

Initialization Method for Communication and Data Sharing in P2P Environment Between Wireless Sensor Nodes

M. Asif Jamal¹

Department of Computer Science
& Engineering,
Air University,
Islamabad, Pakistan

Aziz Ur Rehman²

Department of Computer Science & IT
The University of Lahore,
Gujrat, Pakistan

Moonisa Ahsan³

Department of Computer Science & IT
The University of Lahore,
Gujrat, Pakistan

M. S. Riaz⁴

Department of Computer Science
& Engineering,
Air University,
Islamabad, Pakistan

M. S. Zafar⁵

Department of Computer Science
Beihang University of Aeronautics
& Astronautics (BUAA),
Beijing, China

Abstract—Wireless Sensor Networks have increased noteworthy thought nowadays, rather than wired sensor systems, by presenting multi-useful remote hubs, which are littler in size. However, WSNs correspondence is inclined to negative impacts from the physical environment, like, physical hurdles and interference. The reason for this work is to outline a testbed, to introduce method for communication startup and data sharing in a peer to peer (p2p) environment between wireless sensor nodes. The work is directed on both the IEEE 802.15.4 physical and the application layers. In this testbed, one channel, from the IEEE 802.15.4 channels range is devoted as an “emergency channel” which is utilized for handshaking or in case there is communication failure between the Transmitter (Tx) and Receiver (Rx) nodes. The remaining 15 channels are called “data channels” and are utilized for real information transmission and control signals. Linux based TinyOS-2.x is utilized as a working framework for low power sensors. MICAz bits are utilized as nodes and a MIB520 programming board is utilized for burning the codes and for the purpose of gateways.

Keywords—TinyOS; peer-to-peer; motes; testbed; nesC; MICAz; MIB520; handshaking

I. INTRODUCTION

An awesome improvement is going on these days with the generation of low power remote sensors. However, during the communication of these low power enabled sensor nodes, interferences must be avoided from any source internally or externally. Therefore, it is necessary to make a testbed to evaluate result in an environment that eliminates the internal or external source of interference. Testbed for wireless sensor networks can play an important role in academia because, theoretical study and simulation show results in ideal situation. A wireless sensor network consists of low power sensor nodes, which have the responsibility to sense the task assigned to them and report this sensed information via some wireless link to gateway. Usually these nodes comprise of a microchip, which is responsible for transmission and reception of data [1], [2].

In WSNs a good testbed ought to have these properties,

- 1) Maintain synchronization in the occurrence of communication failure.
- 2) Should have the ability of P2P communication symmetry.

Section 2 presents literature review and background study whereas Section 3 described the methodology followed by Section 4 that includes implementation and results. Section 5 shows conclusion and future work.

II. LITERATURE REVIEW

This work is planned to make a testbed which conveys improvement to a current testbed created at Mid-Sweden University (MIUN) “An Empirical Study of Low Power Multichannel correspondence in WSN”, created by authors [3], which, over the long haul, will be advantageous for outlining new conventions. On account of correspondence misfortune, a calculation is intended for the synchronization of nodes. For transmitter-receiving symmetry the calculation is outlined so that after a particular number of packets have been sent by the transmitter, both nodes change their part.

The correspondence depends on the IEEE 802.15.4. It uses carrier sense multiple access with collision avoidance (CSMA/CA) as an access provision method. As CSMA/CA works on low data application, so it provides enough throughput without severe interference and delay [4]. This IEEE 802.15.4 gives an aggregate 26 channels [5] and from this, one lies in the 868 MHz band (utilized as a part of Europe later extended to three in 2006 [6]), 10 lies in the 915MHz band (utilized in North America extended to thirty in 2006 and 16 lies in the 2.4GHz band (utilized around the world). The 2.4GHz variant of IEEE 802.15.4 offers the most astounding throughput of 250 kbps, and is, hence, covers long distance [7]. Subsequently this testbed concentrates on the 2.4 GHz

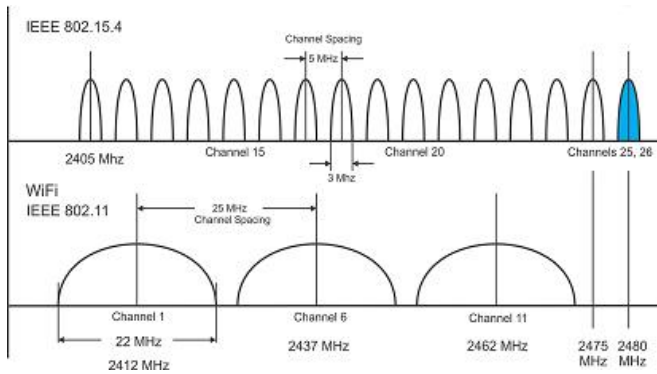


Fig. 1. IEEE 802.15.4 and Wi-Fi Spectrum [8].

range of 16 channels ranging from channel no 11 to 26. Fig. 1 shows the IEEE 802.15.4 and IEEE 802.11 Spectrum.

The existing MIUN testbed [3], for WSN has shown problems. The main issue was the correspondence failure between the gateway and the nodes for longer interval of time, which results loss of important time during the field trial. The objective of this work has been to respond to the following questions.

- 1) How to synchronize the nodes when the communication is lost due to hardware problem or inter-channel interference.
- 2) Implementation of peer to peer communication between transmitter and receiver nodes.

A. Definition and Implementation of a New Initialization Algorithm

In previous studies [3], after five unsuccessful beacon message transmissions, the transmitter node (Tx) switches to the next channel, and the Receiver Node (Rx) will remain on the current channel as it is not receiving beacon, meaning that it must wait until the Tx has made a full sweep over all channels, for a new communication attempt to occur. The new algorithm should eliminate this problem. The algorithm can be in the form of a handshake or any other method which is robust to packet loss. The new algorithm should ensure that the effort to establish the communication will be made more efficient.

B. Handshaking

In communication, handshaking mechanism between nodes is considered to be most significant for connection establishment. Whenever we wish to establish or re-establish a connection, handshaking is the foremost step. Therefore, handshaking can be defined as the process in which Tx broadcasts a number of beacons, which, if the signal is received efficiently to the Receiver node (Rx), then the Rx will response with an acknowledgment (ACK) message. The ACK message shows the successful agreement between sender and receiver nodes for conducting efficient handshaking for peer-to-peer communication [9]. The same technique will be implemented in this work and one channel from the IEEE 802.15.4 spectrum (channel # 26) with proficiency of 2.4 GHz bandwidth will be dedicated to establish hand-shaking path between nodes.

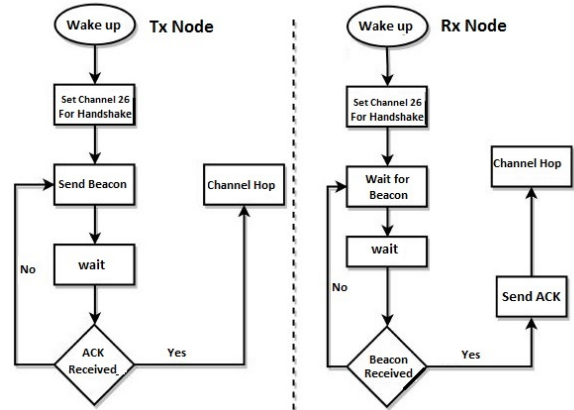


Fig. 2. Flowchart for Tx & Rx Hand Shake.

C. Introducing Peer to Peer (P2P) Symmetry between the Transmitter-receiver Pair

In the previous setup [3], the Tx triggers the communication with Rx via beacons. After sending 5 beacons Tx starts the transmission of data packets, while the Rx stores the packet logs. In our proposed setup, the code should be modified so that, in the first round after a predefined number of packets transmitted by Tx, the two communicating nodes should switch roles and the direction of communication changes. In other words, the node that has been receiving and logging data should, in the following round, take up the transmitter role, and the former transmitter should maintain packet logs. This task also comprises the introduction of acknowledgment packets.

D. Channel Hopping/Frequency Hopping

Channel Hopping is the consecutive change of channel in the available frequency range. Because of obstruction in the remote medium [10], channel hopping must be performed in a manner that both Tx and Rx might change channels at the same time. In the testbed channel hopping calculation must be characterized for both Tx and Rx.

III. METHODOLOGY

In the IEEE 802.15.4 range, channel 26, is termed as an “Emergency Channel” in this testbed, while the remaining channels (11 to 25) are called “Data Channels”. At first, for peer to peer correspondence, it is vital that the two nodes must handshake with each other before an information trade can happen. In this way, the emergency channel is dedicated for handshaking between the Tx and Rx. The data communication channels are devoted for actual information transmission. Fig. 2 demonstrates the handshaking process.

In Fig. 2 when the Tx is started transmission at the emergency channel it then begins sending beacons occasionally and waits for an affirmation in the form of ACK from Rx. If Rx receives the beacon, then it replies with an acknowledgment and jumps to the first data channels (e.g. Channel 11). Tx likewise hops to the same data channel, predefined to both Tx and Rx.

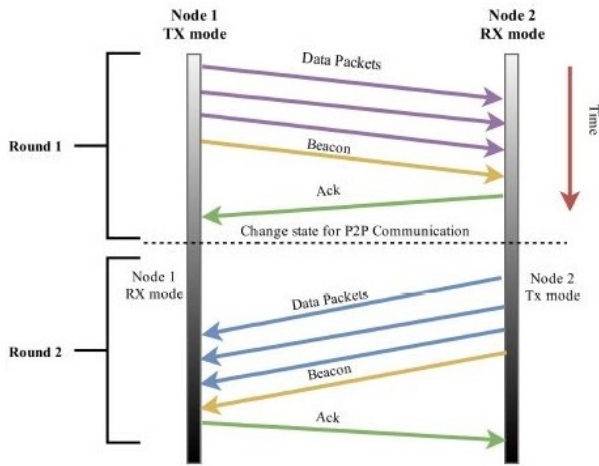


Fig. 3. P2P data transmission symmetry in a single data channel.

Keeping in mind the end goal to acquaint the symmetry with the P2P pair at the data communication channels, a packet counter is utilized to record the number of packets transmitted by the Tx. When the maximum packet counter limit is reached, a beacon is sent by Tx to remind Rx, to change its transmission direction (e.g. from Rx to Tx). Upon the successful reception of beacon message by Rx, it begins sending data packets to Tx to form a peer to peer communication. At the point when the maximum packet counter point is reached, again a beacon is sent by Rx and it waits for an ACK from Tx. After receiving the beacon by Tx, it sends an ACK to Rx, and hops to the next channel. At this point, successful reception of ACK, Rx likewise hops to the following data channel. The above methodology is reshaped in all data channels. Fig. 3 demonstrates a diagrammatic representation of the Tx and RX P2P transmission symmetry. If a packet/channel loss or hardware failure occurs, then the nodes likewise hop back to the emergency channel for handshaking.

For explaining the Task 2 more deeply, flow-chart diagrams and following terms will be considered.

At the emergency channel:

- 1) The Tx is capable of sending beacon only to check the communication path with Rx for hand-shaking process.
- 2) The ACK is only send by the Rx after the successful receiving of the beacon packet.

At the data channel:

- 1) After the last data packet (i.e. the final packet preceded by the beacon), beacon is sent to the destination. If the beacon is successfully received by Rx, it shows that a shift of the transmission roles should occur, and if the beacon is received by Tx, it shows that it should hop to the next available data channel in network.
- 2) ACK is sent upon successful reception of the Beacon.

Peer-to-peer communication among two motes is implemented in such a way that the communication path will alter whenever the beacon packet is sent by Tx, after sending a

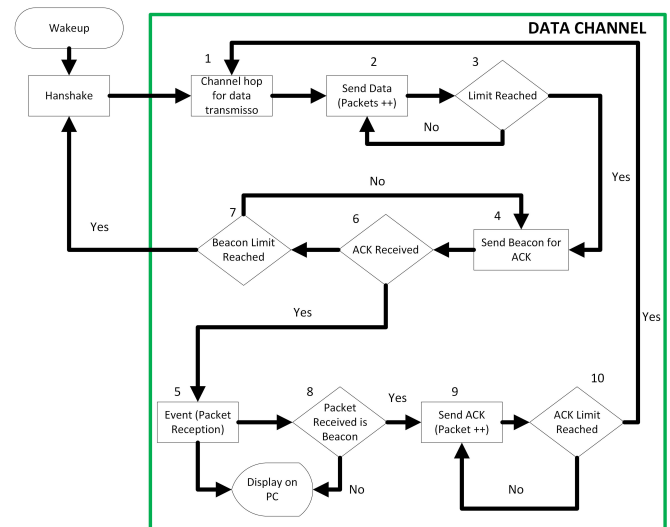


Fig. 4. Tx flow chart for p2p communication.

specified number of data packets and is successfully received by the Rx Channel. Fig. 4, the flow-chart diagram of Tx Peer-to-peer communication is illustrated. When the Tx hops from emergency channel to the initial channels by means of the channel hopping task (1), the Tx initiates the process of sending data packets (2). When the defined maximum packet limit is reached, Tx then halts the process of sending further data packets (3) and sends a beacon (4) to alert Rx to take its turn of communication. If a successful ACK is received (6) from Rx then the Tx node gets into waiting mode for receiving data packets from Rx (5). During this process, while Tx is receiving data packets, if a beacon is received (8) from Rx then Tx responds back with an ACK Alert (9) (according to our testbed criteria, it will send 3 ACK (10)) and then hops to the next available data channel in the network by altering its path using channel hopping task (1).

A. Hardware Failure or Communication Loss

In Fig. 4 if Tx is not receiving an ACK (6) from Rx, it means that communication loss or hardware failure has occurred, then Tx will keep sending beacons until it reaches its maximum beacon limit (maximum beacon limit = 3). If no ACK is received by Tx based on the last beacon sent (7), then it will jump to the emergency channel.

B. RX Flowchart

In Fig. 5 below the flow charts inside green border shows RX P2P communication at the data channel where the Rx receives packets from Tx. Every packet received from Tx is forwarded to a serial port for PC logging. If, during the packet receiving, a beacon is received (3) then Rx will reply with an ACK (4) and starts sending data packets (5) until the maximum number of data packet limit, is reached and a beacon is sent (6), to inform Tx that it is time to jump to the next data channel and wait for an ACK from Tx. If an ACK is received (7) then Rx jumps to the next data channel by means of the channel hopping task (1).

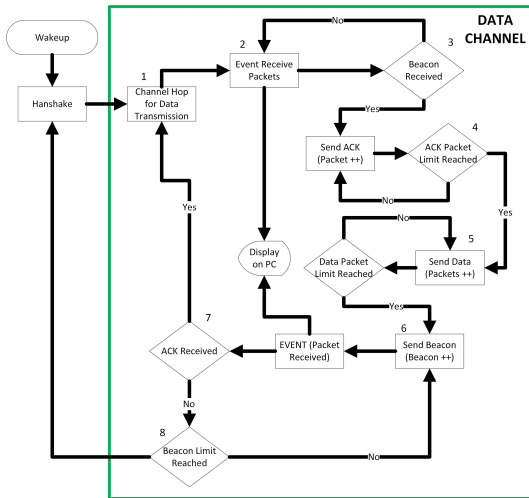


Fig. 5. Rx flow chart for p2p communication.

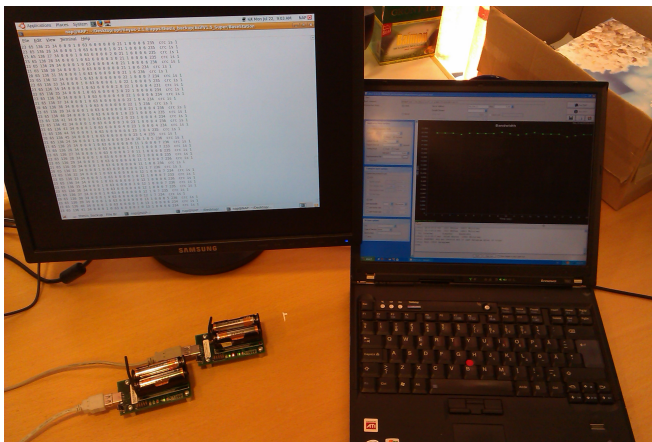


Fig. 6. Testbed Setup.

C. Hardware Fail, Channel Loss or Packet Loss

In Fig. 5, if Rx is not receiving an ACK (6) from Tx, it means that communication loss or hardware failure has occurred, then Rx will continue to send beacons until it reaches its maximum beacon limit (maximum beacon limit = 3). If no ACK is received by Rx after the last beacon has been sent (7), then it will jump to the emergency channel.

IV. IMPLEMENTATION OF THE RESULTS

To meet the testbed specifications and requirement, we have implemented Linux based TinyOS-2.x. MICAz motes [11] from Crossbow are used as hubs in our testbed and to enable communication among nodes, a MIB520 programming board [12] is used as gateway. For scripting and coding, NesC programming language is used [13], [14]. The IEEE 802.15.4 physical (PHY) packet format consists of a PHY header, PHY payload and PHY footer [15]. The following image, Fig. 6 shows the practically implemented testbed setup for this work.

In this testbed, the PHY header 12 bytes, 11bytes of payload (for data packet) and 2 bytes of PHY footer shown in Fig. 7.

```

BaseStation Program's Output Started
25 65 136 0 34 0 255 255 1 0 63 6 0 0 0 0 0 25 1 0 0 0 0 0 13 236 1
Next Packet
25 65 136 0 34 0 255 255 1 0 63 6 0 0 0 1 0 25 1 0 0 0 0 0 14 233 1
Next Packet
25 65 136 0 34 0 255 255 1 0 63 6 0 0 0 2 0 25 1 0 0 0 0 0 14 235 1
Next Packet
25 65 136 0 34 0 255 255 1 0 63 6 0 0 0 3 0 25 1 0 0 0 0 0 13 236 1
    
```

Fig. 7. PHY Packet Format.

```

21 65 136 107 34 0 0 0 1 0 63 6 0 0 0 0 0 26 1 0 239 235 crc is 1 Beacon
23 65 136 108 34 0 0 0 1 0 63 6 0 0 0 0 0 11 1 0 0 0 1 235 crc is 1
23 65 136 109 34 0 0 0 1 0 63 6 0 0 0 1 0 11 1 0 0 0 1 235 crc is 1
23 65 136 110 34 0 0 0 1 0 63 6 0 0 0 2 0 11 1 0 0 0 1 234 crc is 1
21 65 136 111 34 0 0 0 1 0 63 6 0 0 0 0 0 11 1 0 2 233 crc is 1
20 65 136 112 34 0 0 0 1 0 63 6 0 0 0 0 0 11 1 0 235 crc is 1
23 65 136 113 34 0 0 0 1 0 63 6 0 0 0 0 0 12 1 0 0 0 1 235 crc is 1 Data packets
23 65 136 114 34 0 0 0 1 0 63 6 0 0 0 1 0 12 1 0 0 0 0 236 crc is 1
23 65 136 115 34 0 0 0 1 0 63 6 0 0 0 2 0 12 1 0 0 0 1 234 crc is 1
21 65 136 116 34 0 0 0 1 0 63 6 0 0 0 0 0 12 1 0 1 235 crc is 1
20 65 136 117 34 0 0 0 1 0 63 6 0 0 0 0 0 12 1 1 235 crc is 1 ACK
23 65 136 118 34 0 0 0 1 0 63 6 0 0 0 0 0 13 1 0 0 0 0 236 crc is 1
23 65 136 119 34 0 0 0 1 0 63 6 0 0 0 1 0 13 1 0 0 0 0 235 crc is 1
23 65 136 120 34 0 0 0 1 0 63 6 0 0 0 2 0 13 1 0 0 0 0 233 crc is 1
21 65 136 121 34 0 0 0 1 0 63 6 0 0 0 0 0 13 1 0 0 233 crc is 1
    
```

Fig. 8. Beacon, ACK and Data packets format.

In Fig. 7, the green border shows a correct packet received by the BS. The first 12 bytes, with gray background (from 25 to 6) constitutes the packet header, the second 12 bytes, with the blue background (from 0 to 0) form the packet payload, and the last two bytes, with the yellow background (14 and 235) are the packet footer Received Signal Strength Indicator (RSSI) and Chip Correlation Indicator (CCI). The last bit (1) is the Cyclic Redundancy Check (CRC), which is the first bit of CCI byte (235), which is simply extracted to differentiate between a corrupted and uncorrupted packet. The first byte (25) in the packet header is the Frame length byte which in forms the CC2420 chip (transceiver chip used by MICAz) that the total length of the packet is 26 bytes (Frame length byte + 25 bytes). The payload part is the actual packet data sent by Tx in which, according to this testbed message type, the first four bytes (0002) are the packet counter, the 6th byte (25) is reserved for the IEEE 802.15.4 channel information (from channel 11 to channel 25) and the next byte (1) is the node Id (Tx) from which this packet is received. The first byte in the footer (14) is the RSSI value and the second byte (235) is the CCI. The last bit (1) in the red border, shown next to the packet, is the CRC which is the left most significant bit of the last byte (23510 = 111010112, Link Quality Indicator (LQI) is 235-128 =107)

In this testbed, three types of packets (Beacon, Acknowledgment and Data packets) are used which is shown in Fig. 8 with different frame length byte of 21, 20 and 23.

In case of a hardware/communication failure, both motes (Tx and Rx) can returned to the emergency channel and can reestablish communication. Fig. 9 shows the output of the Tx in the case of communication loss at channel 23.

The red border shows, the point at which the communication loss occurs. This communication loss occurred when the reset button at the Tx mote was pressed and hence the packet sequence number (4th byte in the header) was restarted (changed from 42 to 0). As it is cleared that the communication loss occurred at channel 23 then, instead of going to other channels (i.e. 24, 25), the Tx has jumped to the emergency channel (channel 26) and then, after handshaking (receiving ACK from Rx), communication is maintained in the next data


```

TX interrupted
nap@NAP: ~/Desktop/opt/tinyos-2.1.0/apps/t
File Edit View Terminal Help
21 65 136 37 34 0 1 0 0 0 63 6 0 0 0 0 0 22 0 0 6 236 crc is 1
20 65 136 38 34 0 1 0 0 0 63 6 0 0 0 0 0 23 0 4 236 crc is 1
23 65 136 39 34 0 1 0 0 0 63 6 0 0 0 0 0 23 0 0 0 5 236 crc is 1
23 65 136 40 34 0 1 0 0 0 63 6 0 0 0 0 1 0 23 0 0 0 5 236 crc is 1
23 65 136 41 34 0 1 0 0 0 63 6 0 0 0 0 2 0 23 0 0 0 6 236 crc is 1
21 65 136 42 34 0 1 0 0 0 63 6 0 0 0 0 0 23 0 0 5 236 crc is 1
20 65 136 0 34 0 1 0 0 0 63 6 0 0 0 0 0 26 0 6 234 crc is 1
20 65 136 2 34 0 1 0 0 0 63 6 0 0 0 0 0 11 0 8 231 crc is 1
23 65 136 3 34 0 1 0 0 0 63 6 0 0 0 0 0 11 0 0 0 8 235 crc is 1
23 65 136 4 34 0 1 0 0 0 63 6 0 0 0 0 1 0 11 0 0 0 8 234 crc is 1
23 65 136 5 34 0 1 0 0 0 63 6 0 0 0 0 2 0 11 0 0 0 8 233 crc is 1
21 65 136 6 34 0 1 0 0 0 63 6 0 0 0 0 0 11 0 0 8 235 crc is 1
20 65 136 7 34 0 1 0 0 0 63 6 0 0 0 0 0 12 0 8 235 crc is 1
23 65 136 8 34 0 1 0 0 0 63 6 0 0 0 0 0 12 0 0 0 9 236 crc is 1
23 65 136 9 34 0 1 0 0 0 63 6 0 0 0 0 1 0 12 0 0 0 8 236 crc is 1
23 65 136 10 34 0 1 0 0 0 63 6 0 0 0 2 0 12 0 0 0 8 236 crc is 1
21 65 136 11 34 0 1 0 0 0 63 6 0 0 0 0 0 12 0 0 8 235 crc is 1
20 65 136 12 34 0 1 0 0 0 63 6 0 0 0 0 0 13 0 6 234 crc is 1
    
```

Fig. 9. Tx interrupted at channel 23.

channels (11 to 13).

Fig. 10 shows the Rx output in the case of a communication loss with Tx at channel 23 as discussed above. The red square shows the point at which the communication loss has occurred. It is cleared from the beacon packet inside the red border that, due to communication loss, 23 beacons are lost which were sent by Tx, while the handshaking is conducted on the 24th beacon.

A. Performance Evaluation

To evaluate the performance of the existing MIUN testbed [3] with the testbed developed in this work the communication reestablishment time in the case of communication loss will be calculated. It is assumed that the data packets transmitted in each channel are 10,000 the time interval between two consecutive packets is 100ms and the total number of channels are 16. At the time when both Tx and Rx were jumping to channel 11, the communication loss occurs due to the reset button of Rx mote being pressed. When the reset button is pressed, the Rx will jump to the default channel, which is channel 26.

In the MIUN testbed [3], Tx will go through all the

```

nap@NAP: ~/Desktop/opt/tinyos-2.1.0/apps/testis
File Edit View Terminal Help
20 65 136 38 34 0 0 0 1 0 63 6 0 0 0 0 0 22 1 5 236 crc is 1
23 65 136 39 34 0 0 0 1 0 63 6 0 0 0 0 0 23 1 0 0 0 4 236 crc is 1
23 65 136 40 34 0 0 0 1 0 63 6 0 0 0 0 1 0 23 1 0 0 0 4 235 crc is 1
23 65 136 41 34 0 0 0 1 0 63 6 0 0 0 2 0 23 1 0 0 0 4 234 crc is 1
23 65 136 42 34 0 0 0 1 0 63 6 0 0 0 3 0 23 1 0 0 0 4 236 crc is 1
23 65 136 43 34 0 0 0 1 0 63 6 0 0 0 4 0 23 1 0 0 0 4 234 crc is 1
21 65 136 44 34 0 0 0 1 0 63 6 0 0 0 0 23 1 0 4 235 crc is 1
20 65 136 45 34 0 0 0 1 0 63 6 0 0 0 0 23 1 4 235 crc is 1
21 65 136 24 34 0 0 0 1 0 63 6 0 0 0 0 24 0 26 1 0 5 236 crc is 1
23 65 136 25 34 0 0 0 1 0 63 6 0 0 0 0 0 11 1 0 0 0 7 236 crc is 1
23 65 136 26 34 0 0 0 1 0 63 6 0 0 0 0 1 0 11 1 0 0 0 7 233 crc is 1
23 65 136 27 34 0 0 0 1 0 63 6 0 0 0 2 0 11 1 0 0 0 8 235 crc is 1
23 65 136 28 34 0 0 0 1 0 63 6 0 0 0 3 0 11 1 0 0 0 8 235 crc is 1
23 65 136 29 34 0 0 0 1 0 63 6 0 0 0 4 0 11 1 0 0 0 7 236 crc is 1
21 65 136 30 34 0 0 0 1 0 63 6 0 0 0 0 0 11 1 0 8 236 crc is 1
20 65 136 31 34 0 0 0 1 0 63 6 0 0 0 0 0 11 1 7 236 crc is 1
23 65 136 32 34 0 0 0 1 0 63 6 0 0 0 0 0 12 1 0 0 0 7 234 crc is 1
23 65 136 33 34 0 0 0 1 0 63 6 0 0 0 1 0 12 1 0 0 0 7 233 crc is 1
23 65 136 34 34 0 0 0 1 0 63 6 0 0 0 2 0 12 1 0 0 0 6 236 crc is 1
23 65 136 35 34 0 0 0 1 0 63 6 0 0 0 3 0 12 1 0 0 0 7 236 crc is 1
23 65 136 36 34 0 0 0 1 0 63 6 0 0 0 4 0 12 1 0 0 0 7 235 crc is 1
    
```

Fig. 10. Rx output in case of Communication Loss.

TABLE I. PERFORMANCE EVALUATION

No. of Packets	Re-establishment Time(ms)		Packet loss	
	MIUN testbed	Testbed (modified)	MIUN testbed	Testbed (modified)
10	$14 * 10^3$	800	140	5
100	$14 * 10^4$	$5.3 * 10^3$	1400	50
1000	$14 * 10^5$	$5.03 * 10^4$	14000	500
10000	$14 * 10^6$	$5.003 * 10^5$	140000	5000
100000	$14 * 10^7$	$5.0 * 10^6$	1400000	50000

channels and will send packets without knowing whether or not the packet has been received by Rx. Tx will jump to next channels until it reaches at channel 26. Hence, the maximum time for communication reestablishment would be,

$$\text{Time} = (\text{No. of packets sent by Tx in each channel}) * (\text{time interval between two packets}) * (\text{No. of channel Tx will jump})$$

$$\text{Time} = 10,000 \times 100\text{milliseconds} \times 14$$

$$\text{Time} = 1, 40, 00,000 \text{ milliseconds}$$

$$\text{Time} = 14,000 \text{ sec}$$

Now to calculate the reestablishment time taken by modified testbed, reviewing Fig. 3, the data packets sent in each round would be 5,000 plus 3 beacons. When a communication loss occurs, the Tx will know this after sending the 3rd beacons at channel 12 and will jump to the Emergency channel (channel 26). Hence, the maximum time for communication reestablishment will be,

$$\text{Time} = (\text{No. of packets sent by Tx in each channel}) * (\text{Time interval between two packets}) * (\text{No. of channel Tx will jump}) + (\text{No. of beacons sent by Tx in each channel}) * (\text{Time interval between two beacons})$$

$$\text{Time} = (5,000 \times 100\text{milliseconds} \times 1) + (3 \times 100\text{milliseconds})$$

$$\text{Time} = 5, 00,300\text{milliseconds}$$

$$\text{Time} = 500.3 \text{ sec}$$

Table I below shows the performance evaluation of the MIUN testbed [3] and its modified version (developed in this work). The evaluation shows that the communication reestablishment time for the modified MIUN testbed is 28 times less than the existing MIUN testbed [3] i.e. a 28 times faster recovery in the case of communication loss.

V. CONCLUSION AND FUTURE WORK

The primary center in this work, as of now talked about in beginning, was in designing a testbed that makes it possible to study the channel properties during the communication of motes at IEEE 802.15.4 in the 2.4GHz band. A new initialization procedure is implemented at communication start up. The initialization algorithm assists in producing a 28 fold increase in time for the communication re-establishment than is possible when using the existing MIUN testbed, in the case of communication loss between the wireless sensor nodes. To study the link symmetry, P2P communication is achieved between Tx and Rx by introducing a beacon and ACK pair. In the case of communication/hardware failure at any data channel, both Tx and Rx will switch to the emergency channel (IEEE 802.15.4 channel # 26) for handshaking.

The future work could be suggested as to reserve 2 channels (both 25 and 26) instead of only 1 channel (26) for the emergency channel, to provide a backup for the emergency channel. Secondly, P2P communication can be performed using timers instead of packet counter.

REFERENCES

- [1] Prabhu, Boselin, M. Pradeep, and E. Gajendran. "Military Applications of Wireless Sensor Network System." (2017).
- [2] Rao, Raghu Mysore, et al. "Multi-antenna testbeds for research and education in wireless communications." *IEEE Communications Magazine* 42.12 (2004): 72-81.
- [3] Liu, Zuguang. "An empirical study of low power multi-channel communication in WSN Zuguang Liu." (2011).
- [4] Caione, Carlo, Davide Brunelli, and Luca Benini. "Distributed compressive sampling for lifetime optimization in dense wireless sensor networks." *IEEE Transactions on Industrial Informatics* 8.1 (2012): 30-40.
- [5] Zheng, Jianliang, and Myung J. Lee. "A comprehensive performance study of IEEE 802.15. 4." *Sensor network operations* 4 (2006): 218-237.
- [6] Baronti, Paolo, et al. "Wireless sensor networks: A survey on the state of the art and the 802.15. 4 and ZigBee standards." *Computer communications* 30.7 (2007): 1655-1695.
- [7] Del Prete, Massimo, et al. "A 2.4 GHz-868 MHz dual-band wake-up radio for wireless sensor network and IoT." 2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). IEEE, 2015.
- [8] Balodhi, Meenu, Vishwanath Bijalwan, and Banit Negi. "Zigbee & IEEE 802.11 b (WLAN) coexistence in ubiquitous network environment." arXiv preprint arXiv:1407.0462 (2014).
- [9] Mzid, Rania, et al. "Adapting TLS handshake protocol for heterogeneous IP-based WSN using identity based cryptography." *Communication in Wireless Environments and Ubiquitous Systems: New Challenges (ICWUS)*, 2010 International Conference on. IEEE, 2010.
- [10] Watteyne, Thomas, et al. "Mitigating multipath fading through channel hopping in wireless sensor networks." *Communications (ICC)*, 2010 IEEE International Conference on. IEEE, 2010.
- [11] Kodali, Ravi Kishore, and Narasimha Sarma. "Test bed for wireless sensor networks using XMesh networking protocol." *Advances in Computing, Communications and Informatics (ICACCI)*, 2013 International Conference on. IEEE, 2013.
- [12] Steenkamp, Leon du T., Shaun Kaplan, and Richardt H. Wilkinson. "Wireless sensor network gateway." *AFRICON*, 2009. AFRICON'09.. IEEE, 2009.
- [13] Gay, David, et al. "The nesC language: A holistic approach to networked embedded systems." *Acm Sigplan Notices* 49.4 (2014): 41-51.
- [14] LAN/MAN Standards Committee. "Part 15.4: wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANs)." *IEEE Computer Society* (2003).
- [15] IEEE P802. 15.4 e-2012. "Part 15.4: Low-Rate Wireless Personal Area Networks (WPANs), Amendment 1: MAC sub-layer."