An Energy-efficient Multihop Scatternet Formation for Bluetooth Networks

LI XIANG, YANG XIAO ZONG School of Computer Science and Technology Harbin Institute of Technology No.92, West Da-Zhi Street, Harbin, Heilongjiang PR CHINA

Abstract: Bluetooth is an emerging wireless technology that supports ad hoc networking. With increasing interest in energy constrained multi-hop wireless networks, a fundamental problem is to determine energy efficient communication strategies over these multi-hop networks. In this paper, an asynchronous and distributed energy-efficient Bluetooth scatternet construction algorithm is proposed. First, all random distributed nodes are self-organized into piconets with bounded number of slaves within k; then select bridge nodes to interconnect piconets into a scatternet. The election of the master or bridge is driven by a node which has the plenteous energy and strong receiving signals. Any two adjacent piconets is connected by one route, and every bridge is assigned an exact role by the role transition diagram. The resulting scatternet is a connected mesh-like network, master and bridge nodes constitute a connected dominating set of the scatternet. Route discovery for the scatternet can be realized by using the similar dynamic source routing protocol through masters and bridges which have relative good duty time. The simulation results confirm good functionality of scatternet including a considerable gain in network lifetime.

Key words: Ad hoc Network; Bluetooth; Scatternet; Piconet; Power Control; Network Lifetime

1 Introduction

Bluetooth is an emerging low-power, low-cost short-range radio technology [1]. It is considered a promising technology for short range wireless networks and wireless personal area network, which is also another important platform for ad hoc networking. With increasing interest in energy constrained multi-hop wireless networks, a fundamental problem is one of determining energy efficient communication strategies over these multi-hop networks.

In the Bluetooth specification, only the principle of building a piconet is given, how to construct an efficient scatternet is still an open issue. Although some existing ad hoc networks formation protocols can be used but they aren't be completely suitable for Bluetooth network for its particularity. There are a few papers addressing this problem in the current literature. In [2], a BTCP protocol is proposed, which is a leader-election type scatternet construction protocol and need all nodes within proximity of each other, the number of nodes is limited to within 36 nodes. Protocol given in [3] is similar to the one proposed in [2], each node's role, including a super-master (leader) is elected by voting but not designated. In [4-9], scatternet formation protocols are all based on tree topology, are processes of constructing Bluetooth spanning tree. The tree structure is shown to be simple to realize and efficient for packet scheduling and routing, but lacks efficiency and robustness. In [10-12], an asynchronous and distributed scatternet protocol is given respectively. Protocols produce a few independent piconets and interconnect these piconets into a scatternet. The Bluenet protocol in [10] produces a scatternet whose piconets have a bounded number of slaves, but it is unable to always guarantee the connectivity of the resulting mesh. In [11], a three-phase BlueConstellation protocol is proposed. It hasn't limited the number of slaves in a piconet and is actually a formation protocol of clustering. In [12], a TPSF protocol is proposed to support dynamic topology changes. The scatternet is on-demand created whenever a node wants to initiate data communications with another node, thus achieving high aggregate throughput at the expense

of connection setup delay.

In this paper we propose a two-stage energy-efficient Bluetooth scatternet construction protocol. It is similar to the protocols given in [10-12], but extends the network lifetime by selecting the nodes with greater energy and stronger received signal strength acting as masters or bridges which usually carry heavier burdens than the pure slaves.

2 Scatternet Formation

2.1 Energy-efficient Consideration

Bluetooth units are mostly used by mobile devices, and energy is supplied by batteries that have limited lifetime. Battery depletion results in failure of the node, which is an unwanted situation looking from a user perspective, and also it may require reorganization of the whole scatternet. So we proposed an energy-efficient scatternet formation algorithm based on device and link characteristics.

2.1.1 Device Energy Grade

Since master and bridge nodes are loaded more compared to slaves, a device with high battery capacity and high traffic generation rate is better to be chosen as master or bridge. If a device, having a high battery capacity, is chosen to be a master or a bridge, the resulting scatternet will be more stable and energy-efficient. For example, for the scenario illustrated in Fig.1, choice of a mobile phone as the master of several laptops is not an intelligent decision for a robust network.



Fig.1 Scatternet formation based on device class

Device class information is used in the scatternet formation, i.e., scatternet is formed by taking into consideration whether the device is a laptop, a desktop or PDA etc, for device class denotes the energy supply capacity in a way. Device class is known to Bluetooth model and is exchanged with neighboring devices during connection establishment procedure by the class of device/service field of the frequency hop synchronization packet [13].

Algorithm assigns a Device Energy Grade (DEG) to each node to make use of the class of device information together with battery power level of devices. DEG is calculated using device classes and the battery power level as:

$DEG = BC * BPL \qquad (1)$

where, BC (Battery Capacity) indicates the power capacity of the battery and the BPL (Battery Power Level) represents the fraction of remaining battery, $0 \le BPL \le 1$.

DEG is calculated by using the assigned Battery Capacity for different classes of devices, shown in Table 1. These assignments are based on rough estimations, where Battery Capacity is predicted from the average lifetime of the devices in terms of hours.

Battery	Lifetime if	Device Class		
Capacity	active (hours)			
5	infinite	NAP, desktop,		
		projector,printer,scanner		
4	4	laptop, camera		
3	3	Mobile phone		
2	2	PDA		
1	1	peripherals, headset, sensor		

Table 1 Assignment of Battery Capacity

2.1.2 Received Signal Strength Grade

Bluetooth modules have power control abilities [1]. Power control can be used not only to reduce interference but also to extend the life of battery in a device. The Bluetooth standards define three power classes each with a different transmit power range. Transmit power step sizes in the range 2 to 8dB have been specified. If devices receiving strong signals from each other are connected, less power is consumed for data transmitting, thereby increasing the lifetime of scatternet and reducing interference as illustrated in Fig.2. Furthermore, if needed, it could be possible to reduce the transmit power.



Fig.2 Scatternet formation based on RSSI

Bluetooth module has Received Signal Strength Indication (RSSI) that measures the received signal strength of the synchronization messages sent by its neighbors [1]. If all nodes transmit these messages at the same power level, the signal will be stronger for a neighbor that is closer. We assign a Received Signal Strength Grade (RSSG) to a device based on the measured RSSIs for the links to all its neighboring devices.

Since the transmit distance can be known by measuring the received signal strength, RSSG is used to construct the scatternet in such a way that the links are established between closer nodes in the formed scatternet topology. It is assumed that the distance between a master and a slave is known to both of them. The received signal strength quantization can be roughly made based on distance as depicted in Table 2, combined with the transmit power model based on distance introduced in [14,15].

Distance,	Signal strength	RSSI	Maximum
d(m)			Transmit
			Power(dBm)
0≤d≤1	VS(Very strong)	5	P _{max} -24
1 <d≤2< td=""><td>S(Strong)</td><td>4</td><td>P_{ma}-18</td></d≤2<>	S(Strong)	4	P _{ma} -18
2 <d≤4< td=""><td>M(Medium)</td><td>3</td><td>P_{max}-12</td></d≤4<>	M(Medium)	3	P _{max} -12
4 <d≤8< td=""><td>W(Weak)</td><td>2</td><td>P_{max}-6</td></d≤8<>	W(Weak)	2	P _{max} -6
8 <d≤10< td=""><td>VW(Very Weak)</td><td>1</td><td>P_{max}</td></d≤10<>	VW(Very Weak)	1	P _{max}

Table 2	Assignment	of RSSI
1 4010 2	rissignment	01 1001

Using RSSI from Table 2, RSSG of a node i can be calculated as:

 $RSSG(i) = \sum_{i \in J} RSSI(i,j) / |J|$ (2)

J is the set of nodes building links with i; RSSI(i,j) denotes the RSSI on the link between i and j.

2.2 Algorithm

To a given set of Bluetooth units, randomly distributed in a graphical area. If two nodes are in radio range of each other, they can discover each other and become neighbors. Any two neighboring nodes can communicate with each other, and consider there is an underlying link between them. Suppose that such a network topology graph ("visibility graph") is connected in physical, algorithm studies how to interconnect these nodes into a scatternet.

2.2.1 Initial Piconets Formation

First, all Bluetooth nodes are void; i.e., no master and no slave in any piconet. Each node senses for adjacent nodes by alternating between INQUIRY and INQUIRY SCAN states randomly. This procedure is usually called "Device Discovery", a detail description can be seen in [2,16]. By inquiry operations, every node collects information about its neighbors within radio range, i.e., forms a local visibility graph. Suppose that V, |V|=n, is the set of nodes, the set of the neighbors of a node $v \in V$ will be denoted by $\Gamma(v), v \notin \Gamma(v)$.

Node v will win compared to node u if one of the following conditions holds:

(a) DEG(v)>DEG(u).

(b) RSSG(v)>RSSG(u) when DEG(v)=DEG(u).

(c) BD_ADDR(v)<BD_ADDR(u), when DEG(v)= DEG(u) and RSSG(v)=RSSG(u).

where, BD_ADDR refers to the unique Bluetooth device address; RSSG of a node v is calculated by formula (2) in section 2.1.2 using $J=|\Gamma(v)|$. We say that v is bigger than u, and denote as: v>u.

Based on information gathered during the "Device Discovery" phase, each void node v starts the execution of the algorithm INITIAL() at the same time. Only nodes that are the biggest among all the nodes in their neighborhood win and become a master. All the other nodes just wait to receive a page message. Once a node v is elected as master, it calls HOST() to invite at most k nodes to join its piconet, and informs the rest neighbors stating that it is a master. While inviting neighbors who are void to join its piconet, node v first pages u which has the biggest RSSI(v,u) among nodes that haven't been paged by v. So links are built between these closer nodes. In the procedures, we use the following notations:

• Pico(v), the set of nodes in v's piconet. It is initialized to \emptyset , and updated only if v is a master.

• Host(v), the variable in which every node v records master that it joins. It is initialized to nil.

INITIAL(){	
if (for each node $u \in \Gamma(v): u < v$)	{
call HOST();	
if (v has been paged by all its neighbors)	
exit the execution of this phase;	
else exit and wait for the following page;	}
else go to page scan mode to wait for a page;}	

Then, we have the following page message triggered procedure OnReceivingPage(). On receiving a page message from a neighbor u, node v checks if the page is sent by a master (Flag=true). If

HOST() {
Host(v)=v; Pico(v)=Pico(v) \cup {v}; int m=0;
go to page mode;
while ($(\exists_{u \in \Gamma(v), u \text{ has the biggest RSSI(v, u)}}$ among nodes who haven't been
$p_{aged by v}u \text{ is void})\&\&(m < k)) \mathbf{do}$
send Page(v,Host(v), true) to u;
record that v has paged u;
if (u join the piconet of v successfully)
{ $Pico(v) = Pico(v) \cup \{u\}$; $Host(u) = v$; $m + +$;}
else get u's master;
record v has been paged by u; }
send Page(v,Host(v),false)to all neighbors who
haven't been paged by v;
go to page scan mode;}

this is the case, and v hasn't joined any piconet yet, it joins page node u's piconet. If node v has already joined a piconet instead, it informs u about this, also communicating the ID of its master. Then v proceeds to check if all its bigger neighbors have paged it. If this is not the case, it keeps waiting for another page.

```
OnReceivingPage(u,Host(u),Boolean Flag){
  record that u has paged; record Host(u);
if (Flag==true)
 if (v is void)
  v join the piconet of u; Host(v)=Host(u);
  send Page(v,Host(v),false)to all neighbors
      except for u in page mode;
  go to page scan mode;
                                    }
else inform u about my master;
if (some bigger neighbor has to page yet)
      exit and wait for the next page;
else if (v is void) call HOST();
if (v has been paged by all its neighbors)
     exit the execution of this phase;
else exit and wait for the next page;}
```

When successfully paged by all bigger neighbors, node v knows whether it has already joined the piconet of a bigger master or not. In the first case, v is a slave of the bigger master that paged it first. In the latter case v itself is going to be a master and tries to invite at most k nodes to join its piconet. In any case, v goes to page mode, and communicates its decision to all its neighbors that haven't known this. In fact, whenever a node has made a decision on its role of a master or slave, it therefore pages its neighbors and communicates its decision, along with its master ID. This exchange of information is necessary to implement the following phase of bridge selection for obtaining a connected scatternet.

Whatever a master or a slave node, once it has received pages from all its neighbors, it terminates the execution of this phase of the algorithm. Finally all nodes terminate the algorithms of this phase being either a master or a slave, and a series of piconets are produced. A piconet denoted by $Pico(v)(\neq \emptyset)$ is consists of all nodes in set Pico(v), with v as master and others as slaves.

2.2.2 Interconnecting Piconets

The purpose of this phase is to interconnect adjacent piconets by selecting bridges. For any two adjacent piconets, only one route is built between them. If there exists more than one connection link between any two adjacent piconets, more routes can be supported for data packet relay, but will increase the number of links in the scatternet, and the complexity of scatternet construction and route selection will be increased which degrades the network performance. Just using one route to connect any pair of adjacent piconets is a good tradeoff.

1. Selecting Bridge Nodes.

Based on information collected during the initial piconets formation, each node deals with its neighbors list according to the rules (1) to (3) in turn.

(1)For each node i, $i \in Pico(v)$, $\Gamma(i)=\Gamma(i)\setminus Pico(v)$.

(2)A node i is called "possible bridge" if $|\Gamma(i)| > 0$. For a possible bridge node i, it has a set of neighbors $J = \{j | j \in \Gamma(i) \land (Host(j) = v, v \in V)\}$, i.e., all nodes in J belong to the same piconet: Pico(v). If $|J| \ge 2$, then $\Gamma(i) = \Gamma(i) \setminus J \cup \{z\}$, where $z = Biggest_j(J \setminus \{v\})$ (Biggest is the operator to select the biggest element in a set). In order to reduce the burden of a master, we prefer to select slave as bridge. Do symmetrical modification, that is, for any node $k \in J \setminus \{z\}$, $\Gamma(k) = \Gamma(k) \setminus \{i\}$.

(3)If there exists more than one underlying route between two adjacent Pico(u) and Pico(v), we try to keep only one route between them proceeding in the following ways in turn:

(i)Select the two-hop path: u-w-v, if $w \in Pico(u)$ and |Pico(v)| < k, otherwise $w \in Pico(v)$ and |Pico(u)| < k. We bound the number of slaves in a piconet within k in order to avoid putting nodes into inactive state while wasting intra piconet overhead for the master, and also select the shorter path as possible in order to shorten the communication path length. If there exists more than one such route, we select the one where w is the biggest node on these routes except for master u and v. (ii)Select the three-hop path: u-x-y-v. If there exists more than one such route, we select the one including the biggest possible bridge among all nodes on these three-hop routes except for masters.

(iii)Select the two-hop path: u-w-v just like doing in (i), but don't judge if |Pico(u)| or |Pico(v)| < k. It causes the degree of a master in excess of k, but this is the final method not to select single hop path between u and v where two masters connect directly.

2. Building master-slave Connections.

Four kinds of bridges: S/M, S/S, M/S, and M/S/S bridge are given here, and a role transition diagram is showed in fig.3. To a possible bridge with role master it has great chance to win and keep its role as master, and it is possibly being selected and become 'S/M' bridge when it directly connects to another master. In all other cases, a possible bridge becomes one of the four kinds of bridges.



where:

(1) Win: the biggest possible bridge among all its possible bridge neighbors who haven't built master-slave connection relations with it will win, it becomes master to connect all the remaining bridges.

(2) Selected: possible bridge neighbor of a winner will be selected, it becomes slave of the winner.

A page message WIN(v) is used by a node v to let its neighbors who haven't built master-slave connection relations with it know that it is the winner in this turn. Two variables are used here:

 \blacksquare R_i, the variable with which a possible bridge records its role. It is initialized to master or slave according to its roles assigned in the initial piconet.

• W_i , integer variable. It is initialized to $|\Gamma(i)|$ in the related record of a possible bridge i.

Every possible bridge v calls COMPETE() at the same time in the beginning, only the winner sends a WIN message to the neighbors who haven't built master-slave connection relations with it, and terminates the algorithm, otherwise exits and waits for the next page.

On receiving a WIN page message from a neighbor u, node v triggers the procedure OnReceivingWIN(). Node v confirms that u is its

 $\begin{array}{l} \text{COMPETE()} \{ \\ \text{if (for each node } u \in \Gamma(v) \&\& 0 < W_u < n: v > u) \\ W_v = -1; \\ \text{if } (R_v = = \text{slave}) \quad R_v = M/S; \\ \text{else if } (R_v = = S/S) \quad R_v = M/S/S; \\ \text{for } (\forall u \in \Gamma(v): 0 < W_u < n) \\ \{ W_u = 0; \text{ send } WIN(v, R_v, W_v) \text{ to } u \text{ in page mode;} \} \\ \text{exit the execution of this phase; } \\ \text{exit the execution of this phase; } \\ \text{exit and wait for the next page;} \} \end{array}$

master and responses for it, including decreasing W_v by 1 and ascertaining its role. If v has built connections with all its neighbors, it terminates the execution of this phase of the algorithm, otherwise it calls COMPETE() to build connections with the remaining possible bridge neighbors.

OnReceivingWIN(u, R_u , W_u) { update the record about u using R_u , W_u ; W_v --; if (R_v ==slave) R_v =S/S; else if (R_v ==master) R_v =S/M; if (W_v ==0) exit the execution of this phase; else call COMPETE();}

Finally each bridge v can confirm its master-slave connection relation with neighbor u belongs to other piconet, i.e., if W_u =-1, v is slave of u; W_u =0, v is master of u. Some new piconets are produced and form a scatternet.

2.3 Properties of the Scatternet

In this section we give some useful properties of the scatternet constructed in forms of proposition.

Definition 1: A scatternet can be represented as a undirected graph G=(V,E).V is the set of nodes, $V=M\cup S\cup B$, M is the set of master nodes, S is the set of slave nodes, B is the set of bridges; E is the set of edges, $(u,v)\in E$ is a link between u and v.

Proposition 1 The scatternet formed is connected.

Proof: Let u and v be two nodes in V. In the underlying visibility graph, there exists a path between u and v (since the network connectivity is physical achievable by assumption). Let P be the shortest path of a certain length m (m<n) that connects u and v in the visibility graph. We prove the correctness of the proposition induced by m.

(1) m=1, two nodes can communicate with each other directly in the underlying visibility graph, then

u and v are neighbors. The distribution of u, v in the scatternet may be one of the two following cases:

(a) Belong to the same piconet;

(b) Belong to different piconets Pico(x) and Pico(y) respectively.

In the former case, they can communicate with each other obviously. In the latter case, two piconets Pico(x) and Pico(y) must be adjacent piconets (they can be connected by at least one underlying path (u,v)), and there exists unique connection between Pico(x) and Pico(y). Thus communication path can be built between u and v too. So whatever the case the algorithm guarantees successful communication between u and v.

(2) Assume that the proposition is correct when m=k. We will prove that the proposition is also correct when m=k+1.

Suppose that there are two nodes x_0 and x_{k+1} , they can reach each other by a shortest path P: $x_0 - x_1 - x_2$ $- \dots - x_k - x_{k+1}$ in the underlying visibility graph. On this path any single hop nodes are neighbors, such that x_i and x_{i+1} , $0 \le i \le k$. According to the proof in (1), the algorithm guarantees successful communication between x_i and x_{i+1} , for the rest hops $x_0 - x_1 - x_2 - \dots - x_i$, $x_{i+1} - x_{i+2} - \dots - x_k - x_{k+1}$, their path lengths are both shorter than k, by assumption, x_0 can reach x_i , and x_{i+1} can reach x_{k+1} , then u can reach v while n=k+1. So proposition is proved.

Proposition 2 C=M \cup B is a connected dominating set of graph G.

Proof: $\forall s \in S=V/C$ in G, $s \in Pico(v)$, $v \in M$ and $v \in C$. Node s is a slave of master v in Pico(v), s is adjacent to v obviously, so set C is the dominating set of G. Next, prove that C is connected.

C is connected if and only if any two nodes in C can reach each other by nodes in C. Node u reaching v by C is defined as existing a path: $u-c_1-c_2-...$ $-c_m-v$ which meets $c_i \in C$, $1 \le i \le m$ (m is a finite positive integer). Using reduction to absurdity, suppose that C is not connected, node u can't reach v by C. According to proposition 1, u can reach v in G, such that $u-v_1-v_2-...-v_k-v$. Then there exits a node, suppose to be node v_j , $v_j \in S=V/C$, $1 \le j \le k$, the degree of v_j is 2. But for any slave node, its degree equals 1, so the contradiction is produced. Suppose is wrong, C is connected.

So C is a connected dominating set of G, thus proving the proposition.

Proposition 3 D=M is a dominating set of G.

Proof: Omitted.

Proposition 2 and 3 can provide supports for the research of routing in Bluetooth scatternet based on dominating set.

3 Route Discovery in the Scatternet

Whenever a node wants to initiate data communication with another node, a route is constructed on-demand between the communicating nodes and broken down when the data transmission is finished. Recall that after the scatternet formation, each master node maintains all the information of its slaves and bridges nodes within its piconet, and each bridge node maintains all the information of its neighbors of adjacent piconets. The route selection mechanism for our scatternet can be based on any on-demand source routing protocols proposed for wireless mobile ad hoc networks. In this paper, we use the dynamic source routing (DSR) protocol for route selection [17]. The routing procedures in the scatternet include the following steps:

(1) The source node S sends a Route Request (RREQ) message to its master node M.

(2) If master node M has the destination's route in its routing cache, it sends a Route Reply (RREP) message to the source node S. Otherwise, the master node M sends a RREQ to all of its bridge nodes by adding its BD ADDR in the RREQ message.

(3) When bridge node B receives a RREQ, it forwards the message to all its neighbors including master nodes or bridge nodes belong to other piconets. Note that whenever a bridge or master node receives a RREQ, it checks the request ID and discards the message if it indicates a duplicate.

(4) If a master node has the destination node information, it will send a RREP on the reverse path instead of forwarding the RREQ to the destination. The RREP includes piconets route information which is represented by masters' BD_ADDRs, and it is preserved in each master's route cache.

(5) Finally, the master node (to which the source node is connected) receives the RREP. This RREP includes the piconets route. The master node forwards this RREP message to the source node.

Master and bridge nodes constitute a connected dominating set of the scatternet, a route between any two nodes can be found successfully. If more than one route is built, the shortest one is selected.

4 Simulation Results

We developed a C++ based extension using the Bluetooth software stack BlueHoc [18]. In our simulation model, different numbers of nodes, ranging from 10 to 100(step=10), are deployed randomly on an area of 30m by 30m. We used class 3 Bluetooth units, which have maximum transmit power P_{max} of 0dBm (i.e., transmission radius is 10m). Since some of nodes may not have any neighboring nodes after random deployment, we get rid of those nodes and only consider the nodes that have at least one neighbor. For each simulation experiment, we repeat running the simulation 100 times and then take the average of these 100 measurements. Nodes are assigned random device classes where all of the devices are battery fed. The batteries are assumed to be initially full (BPL=1, hereafter). Scatternets are built with k=5 and k=100 (having no limitation on the number of slaves in a piconet) respectively or the Bluenet [10] and BlueConstellation [11] scheme. The connectivity of the resulting scatternet produced by Bluenet is not guaranteed, this exists in the case when two neighboring nodes both become masters, but the slaves of the two piconets can't set outgoing links between the two neighboring piconets to form a scatternet. In this case, a link is built between the two neighboring masters (links between adjacent piconets can be built only by slaves in Bluenet). While configuring Blueconstellation decide to keep only one route to interconnect adjacent piconets. Finally, the performance of resulting scatternets is evaluated and compared.

Fig.4 presents the total number of the algorithmic messages without caring about the device discovery process. It shows that all messages are linear against the number of nodes. As expected, the scatternet (k=100) and BlueConstellation cost few traffic messages, due to their simple operations and the fact that a node needs to exchange very little information with its one-hop neighbors; however, pay little attention to configure BlueConstellation, the route between two masters is built by the bigger master instructing its gateway slave, and a switch of role needed to be performed to acquire slave/slave bridge when a gateway slave becomes the master of the smaller master. In fact more slave/slave bridges are produced in BlueConstellation than those in the

scatternet (k=100) for different rules for selecting bridges. So BlueConstellation costs more traffic messages than the scatternet with k=100. Bluenet acquires a higher number of messages compared to the scatternet with k=5, since a great number of smaller piconets are formed, and more inter-piconet links are set up and requires the exchange of a piconet composition among neighboring nodes. In our scatternet, bigger piconets formed when k=100 and the number of piconets will be reduced, so few messages needed to transmit during the second phase of interconnecting piconets.



In Figure 5, we exhibit the average number of piconets generated. The number of piconets increases with the number of nodes. The number of piconets in the scatternet using Bluenet scheme further increases, this is because of the limit of the number of slaves per piconet within 5 and the random choosing of a master in phase 1, which generates more masters. Furthermore, more than one outgoing link can be set instructed by master to adjacent piconets in Bluenet, which will form more extra piconets and lead to great increase in the number of piconets. So the number of piconets is much lower in our scatternet (k=5) by just keeping one connection route between any two adjacent piconets. Because of lack of the bound on the number of slaves per piconet which leads to forming a smaller number of bigger initial piconets, only selecting one connection route between any two adjacent piconets also greatly decreases the number of piconets, the number of piconets produced in BlueConstellation and our scatternet (k=100) is much less; for the sake of preferring to select slave as bridge in our scatternet, decrease the chances of using slave/slave bridge to interconnect adjacent piconets, so the number of piconets in our scatternet is much more than that in BlueConstellation.

Fig.6 shows the average number of roles per node in the scatternet. A high number of roles that per



node translates into reduced throughput performance due to the overhead associated to piconet switching. Increasing with n, bridges quickly increase for piconets interconnection. In Bluenet, almost all the nodes are bridges which leads to a possibly high number of piconets, the percentage of nodes with an extremely high number of roles quickly increases with number of nodes to take into account the increased number of extra piconets to which nodes affiliate; correspondingly, bridges in the scatternet (k=5) are few, so the average number of roles per node is lower than those in BlueNet. A noticeable variation occurs in the scatternet (k=100) and BlueConstellation, that gives the higher density of the BT topologies and the corresponding lower number of piconets (more bigger piconets produced), requires a lower number of bridges which are likely to be slave/slave bridges in BlueConstellation, so the average number of roles per node in them slightly decreases with number of nodes.



Fig.7 shows the average shortest path length of the scatternet produced. The average shortest path length of scatternet has great increase compared to the route length in visibility graph. This is essentially due to the principle of building a piconet or scatternet, which may force nodes in a piconet to communicate through their common master as a transit node, or through an inter-piconets route if they belong to different piconets. In Bluenet, as the node density increases resulting in shorter links because of the

8

higher number of roles per node, though piconets in Bluenet have a bounded number of slaves; however, the path length has a slight increase than number of nodes in other scatternets. In the scatternet (k=5), routes are longer than routes in BlueConstellation and the scatternet (k=100) because of the higher number of piconets generated, to reflect the higher number of piconets that have to be crossed; similarly, path length in the scatternet (k=100) are little longer than that in BlueConstellation.



It is assumed that the scatternets have been formed, we illustrate the average lifetime as a function of the network size in fig.8. Lifetime of the scatternet is defined as the time it takes untill any Bluetooth device in the network exhausts all its battery power. The power consumption for sending is 1 unit per slot; for receiving 0.5 unit per slot; 0.05 unit per slot during inactive or standby mode. Nodes are equipped with full battery initially at an assumed bit rate of 200 kbps. We can see that the lifetime of the scatternet decreases with the number of nodes, this is because of the increase overhead of route discovery and the increase of average number of hops between source-destination pairs as the number of nodes increases. As the number of nodes increases, a packet may reach its destination at the expense of decreasing the battery of more intermediate nodes. Thus batteries of bottleneck nodes are depleted faster as the scatternet grows. So Bluenet has longer network lifetime than BlueConstellation, and the scatternet (k=100) has longer network lifetime than the scatternet (k=5). Since devices with higher DEG and RSSG are assigned as the masters and bridges, our scatternet can carry more traffic than other scatternets before the first battery fails and less transmit power are consumed by power control in our scatternet, the lifetime of the our scatternet have a great improvement. For the scatternet (k=100), a total 92% and 45% increase are achieved compared

to BlueConstellation and BlueNet respectively.



5 Conclusion

In this paper we introduce an asynchronous and distributed algorithm for energy-efficient scatternet construction in large Bluetooth-based ad hoc network. The main problem we discuss with our scatternet is how to select bridges and build just one route between any adjacent piconets pair while gaining high power efficiency. We take two energy-efficient techniques: (a) device class information is used to assign the battery capacity of the device; (b) RSSI on a link is rough assigned according to the distance, hence power control scheme can be employed to the link. The devices having higher battery capacity and stronger received signal strength have great chances to be selected as masters or bridges which have relative heavy burdens, thus carrying more traffic and prolonging the network lifetime.

References:

- [1] Bluetooth SIG, Specification of the Bluetooh System, version 1.1, Feb.2001.
- [2] T. Salonidis, P. Bhagwat, L. Tassiulas, R. LaMaire, *Distributed Topology Construction of Bluetooth Personal Area Networks*, in: Proc. of IEEE INFOCOM'2001, pp. 1577-1586.
- [3] R. Lakshmi, K. Manika, S.Abhinanda, A.Alok, *Clustering Algorithms for Wireless Ad hoc Networks*, in: Proc. of DIAL-M 2000, pp. 54-63.
- [4] H. LIN, W.F. DING, Q. GAO, A Distributed Algorithm of Topology Construction and Network Routing of Bluetooth Scatternet, Journal of Computer Research and Development, Vol.40, No.2, pp. 201-207.
- [5] G.V. Zaruba, S. Basagni, I. Chlamtac, Bluetrees-Scatternet Formation to Enable

Bluetooth-Based Ad Hoc Networks, in: Proc. of IEEE ICC'01, pp. 273-277.

- [6] M.T. Sun, C.K. Chang, T.H. Lai, A Self-Routing Topology for Bluetooth Scatternets, in: Proc. of ISPAN'02, pp.13-18.
- [7] F. Cuomo, G. Di Bacco, T. Melodia, SHAPER: A self-healing algorithm producing multi-hop Bluetooth scattERnets, in: Proc. of IEEE Globecom 2003, pp. 236-240.
- [8] C. Law, K.Y. Siu, A Bluetooth Scatternet Formation Algorithm, in: Proc. of IEEE SAWN 2001, pp. 2864-2869.
- [9] E. Vergetis, R. Guérin, S. Sarkar, J. Rank, *Can Bluetooth Succeed as a Large-Scale Ad Hoc Networking Technology?* IEEE Journal on Selected Areas in Communications, Vol.23, No.3, 2005, pp. 644-656.
- [10] Z.F. Wang, R.J. Thomas, Z. Hass, *Bluenet-a new* scatternet formation scheme, in: Proc. of HICSS-35'02, pp. 61-69.
- [11] S. Basagni, C. Petrioli, Multihop Scatternet Formation for Bluetooth Network, in: Proc. of IEEE VTC'2002, pp. 425-428.
- [12] Y. Kawamoto, V.W.S. Wong and V.C.M. Leung, A Two-Phase Scatternet Formation Protocol for Bluetooth Wireless Personal Area Networks, in: Proc. of IEEE WCNC'03, pp. 1242-1247.
- [13] <u>https://www.Bluetooth.org/foundry/assignnumb/</u> do cument/baseband.
- [14] B.J. Prabhu, A. Chockalingam, A Routing Protocol and Energy Efficiency Techniques in Bluetooth Scatternets, in: Proc. of IEEE ICC 2002, pp. 3336-3340.
- [15] A. Karnik, A. Kumar, Performance analysis of the Bluetooth physical layer, in: Proc. of ICPWC'2000, pp. 70-74.
- [16] C. Law, A.K. Mehta, K.Y.Siu, *Performance of a New Bluetooth Scatternet Formation Protocol*, in: Proc. of MobiHoc'2001, pp. 183-192.
- [17] D.B. Johnson, D.A. Maltz, Y.C. Hu, *The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)*, Internet Draft, draft-ietf-manet-dsr-10.txt, 2004, July.
- [18] BlueHoc: Bluetooth performance evaluation tool, http://oss.software.ibm.com/developerworks/opens ource/bluehoc/.