New Bluetooth Interconnection Methods: Overlaid Bluetooth Piconets (OBP) and Temporary Scatternets (TS)

Sewook Jung [a], Alexander Chang [a], and Mario Gerla [a]

[a] *Department of Computer Science, University of California, Los Angeles, Los Angeles, CA, USA*

Abstract

In a large scale Bluetooth network, the permanent Scatternet is regarded as the only method to interconnect Piconets. But, many Bluetooth devices do not support Scatternet. When they support it, features are limited. Moreover, in high mobility situations, permanent Scatternet is not useful because of the extremely high overhead caused by frequent disconnections and reconnections. We propose Overlaid Bluetooth Piconets (OBP) and Temporary Scatternets (TS) to interconnect Piconets and form a virtual Scatternet. In OBP, every Piconet continuously changes its stages and collects metadata from Piconets within communication range. If metadata shows the existence of useful data to transfer, an inter-piconet connection is made and data is transferred. TS can be used instead of using permanent Scatternet to interconnect Piconets when needed. TS does not require large Scatternet formations and complex maintenance schemes. Moreover, it does not keep routing information. In this paper we introduce and illustrate the OBP and TS concepts. We then compare throughput and efficiency of OBP and TS with respect to Scatternet.

Key words: Bluetooth, Piconet, Scatternet, OBP, TS, metadata

1 Introduction

Bluetooth is a short-range wireless network technology that supports ad-hoc network. Up to eight nodes are organized in a star-shaped cluster, called Piconet. The cluster head is called master and the other nodes are called slaves. Two slaves cannot transfer packets directly. So, master should intervene between two slaves when a slave transmits packets to the other slave. Piconets are interconnected through bridge nodes and interconnected Piconets form a Scatternet. Bridges are the nodes participating in more than one Piconet with

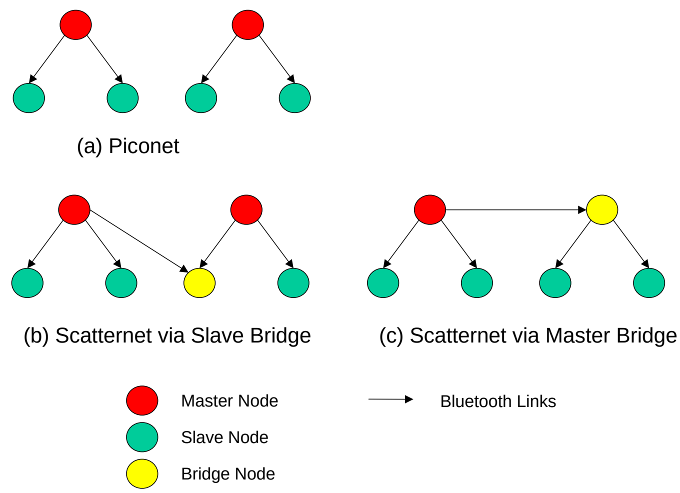


Fig. 1. Piconet and Scatternet

a time-sharing method. When a node is acting as a master for a certain Piconet and acting as a slave for the other Piconet, we call it a master bridge. When a node is acting as a slave for more than one Piconet at the same time, we call it a slave bridge. Fig. 1 shows examples of Piconet and Scatternet. Fig. 1(a) shows two Piconets that has one master and two slaves. These two Piconets are interconnected via slave bridge (Fig. 1(b)) and via master bridge (Fig. 1(c)).

Bluetooth data communication usually uses Asynchronous Connectionless Links (ACL) that has time slots of $625\mu s$. Data packets may use 1, 3, or 5 slots and they may be Forward Error Coded (FEC). FEC packets are DM1, DM3, and DM5 (with the digits indicating the number of slots used). The non-error coded ones are DH1, DH3, and DH5. The latest Bluetooth Specification 2.0 introduces Enhanced Data Rate (EDR) packets and they are 2-DH1, 2-DH3, 2-DH5, 3-DH1, 3-DH3, and 3-DH5. The 2-DH(1,3,5) and 3-DH(1,3,5) packets are similar to DH(1,3,5) but uses p/4-DQPSK and 8DPSK modulations, respectively [2]. Bluetooth packet information is described in Table 1.

Bluetooth uses frequency hopping scheme that has pseudo-random ordering of 79 frequencies in the industrial, scientific and medical (ISM) band. The hopping pattern may be adapted to exclude a portion of the frequencies that are used by interfering devices. The adaptive hopping technique improves Bluetooth devices' co-existence with static (non-hopping) ISM systems such as 802.11 when these are co-located [2].

Many Bluetooth chips are produced and already installed in many personal

Type	Payload (bytes)	FEC	Symmetric Max Rate (Kbps)	Asymmetric Max Rate (Kbps) Forward	Asymmetric Max Rate (Kbps) Backward
DM1	0-17	2/3	108.8	108.8	108.8
DH1	0-27	No	172.8	172.8	172.8
DM3	0-121	2/3	258.1	387.2	54.4
DH3	0-183	No	390.4	585.6	86.4
DM5	0-224	2/3	286.7	477.8	36.3
DH5	0-339	No	433.9	723.2	57.6
2-DH1	0-54	No	345.6	345.6	345.6
2-DH3	0-367	No	782.9	1174.4	172.8
2-DH5	0-679	No	869.7	1448.5	115.2
3-DH1	0-83	No	531.2	531.2	531.2
3-DH3	0-552	No	1177.6	1766.4	235.6
3-DH5	0-1021	No	1306.9	2178.1	177.1

Table 1
Bluetooth ACL Packets

devices such as Laptop, PDA, and Cellular phone. Support of Scatternet connection is defined as optional in all Bluetooth specifications, therefore many Bluetooth chips do not support Scatternet. Even if Scatternet connection is supported in Bluetooth devices, there is a limit in the number of simultaneous masters a slave can connect to, and also forming and keeping Scatternet requires special applications. Because of these reasons, temporary interconnection of Piconets is more useful than a permanent Scatternet in mobile situations.

We propose Overlaid Bluetooth Piconets (OBP) which enables network services for mobile users without Bluetooth Scatternet. Bluetooth nodes first form several Piconets, and OBP forms a virtual Scatternet later. OBP does not form a permanent interconnection of Piconets. Instead, it virtually interconnects Piconets when they are in the communication range. By using OBP, each Bluetooth Piconet can collect metadata from the Piconets in the communication range. Metadata contains information on transmission nodes, file names, and synchronization times. If there is real data to transfer between Piconets, it will be transferred after the metadata exchange.

We also propose Temporary Scatternets (TS) that forms Scatternets when Piconets are in the communication range. These Scatternets only last during transmission period. TS assumes at least one node in each Piconet has a Scatternet capability and can change its role as a master bridge or a slave bridge. After Scatternet is made, metadata is transferred first and then real data is transferred later as in OBP.

This paper has two main contributions. First, we describe the idea of Overlaid

Bluetooth Piconets (OBP) and Temporary Scatternets (TS). We show how they can be applied to Bluetooth devices already in use. Second, we describe the feasibility of OBP and TS by simulation results which are compared to those of Bluetooth Scatternet.

2 Related Works

In [5], overlay architecture is used to operate on top of the existing protocol stacks in various network architectures and to provide a store-and-forward gateway function between them when a node physically touches two or more dissimilar networks.

In ZebraNet [7], wireless sensor nodes are attached to animals and collect location data. This data is opportunistically transferred when the nodes are in the radio range of base stations. They show the effect of mobile base stations and sensor devices, and the use of two flooding-based routing protocols. In DataMules [15], "mule" travels among low-power sensor nodes and provides non-interactive messages periodically to allow sensor nodes save power.

In Pocket switched Network [6], Bluetooth devices are used in conference situations and measure real-world mobility patterns. They used Intel iMote Bluetooth platform to find out human mobility patterns. They check contact and inter-contact time and show many characteristics such as contacts with group of nodes, distribution of contacts among nodes, and influence of the time of day. These results are helpful to determine proper store-and-forward techniques.

In [9], Overlaid Bluetooth Piconets (OBP) is introduced and shows the possibility of interconnecting Bluetooth Piconets without Scatternet. OBP continuously changes its state and collects metadata from probed Piconet and uses multiple transfers to increase throughput. This state change makes virtual Scatternet among Piconets and in the viewpoint of application layer, Piconets are seen as interconnected. In this paper, OBP shows better throughput and efficiency than Scatternet.

In [8], Blueprobe, a capacity measurement tool for TDMA protocol, measures allocated capacity of a certain link or a multi-hop path. Moreover, capacity is compared in various situations and shows the effect of hop length and interconnection types (master bridge or slave bridge). As a result, interconnection type affects more than hop length. Based on comparisons among capacities of multiple one-to-one connections, interconnection via master bridge, and interconnection via slave bridges, multiple one-to-one connections case has the maximum capacity. Interconnection via master bridge is the second, and interconnection via slave bridge is the last.

3 Overlaid Bluetooth Piconets (OBP)

Overlaid Bluetooth Piconets (OBP) does not require Scatternet connection. So, all Bluetooth devices used in the world can use OBP as a Piconet interconnection method and form a virtual Scatternet, even if they do not support Scatternet. OBP can be used for the network that has challenging conditions, such as frequent disconnections, or long delays due to mobility of nodes. Instead of using Scatternet connection, OBP uses multiple one-to-one connections at the same time. Because of the frequency hopping scheme, several one-to-one links can be made and used to transfer at the same time without interference. And this interference-free feature increases total capacity.

Forming a Scatternet requires special Scatternet formation algorithms. Even if a Scatternet is formed, user's mobility disconnects the initial Scatternet, and thus frequent reconnections are needed. Many Scatternet algorithms [1,12,16,14] are developed and they help keeping connectivity of each device. However, Scatternet connection increases the average hop length and the number of links connected to a certain node, therefore it decreases capacity [11]. To increase capacity, Scatternet optimization method is needed [10]. Scatternet also has a scalability problem. As number of nodes increases, Scatternet is hard to maintain because Scatternet maintenance algorithms often use centralized methods. Because of these problems, Scatternet connections are not always useful, especially in high mobility situations.

Consider that we are using Scatternet unsupported Bluetooth devices. When a Piconet is formed, slave nodes cannot communicate with outside Piconet nodes. Master nodes can do inquiry and look for free nodes (unconnected nodes) in the communication range. Slave nodes cannot do inquiry-scan after their connections to a master. So, to do an inquiry-scan or to be connected to another master, a slave node should disconnect from its master node and become a free node. Therefore, each Piconet continuously changes its stages. Slave stage, Probe stage, Return stage, and Transfer stage are used in this sequence, and they form OBP Period as shown in Figure 2.

In Slave stage, every node keeps its original Piconet connection and intra-piconet transfers are made. Some nodes may not have any Piconet connection. These nodes remain as free nodes and are denoted as singleton nodes.

In Probe stage, one slave is randomly selected and disconnected from each Piconet and performs inquiry-scan and we denote this slave as probe node. Master nodes perform inquiry and find out which probe nodes are available

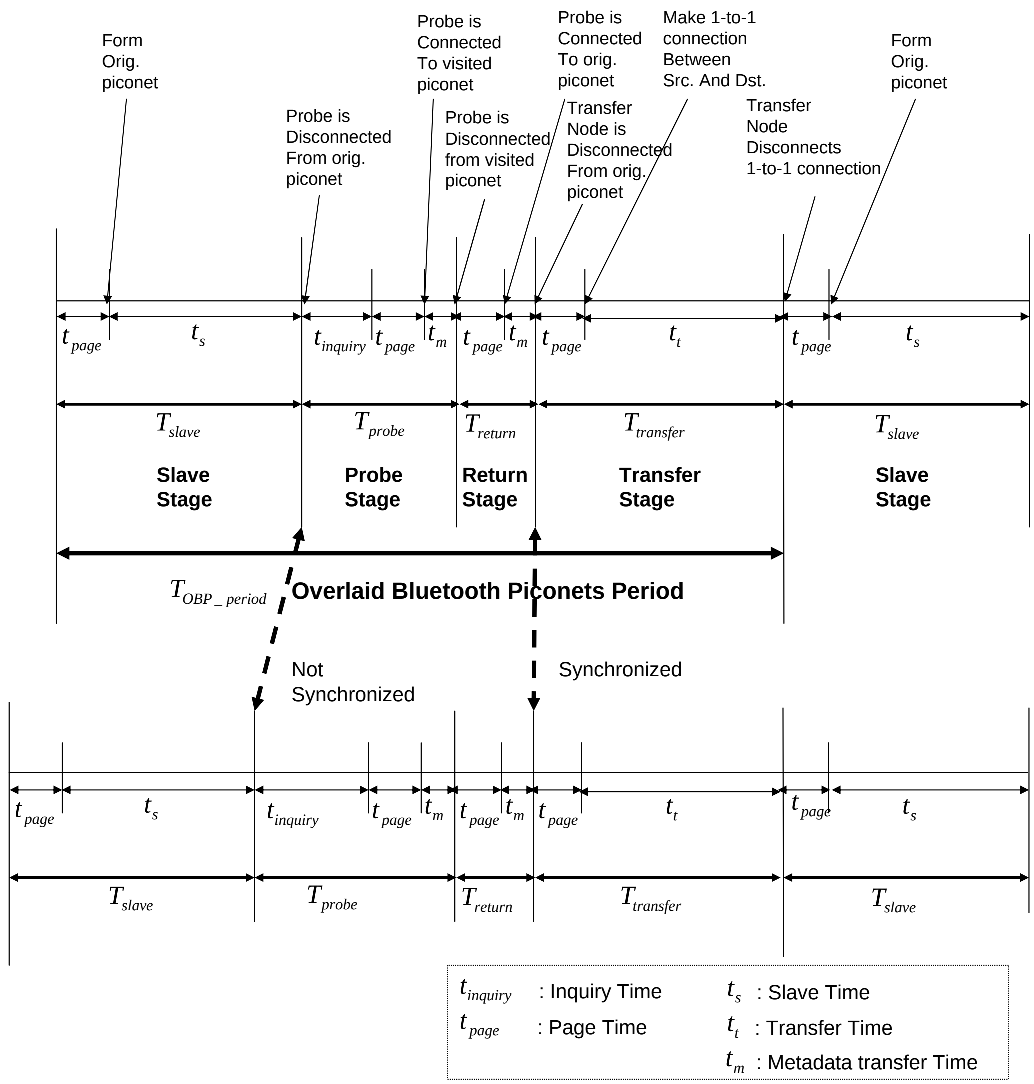


Fig. 2. Overlaid Bluetooth Piconets (OBP) Period

in the communication range. If a master node finds a probe node, master connects to it. Several probe nodes may be detected at the same time. In this case, master node should decide which one to choose among them. At the first Probe stage, master node randomly chooses one probe node and connects to it. At the later Probe stages, master chooses a probe node that is not connected before. If all probe nodes are connected before, master chooses the probe node that is connected earlier than other nodes. Master node keeps probe node connection log (bd-address and connection timestamp). Singleton nodes have 50% chance of doing an inquiry-scan (acting as a probe node) and 50% chance of doing an inquiry (acting as a master node). Thus in this stage, probe nodes are created to be connected to other Piconets (probed Piconets). After the connection, a probe node transfers metadata (containing sources and destinations of application flows) collected from the original Piconet to nodes in the probed Piconet and collects metadata from the probed Piconet nodes. After this, the probe node finds out whether there is useful data or not. If there is data to transmit, probe node and probed Piconet nodes synchronize transfer start time and decide which node will send and receive.

In Return stage, probe nodes are disconnected from the probed Piconets and return to their original Piconets. Inquiry is not included in this stage because master node already knows that probe node (that was slave of this master in slave stage) is in the communication range. So, master can connect to probe node with BD_ADDR. After connection to the original Piconet, the probe node conveys metadata received from the probed Piconet to all original piconet nodes and information about which nodes are used in the Transfer stage and when it is started.

In Transfer stage, inter-piconet transfer related nodes are disconnected from the original Piconets. If a master is related to this transfer, it will disconnect all of its slaves. After the disconnection, source nodes connect destination nodes (form a 1-to-1 connection) and transfer data. Inquiry is also not needed for this because source nodes already know that destination nodes are in the communication range and source node can connect to destination node with destination node's BD_ADDR.

After Transfer stage, source and destination nodes return to their original Piconets and OBP enters Slave stage. This returning is made almost same as Return stage but at this time, more than one node may be returned to the same Piconet.

Two Piconets may not be synchronized in the Slave stage. However, after a probe node is connected to the probed Piconet, the probe node will receive exact synchronization point from the probed Piconet. Two Piconets can be synchronized after the Transfer stage. Figure 2 shows how to synchronize between Piconets in Probe stage and Return stage.

Each node in the Piconet changes its role according to stages in OBP Period. Figure 3 shows each stage. There are three application flows: from S1 to D1, from S2 to D2, and from S3 to D3. S1, S2, and S3 denote source nodes and D1, D2, and D3 denote destination nodes. Figure 3 (a) shows Slave stage. In this stage, only intra-piconet transfer is possible because there is no link between different Piconet nodes. So, only the flow from S3 to D3 can be

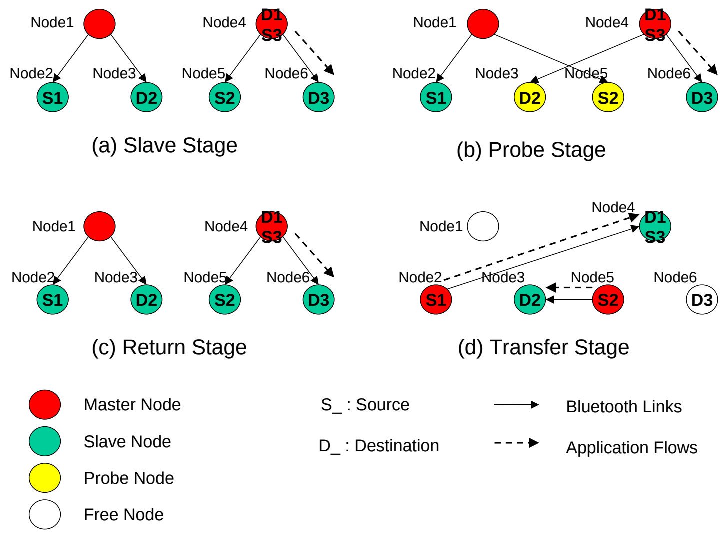


Fig. 3. Overlaid Bluetooth Piconets Stages

transferred. The flow will remain until Transfer stage is started because link from S3 to D3 is remained as connected until Transfer stage. Figure 3 (b) shows Probe stage in which probe nodes (node 3 and 5) are disconnected from their original Piconets and are connected to probed Piconets. After these connections, the probe nodes and the nodes in the probed Piconets exchange metadata. Synchronized transfer time will be assigned at this time. Figure 3 (c) shows Return stage and the probe nodes return to their original Piconets and convey the metadata to their Piconet nodes. Figure 3 (d) shows Transfer stage. In this stage, source and destination nodes are disconnected from their original Piconets. Source nodes make connection to destination nodes and start inter-piconet transfers such as S1 → D1 and S2 → D2.

4 Temporary Scatternets (TS)

Temporary Scatterenets (TS) assumes at least one node in each Piconet has the Scatternet capability. Each Piconet finds out existence of other Piconets from inquiry. Scatternet capable node can do inquiry or inquiry-scan when it is acting as master or slave. If more than one node responds to inquiry, inquiry node should select one among them. At the first Scatternet stage, inquiry node randomly chooses one inquiry-scan node and connects to it. At the later Scatternet stages, inquiry node chooses an inquiry-scan node that is not connected before. If all inquiry-scan nodes are connected before, inquiry node chooses one that is connected earlier than other nodes. Inquiry node

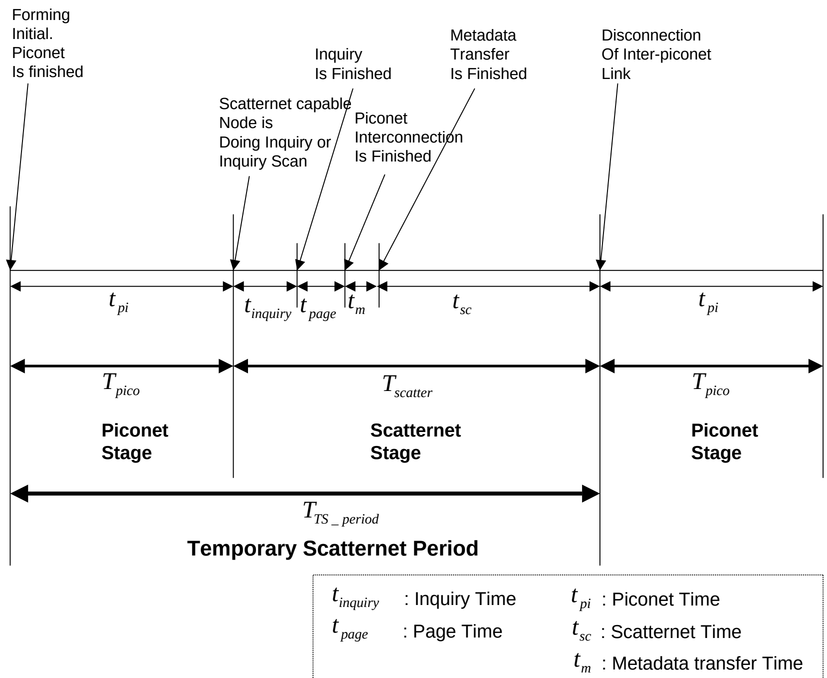


Fig. 4. Temporary Scatternet (TS) Period

keeps connection log (bd-address and connection timestamp).

After inquiry, each Piconet makes temporary interconnection when other Piconets are found and forms a Scatternet. If there is more than one Scatternet capable node, Piconet master is the best choice among them because connection between Piconet masters can form a Scatternet that has the maximum 3 hop length. Moreover, it is connected via master bridge that showed better performance than slave bridge [8]. If all Scatternet capable nodes are slave nodes, then choose one among them.

Figure 4 shows Temporary Scatternet (TS) Period. It contains Piconet stage and Scatternet stage. Before starting of first Piconet stage, initial Piconets should be formed and this connection requires connection time. After that Piconet stage can be started. During Piconet stage, intra-piconet transfers are made. After finishing Piconet stage, Scatternet stage is started. In the beginning of Scatternet stage, inquiry time is needed to find out proper Piconet to connect and page time is needed to inter-connect Piconets. Metadata transfer time is used for transmission of metadata among nodes in the original Piconet and connected Piconet. After metadata transfer, every node can have information of real data. After that, inter-piconet data is transferred during Scatternet time. After finishing Scatternet stage, inter-piconet link is disconnected and Piconet stage is started again.

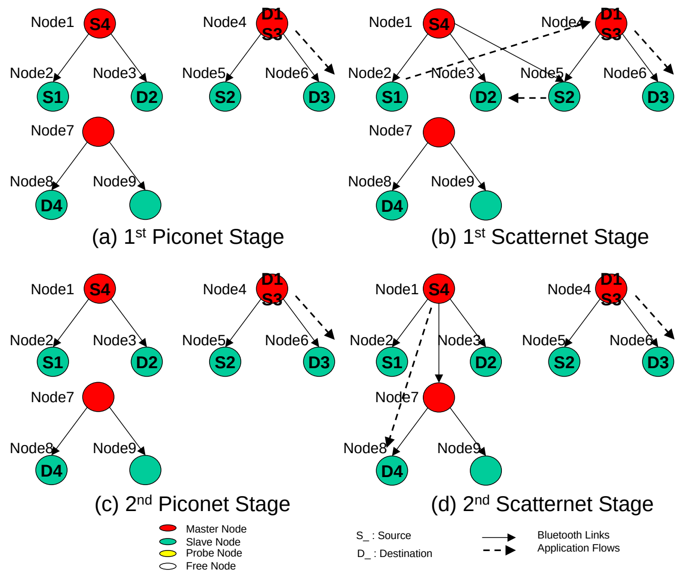


Fig. 5. Temporary Scatternet (TS) Stages

Figure 5 shows connection status and transfer status for each stage. In the first Piconet Stage, only intra-piconet transfer is possible. There exists only one intra-piconet transfer (S3 → D3) and it is transferred during this first Piconet stage. In the first Scatternet stage, Node1 connects Node5 and form a temporary Scatternet. After the connection is made, metadata is transferred to connected Piconet node. For example, Node1 transfers metadata to Node4, Node5, and Node6. After transferring metadata, all nodes can find out about application flows and start transmission. In the first Scatternet stage, two inter-piconet transfers (S1 → D1 and S2 → D2) and one intra-piconet transfer(S3 → D3) are possible. By disconnecting link from Node1 to Node5, 1st Scatternet stage is ended and moves to the 2nd Piconet stage. 2nd Piconet stage is same as 1st Piconet stage. In the 2nd Scatternet stage, Node1 connects to Node 7 and forms a different Scatternet. At this time, there is one inter-piconet transfer (S4 → D4) and one intra-piconet transfer (S3 → D3).

5 Throughput and Power Estimation

Throughput and Power are estimated to make comparison among OBP, TS and Scatternet.

Slave stage, Probe stage, Return stage, and Transfer stage durations are denoted as (1)-(4) and OBP Period duration is the sum of all stages' durations and denoted as (5).

$$T_{slave} = t_{page} + t_s \tag{1}$$

$$T_{probe} = t_{inquiry} + t_{page} + t_m \tag{2}$$

$$T_{return} = t_{page} + t_m \tag{3}$$

$$T_{transfer} = t_{page} + t_t \tag{4}$$

$$T_{OBP\_period} = T_{slave} + T_{probe} + T_{return} + T_{transfer} \tag{5}$$

t_{page} and $t_{inquiry}$ are page time and inquiry time, respectively. t_m is metadata transfer time in Probe stage and Return stage. t_s is slave time in Slave stage and used only for intra-piconet transfer. t_t is transfer time in Transfer stage and used for inter-piconet transfer. But, intra-piconet transfer is still possible during Transfer stage because not all the Piconet links are disconnected every time. If source and destination nodes are not used for inter-piconet transfer, they can be used for intra-piconet transfer.

Intra-piconet throughput in OBP is calculated as follows.

$$\theta^{sd}_{OBP\_intra} = C \cdot q_{sd} \cdot f_{sd} \cdot p_i \cdot (\frac{t_s}{T_{OBP\_period}} + (1 - p_e) \cdot \frac{T_{transfer}}{T_{OBP\_period}}) \tag{6}$$

Intra-piconet transfer is possible during t_t when source and destination are in the same Piconet. It is also possible during $T_{transfer}$ when source and destination remain in the same original Piconet because they are not used for inter-piconet transfer. C is the maximum capacity of a Bluetooth radio link, specified in Table 1. f_{sd} is usage percentage of capacity. It is calculated by 1 over the number of intra-piconet flows in one Piconet for intra-piconet case and is calculated by 1 over the number of inter-piconet flows located at same node for inter-piconet case. q_{sd} is the Link Quality (LQ) of the link (s, d) that can be obtained from the packet error rate (PER),denoted by q, as (A.1), while PER, denoted by r, can be calculated as a function of the bit error rate (BER), using the formulae (A.2) and (A.3), for DH and DM packet types, respectively [4].

Inter-piconet throughput is calculated as follows.

$$\theta^{sd}_{OBP\_inter} = C \cdot q_{sd} \cdot f_{sd} \cdot p_e \cdot (\frac{t_t}{T_{OBP\_period}}) \tag{7}$$

Total throughput is the sum of intra-piconet transfer and inter-piconet transfer and it is calculated as follows.

$$\theta_{OBP} = \sum_{(s,d)\in F} (\theta^{sd}_{OBP\_intra} + \theta^{sd}_{OBP\_inter}) \tag{8}$$

Power consumption for OBP is calculated as follows.

$$P_{OBP} = \sum_{(s,d)\in F} (P_t + P_r) \cdot h_{sd} \cdot f_{sd} + P_{OBP\_con} \tag{9}$$

h_{sd} is the hop distance between source and destination. For the intra-piconet transfer, the hop distance is 1 (master and slave) or 2 (slave and slave), and for the inter-piconet transfer, it is 1. In [11], P_t and P_r are assumed as transmitting and receiving power consumption at the full capacity of a radio link. P_{OBP_con} is the power consumed for connection and disconnection in various stages.

5.2 Temporary Scatternets (TS)

Piconet stage and Scatternet stage durations are denoted as (10), (11), respectively and TS period duration is the sum of all stages' durations and denoted as (12).

$$T_{pico} = t_{pi} \tag{10}$$

$$T_{scatter} = t_{inquiry} + t_{page} + t_m + t_{sc} \tag{11}$$

$$T_{TS\_period} = T_{pico} + T_{scatter} \tag{12}$$

t_{page} and $t_{inquiry}$ are page time and inquiry time, respectively. t_m is metadata transfer time in Scatternet stage. t_{pi} is Piconet time in Piconet stage and used only for intra-piconet transfer. t_{sc} is Scatternet time in Scatternet stage and used for inter-piconet transfer. But, intra-piconet transfer is still possible during Scatternet stage because Piconet link is not disconnected in Scatternet stage.

Intra-piconet throughput in TS is calculated as follows.

$$\theta^{sd}_{TS\_intra} = C \cdot q_{sd} \cdot f_{sd} \cdot p_i \cdot \frac{t_{pi} + T_{scatter}}{T_{TS\_period}} \tag{13}$$

Intra-piconet transfer is possible during t_{pi} and $T_{scatter}$ when source and destination are in the same Piconet. C, q_{sd}, f_{sd}, p_i, and p_e are defined same as in OBP case.

Inter-piconet throughput is calculated as follows.

$$\theta^{sd}_{TS\_inter} = C \cdot q_{sd} \cdot f_{sd} \cdot p_e \cdot (\frac{t_{sc}}{T_{TS\_period}}) \tag{14}$$

Total throughput is the sum of intra-piconet transfer and inter-piconet transfer and it is calculated as follows.

$$\theta_{TS} = \sum_{(s,d)\in F} (\theta^{sd}_{TS\_intra} + \theta^{sd}_{TS\_inter}) \tag{15}$$

Power consumption for transfer in TS is calculated as follows.

$$P_{TS} = \sum_{(s,d)\in F} (P_t + P_r) \cdot h_{sd} \cdot \min_{(i,j)\in(s,d)} f_{sd} + P_{TS\_con} \tag{16}$$

h_{sd} is the hop distance between source and destination. Notice that the factor $\min_{(i,j)\in(s,d)} f_{sd}$ in (16) adapts the power consumption to the bandwidth of the bottleneck link along the path. P_{TS_con} is the power consumed for connection and disconnection of inter-piconet link.

5.3 Bluetooth Scatternet

In [11], throughput is calculated as follows.

Number of Piconet member ($n_{Piconet}$ **or** $n_{probed\_Piconet}$)	[1, 4]
Number of Nodes($\|N\|$)	50
Number of Flows($\|F\|$)	100
Capacity (C)	723Kbps
Inquiry time and Page time ($t_{inquiry}$, t_{page})	(4, 2) sec
Slave time and Transfer time (t_s, t_t) **or** **Piconet time and Scatternet time** (t_{pi}, t_{sc})	(5, 5) sec

Table 2
Comparison Parameters

$$\theta_{scatter} = \sum_{(s,d)\in F} C \cdot \min_{(i,j)\in(s,d)} (f_{ij}^{sd} q_{ij}) \tag{17}$$

$\min_{(i,j)\in(s,d)}(f_{ij}^{sd} q_{ij})$ denotes the smallest usable bandwidth portion on the links of a connection (s, d) (i.e the bottleneck), while q_{sd} is the link quality (LQ) of the link (i, j).

In [11], power consumption is calculated as follows.

$$P_{scatter} = \sum_{(s,d)\in F} (P_t + P_r) \cdot h_{sd} \cdot \min_{(i,j)\in(s,d)} f_{sd} + P_{recon} \tag{18}$$

P_{recon} is the power consumed for reconnection of link when Scatternet is partitioned.

5.4 Throughput comparison

Throughputs of OBP, TS, and Scatternet are calculated as (8), (15), and (17), respectively. We assume parameters as in Table 2. And then, p_i and p_e are calculated as follows.

$$p_i = \frac{100}{50} \cdot \frac{2.5 - 1}{49} = 0.061224 \tag{19}$$

$$p_e = \frac{100}{50} \cdot \frac{2.5}{49} \cdot p_{probe} = 0.102041 p_{probe} \tag{20}$$

We assume Link Quality q_{sd} as 0.25, and Usage Percentage f_{sd} as 0.2 for intra- and inter-piconet transfers in OBP. Link Quality is set as same value in OBP and TS cases, but for Scatternet case, it is set to lower values because Scatternet increases retransmission based on disconnection. f_{sd} is calculated by average number of flows in same Piconet or Scatternet. Average number of nodes in the Piconet, $\overline{n_{Piconet}} = 2.5$, therefore Average number of Piconet is calculated as $50/2.5 = 20$. So, number of flows in each Piconets is calculated as $100/20 = 5$. If all flows passes same node and then $f_{sd} = 1/5 = 0.2$. And then throughput of OBP is calculated as

$$\begin{aligned}\theta^{sd}_{OBP\_intra} &= 723Kbps \cdot 0.25 \cdot 0.2 \cdot 0.061224 \cdot (\frac{5}{23} + (1 - 0.102041 \cdot p_{probe}) \cdot \frac{7}{23}) \\ &= (1.154737878 - 0.068735p_{probe})Kbps\end{aligned} \tag{21}$$

$$\begin{aligned}\theta^{sd}_{OBP\_inter} &= 723Kbps \cdot 0.25 \cdot 0.2 \cdot 0.102041 \cdot p_{probe} \cdot \frac{5}{23} \\ &= 0.801909p_{probe}Kbps\end{aligned} \tag{22}$$

$$\begin{aligned}\theta_{OBP} &= 100 \cdot (\theta^{sd}_{OBP\_intra} + \theta^{sd}_{OBP\_inter}) \\ &= (115.4737878 + 77.3174p_{probe})Kbps\end{aligned} \tag{23}$$

We assume Link Quality q_{sd} as 0.25, and Usage Percentage f_{sd} as 0.2 and 0.1 for intra- and inter-piconet transfers in TS, respectively. f_{sd} for intra-piconet transfer, it is calculated same as that of OBP. For inter-piconet transfer, two Piconets are merged and it doubles the average number of flows. So, $f_{sd} = 1/10 = 0.1$. And then, throughput of TS is calculated as

$$\begin{aligned}\theta^{sd}_{TS\_intra} &= 723Kbps \cdot 0.25 \cdot 0.061224 \cdot \frac{0.2 \cdot 5 + 0.1 \cdot 11.5}{16.5} \\ &= 1.441964Kbps\end{aligned} \tag{24}$$

$$\begin{aligned}\theta^{sd}_{TS\_inter} &= 723Kbps \cdot 0.25 \cdot 0.1 \cdot 0.102041 \cdot p_{probe} \cdot \frac{5}{16.5} \\ &= 0.558901p_{probe}Kbps\end{aligned} \tag{25}$$

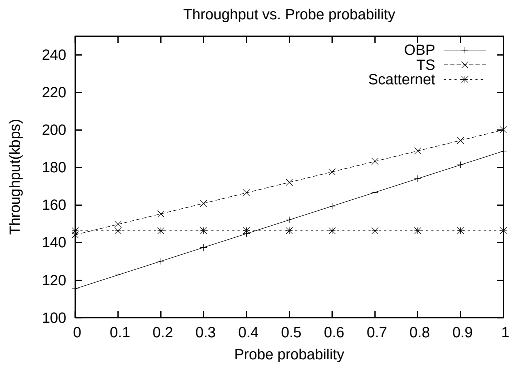


Fig. 6. Throughput vs. Probe probability

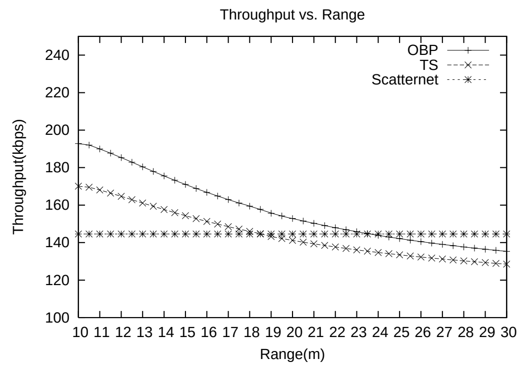


Fig. 7. Throughput vs. Range

$$\begin{aligned}\theta_{TS} &= 100 \cdot (\theta^{sd}_{TS\_intra} + \theta^{sd}_{TS\_inter}) \\ &= (144.1964 + 55.8901 p_{probe}) Kbps\end{aligned} \tag{26}$$

We assume Link Quality q_{sd} as 0.2, and Usage Percentage f_{sd} as 0.01 for Scatternet. If all flows go through one node, then $f_{sd} = 1/100 = 0.01$. Therefore, throughput of Scatternet is calculated as

$$\begin{aligned}\theta_{scatter} &= 100 \cdot 723Kbps \cdot 0.2 \cdot 0.01 \\ &= 144.6Kbps\end{aligned} \tag{27}$$

Figure 6 shows throughputs of OBP, TS, and Scatternet versus probe probability (p_{probe}). When probe probability is increased, throughput of OBP and TS are increased. Based on our assumption, in the higher probe probability, OBP and TS show better performance than that of Scatternet.

Figure 7 shows throughputs of OBP, TS, and Scatternet versus node's moving range. When the range is less relatively small, OBP and TS show better performance than that of Scatternet because p_probe is high. As range is increases, p_probe decreases therefore throughput also decreases. Compared to our simulation result (Figure 8) when range is $15.1 \cdot 15.1m^2$, throughput of OBP is almost same. Throughputs of TS and Scatternet are slightly higher than simulation result.

6 Simulator

In this section, we present the simulation environment that we used for evaluating our approach.

6.1 UCBT Simulator

For evaluation purposes, we implemented OBP algorithms in the UCBT ns-2 [13] based Bluetooth simulator [17], because it is the only publicly available open source Bluetooth simulator that supports mesh-shaped Scatternets.

UCBT implements the majority of the protocols in the Bluetooth. The simulator has recently added support for mesh-shaped Scatternets, but it assumes that all nodes are in the communication range. Therefore, we also added to UCBT a simple Scatternet formation protocol (described in section 6.3), besides our OBP algorithms.

6.2 Mobility

We assume Bluetooth devices are used in a conference room that has fixed boundary. Group of people are moving together with specific waypoint. For simulating mobility, we use the revised random waypoint model and Nomadic community mobility model in [3]. Because Piconets are moving together, we assume a Piconet master is moving according to the random waypoint model and slaves are staying in the short range (< 3m) of their master. Therefore, all Piconet members are moving to randomly chosen direction and speed. Maximum speed (0.0, 0.3, 0.6. 0.9, or 1.2 m/s) is predefined to limit node's speed. 1.2 m/s is selected because this speed is same as 4.32 km/s and just above walking speed. To add random factor, direction is changed periodically with an offset in the range [-10, 10] degrees with respect to the original direction. When a node reaches the boundary of the simulation area, it is mirrored back into the simulation area.

6.3 Scatternet Formation

We implemented a Scatternet algorithm based on [1,12,11]. On the first phase, nodes execute inquiry or inquiry-scan with a probability of 1/4 and 3/4, respectively. When an inquiry node discovers an inquiry-scan node, it will page the inquiry-scan node. This way, the inquiry node becomes a master of the other node in the newly formed Piconet. After this first phase, Piconets are formed. On the second phase, master nodes execute inquiry and slave nodes execute inquiry-scan. When master detects nodes that have hop distance longer than MAX_HOP distance (we define it as 4), master connects them and a Scatternet is formed.

Node's mobility can disconnect certain link and make hop distance longer than

Moving Area(Xr, Yr)	$15.1 \times 15.1m^2$, $21.28 \times 21.28m^2$
Number of Piconet member ($n_{Piconet}$ **or** $n_{probed\_Piconet}$)	[1, 4]
Number of Nodes$(\|N\|)$	50
Number of Flows$(\|F\|)$	100
Moving speed of nodes(S)	0.1, 0.3, 0.6, 0.9, 1.2 m/s
Packet type(P)	DH5, 2-DH5, 3-DH5
Inquiry time and Page time $(t_{inquiry}, t_{page})$	(4, 2) sec
Slave time and Transfer time (t_s, t_t) **or** **Piconet time and Scatternet time** (t_{pi}, t_{sc})	(5, 5) sec (7, 7) sec (10, 10) sec

Table 3
Simulation Parameters

4 (if partition is made, hop distance is set as ∞). For healing partition and long hop distance, Scatternet reconfiguration procedure makes reconnections and reduces hop distance.

6.4 Parameters

Parameters are described in Table 3.

6.5 Pseudo-codes

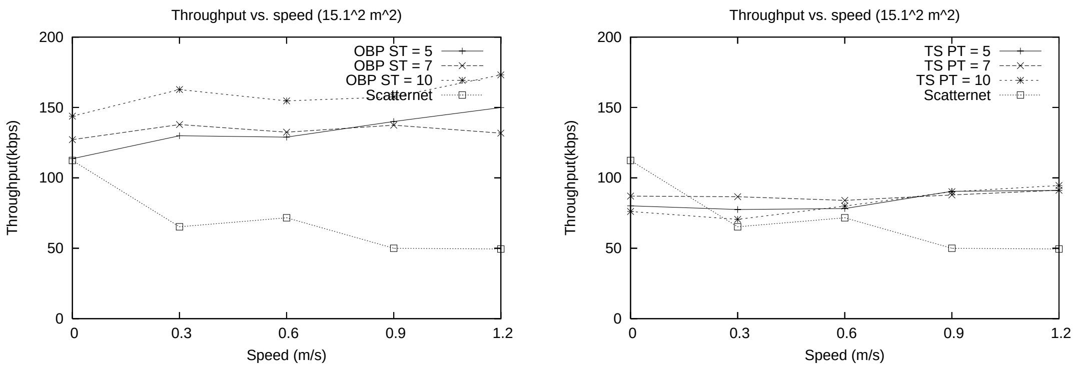
The Pseudo-codes in Appendix B show the basic algorithms of Overlaid Bluetooth Piconets (OBP), Temporary Scatternets (TS), and Scatternet.

6.5.1 Overlaid Bluetooth Piconets (OBP)

Pseudo-code for Overlaid Bluetooth Piconets (OBP) is shown in Algorithms 1 through 5. Main algorithm is described in Algorithm 1 and it calls each stage (Slave, Probe, Return, and Transfer stages) which are in Algorithms 2, 3, 4, and 5, respectively.

6.5.2 Temporary Scatternets (TS)

Pseudo-code for Temporary Scatternets (TS) is shown in Algorithms 6 through 8. Main algorithm is described in Algorithm 6 and it calls each stage (Piconet



(a) OBP (b) TS

Fig. 8. Throughput vs. Speed

and Scatternet stages) which are in Algorithms 7 and 8, respectively.

6.5.3 Scatternet

Pseudo-code for Scatternet formation is shown in Algorithm 9.

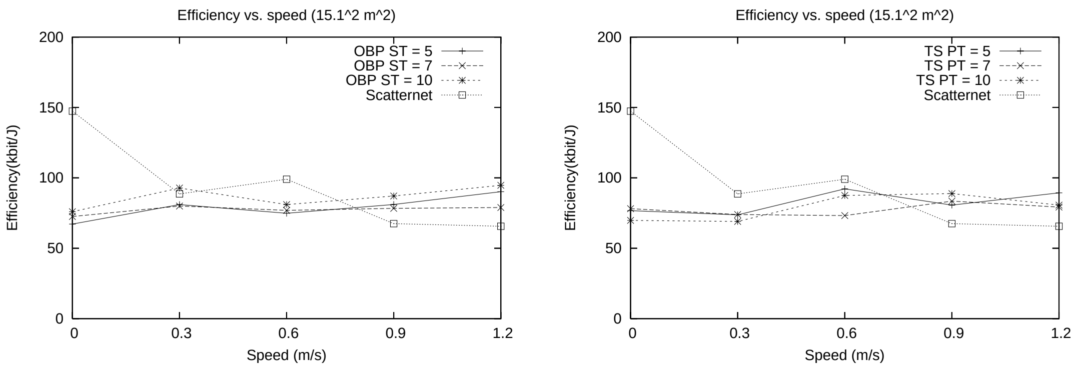
7 Results

We evaluate throughput and efficiency (throughput / power consumption) versus speed, data rate, and time. We also check number of distinct probed Piconets per second versus slave and transfer times. For all simulations, we set transfer time value same as slave time value, and Piconet time value same as Scatternet time value.

7.1 Throughput vs. Speed

Figure 8 shows throughput vs. speed results. We use maximum moving speed varying from 0 to 1.2 m/s to evaluate the throughput versus speed. DH5 packets and $15.1 \times 15.1 m^2$ area are used for this test.

As the speed increases the throughput of Scatternet decreases. When nodes are moving, nodes can be moved out of communication range. At this time, supervision timeout will happen and therefore that link is disconnected. Disconnection will make Scatternet partition and requires reconnection. Until reconnection, application flow should be stopped. These frequent link disconnections and reconnections reduce throughput.



(a) OBP (b) TS

Fig. 9. Efficiency vs. Speed

However, the throughputs of OBP cases stay the same or increase as the speed increases. OBP uses opportunistic transfers, therefore meeting chance is the most important factor of throughput. High mobility makes higher chance of meeting other Piconets, which produces more inter-piconet transfers in OBP and thus increases throughput. Moreover, OBP uses multiple one-to-one connections to fully utilize frequency hopping method. Frequency hopping method uses pseudo random frequencies, and therefore multiple one-to-one transmissions via multiple links can be possible without interference.

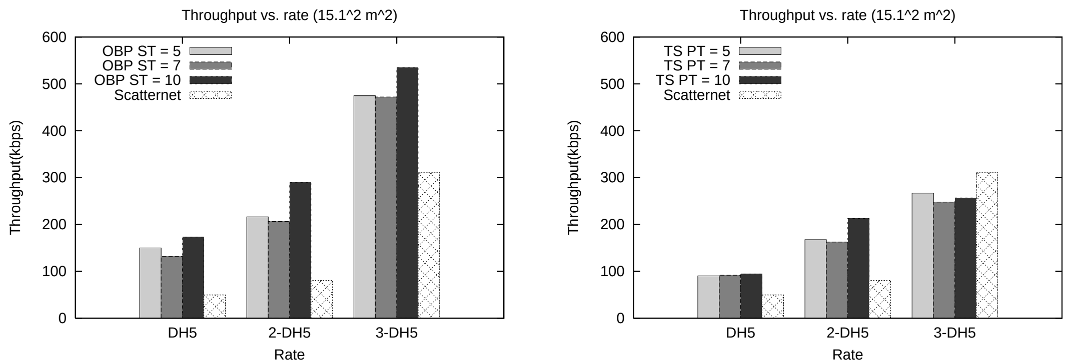
The throughput of TS is lower than that of Scatternet when nodes are not moving (speed = 0 m/s). But, it stays the same or increases as speed increases. In mobile situations, throughput of TS is better than that of Scatternet. TS uses single bridge (master bridge or slave bridge) instead of using multiple one-to-one connections which are used in OBP. In this case, this bridge is the bottleneck and it prevents total throughput increase.

7.2 Efficiency vs. Speed

Figure 9 shows efficiency vs. speed results. The same testing environment is used as in section 7.1.

OBP and TS shows almost same pattern. The power consumptions in OBP cases are higher than that of Scatternet because of higher throughput, frequent connections and disconnections, and metadata transfers. Even though the power consumption is higher in OBP, the throughput is much higher than that of Scatternet, which results in better efficiency in high mobility cases for OBP.

TS consumes less energy than OBP because of lower throughput and less connections and disconnections. But, TS consumes more than Scatternet cases.



(a) OBP (b) TS

Fig. 10. Throughput vs. Rate

Throughput is also lower than that of OBP case and higher than that of Scatternet case, therefore efficiency is almost same as that of OBP case and better than that of Scatternet in high mobility cases.

7.3 Throughput vs. Rate

Figure 10 shows throughput vs. rate results. For this test, DH5, 2-DH5, and 3-DH5 packets are used. The speed is set to 1.2 m/s speed and the area is set to $15.1 \times 15.1 m^2$ for this test.

When higher capacity packets are used, throughput increases as we expected in all cases. All OBP cases' throughputs are better than those of Scatternet because of multiple one-to-one transfers in OBP. Throughputs of 2-DH5 and 3-DH5 are not increased as twice or three times of DH5 case because OBP requires probe and connection.

In TS case, throughput of 3-DH5 case is lower than that of Scatternet. TS makes temporary Scatternet and does not keep routing information. This temporary Scatternet finds out routing path after Piconet interconnection. We used AODV as a routing protocol and it requires more setup time than Scatternet. Link Quality and Usage Percentage of TS are higher than those of Scatternet cases. So, TS shows better performance when DH5 and 2-DH5 packets are used. When 3-DH5 packet is used, the affects of setup time is greater than Link Quality and Usage Percentage, and thus Scatternet outperforms TS for this case.

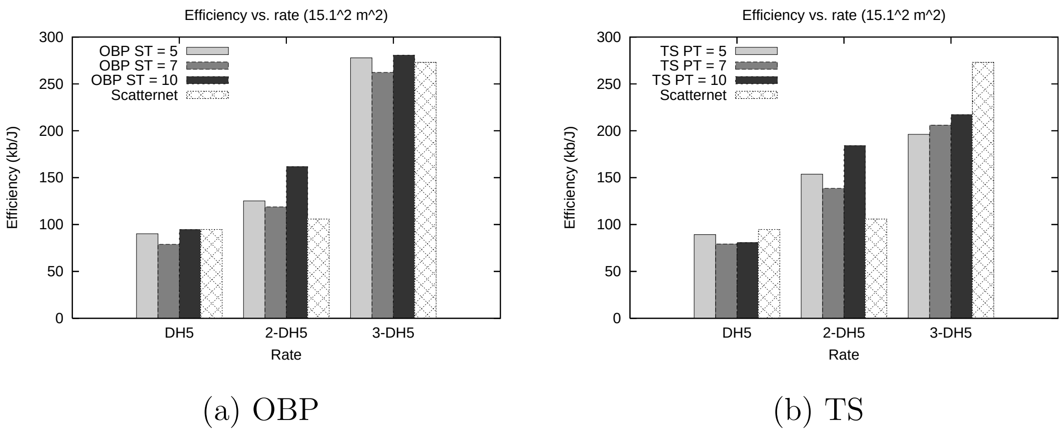


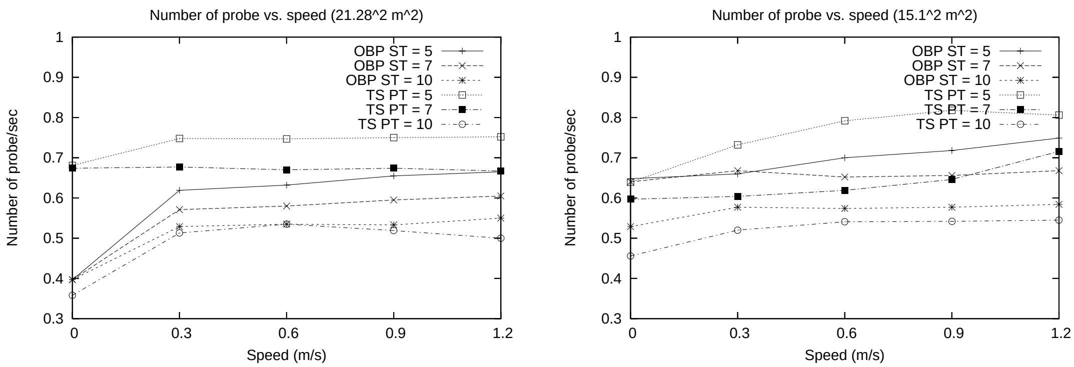
Fig. 11. Efficiency vs. Rate

7.4 Efficiency vs. Rate

Figure 11 shows efficiency vs. rate results. With the same testing environment as in section 7.3, the efficiencies of OBP, TS, and Scatternet do not vary a lot for a particular rate. As the rate increases, the efficiencies increase as well following the same pattern of throughput in section 7.3, because the power consumptions do not vary very much among different rates. TS shows the best efficiency when 2-DH5 packet is used and OBP shows the best efficiency when 3-DH5 packet is used.

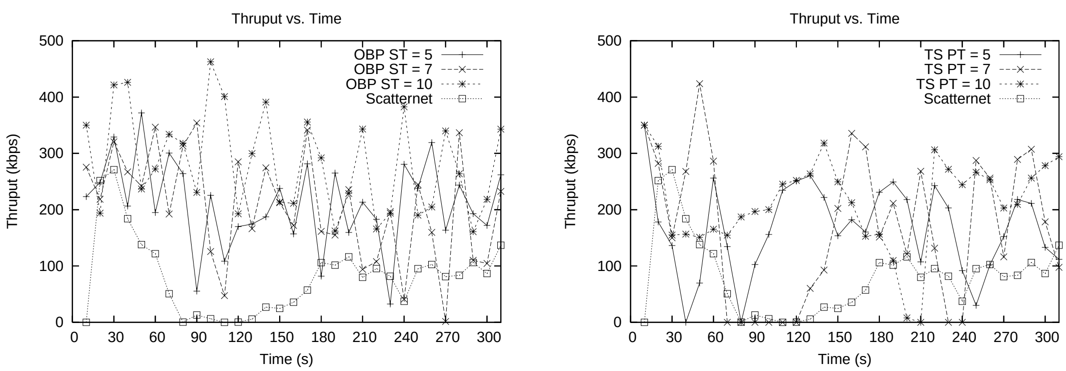
7.5 Probe Rate vs. Speed

Figures 12 shows the number of distinct probed Piconets per second with varying speeds in the areas of $21.28 \times 21.28 m^2$ and $15.1 \times 15.1 m^2$, respectively. When speed increases, the percentage of probed Piconets increases in both areas. And this increase reflects the increase in throughput shown in section 7.1. Also, in the larger area, the percentage increase between the speeds of 0 and 0.3 m/s is significant compared to other speed differences as expected, because nodes start moving increases the chance of meeting other Piconets. This is not the case for the smaller range as more Piconets are already in the communication range even if speed is 0 m/s. Among different slave times and Scatternet times, shorter ones have higher probe rate than longer ones as we expected, because total OBP or TS periods are directly proportional to the times and thus decreases the number of probe.



(a) $21.28 \times 21.28\ m^2$ (b) $15.1 \times 15.1\ m^2$

Fig. 12. Probe rate vs. Speed



(a) OBP (b) TS

Fig. 13. Throughput vs. Time

7.6 Throughput vs. Time

Figure 13 shows every 10 seconds' average throughput. We use 1.2 m/s speed, 2-DH5 packets, and $15.1{\times}15.1m^2$ range for this test.

In OBP, throughput varies a lot during the test time, because inter-piconet transfers (which is the main part of the throughput) are only possible during Transfer stage. During this stage, the throughput is high and in other stages it is low, and this is reflected in the oscillation of the throughputs in the figure. Shorter slave time one has shorter Transfer stage and thus has shorter oscillation period where as the longer one has longer period.

In TS case, throughput drops more than in OBP case. TS uses temporary Scatternet and when Scatternet is formed, a transfer path should be discovered with routing protocol. During path finding period, data packets cannot be transferred, and therefore throughput varies more than that of OBP.

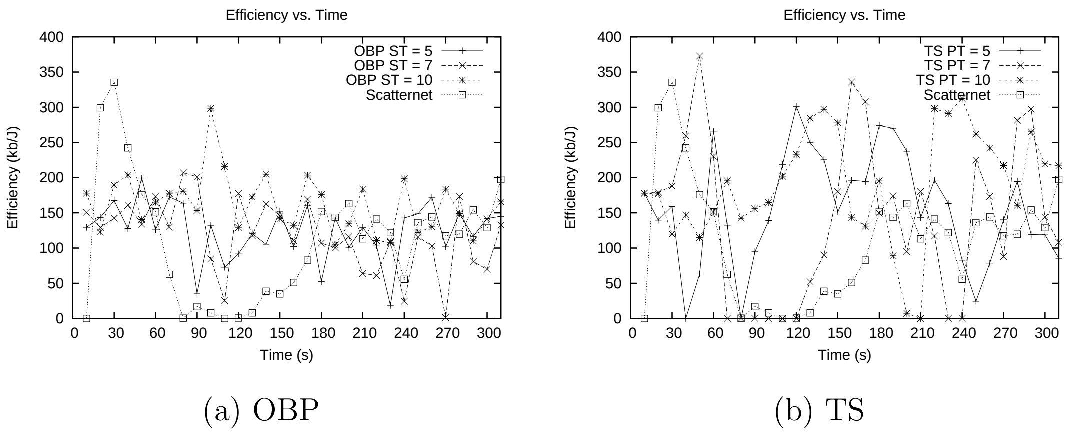


Fig. 14. Efficiency vs. Time

In Scatternet, node's movement disconnects some links, and thus decreases throughput at certain times, and reconnection regains the throughput.

7.7 Efficiency vs. Time

Figure 14 shows every 10 seconds' average efficiency. Same parameters in section 7.6 are used.

During Probe and Return stages of OBP, power for inter-piconet transfers disappears, instead, power for connections and disconnections is consumed. So, power consumption does not decrease as throughput decreases. In Scatternet stage of TS, additional power consumption for inter-piconet link connection and disconnection happens, but this effect is negligible since relatively small number of connections and disconnections are used compared to OBP case. In Scatternet, power consumption is almost constant throughout the simulations. Thus, the efficiency follows the throughput pattern in section 7.6.

8 Conclusion

In this paper, we presented several approaches to interconnect Bluetooth Piconets without using a permanent Scatternet in mobile environments. Overlaid Bluetooth Piconets (OBP) show resilience to mobility compared to traditional Scatternet and produce significantly higher throughput. Temporary Scatternets (TS) show only slightly higher throughput than Scatternet but are much simpler to set up and manage. The Scatternet topology requires Scatternet formation and reconfiguration as nodes are moving. OBP and TS instead create virtual Scatternets that do not require persistent connections. Moreover, OBP always uses multiple one-to-one connections therefore the routing protocol is not needed. Thus, it is very well suited for mobile environments. The efficiency of OBP and TS is comparable to that of Scatternet while keeping higher throughput.

OBP and TS use fully decentralized algorithms. They do not have to keep information of topology or global routing. OBP and TS only save connection log for connection fairness. TS may use routing, but its maximum hop ength is limited to 4 when using slave bridge. Also with higher mobility, in OBP and TS, the chance of meeting other Piconets increases and thus various application flows can be supported which in turn increase the throughput. OBP is applicable to all currently available Bluetooth devices even if they do not support Scatternet. TS is applicable if there is at least one Scatternet capable device in each Piconet. It has the same bottleneck link problem as Scatternet, but it has simper algorithms than OBP and Scatternet.

In summary, OBP and TS are more practical for networking Bluetooth devices than Scatternets. In the future, we will add store-and-forward method to increase transfer opportunity, and metadata flooding to keep the virtual Scatternet up-to-date.

References

[1] S. Basagni and C. Petrioli. A scatternet formation protocol for ad hoc networks of bluetooth devices. In *IEEE Vehicular Technology Conference (VTC)*.

[2] Bluetooth specification v2.0, 2004. Bluetooth SIG.

[3] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. In *IEEE Mobicom*, 2002.

[4] L.-J. Chen, R. Kapoor, M. Y. Sanadidi, and M. Gerla. Enhancing bluetooth tcp throughput via link layer packet adaptation. In *IEEE International Conference on Communications (ICC)*, 2004.

[5] K. Falls. A delay-tolerant network architecture for challenged internets. In *ACM SIGCOMM*, 2003.

[6] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot. Pocket switched networks and human mobility in conference environment. In *SIGCOMM Workshops*, 2005.

[7] P. Juang, H. Oki, Y. Wang, M. Maronosi, L. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. In *ASPLOS*, 2002.

[8] S. Jung, A. Chang, and M. Gerla. Comparison of bluetooth interconnection methods using blueprobe. In *The Second International Workshop On Wireless Network Measurement (WiNMee)*, 2006.

[9] S. Jung, A. Chang, and M. Gerla. Performance comparison of overlaid bluetooth piconets (obp) and bluetooth scatternet. In *IEEE Wireless Communications and Networking Conference (WCNC)*, 2006.

[10] S. Jung, C. K. Kalló, M. Gerla, and M. Brunato. Decentralized optimization of dynamic bluetooth scatternets. In *Proceedings of the Second Annual International Conference on Mobile and Ubiquitous Systems: networking and services (MobiQuitous)*, 2005.

[11] C. K. Kalló, S. Jung, L.-J. Chen, M. Brunato, and M. Gerla. Throughput, energy and path length tradeoffs in bluetooth scatternets. In *IEEE International Conference on Communications (ICC)*, 2005.

[12] C. Law, A. K. Mehta, and K.-Y. Siu. A new bluetooth scatternet formation protocol. In *Mobile Networks and Applications*, 2003.

[13] ns-2 (The Network Simulator). http://www.isi.edu/nsnam/ns/.

[14] C. Petrioli, S. Basagni, and I. Chlamtac. Bluemesh: Degree-constrained multi-hop scatternet formation for bluetooth networks. In *mobile Networks and Applications*, 2004.

[15] R. Shah, S. Roy, S. Jain, and W. Brunette. Data mules: Modeling a three-tier architecture for sparse sensor networks. In *IEEE SNPA Workshop*, May 2003.

[16] I. Stojmenobic. Dominating set based bluetooth scatternet formation with localized maintenance. In *Proceedings of the 16th International Parallel and Distributed Processing Symposium*, 2002.

[17] UCBT simulator. https://www.ececs.uc.edu/cdmc/ucbt.

Appendix

A Additional Formulae

Packet error rate (PER), denoted by q, is calculated as (A.1), while PER, denoted by r, can be calculated as a function of the bit error rate (BER), using the formulae (A.2) and (A.3), for DH and DM packet types, respectively [4].

$$q = 1 - r \tag{A.1}$$

$$r = 1 - (1 - b)^s \tag{A.2}$$

$$r = 1 - ((1 - b)^{15} + 15b(1 - b)^{14})^{s/15} \tag{A.3}$$

p_i is the probability of intra-piconet (internal) flow existence and p_e is the probability of inter-piconet (external) flow existence.

Assume that N is the set of nodes in the conference room and F is the set of all flows in all nodes. In that case, $|F|$ sources and $|F|$ destinations exist. So, the possibility of having a source or a destination at a certain node is $|F|/|N|$. And then, p_i and p_e are calculated as follows.

$$p_i = \frac{|F|}{|N|} \cdot \frac{n_{Piconet} - 1}{|N| - 1} \tag{A.4}$$

$$p_e = \frac{|F|}{|N|} \cdot \frac{n_{probed\_Piconet} - 1}{|N| - 1} \cdot p_{probe} \tag{A.5}$$

$n_{Piconet}$ and $n_{probed\_Piconet}$ are the number of nodes in original Piconet and in probed Piconet, respectively. p_{probe} is probability that at least one Piconet is probed. It depends on the communication range and nodes' moving range. If all nodes are in the communication range, all Piconets are in the same range. So, at least one Piconet detects probe node and connects to it. In this case, p_{probe} is 1. If all nodes are not in the communication range, p_{probe} is communication area divided by moving area. Near the boundary, communication area will be decreased because it is not a full circle. So, p_{probe} can be calculated as follows. When all nodes are in the communication range (A.6) is applied and when all nodes are not in the communication range (A.7) is applied.

$$p_{probe} = 1 \tag{A.6}$$

$$p_{probe} \simeq 1 - (1 - \frac{10^2\pi}{X_r \cdot Y_r})^{(|N|/\overline{n_{Piconet}})-1} \tag{A.7}$$

$|N|/\overline{n_{Piconet}}$ is average number of Piconets, and $(1 - \frac{10^2\pi}{X_r \cdot Y_r})^{(|N|/\overline{n_{Piconet}})-1}$ is the probability that all other Piconets are not in the communication range of $10m$ in the moving area of X_r by Y_r.

B Algorithms

Algorithm 1 **OBP Process**

set $t_m, t_{pi}, t_{sc}, sim\_time$
read node parameters
form Piconets
while $t < sim\_time$ **do**
  call *Slave_stage* procedure
  call *Probe_stage* procedure
  call *Return_stage* procedure
  call *Transfer_stage* procedure
end while

Algorithm 2 ***Slave_stage***

Master pages slaves
if intra-piconet flow exists **then**
  start intra-piconet flow
end if
while $t\_stage < t_s$ **do**
  keep transferring intra-piconet data
end while

Algorithm 3 ***Probe_stage***

```
if num_slave > 0 then
    master chooses one of its slave as probe node
    master disconnects probe node
else
    random_number = random[0,1]
    if random_number < 0.5 then
        master is set as probe node
    else
        master remains as master node
    end if
end if
if master_node then
    start inquiry
else if probe_node then
    if probe_node has intra Piconet flow then
        stop intra-piconet flow
    end if
    start inquiry-scan
end if
if inquiry_response > 1 then
    master chooses one BD_ADDR based on previous log
else
    master chooses responded BD_ADDR
end if
master pages chosen BD_ADDR
while t_stage < t_s do
    transfer metadata
end while
set transfer time
master disconnects probe node
```

Algorithm 4 ***Return_stage***

```
if num_slave > 0 then
    master pages probe_node
else
    probe_node returns as master_node
end if
transfer metadata
```

Algorithm 5 ***Transfer_stage***

if inter-piconet flow exists **then**
  if source and destination node has intra-piconet flow **then**
    stop intra-piconet flow
  end if
  if source and destination nodes are slave nodes **then**
    master disconnects source and destination nodes
  else
    master disconnects all slaves
  end if
end if
source node connects destination node
start inter-piconet flow
while $t\_stage < t_t$ **do**
  keep transferring inter-piconet data
end while
stop inter-piconet flow
source node disconnects destination node

Algorithm 6 TS Process

set $t_m, t_{pi}, t_{sc}, sim\_time$
read node parameters
form Piconets
while $t < sim\_time$ **do**
  call *Piconet_stage* procedure
  call *Scatternet_stage* procedure
end while

Algorithm 7 ***Piconet_stage***

if intra-piconet flow exists **then**
  star intra-piconet flow
end if
while $t\_stage < t_{pi}$ **do**
  keep transferring intra-piconet data
end while

Algorithm 8 ***Scatternet_stage***

```
if num_Scatternet_capable_node > 0 then
    if master is Scatternet_capable_node then
        choose master as bridge node
    else
        choose one slave as bridge node
    end if
end if
random_number = random[0,1]
if random_number < 0.5 then
    bridge node is set as bridge master
    bridge master starts inquiry
else
    bridge node is set as bridge slave
    bridge slave starts inquiry-scan
end if
if inquiry_response > 1 then
    bridge master chooses one BD_ADDR based on previous log
else
    bridge master chooses responded BD_ADDR
end if
bridge master pages chosen BD_ADDR
while t_stage < t_m do
    transfer metadata
end while
start inter-piconet flow
while t_stage < t_sc do
    keep transferring inter-piconet data
end while
stop inter-piconet flow
bridge master disconnects bridge slave
```

Algorithm 9 Scatternet Process

```
set sim_time
read node parameters
form Piconets
calculate hop_distance matrix
if hop_distance > 4 then
    make direct connection between two Piconets whose hop_distance > 4
end if
start application flow
while t < sim_time do
    keep transferring application data
    calculate hop_distance matrix
    if hop_distance > 4 then
        make direct connection between two Piconets whose hop_distance > 4
    end if
end while
```