Then the average minimum transmission distance (AMTD) is computed as: $AMTD = \left(\sum_{i=1}^{N} \text{min\_hop}(i, 0)\right) / N$. Obviously, *AMTD* determines the energy cost and data transmission delay of a BGN. Given an arbitrary connected network, it is not hard to calculate the optimal *AMTD*.

Figure 7 shows the *AMTD* values of the scatternets constructed by BGN, BlueNet and BlueTree algorithms under different node densities. The simulation results show that the BlueTree algorithm has the best performance in terms of AMTD while BGN algorithm achieves very similar performance when $\theta = 1$. The scatternet constructed by BlueNet algorithm has the largest *AMTD*, worst among all algorithms evaluated. We also calculated the lower bound of *AMTD*, demonstrated as *AMTD*$_{\text{min}}$ in Fig. 7.

4.3. Network reliability

Traditionally, the reliability of a network is measured by the number of dividing nodes in this network. Given a connected network, if cutting edges connecting to a node in the network will divide the network into two non-intersected sub-networks, this node is a dividing node of the network. In general, the more dividing nodes in a network, the less reliable it is. The scatternet constructed by BlueTree algorithm is a tree structure. All of its branch nodes are dividing nodes. The ratio of dividing nodes to branch nodes is 1 in such a network. This network is least reliable.

Figure 8 depicts the number of dividing nodes among the branch nodes of the scatternets constructed by different algorithms under various node densities. Figure 9 shows the ratio of dividing nodes to branch nodes in the scatternets constructed by different algorithms under various node densities.

As shown in Figs 8 and 9, the number of dividing nodes and the ratio of dividing nodes to branch nodes decrease as the node density increases. Although the number of dividing nodes and the ratio of

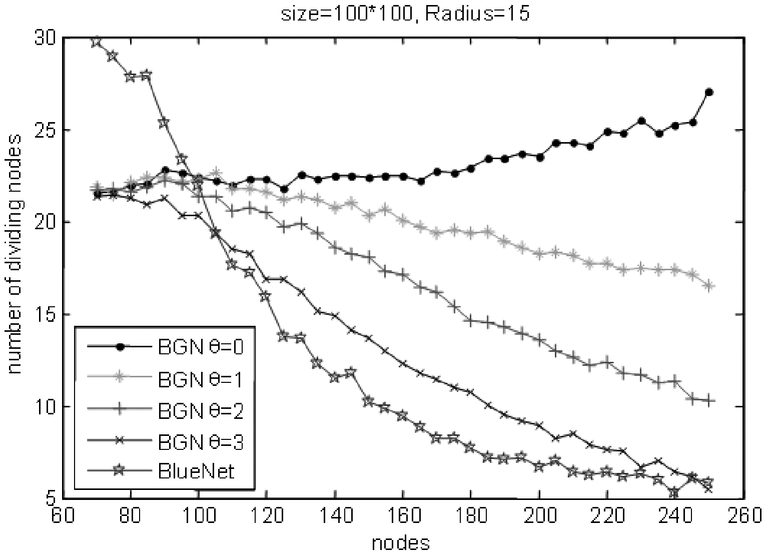


Fig. 8. The number of dividing nodes in branch nodes.

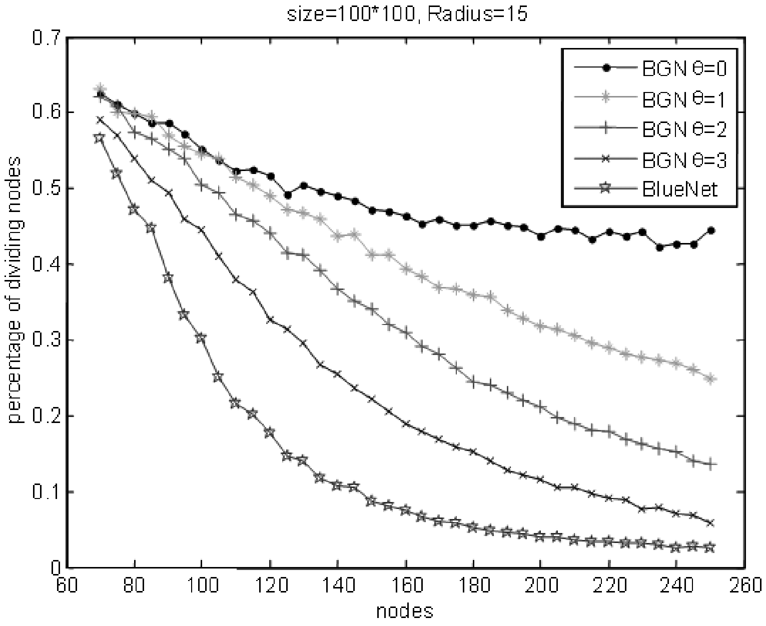


Fig. 9. The percentage of dividing nodes among the branch nodes.

dividing nodes to branch nodes in the scatternet constructed by BlueNet algorithm are slightly less than those by BGN algorithms, the number of branch nodes in the scatternet constructed by BGN algorithm is far less than that by BlueNet algorithm. Therefore, BGN algorithm can maintain a higher degree of network connectivity while avoiding excessive redundant links.

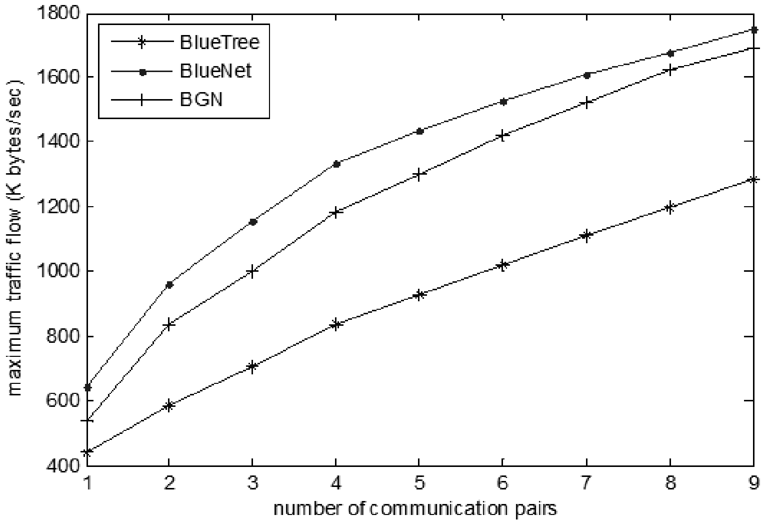


Fig. 10. Maximum traffic flows under various numbers of concurrent flows.

4.4. The maximum traffic flows

In most wireless sensor networks, nodes need to send the collected data to the clustering node in a short period of time. As wireless sensor networks often have only one clustering node, node resource constraints and link capacity limitations limit the amount information a scatternet can transfer within a unit time. Therefore, it is important to study the maximum information can be transmitted by the scatternet within a unit time. We call this the Maximum Traffic Flow.

Figure 10 presents the Maximum Traffic Flows of Scatternets constructed by BlueTree, BlueNet and BGN algorithms under different numbers of concurrent flows. Experimental environment is set as follows: There are 40 nodes randomly distributed in a square area of 20 m × 20 m. We assume the communication link radius $R = 10$ m and each node has a transmission bandwidth of B_0 = 1000 Kbps. For a branch node, the bandwidth cost of maintaining a link with its subordinate node is $\mathrm{B}_1 = 10$ Kbps and the bandwidth cost for maintaining a bridge between two Piconets is 100 Kbps. Thus, the communication capacity of a node i can be computed as:

$$C_i = \begin{cases} B_0 - n_s^i B_1 - (n_b^i - 1)B_2 & \text{i is a branch node} \\ B_0 & \text{otherwise} \end{cases} \tag{2}$$

where n_s^i denotes the number of the subordinate nodes for branch node i, and n_b^i denotes the number of piconets connected to this branch node.

As shown in Fig. 10, the scatternet constructed by BlueTree algorithm has the smallest maximum traffic flow while the scatternet constructed by BlueNet algorithm has the largest maximum traffic flow. The maximum traffic flow in the scatternet constructed by our BGN algorithm is only slightly smaller than that by BlueNet algorithm.

5. Conclusion and future studies

This paper proposed a novel algorithm, BGN algorithm, to construct Scatternet for Bluetooth-based sensor networks. By adding links among the branch nodes of a Bluetooth Growing Tree, the scatternet constructed by our BGN algorithm can maintain a high degree of connectivity while avoiding excessive redundant links among the branch nodes. Because our BGN algorithm has the merits of both BlueTree algorithm and BlueNet algorithm, the constructed scatternet demonstrates well balanced performance in terms of average minimum transmission distance, network reliability, and maximum traffic flow.

We are currently designing an efficient routing protocol for wireless sensor networks based on our BGN algorithm. We will also study the self-stabilization protocol of constructing the scatternet in a dynamic wireless sensor network environment where some sensor nodes may fail due to power failure or environment impact, or some sensor nodes may move around the targeted sensing area.

Acknowledgement

James Z. Wang's work is partially supported by NSF grant DBI-0960586. Yongjun Li's work is partially supported by Guangdong Science and Technology Department grant 2008B080703004, 2009A030302009 and Guangdong Administration of Work Safety grant x2jsB2080910.

References

[1] K. Xuan, G. Zhao and D. Taniar, *Bala Srinivasan: Continuous Range Search Query Processing in Mobile Navigation*, ICPADS, 2008, 361–368.

[2] Agustinus Borgy Waluyo, Bala Srinivasan, David Taniar: A Taxonomy of Broadcast Indexing Schemes for Multi Channel Data Dissemination in Mobile Database. AINA (1) 2004: 213-218

[3] Agustinus Borgy Waluyo, Bala Srinivasan, David Taniar: Research in mobile database query optimization and processing. Mobile Information Systems 1(4): 225-252 (2005)

[4] James Jayaputera, David Taniar: Data retrieval for location-dependent queries in a multi-cell wireless environment. Mobile Information Systems 1(2): 91-108 (2005)

[5] James Z. Wang, Zhidian Du and Pradip K Srimani. Cooperative Proxy Caching for Wireless Base Stations, International Journal of Mobile Information Systems, Volume 3, Number 1, pp. 1-18, 2007.

[6] X. H. Wang, M Iqbal, Bluetooth: Opening a blue sky for healthcare, Mobile Information Systems, vol 2 (2006), pp. 151-167.

[7] Barolli, L., Hsu, H., and Shibata, Y. Mobile Systems and Applications. Mobile Information Systems. vol 4, (2008), pp. 77-79.

[8] Vojislav B. Misic, Jelena V. Misic: Improving sensing accuracy in cognitive PANs through modulation of sensing probability. Mobile Information Systems 5(2): 177-193 (2009)

[9] A.B. Waluyo, R. Hsieh, D. Taniar and J.W. Rahayu, Bala Srinivasan: Utilising Push and Pull Mechanism in Wireless E-Health Environment, *EEE* (2004), 271–274.

[10] G.V. Zaruba, S. Basagni and I. Chlamtac, Bluetrees- Scatternet Formation to Enable Bluetooth-Based Ad Hoc Networks, *Proceedings of IEEE International Conference on Cominunications*, Helsinki, Finland, June 2001, pp. 273–277.

[11] R.B. Muhammad, Range assignment problem on the Steiner tree based topology in ad hoc wireless networks, *Mobile Information Systems* **5**(1) (2009), 53–64.

[12] F. Tang, I. You, M. Guo, S. Guo and L. Zheng, Balanced bipartite graph based register allocation for network processors in mobile and wireless networks, *Mobile Information Systems* **6**(1) (2010), 65–83.

[13] S. Saginbekov and I. Korpeoglu, *An Energy-Efficient Scatternet Formation Algorithm for Bluetooth-Based Sensor Networks*, Proceedings of the 2nd European Workshop on Wireless Sensor Networks, Istanbul, 2005, pp. 207–216.

[14] T. lin, Y. Tseng, K. Chang and C. Tu, Formation, Routing, and Maintenance Protocols for the BlueRing Scatternet of Bluetooths, *Proceedings of the 36th Annual Hawaii International Conference on System Sciences* (*HICSS03*), January 2003, pp. 313–322.

[15] T. Hassan, *Ring of Masters* (*ROM*) *A New Ring Structure for Bluetooth Scattemets*, Master' s thesis, American University of Beirut, Department of Electrical and Computer Engineering, February 2004.

[16] B. Huang, S. Ngah, H. Zhu and T. Baba, Solidring: A Novel Bluetooth Scatternet Structure, *International Journal of Computer Science and Network Security* **8**(9) (Sep 2008), 93–103.

[17] Z. Wang, R.J. Thomas, and Z.J. Haas, *Bluenet- a new scatternet formation scheme, Proceedings of 35th Hawaii International Conference on System Science* (*HICSS-35*), Big Island, Hawaii, Jan 2002, pp 61–69.

[18] W.Z. Song, X.Y. Li, Y. Wang and W. Wang, dBBlue: low diameter and self-routing Bluetooth scatternet, *Journal of Parallel and Distributed Computing* **65**(2) (2004), 178–190.

[19] Ralston, A. de Bruijn Sequences-A Model Example of the Interaction of Discrete Mathematics and Computer Science, *Math Mag* **55** (1982), 131–143.

[20] C.Y. Chang, C.T. Chang, S.C. Lee and C.H. Tseng, TARP: A Traffic-Aware Restructuring Protocol for Bluetooth Radio Networks, *International Journal of Computer and Telecommunications Networking* **51**(14) (2007), 4070–4091.

[21] N. Zaguia, Y. Daadaa and I. Stojmenovic, Simplified bluetooth scatternet formation using maximal independent sets, *Integr Comput -Aided Eng* **15**(3) (2008), 229–239.

[22] C. Petrioli, S. Basagni and I. Chlamtac, BlueMesh: Degreeconstrained multi-hop scatternet formation for Bluetooth networks, *Mobile Networks and Applications* **9** (2004), 33–47.

[23] I. Al-Kassem, S. Sharafeddine and Z. Dawy, *BlueHRT: Hybrid Ring Tree Scatternet Formation in Bluetooth Networks*, Computers and Communications, 2009. ISCC 2009, IEEE Symposium on 5–8 July 2009, pp. 165–169.

[24] Y. Li and J.Z. Wang, Cost analysis and optimization for IP multicast group management, *Computer Communications* (8) (2007), 1721–1730.

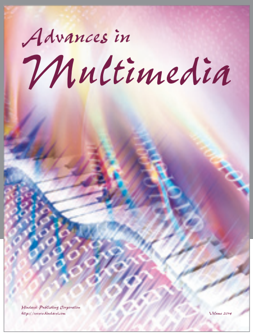
[25] R. Xie, D.Y. Qi, Y.J. Li and J.Z. Wang, A novel distributed MCDS approximation algorithm for wireless sensor networks, *Wireless Communications and Mobile Computing* **Issue 3**(9) (2009), 427–437.

Yongjun Li received his B.E. of science, M.E. of engineering and Ph. D. of science degrees from East China Normal University, National University of Defense Technology, and Sun Yat-sen University, respectively. He has been conducting postdoctoral research since 2004 at South China University of Technology (SCUT). He worked first at Hunan Normal University from Jul., 1991 to Oct.,1999, and then at Guangzhou Maritime College from 1999 to 2002, later at Guangzhou Radio & TV University from 2002 to 2005. Now he is an associate professor and continues his teaching and research work at the College of Computer Science & Engineering of SCUT. His research interests include new generation of network technology, intelligent initiative network and computer application technology. He served as the principle investigator in several projects supported by National and Province Science Fund and participated in many software development projects. He has published three text books and more than twenty papers, and received one patent.

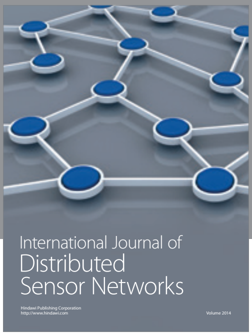
Rong Xie received his B.S. M.S. and Ph.D. degrees in Computer Science from South China Normal University, Guangdong University of Technology, and South China University of Technology. He is with ZhaoQing University and currently an Assistant Professor in the School of Computer. His current research interests include computer networks, wireless networks, and next generation Internet.

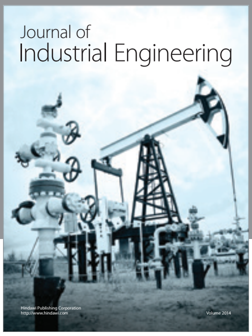
Hu Chen is now studying for his M.S. degree in South China University of Technology.

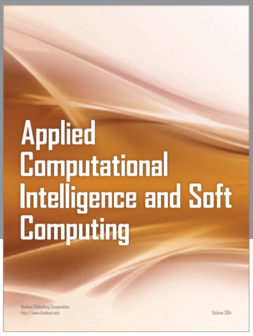
James Z. Wang received the B.S. and M.S. degrees in Computer Science from University of Science and Technology of China, and the Ph.D. degree in Computer Science from University of Central Florida. He is currently an associate professor in the School of Computing at Clemson University. He previously worked as senior software engineer in Veritas Corp. and Computer Associate. His research interests include multimedia systems, database, distributed computing, Internet technologies, data mining, information retrieval and bioinformatics. He has published nearly 60 papers in refereed journals and conference proceedings. He is a senior member of IEEE and a senior member of ACM. He is an associate editor for International Journal of Data Mining and Bioinformatics.

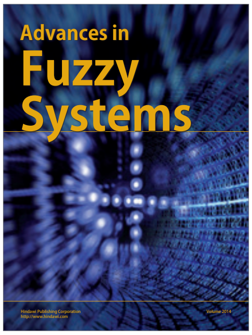

Advances in
Multimedia

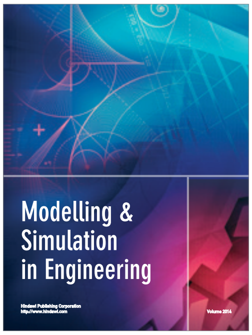
The Scientific
World Journal

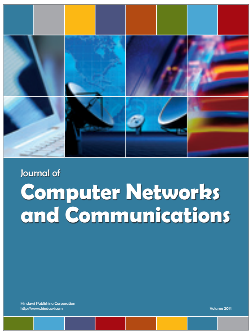

International Journal of
Distributed
Sensor Networks

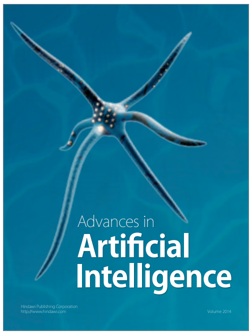

Journal of
Industrial Engineering

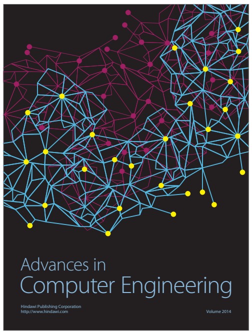

Applied
Computational
Intelligence and Soft
Computing


Advances in
Fuzzy
Systems

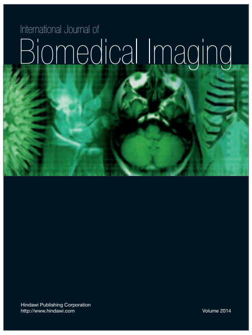

Modelling &
Simulation
in Engineering

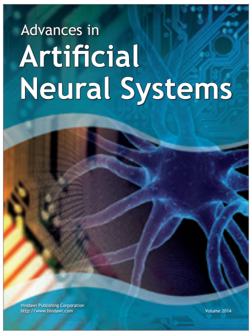

Journal of
Computer Networks
and Communications

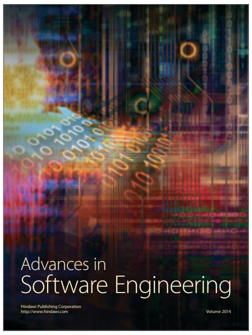

Advances in
Artificial
Intelligence


Advances in
Computer Engineering

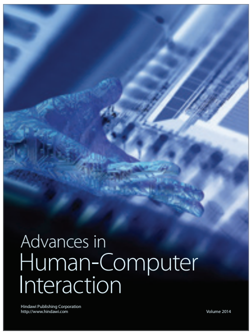
International Journal of
Computer Games
Technology

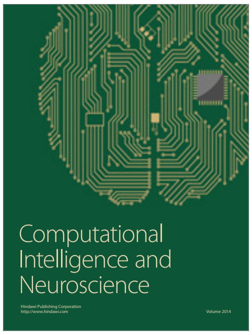

International Journal of
Biomedical Imaging


Advances in
Artificial
Neural Systems


Advances in
Software Engineering

Journal of
Robotics


Advances in
Human-Computer
Interaction


Computational
Intelligence and
Neuroscience

International Journal of
Reconfigurable
Computing

Journal of
Electrical and Computer
Engineering