Link Repairing for Inter-Piconet Communication Technique in Bluetooth Scatternet

Sabeen Tahir, Abas Md Said, and Sheikh Tahir Bakhsh

***Abstract*—A mobile ad-hoc network (MANET) is a collection of wireless mobile nodes, which creates a temporary network. Bluetooth is one of the emerging communication standards of MANET. Bluetooth allows communication through a basic network, which is called piconet. A piconet has one master and at most seven active slaves. There can be multiple piconets in a Bluetooth network, which is called a scatternet. Communication among multiple piconets is achieved through a common node known as a relay. Bluetooth allows synchronize transmission, therefore, packet transmission may always have unreliable links caused by relay node mobility. This degrades the system performance. Devices follow broadcast mechanisms to find an alternative route when a link breaks. A large number of broadcast route requests increase control overheads and consume a large amount of power to send and receive control messages. Thus, this paper proposes a Link Repairing for Inter-piconet Communication (LRIC) technique for a Bluetooth Scatternet to recover lost links efficiently. The proposed LRIC achieves several advantages i.e. minimizing the number of control messages, increasing throughput and reducing network delay. The simulation results show that the proposed technique outperforms existing techniques; the LRIC reduces 20% to 30% message overhead and increases 15% to 20% throughput.**

***Index Terms*—Bluetooth, broken link, link repair, packet reliability, scatternet.**

I. Introduction

Bluetooth is playing a key role in communications with electronic devices and it is an emerging standard for Wireless Personal Area Networks (WPANs) [1]. Bluetooth is a low power, low-cost and short range wireless technology. Bluetooth uses a set of 79 radio frequencies (RF) in the ISM band at about 2.4GHz. The Frequency Hopping Spread Spectrum (FHSS) technique is used to avoid interference. A Bluetooth hoping sequence is determined from the Bluetooth device address that is known as a channel [2]. Each channel is divided into time slots of $T = 625\mu s$, which is synchronized to the master clock. Bluetooth devices employ the Time Division Duplex scheme channel access. Initially, Bluetooth was launched as a wireless cable replacement; therefore, initially the RF range of Bluetooth devices was 10 meters [3].

A Bluetooth basic communication unit consists of up to eight active Bluetooth devices; the devices play different roles that include one master and seven active slaves at a time. It is the duty of the master in a piconet to manage and schedule data transmission and channel allocation for slaves [4]. Slaves are synchronized with the master and listen to the master; a slave may reply to the response of the master if and only if the slave is addressed explicitly by the master. The master sends packets to slaves in even-numbered slots and the slave responds to the master in subsequent slots (odd). Slaves are allowed to send packets only in response to the master node. Data are transmitted in the network using the structure of packets; each packet can be as long as 1, 3 or 5 slots [5]. The frequency remains unchanged during the transmission of a packet; ahead of the transmission of the next packet the frequency is altered, which is called fast hopping and enhances the communication reliability. The master node has a unique queue for each of its slaves, whereas a slave maintains a queue of packets for its master. If the master has no data for the slave, it sends a zero payload POLL packet to the slave; in response, the slave sends a NULL packet to the master [6].

If two Bluetooth devices belong to different piconets, they cannot directly communicate with each other although their distance is less than the radio communication range; this is because their channel hopping sequences are different[7]. To join piconets is a more complex structure; it is achievable through a scatternet which consists of many piconets. A node plays the role of a relay/ bridge if it joins multiple piconets. A relay uses different hopping patterns for different piconets; the relay forwards data from one piconet to another. The relay time in each piconet, and the switch-over times are arranged by the selected bridge scheduling algorithm [8],[9]. All transmissions in Bluetooth are synchronized; the slave frequency and phase are synchronized with the time slot clock of the master device. The relay adjusts its frequency before switching to the next piconet to join it and the master sends responses in even slots. For synchronization, the relay needs about 3 slots. The relay is an important node in the network as its mobility and degree affects the system performance. It is observed that needless relays increase packet loss rate and transmission time due to switching in different piconets, which raises the scheduling overhead. Frequent movements of the relay disconnect multiple links which increases control overhead. Forwarding and routing are not defined in the Bluetooth specification from one piconet to another [2]. The bridging function is handled by the layer protocol stack; it does not exist in Bluetooth specification which effects performance.

The main contributions of our work are summarized as follows:

In summary, the relay provides connection among different piconets and the mobility of the relay disconnects scatternets. So Bluetooth devices follow broadcast mechanisms to re-build broken links; broadcasts increase control overheads, and decrease network reliability and the packet delivery rate in a Bluetooth Scatternet. Therefore,

Manuscript received June 14, 2012; revised July 13, 2012.

The authors are with the Department of Computer and Informaiton Sciences, Universiti Teknologi PETRONAS, Bandar Seri Iskandar, 31750 Tronoh, Perak, Malaysia

this motivates the proposal for a local repair technique that efficiently repairs broken links in Bluetooth. The main concept is summarized below. First, each master node maintains the relay nodes information based on an existing Hello message. Second, a unicast-type repair message is sent to repair the broken wireless links that is caused by the relay node mobility.

The rest of the paper is organized as follows: Section 2 discusses the overview of background and related works. The proposed LRIC technique is discussed in Section 3. The performance analysis of the proposed technique and its comparison with some standard Bluetooth protocols are discussed in Section 4. Finally, the conclusion is drawn in section 5.

II. Background and Related Works

In Bluetooth each node acts as a wireless router that has the ability to forward packets. The main features of Bluetooth include supporting convenient access to the shared resources via mobile nodes and Bluetooth dynamically forming mobile nodes into a network where each node has the capabilities of processing and wireless forwarding. The limitations of Bluetooth include node mobility and limited resources, which cause a dynamical topology change. As a result, unreliable links and inefficient data transmission degrades the system performance.

Bluetooth has two types of routing protocols; these can be classified into the proactive and on-demand routing protocols. In the proactive protocols, the Topology Dissemination is based on the Reverse-Path Forwarding protocol. In this type, the node maintains a routing table and periodically exchanges routing state information. Proactive protocols need extensively high power consumption for maintaining the routing table, and nodes do not require a path discovery procedure before transmitting packets. On the contrary, on-demand protocols need to discover a routing path when a node wants to communicate with another node. Although On-demand protocols need more time to discover the route, these are preferred for Ad hoc networks.

Bluetooth devices have to play one of the three roles; that is master, relay or slave. The master controls the piconet, the slave is a dependent node attached to the master, and the relay node joins multiple piconets. A relay has to synchronize each time with a master before communication. The relay is a member of more than one piconet; a master switches to the inquiry state and gets the Bluetooth address (48-bit) and the clock information before adding a new slave. When the master changes to the paging state, it allocates a unique 3-bit Active Member Address (AM_Addr) to the active slaves. The master has only 7 active member addresses ranging from 1-7 or 000-111; if the AM_Addr is increased, then no more slaves are allowed by the master. Used as an example, Fig. 1 shows R_1 and R_2 concurrently taking part and connecting piconets P_1 and P_2. The packet transmission from piconet P_1 to P_2 is shown in time slot t_1 where R_1 synchronizes with piconet P_1 and receives a data packet from master M_1. Once, R_1 time completes in P_1 it enters into sniff mode (low power) in P_1 and switches to P_2 at the same time. Then R_1 synchronizes with P_2, and waits for M_2 to allow the communication. R_1 receives data from master M_2 in the even slot and transmits to M_2 the data received from M_1 in the next odd slots.

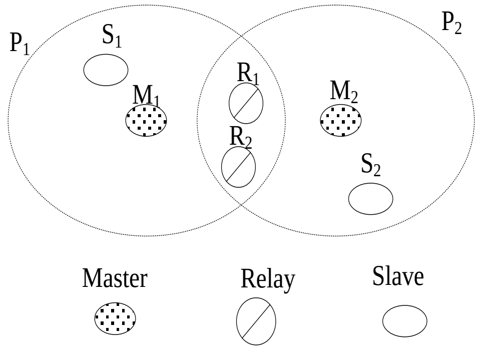

Fig. 1. Scatternet structure

The above example shows that R_2 is an unnecessary consumption of the AM_Addr, as R_1 is sufficient for the communication between P_1 and P_2. The AM_Addr is a useful resource if a relay participates in (n) piconets it consumes (n) AM_Addrs that increases overhead. The Bluetooth specification does not provide any solution for a scatternet configuration; therefore, different approaches have been used for scatternet configuration. We consider the most related work for the proposed technique.

A Routing Vector Method (RVM) [10] constructs a routing path for a pair of Bluetooth devices that is based on a flooding scheme. The RVM does not reduce the relays; due to this inefficient utilization of relay system resources causes them to be wasted. As shown in Fig.1, the RVM utilizes maximum relays, which decreases the system performance. The RVM problem was solved by a distributed relay reduction DRR [11], which performs relay reduction and disjoint routes construction for a scatternet over a Bluetooth radio system. However, the DRR still has some shortcomings; due to mobility there are many chances that selected routes can break. It is analyzed that the RVM and DRR do not consider the mobility of nodes.

While a relay is needed for inter-piconet communication [4], [12], [13], and large numbers of relays are useful for fast flooding, however, unnecessary relays have many drawbacks. One disadvantage is the consumption of the AM_Addr, and also an increase in maintenance overhead. When the relay moves, it breaks the connected links, so Bluetooth devices restart broadcasting to find the substitute route, which degrades the system performance.

This paper proposes a Link Repairing for Inter-piconet Communication (LRIC) technique to overcome control overheads by utilizing the unnecessary relays for route repairing. The proposed work focuses on the On-demand routing to improve the unreliable link problem. The LRIC technique tries to repair a broken link locally instead of broadcasting a route request message.

III. The Proposed Link Repairing

This section consists of the Link Repairing for Inter-piconet Communication (LRIC) technique, which locally repairs broken links.

A. Relay Information

Each master maintains relay information in a relay table, where the master stores the relay ID, clock-offset and degree (connections). In Table I, the first row shows the master of each piconet in ascending order and the first column shows the relay ID. Connection between the masters is represented by 1 and null shows that there is no connection between the masters. A Relay Connection Table (RCT) is maintained by M_2 based on Fig. 2, as shown in Table I R_1 is connecting M_1 and M_2 therefore their correspondence entries are 1, and the rest are null. By considering degree, each master locally decides the relay according to the DRR. The main difference between the proposed technique and the DRR is that the DRR removes the lower degree relay nodes' information from the relay table and assigns them slave roles. While, the proposed technique does not remove the lower degree relay node information, and instead assigns a Backup relay (BR) role to the lower degree nodes. As Bluetooth devices are connected in Ad Hoc Network fashion so the devices can randomly move.

TABLE I: RELAY CONNECTION TABLE (RCT)

ID (Relay/Master)	M_1	M_2	M_3	M_4	clock-offset
R_1	1	1	Ø	Ø	clock-offset (R1)
R_2	Ø	1	1	1	clock-offset (R2)
R_3	Ø	1	Ø	1	clock-offset (R3)

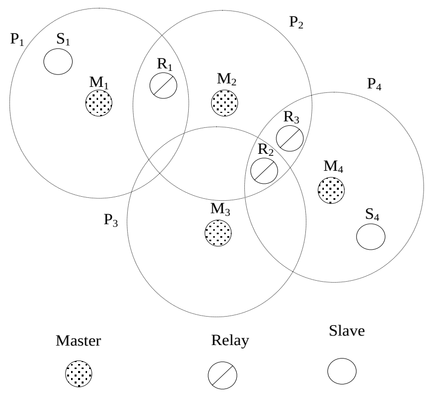

Fig. 2. A connected scatternet through RVM

B. Mobility Model

The proposed technique considers the Random Walk Mobility Model (RWMM), which is based on random speed and directions. It is a simple mobility model**;** the RWMM was first described by Einstein in 1926 through mathematics. In our daily life many entities move in really unpredictable ways, and the RWMM notes the irregular movement [9]. In an RWMM a Mobile Node (MN) moves from one location to another by random direction and speed. Each movement in the RWMM occurs in either an unvarying time interval *t* or a constant distance travelled *d*; when it reaches the end, a new direction and speed is calculated. As 1-D, 2-D, 3-D, and d-D walks derivatives of the RWMM have been developed. Through the RWMM, the movements of entities are tested around their starting points. When MN starts its movement it changes randomly its direction and speed. The previous speed and location is not needed by the RWMM, therefore it does not need memory to maintain the previous pattern values. The RWMM direction and speed does not depend on the previous values [10] so we can generate unlikely schedules such as unexpected stops and quick turns. A Mobile node travels in a predefine time before changing the speed and direction. A node can change its direction after reaching the desired location rather than a specific time. Fig. 4 shows another example in which MN travels for a predefined number of steps instead of time before changing its speed and direction.

C. Backup Relay (BR) Selection

Each relay transmits its Bluetooth Device Address (BD_Addr), clock-offset and degree to its connected master. The relay transmits its link information to all connected masters where it plays the role of a slave; as shown in Fig. 2, R_1 was a pure slave in P_2 with master M_2. When R_1 receives a connection message from M_1, after establishing connection, it sends the new connection information to M_2. Each time after establishing a connection, the relay node updates all masters about its new degree. Since each master maintains a relay table, when the master gets a new message from the relay node it updates the RCT. Once, the RCT is constructed, each master verifies which relay should be assigned the role of the BR. According to the DRR, the higher degree node is selected as the relay, with this proposed work, the same procedure is followed but lower degree nodes are assigned the Backup Relay (BR) roles. Table I is used to illustrate the LRIC technique. *M2* checks the degree of each relay and as R_1 is the only relay between *M1* and *M2* it remains as a pure relay as *R3* and *R2* are both providing connection between *M2* and *M4.* On the other hand, *R2* is a higher degree relay, as it can provide connection among three piconets, so it is selected as a pure relay. In addition, *R3* is assigned the BR role because its degree is lower. According to the DRR, *R3* participates as salve in *P4*, because *P4* has less number of hosts as compared to *P2*. However, according to the LRIC, *M2* assigns the BR role to *R3* and saves its ID and clock-offset in the RCT. After applying the BR procedure, the network topology is shown in Fig. 3 and the updated table of *M2* is shown in Table II. Where the *R3* role has been changed to a BR, and the remaining scatternet is perfectly connected.

TABLE II: AFTER EXECUTION OF CRR OPERATION

ID (Relay/Master)	M_1	M_2	M_3	M_4	clock-offset
R_1	1	1	Ø	Ø	clock-offset (R1)
R_2	Ø	1	1	1	clock-offset (R2)
BR_3	Ø	1	Ø	1	clock-offset (R3)

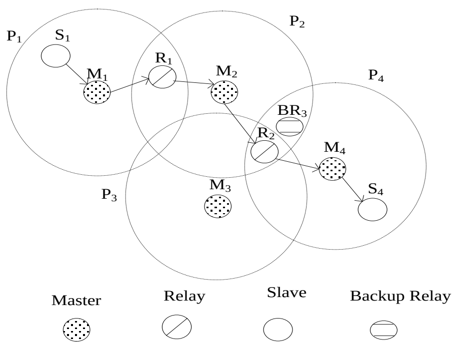

Fig. 3. After execution of relay reduction through LRIC

D. Route Repairing

The proposed LRIC technique is used to locally repair the broken links rather than broadcast a route discovery. Additionally, each master maintains a relay node connection table. The LRIC approach is described as follows.

The proposed LRIC is divided mainly into two steps.

Step1. Determine broken links when a master receives a packet and finds the next hop is missing. The master waits and sends a POLL packet again to the relay node; if the master does not get any reply it realizes that the link is broken.

Step2. If the master detects that a new broken link has occurred, it initializes the repair procedure. Since the master has the BR information, it has ability to execute the repair procedure to repair the link. The master insures that the length of the substitute route should not increase.

For instance, as shown in Fig. 3, a routing path is constructed between S_1 and S_4 through ($S_1 \rightarrow M_1 \rightarrow R_1 \rightarrow M_2 \rightarrow R_2 \rightarrow M_3 \rightarrow S_4$); during transmission, M_2 detects that R_2 is unreachable due to the link being broken. Thus, M_2 starts the route repair process and executes the LRIC. M_2 determines all the BR nodes from the RCT and it finds that R_3 is a node which can provide connection between M_2 and M_3. After selecting R_3 as the backup relay node, M_2 sends a Page message to R_3 to repair the broken link. Once the connection is established M_2 sends a unicast repair message to S_1 and S_4 that contains information about the new route i.e. ($S_1 \rightarrow M_1 \rightarrow R_1 \rightarrow M_2 \rightarrow R_3 \rightarrow M_3 \rightarrow S_4$).

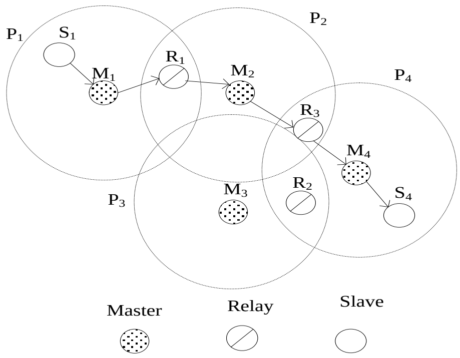

Fig . 4. After route repairing through LRIC

The main contribution of the proposed work is that all the repair messages are unicast-type, which avoids message flooding and thus minimizes control overheads. If the route repair is not successful, the master node notifies the source node with a route failure message.

The steps for the LRIC are given below.

Step 1: Each relay in a Scatternet sends its information (clock-offset, BD_Addr and degree) to all connected masters.

Step 2: The master updates the RCT, when it receives a new message from the relay node.

Step 3: The master receives a data packet and forwards it to the relay node, and waits for a reply.

Step 4: If the master does not get a reply message from the relay, it assumes the link is broken.

Step 5: The master sends a unicast reply to the source node to stop transmission and waits for a route repair packet.

Step 6: The master finds a backup relay from the RCT and sends a Page message to the BR node.

Step 7: On receiving a successful reply, the master sends a unicast route repair message to the source and destination hosts. On receiving the route repair message, the source and destination hosts restart communication through the new route.

Step 8: If the route repair fails, the master sends a route error message to the source node. After repairing the broken route through the backup relay, a new scatternet formation is as shown in Fig. 4, where the scatternet remains perfectly connected.

IV. Performance Study

In this section the performance of proposed LRIC is analyzed and compared with the standard Bluetooth protocols such as the DRR [11] and the RVM [10]. The simulation is performed by using the UCBT [14] that is an extension of the NS-2 [15] for Bluetooth. The UCBT is one of the Bluetooth simulators that offer a rich environment for simulation at different Bluetooth layers. The simulation environment is described as follows. The space size is set to 80x80 m^2 and the number of devices is varied from 15 to 90, and their locations are randomly determined. The total simulation time is 200s; the first 40s are used for link connections and the CBR traffic is generated at the 50^{th} second. Table III shows the performance metrics of the simulation.

TABLE III: Simulation Parameters

Parameter	Value
Number of nodes, N	$15 \leq N \leq 90$
Simulation area	80 x 80 m^2
Radio propagation range	10m
Traffic model	CBR (Constant Bit Rate)
Energy consumption	0.0763 x 10^{-6} J/bit
Packet type	DH3, DH5
Pair of source and destination	15 - 40
Simulation time	200s

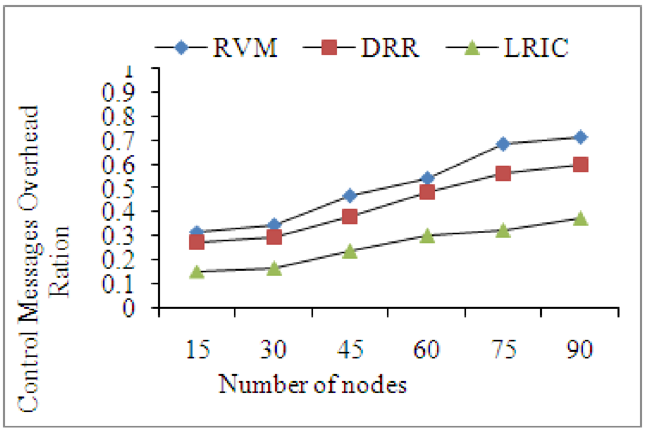

Fig. 5. Control messages overhead vs. number of nodes

A. Control Packets Overhead

Control packets refer to the number of messages used for synchronization and route construction. It is observed that the RVM has the highest overhead because it utilizes the maximum number of relays. Although, the DRR performs relay reduction, it follows the broadcast mechanism to reconstruct the broken link. The relay reduction scheme is run by each relay to decide its role to play as a relay or slave. If the relay plays the role of slave, then it sends requests to all masters to get the number of nodes connected and participate in the less connected nodes. In the LRIC technique, a master locally repairs the disconnected link, which reduces control overhead as compared to the RVM and DRR. As shown in Fig.5, the LRIC has almost 20-30% less control overhead as compared to the RVM and DRR.

B. Throughput

Throughput refers to the average rate of successful messages delivered or successful data bytes received by a node in per unit time. As the number of relays increase in the scatternet, there is also an increased probability of packet loss. Due to relay mobility, the RVM and DRR have less throughput compared to the proposed technique. If a relay node moves frequently, it will increase control overhead and decrease the system performance. The mobility of a relay node not only breaks its connection but also disconnects the communication in different piconets. As the number of relays exceeds what is necessary in the network, it also destabilizes the network; this means it increases the probability of link disconnection and increases packet loss during communication. As shown in Fig. 3 due to the unavailability of the relay node, transmission is blocked for a longer amount of time. In the proposed technique, a broken link is locally repaired. Therefore, the LRIC is more effective than the RVM and DRR in terms of throughput. Fig. 6 shows that the LRIC throughput is 15-20% higher compared to the DRR and RVM.

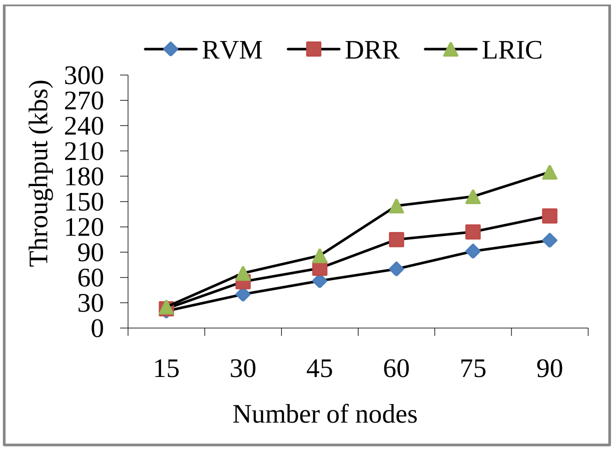

Fig. 6. Throughput vs. number of nodes

C. Network Delay

It is observed that the LRIC reduces delay, as a backup relay is activated when a link is broken due to the relay node. If a relay node moves during transmission, it aborts the transmission. Every time a new procedure starts from the inquiry and the inquiry scans to re-establish the disconnected link, which increases the network delay. In the proposed technique, the BR information is stored locally and due to mobility, a BR is selected without increasing network delay. It is observed through Fig. 7 that the delay in the network increases gradually when the number of connections increases. The RVM and DRR do not have information about the BR; therefore, both techniques' network delays are higher compared to the LRIC.

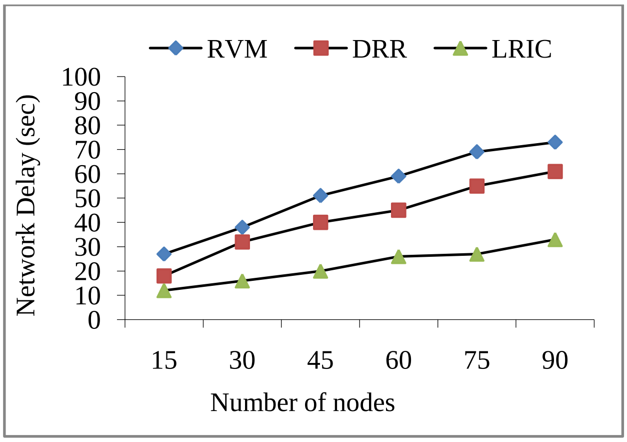

Fig .7. Network delay vs. number of nodes

D. Energy Consumption

The comparison of energy consumption is shown Fig. 8 of RVM, DRR and LRIC for different number of nodes. It is analyzed that both RVM and DRR consume more energy compared to proposed technique. The main reason for this RVM flooding consumes more energy, whereas DRR need bulk of control messages for relay reduction every time when a route request comes. On the other hand, the proposed LRIC utilize in control manner message and only utilize the optimal number relays, which saves the energy for scheduling and unnecessary flooding. It is observed that LRIC almost reduce 30% to 45% energy consumption compared to RVM and DRR. Whenever any link brakes RVM and DRR need to follow conventional Bluetooth procedure to establish the broken link, which causes extra energy consumption.

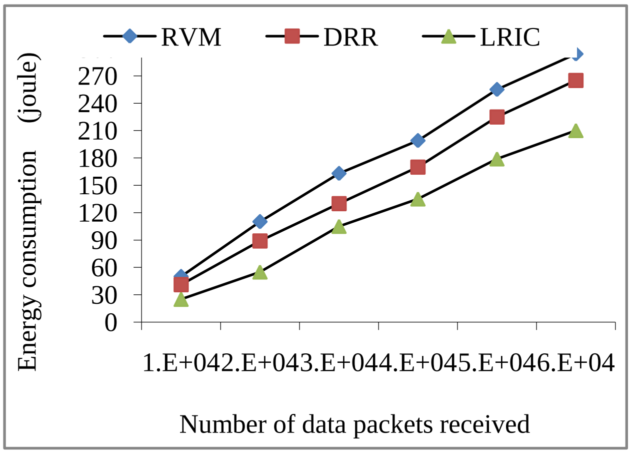

Fig. 8. Energy consumption vs. number of data packets received

V. CONCLUSION

This paper proposed the Link Repairing for Inter-piconet Communication technique that considers the Random Walk Mobility Model. The proposed LRIC technique repairs the broken links to reduce flooding overhead without effecting the scatternet connection. The proposed LRIC technique reduces delays and makes valuable communication possible in at scatternet. The simulation results are compared against the RVM and DRR. While the RVM and DRR are based on a broadcast mechanism to re-establish broken links, the LRIC locally repairs the broken links. Therefore, it is analyzed that the LRIC technique outperforms the RVM and DRR in terms of control message overhead, delay and throughput. It is observed that the LRIC reduces almost 30% of the control overhead and increases 20% throughput as compared to the RVM and DRR. These applications can be used in shopping malls, earthquakes, mobile e-commerce scenarios and supermarkets.

REFERENCES

[1] T. Salonidis, et al., "Distributed topology construction of Bluetooth wireless personal area networks," *Selected Areas in Communications, IEEE Journal on*, 2005, vol. 23, no. 3, pp. 633-643.

[2] Bluetooth_specificaiton. Bluetooth SIG, Specification of the Bluetooth System. [Online]. Available: http://www.bluetooth.com, 2010.

[3] C. D. M. Cordeiro, et al, "A novel architecture and coexistence method to provide global access to/from Bluetooth WPANs by IEEE 802.11 WLANs. in Performance, Computing, and Communications Conference," 2003. *Conference Proceedings of the 2003 IEEE International*, 2003.

[4] H. C. Fang and H. S. Ming, "An Adaptive Interpiconet Scheduling Algorithm Based on HOLD Mode in Bluetooth Scatternets," *Vehicular Technology, IEEE Transactions on*, 2008, vol. 57, no.1, pp. 475-489.

[5] Y. D. Cui, et al., "Bluetooth energy-saving optimization: a queuing theory analysis," *The Journal of China Universities of Posts and Telecommunications*, 2010, vol. 17, pp. 54-57.

[6] L. Jingli, L. Xiangqian, and A. Swami, "Collision analysis for coexistence of multiple bluetooth piconets and WLAN with dual channel transmission," *Communications, IEEE Transactions on*, 2009. vol. 57, no. 4, pp. 1129-1139.

[7] J. Misic and C. J. Fung, "The impact of master-slave bridge access mode on the performance of multi-cluster 802.15.4 network," *Computer Networks*, 2007, vol. 51, no. 10, pp. 2411-2449.

[8] C. Yu, K. M. Yu, and S. Lin, "Efficient Scheduling Algorithms for Bluetooth Scatternets," *Wireless Personal Communications*, 2009, vol. 48, no. 2, pp. 291-309.

[9] B. V. Misic, and M. J. Polling, and B. Scheduling, "Algorithms in Bluetooth, in Technical report TR 03/04 Department of Computer Science, University of Manitoba Winnipeg, Manitoba, Canada R3T 2N2. 2003.

[10] P. Bhagwat and A. Segall, "A routing vector method (RVM) for routing in Bluetooth scatternets. in Mobile Multimedia Communications," *IEEE International Workshop on*, 1999.

[11] G. J. Yu, et al, "Relay reduction and disjoint routes construction for scatternet over Bluetooth radio system," *Journal of Network and Computer Applications*, 2007, vol. 30, no. 2, pp. 728-749.

[12] S. T. Bakhsh, H. Hasbullah, and S. Tahir, "Dynamic Congestion Control through backup relay in Bluetooth scatternet," *Journal of Network and Computer Applications*, 2011, vol. 34, no. 4, pp. 1252-1262.

[13] G. J. Yu and C. Y. Chang, "Congestion control of bluetooth radio system by piconet restructures," *Journal of Network and Computer Applications*, 2008, vol. 31, no. 3, pp. 201-223.

[14] University of Cinicinnati Bluetooth simulator (UCBT), [Online]. Available: http://www.ececs.uc.edu/_cdmc /ucbt/>

[15] The Network Simulator ns-2, [Online]. Available: http://www.isi.edu/nsnam/ns/ns-build.html, Jan 2011.