Meshtastic Infrastructure-less Networks for Reliable Data Transmission to Augment Internet of Things Applications

N. K. Suryadevara(✉) and Arijit Dutta

School of Computer and Information Sciences, University of Hyderabad, Hyderabad, India
nks@uohyd.ac.in

Abstract. This paper has reported a practical implementation of meshtastic networks data transmission via web interface for an IoT theme of the application. The integrated network architecture and the interconnecting mechanisms for reliable, low-cost data transmission of the infrastructure-less network are depicted. Features of meshtastic communication such as addressing and tunneling that can swiftly assimilate with co-systems of IoT were presented. The system setup, transmission, and data gathered in the real-world campus environment are intricacies. The assessment of quality parameters related to data transmission using Heltec Lora devices is described. Results are encouraging as the prototype was tested to generate real-time information exchange using mesh networks rather than a testbed scenario.

Keywords: Wireless mesh networks · Infrastructure-less networks · LoRa · Internet of Things (IoT)

1 Introduction

The technological advancements in media communications is pretty much connected with the Internet of Things (IoT) [1]. However, Telecommunications and the Internet events gave humanity limitless possibilities of information and data trade, which completely revolutionized the world. In the last decade, the growth of the data transfer between people using IoT ecosystems went huge, so did the data transmission technologies [2]. Energy utilization is a significant consideration in the event of communication between sensors for specific applications [3]. Machine to Machine (M2M) communication must ensure that capturing the sense events, nearby data processing of detected events, and packet transmission to the end nodes are efficient [3].

Public Safety and Disaster Recovery (PSDR) communications are entirely dependent on Internet communication infrastructures such as traditional landline, cellular telephony, and infrastructure-based Land Mobile Radio (LMR).

Q. Guo et al. (Eds.): WiSATS 2021, LNICST 410, pp. 622–640, 2022.
https://doi.org/10.1007/978-3-030-93398-2_55

Whereas, in environmental monitoring, the option to transmit data does not depend on existing Internet infrastructures for the applications such as biological and ecological studies, animal tracking, air quality measurements, agriculture, and natural hazards or disasters (e.g., tsunamis, earthquakes, nuclear meltdowns). Recent events have revealed significant shortcomings in PSDR communications [4]. Several aspects need to be considered in these situations when a technical solution is to be considered for effective data transmission through Infrastructure-less communications, such as lack of power supply, harsh weather, low maintenance possibilities, weight restrictions for animal-attached sensors, availability of scenario-specific sensors, and cost factors [5]. The communication technologies that evolved as trendsetters and benchmark in terms of energy efficiency and scalability without Internet infrastructures, such as ZigBee, 6LowPAN, BLE, LoRa, LoRaWAN, Thread [6,7]. The Long Range (LoRa) technology came out as a better alternative for Infrastructure-less low power long-range communication than WiFi and other Low Power Wide Area Network (LPWAN) [8].

When it comes to infrastructure-less communication, off-grid decentralized communication systems are more reliable on a global scale. Among the available Infrastructure-less networks, wireless mesh networks can give coverage to a wide geographical area without depending on a dedicated access point (AP) [9]. Presently, "Disaster.radio" [10] and "Meshtastic" [11] are open source projects that work on disaster-resilient communication networks. Meshtastic allows one to utilize economical GPS radios as extensible, long battery life, secure, network GPS communicator. These radios are extraordinary for those geographical areas where no communication tower is required to install and do not require web access. Every individual in the mesh network can generally see the area and distance of any remaining individuals, and at any instant, they can communicate through text messaging. Meshtastic radios use LoRa technology in their Physical layer component. These radios auto-create a mesh to forward data through packets and follow the publish and subscribe model [11]. The open-source community developed the corresponding firmware and python APIs to be used by the application developers. The message passing between the Meshtastic device and its corresponding application takes place through Google protobuf [12].

The primary objective of this work is to understand how LoRa technology is functioning in the Meshtastic devices for the exchange of little telemetric data that can be considered for IoT data transmission when there is an Internet/telecommunication connection failure. So that, Meshtastic device communication systems can be considered for planning and setup of frameworks that can be utilized in the arrangement of visual control systems in crisis circumstances.

Contribution: This paper aims to describe data communication architecture using meshtastic devices to transmit data in indoor and outdoor environmental scenarios. Furthermore, the network data transmission investigations and their corresponding quality parameters based on the experiments performed using the Meshtastic devices were illustrated. The experimental study analyzes the transfer of packets and received signal level indicator on different channel settings with different radio configurations of 433 MHz Meshtastic devices at various Spreading Factor (SF), Bandwidth (BW), Coding Rate (CR), Signal to Noise ratio (SNR). Accordingly, the Quality of Service (QoS) parameters such as reliability, throughput, and jitters for the Meshtastic data transmission are evaluated and presented.

The rest of the paper is organized as follows; Sect. 2 describes various Wireless Mesh Networks (WMN) and their related communication technologies. Section 3 presents the details for the setup of Meshtastic network architecture and Python APIs considered for the investigations. Furthermore, Sect. 4 describes the intricacies in the implementation, and Sect. 5 presents the results obtained for various experimental settings and their analysis. Section 6 concludes with a discussion of future work.

2 Related Works

Mudathir et al. has presented various wireless communication technologies for Infrastructure-less data communication. Figure 1 depicts the following arrangement followed by corresponding description.

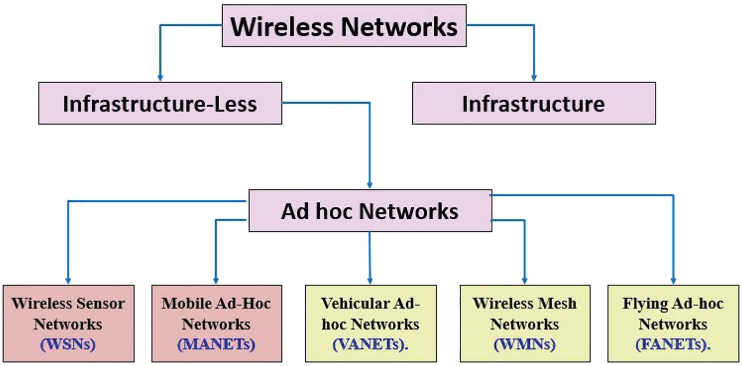

Fig. 1. Classification of Wireless Infrastructure-less Networks

An ad-hoc network comprises remote hubs that can dynamically self-arrange into a flexible and transient topology without essentially utilizing any prior configuration settings. In ad-hoc networks, every hub may convey straightforwardly to one another. Hubs that are not directly associated convey by sending the traffic through halfway hubs through multi-hop communication.

Wireless Sensor Networks (WSNs): The WSNs are interconnected sensor hubs that communicate remotely to gather information about the adjacent environment. Hubs are, for the most part, low power and dispersed in an ad-hoc, and decentralized design. It is a significant innovation empowering the majority of IoT theme of use cases.

Mobile Ad-Hoc Networks (MANET): The MANET organization is a self-governing momentary relationship of wireless devices (hubs) that communicate over remote connections. Hubs that exist within the same range can convey straightforwardly and are responsible for discovering one another. Figure 2 is an illustration of the MANET topology of various devices [13].

Fig. 2. Topology of MANET

Vehicular Ad-Hoc Networks (VANET): This type of wireless organization uses vehicles as versatile hubs, a subclass of MANETs to give interchanges among close-by vehicles and side of the road hardware. The hubs (vehicles) in VANETs are restricted to street geography while moving, so if the street data is accessible, we can anticipate the future situation. This is how street safety and traffic management turn out to be easy.

Wireless Mesh Network (WMN): A WMN comprises two or more wireless radios working together to share routing protocols to create an interconnected Radio Frequency (RF) pathway. No matter how many radios it includes, a wireless mesh network creates only a single name identifier or Single Set Identifier (SSID). Therefore, it is the most reliable network as every node in its range are connected with all the other nodes. Figure 3 depicts a typical WMN.

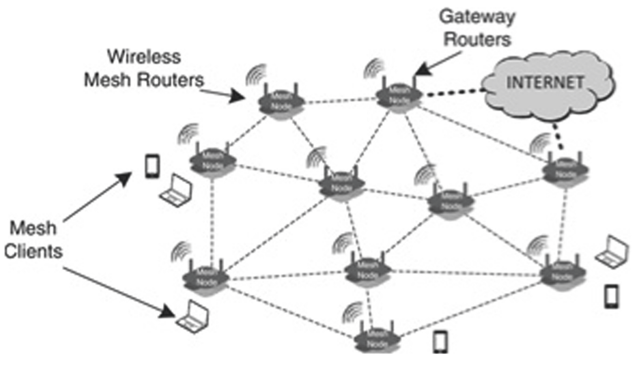

Fig. 3. Wireless Mesh Network (WMN)

Flying Ad-Hoc Networks (FANET): It is majorly used in Unmanned Aerial Vehicle (UAV) network where its importance is in several militaries, commercial and civilian applications like disaster management, crisis management, hostile environment, destroy and search operations, border surveillance, wildfire management, relaying networks, estimation of wind, civil security, agricultural remote sensing, and traffic monitoring [14].

The main advantages of ad hoc networks are flexibility, low cost, and robustness. On the other hand, a wireless mesh network design is exciting and at the same time it is challenging due to noisy channels, limited reach, and insecure wireless transmissions added to mobility and energy constraint.

The **A**LOHAnet protocol used in the computer networking system provides the wireless packet data network used for collision detection in the Medium Access Control (MAC) sublayer for wireless communication technologies [15]. A standard MAC protocol must deal with channel constraints, attenuation, and noise.

In contrast, for efficient wireless transmission in IoT theme of applications, the MAC sublayer should consider requirements, such as quality of service (QoS), low energy consumption, fairness, and scalability, as these critical parameters significantly impact the IoT system performance. Therefore, the popular protocols used in wireless mesh networks consider the QoS, low energy consumption, and scalability parameters for efficient data transmission. The most popular wireless communication technologies used in indoor and outdoor environmental monitoring conditions under the themes of IoT are Zigbee and LoRa.

ZigBee protocol uses the IEEE 802.15.4 as a base standard. One of the most used communication protocols for indoor IoT applications where low energy consumption, low data rate, low cost, and high message throughput performances are required. It can also provide high reliability, security with both encryption and authentication. It can connect up to 65000 nodes to form a WMN. Majorly, it works with the direct-sequence spread spectrum (DSSS) method, capable of transferring data at 250 kbps at 2.4 GHz. The range is around 10 to 100 m, but setting up the network is complex and costlier.

LoRaA bidirectional communication protocol uses the LoRa physical layer to provide low-power long-range communications. To achieve this, LoRa uses Chirp Spread Spectrum (CSS) modulation. With a single base station, it is possible to cover up to hundreds of square kilometers. To guarantee that all communications are completed, the LoRa MAC layer takes responsibility for joining and accepting the end-point and the gateway, scheduling the receiving slots for end-points, and confirming the reception of the received packets. LoRa uses non-licensed 433 MHz to 923 MHz industrial, scientific, and medical (ISM) radio bands according to different world countries and its telecommunication guidelines [16]. The data communication range is up to 2 to 5 km between end nodes and, for Long Range Wide Area Network (LoRaWAN) it goes up to 10 to 15 km. It supports up to 1000 nodes. Table 1 provides the primary differences between the two low-power wireless communication technologies.

Table 1. Comparison between Zigbee and LoRa

Feature	ZigBee	LoRa
Based Data Rate [Mbps]	0.25	0.11
Frequency [GHz]	2.4	433
Range [m]	10 to 100	2000 to 5000
Nodes/Masters Power	65540	1000
Complexity	Medium	Low
Security	AES 128 bit	AES 128 bit

LoRa is better suited for Low Power long-range communication. The open-source community drives the applications in low-poll rate sensing. The technology and accessible gateways for nodes to connect to community-driven networks (such as The Things Network) are suitable for developers and communities.

Considering the advantages mentioned above, the "Meshtastic project" has developed an extremely versatile mesh communicator ecosystem for various applications such as paragliding, hiking, skiing in the field of aviation, and GPS communication using available ISM sub-gigahertz frequencies of LoRa. Furthermore, its applications can be extended to disaster management and agriculture, where environmental conditions of large geographical regions are better monitored.

Islam, B. et al. [17] tested and came out with the examination of the practicality of utilizing LoRa, in indoor (house) climate. Assessments related to the dependability, coverage, strength of signals, responsiveness, and power consumption are presented. It was inferred that LoRa is more competent to use in the indoor industrial warehouse and home monitoring systems using sensor networking.

I. Bobkov, A. et al. [18] considered both 433 MHz and 868 MHz ISM band frequencies and calculated that SNR (signal to noise ratio), RSSI (received signal strength indicator), and PDR (packet delivery ratio). Their work concludes that 433 MHz frequency gives a more stable LoRa signal due to more noteworthy SNR and RSSI values at each spreading factor.

Apparently, no investigation has contemplated and talked about the presentation of LoRa in meshtastic teletype data transmission. In particular, its usage in IoT application domains, in the event of no Internet/telecommunication, the performance of teletype data transmissions over the meshtastic networks.

This paper presents the effective inter-networking mechanisms for transmitting teletype (text) data over infrastructure-less (Meshtastic) networks to support the data transmissions for IoT applications that do not require Internet communication. Reliability and delay in data packet transmission over the Meshtastic networks are presented in the results section.

3 System Description

One of the objectives of this work is to test 'hop' dynamics with a system of three Meshtastic nodes. Initially, a 'base station sender' and a 'target receiver' on opposite sides of a building are positioned to block direct LoRa transmission; and then a 'relay' node on the 'corner' of the building for messages to 'hop' around it is placed. The data transmission is initiated by the 'sender' that was located in room 1 of the building. The device is managed and controlled by the software Meshtastic-python script that would 'reply' to receive messages with the SNR of the incoming message. The idea was that from the field, it could send messages from the 'end node' (controlled via the mobile app), and if the 'relay' node was able to relay the message via a 'hop' to the 'base station sender', then eventually gets a reply relayed back to the sender at the end node.

Figure 4 shows the layout depicting key elements of the integrated system. It consists of i) wireless devices[ ] (currently support devices that use the ESP32 and the nRF52 microcontrollers), ii) Laptop running the software of the Meshtatic device, and iii) android mobile phone as hardware, and iv) meshtastic GUI desktop application v) meshtastic android mobile application as software vi) Any power bank over 500 mAh battery.

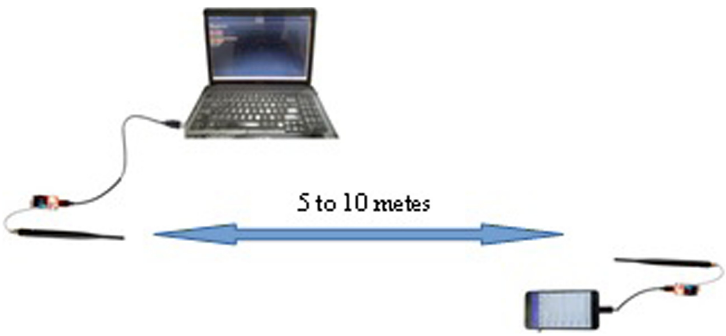

Fig. 4. Peer to Peer meshtastic system setup

There were two different settings to measure the delay, throughput, packet loss, and jitter in the Meshtastic network. For the zero hop (peer-to-peer connection) as depicted in Fig. 4, one ESP32 micro-controller is connected with Laptop where the developed meshtasticGUI software program is running. The other wireless device is at the 5 to 10-m line of sight distance. For a 'single hop' connection (one hop between the two end nodes) as depicted in Fig. 5, the wireless end nodes are kept at two different rooms partitioned by a wall, and another device is placed at a third room on the same floor which has a line of sight communication with the other two devices. The remote devices (that are not connected with the workstation) are attached to the power bank for the power supply. Distance between the hop and the end nodes are 5 to 10 m each. In order to use the Meshtastic network, the ESP32 devices are embedded with meshtastic firmware (Version 1.1.32). The 'Meshtastic' android mobile app (version 1.1.32) is also installed on the mobile.

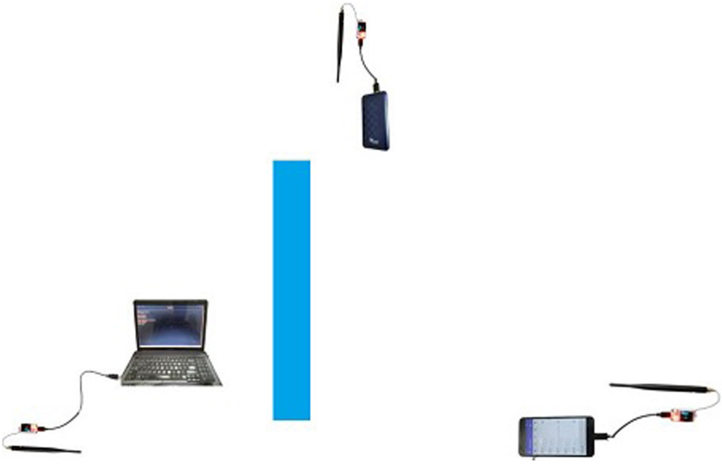

Fig. 5. Single hop meshtastic system setup

Before running the program, the following command as shown in Fig. 6 is used to sync both the devices in the same timestamp:

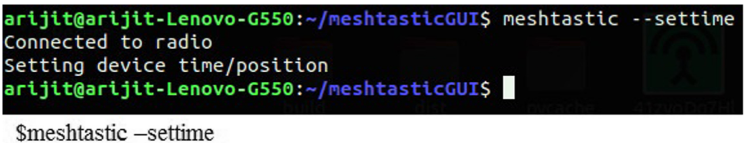

Fig. 6. Setting RTC time to zero

The command sets the RTC inside the device to default zero and also synchronize the other device's RTC with the same clock. The meshtastic python API and other GUI libraries is used to build meshtasticGUI desktop application.

After collecting the data of the transactions, the records of the success and failure of packets are merged together with the file where the sent packet record is stored. By evaluating the difference between the message send time and acknowledgment receive time, the entire round trip time of the packet transmission was calculated.

The communication process is governed by the following steps:

MeshtasticGUI Algorithm::

```
Input: Text File with the collected data
Output: Three CSV file and one JSON file
1. Opening the ports and establishing the connection between the device and the workstation.
2. send some dummy messages by typing on the Textbox of the GUI.
3. Wait till the responses from the remote devices.
4. If responses are coming as success or failure then
4.1 while end of file is true do
4.1.1 Read 223 characters from the file and send through interface.sendText(),
      record it in <filename>.csv file with timestamp;
4.1.2 Receive the incoming acknowledgment packet;
4.1.3 if acknowledgment is success then
4.1.3.1 Record it in the <filename>_success.csv with timestamp;
else if acknowledgment is failure then
4.1.3.2 Record it in the <filename>_failure.csv with timestamp;
end of If
4.1.4 wait till predefined time interval completes;
4.1.5 Record the transaction in the <filename>.JSON file as JSON array;
4.1.6 Close the connection;
end of while
5. else
5.1 Terminate the connection and rerun the program
end of If
```

A. Wireless Devices: LoRa base station consists of Heltec LoRa module connected to a Laptop with USB Type C cable. Each LoRa node for signal transmission consists of a LoRa module based on ESP32 microcontroller and SX1276 transceiver. These modules are provided by Heltec: Wi-Fi LoRa 32 module for 433 MHz and Wi-Fi LoRa 32 (v2) module for 868 MHz experiments respectively. Figure 7 depicts the corresponding Meshtastic device.

During each experiment, LoRa transmitter continuously sends and receives packets containing their identification number, text message, device IDs and acknowledgment (from the receiving node), and the base station. For this experiment, 125, 250 and 500 KHz bandwidth channels with spreading factors (SF) range between 7 to 12 conventional units and a coding rate between 5 to 8. The channels are altered using Command Line Interface. Each time before running the program, in order to gather information regarding a newly embedded device status the following command is used (device should be connected with the terminal). The Fig. 8 depicts the command used to display the Meshtastic information.

The following command is used to change the channel radio settings.

```
$meshtastic --port <port number> --setchan spread_factor <value>
  --setchan coding_rate <value> --setchan bandwidth <value>
```

Fig. 7. Heltec Wi-Fi LoRa 32 (v2) module

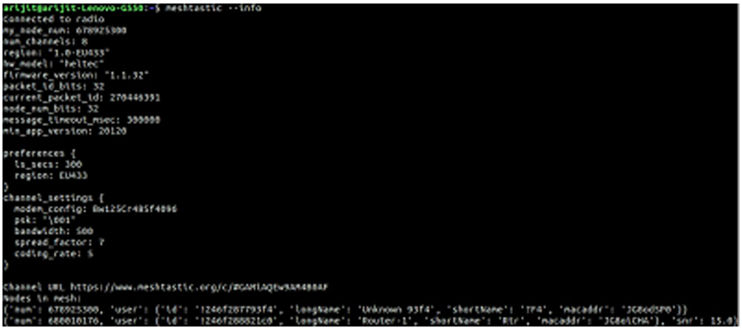

Fig. 8. $meshtastic-info and node information of devices which are using the same radio setting

There are 32 different radio settings applied on Meshtastic to observe each channel's behavior and performance. Table 2 shows the maximum throughput in LoRa for different spreading factors and bandwidth.

Table 2. Max throughput (in bps)

SF/BW	500 KHz	250 KHz	125 KHz
6	37500 bps	18750 bps	9375 bps
7	21875 bps	10937.5 bps	5468.75 bps
8	12500 bps	6250 bps	3125 bps
10	3906.25 bps	1953.125 bps	976.5625 bps
12	1171.875 bps	585.9375 bps	292.9687 bps

B. Adhoc Meshtastic Network: The Adhoc Meshtastic Network can be formed by adding any number of devices on a particular channel. Different channel settings are used in different environments depending upon the distance, speed, power consumption, and reliability. Table 3 explains the most popular channels.

Table 3. Address transformation from device to device Packet.

Channel setting	Alt channel	DR	SF/Sym	CR	BW	Link budget
Short range (but fast)	Short Fast	21.875 kbps	7/128	4/5	500	134 dB
Medium range (but fast)	Medium	5.469 kbps	7/128	4/5	125	140 dB
Long range (but slower)	Long Alt	0.275 kbps	9/512	4/8	31	153 dB
Very long range (but slow)	Long Slow	0.183 kbps	12/4096	4/8	125	154 dB

C. meshtasticGUI Interface: Meshtastic project provides a python API for the developers to built application programs for real-life usage. It can be downloaded and installed as python library and used in the application program. The meshtasticGUI interface provides a meshtastic chat environment which is also used for sending an entire file data. It is made for the computers having Linux operating system. Tkinter library is used to build the GUI interface, where several buttons are provided for some purposes. Clicking on those do the following:

SEND: To send the typed messages.
STOP: To close the connection.
READ: To start reading from a file and send the message.
EXIT: To abort the connection and stop the program.

It uses publish and subscribe model at the application layer and the serial communication between the workstation and device take place at 921600 bps through the USB cable. The field values of Tx power, radio settings, destination address, response, rx_time is set either through API or Meshtastic admin commands through terminal. The transmission packet records are stored CSV and JSON format (Fig. 9).

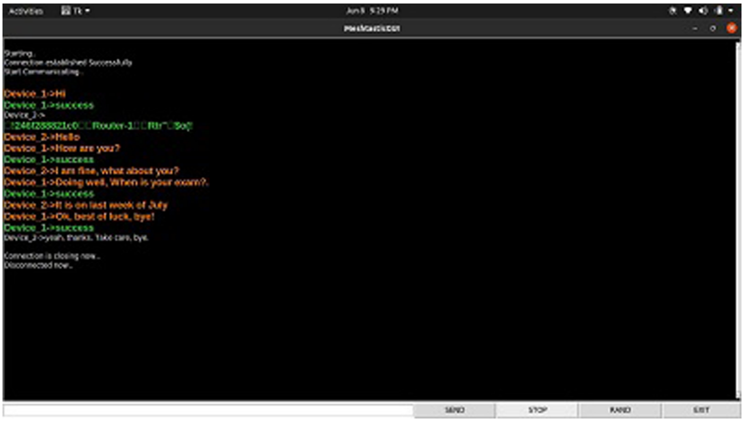

Fig. 9. Developed Meshtastic GUI interface for command execution

4 Implementation Details

This section covers the practical implementation details related to the configuration settings of the network setup.

A. Address Transformation. In Meshtastic network on a particular channel a message can either be unicast or broadcast. The addresses which are 48 byte physical (MAC) address of the data-link layer are used for the recipient identification. For, broadcast the broadcast number 0/xffffffff is used, otherwise the device name can be provided at the time of sending the message. The devices work across various vendors and implementations, using Protocol Buffers pervasively with respect to it's API clients. Before the starting of every packet delivery the channel settings are altered to different spreading factor, bandwidth and coding rate. Initially by using the "settime" option of the command line, the two devices are synchronized to a same timestamp, which is useful for the experiment. Figure 8 describes the nodes information which are on the same channel.

B. Packet Translation. MeshPacket (Meshtastic packet) in peer to peer messaging comes with a 32 bit LoRa preamble. A minimum (8 bit) preamble is used to maximize the amount of time the LORA receivers can stay asleep, which dramatically lowers power consumption.

After the preamble the 16 byte packet header is transmitted. This header is described directly by the PacketHeader class in the C++ source code. But indirectly it matches the first portion of the "MeshPacket" protobuf definition. But notably: this portion of the packet is sent directly as the following 16 bytes (rather than using the protobuf encoding). We do this to both save airtime and to allow receiving radio hardware the option of filtering packets before even waking the main CPU.

- **to (4 bytes):** the unique NodeId of the destination (or 0xffffffff for NodeNum_BROADCAST)
- **from (4 bytes):** the unique NodeId of the sender
- **id (4 bytes):** the unique (with respect to the sending node only) packet ID number for this packet. We use a large (32 bit) packet ID to ensure there is enough unique state to protect any encrypted payload from attack.
- **flags (4 bytes):** Only a few bits are currently used - 3 bits for the "HopLimit" (see below) and 1 bit for "WantAck" After the packet header the actual packet is placed onto the wire. The maximum total size of the LoRa packet is 256 bytes.

The Fig. 10 represents the Meshpacket structure

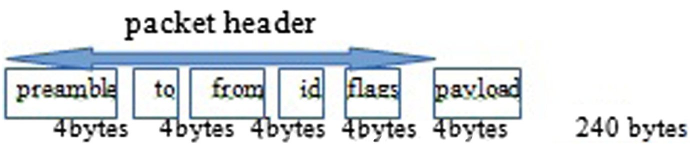

Fig. 10. Meshtastic LoRa packet structure

C. Storage of Data. The fields of the every packet can be accessed through Python program as it is in dictionary structure. While transmitting the data, the information regarding the transmitted packet and it's acknowledgment is received at the base workstation. They are stored as a JSON array which is eventually stored in a JSON type file with the field values of Text message, to, from, message ID, transmission timestamp, Acknowledgment (success or failure) and it's receiving timestamp. The following two figures are examples of a successful and failure message transmission. Then it's extracted into CSV file to do the further analysis (Figs. 11 and 12).

Fig. 11. Meshtastic packet success in JSON format.

Fig. 12. Meshtastic packet failure in JSON format.

5 Experimental Results

Developed system is tested in the school of Computer and Information Science building, University of Hyderabad. LoRa base station connected with the transmitter node was permanently placed on the computer lab in ground floor, while LoRa receiver was being on 10 m distance. The transmitter and receiver was placed on the same room. The experiments were conducted with total of 30 different radio settings and Table 4 shall display the results of the analysis of the data which is captured while transmission of a text file size of 58.4 kB with variable packet size between 33 bytes to 256 bytes. The result contains the total propagation delay, throughput and loss of packet during the transmission.

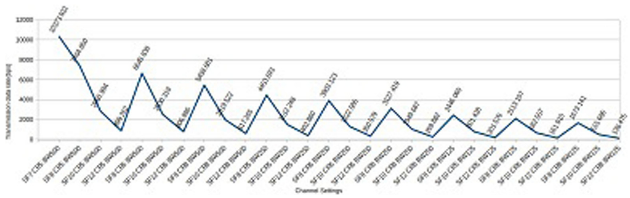

Fig. 13. Real-time graphical representation of transmission delay on different channels

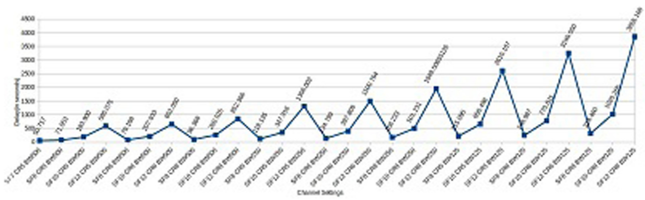

Fig. 14. Real-time graphical representation of throughput on different channels

Figure 13 and Fig. 14 represents the graphical representation of the Table 4 where the X-axis is the different radio settings and Y-axis are delay and throughput respectively. The results are also compared with a different file size to compare and contrast the reliability of the channel.

Table 5 shows the delay and throughput contrast of two different size of files in 4 different channels followed by the graphical representation in Fig. 15, 16, 17 and Fig. 18.

Table 4. Delay, throughput and packet loss of text data transmission in Meshtastic.

SNo	SF	CR	BW	Delay (sec)	Throughput (bps)	Packet loss
1	7	5	500	50.72	10373.92	4
2	8	5	500	71.05	7404.85	1
3	10	5	500	183.9	2860.99	8
4	12	5	500	585.08	899.26	6
5	8	6	500	79.17	6645.84	0
6	10	6	500	207.93	2530.32	1
7	12	6	500	652.05	806.89	3
8	8	8	500	96.37	5459.68	1
9	10	8	500	260.52	2019.52	0
10	12	8	500	852.37	617.27	28
11	8	5	250	118.13	4453.69	3
12	10	5	250	347.92	1512.25	4
13	12	5	250	1306	402.86	3
14	8	6	250	134.8	3903.12	14
15	10	6	250	397.81	1322.59	1
16	12	6	250	1500.76	350.58	22
17	8	8	250	168.23	3127.42	4
18	10	8	250	501.23	1049.69	23
19	12	8	250	1949.51	269.88	1
20	8	5	125	215.1	2446.06	2
21	10	5	125	656.5	801.43	6
22	12	5	125	2610.11	201.58	5
23	8	6	125	248.99	2113.11	23
24	10	6	125	770.83	682.56	0
25	12	6	125	3249.55	161.91	10
26	8	8	125	314.46	1673.14	0
27	10	8	125	1020.27	515.69	3
28	12	8	125	3855.17	136.48	8
29	7	5	125	135.76	4386.65	6
30	9	8	31	2999.75	198.52	2

Table 5. Comparison chart of the data transmission for 58.4 kB and 105.4 kB

SNo	SF	CR	BW	Delay (sec) 58.4 kB	Delay (sec) 105.4 kB	Throughput (bps) 58.4 kB	Throughput (bps) 105.4 kB	Packet loss 58.4 kB	Packet loss 105.4 kB
1	7	5	500	50.72	89.56	10373.92	10762.8	4	4
2	8	5	500	71.05	233.37	7404.85	4130.65	1	0
3	10	5	500	183.9	4783.4	2860.99	201.52	8	9
4	12	5	500	585.08	6917.07	899.26	139.36	6	3

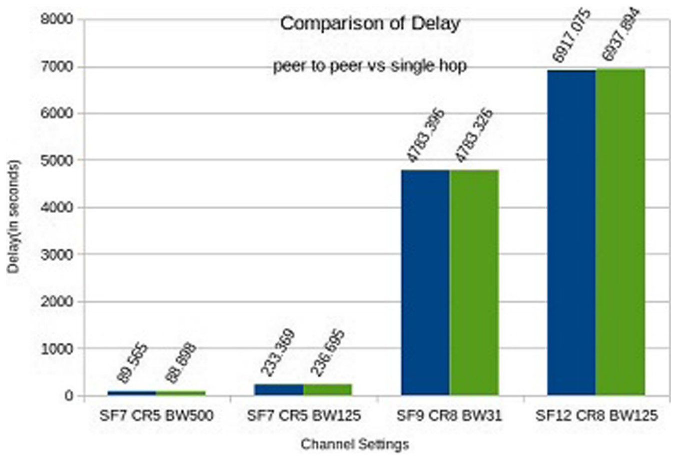

Fig. 15. Comparison graph of the delays of 105.4 kB file (zero hop vs single hop)

Throughput: Throughput is a measure of how many units of information a system can process in a given amount of time.

Jitter: Jitter is a delay variation that puts stress on the receiving endpoint, as it is trying to figure out the right sequence of data packets. There is no network free of Jitter. It affects latency-sensitive applications and hurts the user experience. Jitter introduces inconsistencies which influence the quality of communication and data transfer speeds. The following Figs. 19, 20, 21 and 22 are comparisons of Jitter in the Meshtastic network between zero hop and single hop network.

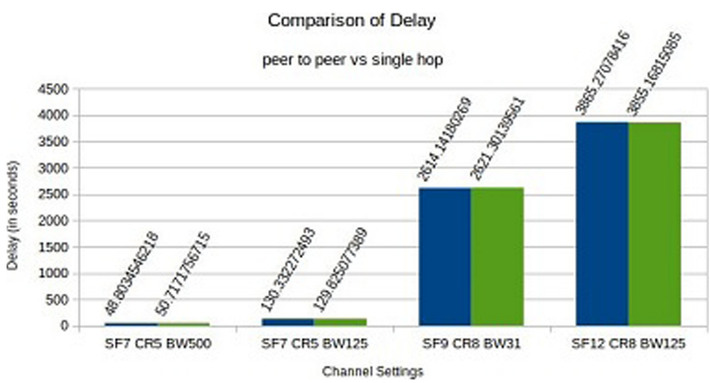

Fig. 16. Comparison graph of the delays of 58.4 kB file (zero hop vs single hop)

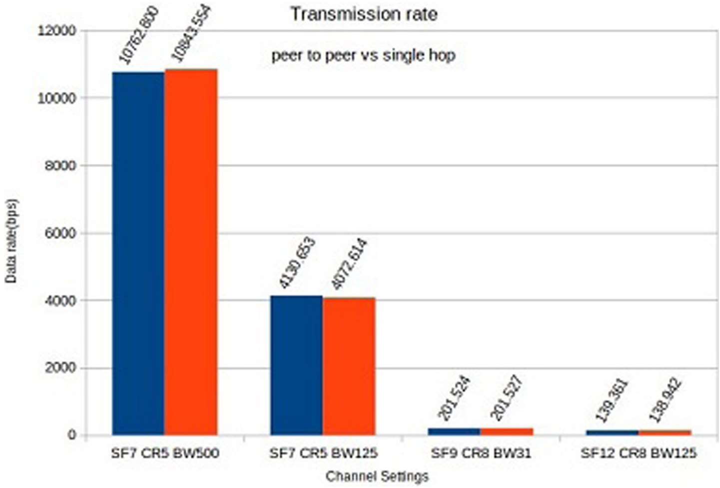

Fig. 17. Comparison graph of the throughput of 58.4 kB file (zero hop vs single hop).

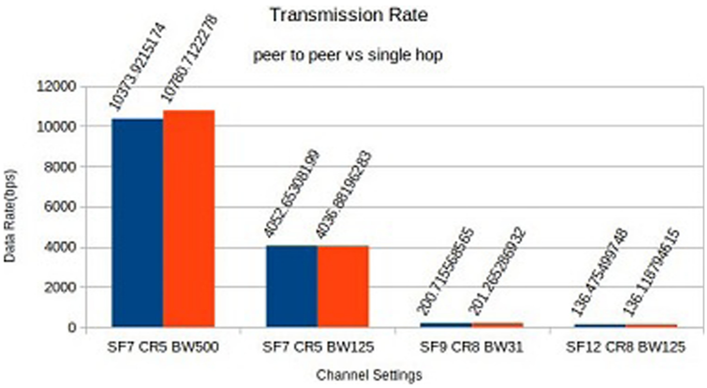

Fig. 18. Comparison graph of the throughput of 108.4 kB file (zero hop vs single hop).

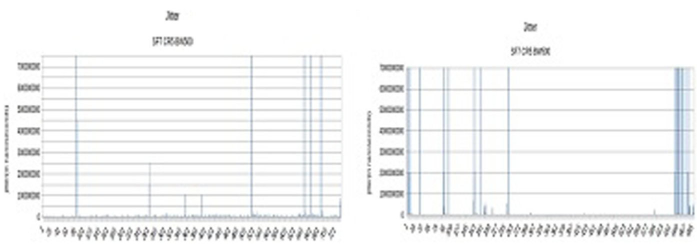

Fig. 19. Comparison of Jitter in Short range (but fast) channel

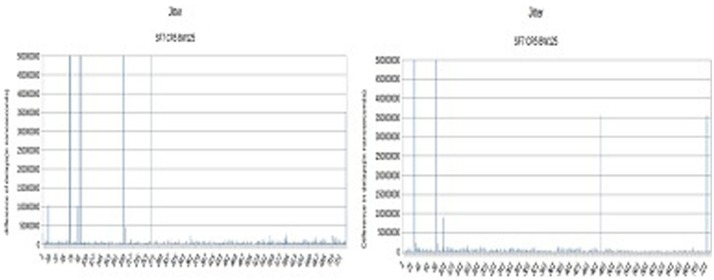

Fig. 20. Comparison of Jitter in Medium range (but fast) channel

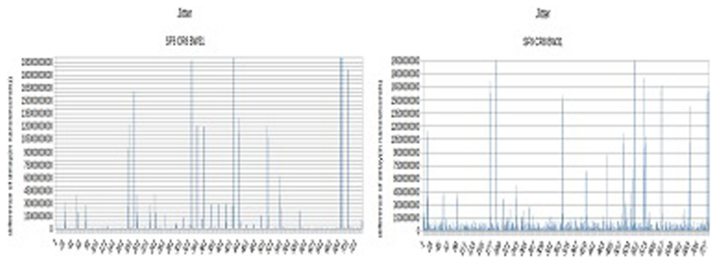

Fig. 21. Comparison of Jitter in Long range (but slow) channel

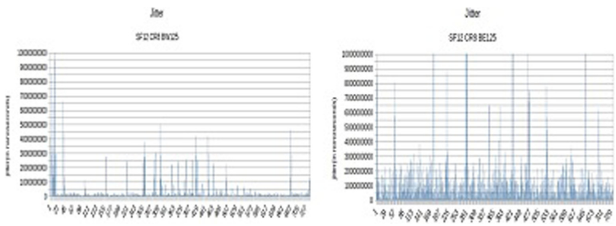

Fig. 22. Comparison of Jitter in Very Long range (but slow) channel

6 Discussion and Future Work

This article provides the experimental study results of teletype data transmission on the basis of LoRa technology using Meshtastic devices. The developed technology can be used in the situations where there is no telecommunication/Internet system for data transfer across long geographical regions. Meshtastic is an ongoing open source community project. There are future improvements that can be implemented on sending and receiving multimedia data like image, audio and video. Multi-hop unicast is a controlled broadcasting through several nodes in a channel. It will be very interesting to explore how the data transmission can be done.

Acknowledgements. The authors would like to thank the IoE PDF grant for the financial support provided in registration of the conference.

References

1. ITU-T Recommendation Y. 2060. Overview of Internet of Things, Geneva, February 2012. https://www.itu.int/en/ITU-T/gsi/iot/Pages/default.aspx
2. Kirichek, R., Kulik, V.: Long-range data transmission on flying ubiquitous sensor networks (FUSN) by using LPWAN protocols. In: Vishnevskiy, V.M., Samouylov, K.E., Kozyrev, D.V. (eds.) DCCN 2016. CCIS, vol. 678, pp. 442–453. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-51917-3_39
3. Bouguera, T., Diouris, J.-F., Chaillout, J.-J., Jaouadi, R., Andrieux, G.: Energy consumption model for sensor nodes based on LoRa and LoRaWAN. Sensors **18**(7), 2104 (2018). https://doi.org/10.3390/s18072104
4. Lopresti, D., Shekhar, S.: A National Research Agenda for Intelligent Infrastructure (2021). https://cra.org/ccc/resources/ccc-led-whitepapers/. Accessed 21 Mar 2021
5. Portmann, M., Pirzada, A.A.: Wireless Mesh networks for public safety and crisis management applications. IEEE Internet Comput. **12**(1), 18–25 (2008). https://doi.org/10.1109/MIC.2008.25
6. https://www.silabs.com/whitepapers/selecting-the-appropriate-wireless-mesh-network-technology
7. Cilfone, A., Davoli, L., Belli, L., Ferrari, G.: Wireless Mesh networking: an IoT-oriented perspective survey on relevant technologies. Future Internet **11**, 99 (2019). https://doi.org/10.3390/fi11040099
8. Mekki, K., Bajic, E., Chaxel, F., Meyer, F.: A comparative study of LPWAN technologies for large-scale IoT deployment. ICT Exp. **5**(1), 1–7 (2019). https://doi.org/10.1016/j.icte.2017.12.005. ISSN 2405-9595
9. Jiang, S.: Introduction and overview. In: Wireless Networking Principles: From Terrestrial to Underwater Acoustic, pp. 1–31. Springer, Singapore (2018). https://doi.org/10.1007/978-981-10-7775-3_1
10. https://communitynetworks.group/t/long-range-low-bandwidth-mesh-projects-for-good/306
11. https://meshtastic.org/docs/about. Accessed 18 Aug 2020
12. https://github.com/meshtastic/Meshtastic-protobufs
13. Tripathy, B.K., Jena, S.K., Reddy, V., Das, S., Panda, S.K.: A novel communication framework between MANET and WSN in IoT based smart environment. Int. J. Inf. Technol. **13**(3), 921–931 (2020). https://doi.org/10.1007/s41870-020-00520-x
14. Khan, M.A., Safi, A., Qureshi, I.M., Khan, I.U.: Flying ad-hoc networks (FANETs): a review of communication architectures, and routing protocols. In: First International Conference on Latest trends in Electrical Engineering and Computing Technologies (INTELLECT) 2017, pp. 1–9 (2017)
15. Abramson, N.: The ALOHAnet - surfing for wireless data. IEEE Commun. Mag. **47**(12) (2009)
16. https://www.thethingsnetwork.org/docs/lorawan/frequencies-by-country/ . Accessed 20 Dec 2020
17. Islam, B., Islam, M.T., Kaur, J., Nirjon, S.: LoRaIn: making a case for LoRa in indoor localization. In: IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops) 2019, pp. 423–426 (2019). https://doi.org/10.3390/ijgi7110440
18. Bobkov, I., Rolich, A., Denisova, M., Voskov, L.: Study of LoRa performance at 433 MHz and 868 MHz bands inside a multistory building. In: Moscow Workshop on Electronic and Networking Technologies (MWENT) 2020, pp. 1–6 (2020). https://doi.org/10.1109/MWENT47943.2020.9067427