Exploring Collision Avoidance during Communication Over Sound for Healthy Environment

Praveen Kumar Sharma
National Institute of Technology
Durgapur, West Bengal, India
praveencs54@gmail.com

Suraj Gupta
National Institute of Technology
Durgapur, West Bengal, India
rocksuraj0912@gmail.com

Argha Sen
National Institute of Technology
Durgapur, West Bengal, India
arghasen10@gmail.com

Tanmay De
National Institute of Technology
Durgapur, West Bengal, India
tanmayd12@gmail.com

Sujoy Saha
National Institute of Technology
Durgapur, West Bengal, India
sujoy.ju@gmail.com

ABSTRACT

The rapid rise in Internet of Things (IoT) devices increases communication overhead. The communication is the most crucial part in IoT. However, most of the the traditional methods of communication emit harmful electromagnetic radiations which have severe impact on human health. To reduce the health hazards on the society due to such radiations, we have proposed a harmless mode of communication using the near-ultrasonic audible acoustic signal. We can use the sound signal as communication medium for sensitive systems like healthcare system, smart classroom system, and so on, where the occupants are vulnerable to harmful radiations. In this work, we have used Chirp Software Development Kit (SDK) as the tool for connecting the IoT devices through acoustic signal, which has the capability of multi path transmission where a device is allocated randomly a channel from a pool channels. However, with the rise in the number of devices in the system, the chances that a channel being allocated to many devices increase. It increases the chances of collision during communication, unless necessary precaution is taken in the channel allocation. By resolving the above issues when multiple node used same channel, we have achieved 80% more network throughput than the existing technique. In addition, we have proposed an approach to increase the communication range with acoustic communication, which is another critical issue in the present context.

CCS CONCEPTS

• **Networks → Network protocol design**.

KEYWORDS

Chirp, Communication, Sound Signal, Collision, Bandwidth

ICDCN 2020, January 4–7, 2020, Kolkata, India

ACM ISBN 978-1-4503-7751-5/20/01…$15.00
https://doi.org/10.1145/3369740.3372765

ACM Reference Format:
Praveen Kumar Sharma, Suraj Gupta, Argha Sen, Tanmay De, and Sujoy Saha. 2020. Exploring Collision Avoidance during Communication Over Sound for Healthy Environment. In *21st International Conference on Distributed Computing and Networking (ICDCN 2020), January 4–7, 2020, Kolkata, India.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3369740.3372765

1 INTRODUCTION

The rise of IoT devices in the home and workplaces have created a world where the data and connectivity are becoming increasingly complex. As IoT technology advances and the demand for efficient ways of communicating data between these devices grows, the world has observed a rise in emerging new data transmission technology. One solution to meet these new demands is "data-over-sound," which harness devices existing or external speakers and microphones to send and receive data over an acoustic channel. Data-over-sound presents a compelling solution for many device-to-device connectivity applications, particularly for use cases that require frictionless, low-cost connectivity with nearby devices. This paper aims to achieve the concept of Data-over-sound and improves the application areas of the Internet of Things, including provisioning smart devices, by facilitating secure near-field communication at low cost, low power scenarios.

Wireless devices have become an integral part of our everyday life. These devices are being used for many purposes such as for Internet and Telecommunication including communicating with other wireless devices, some of them cause harmful impacts on the human body [3] as mentioned in Table 1. The results, collected through interviews and surveys, show the intensity of harm of different wireless devices; Mobile phone is the most effective device with 96%, Bluetooth Device 32%, Laptop 54%, Tablet PC 14% and Wireless router 20% [2].

One of the main advantages of data-over-sound is that the physical infrastructure needed to facilitate sonic data transfer is already largely in place. The voice is gaining momentum as the primary control mechanism for many IoT devices, and as such, microphones

[1]https://www.livescience.com/62533-ultrasonic-ultrasound-health-hearing-tinnitus.html
[2]https://www.lifewire.com/3g-vs-4g-mobile-networks-the-health-factor-2373258
[3]https://www.cdc.gov/niosh/docs/82-109/

Table (1) Radio Waves with their frequency band and health hazards

Signal Types	Frequency Band	Health Hazards
Near ultrasonic Sound waves[1]	17-20KHz	Headache, Dizziness and Nausea
4G Networks [2] [4]	2-8GHz	Blood brain barrier, Cancer, Electromagnetic Hypersensitivity, Affects Glucose metabolism
Wi-Fi frequency[4]	2.4GHz/5GHz	Contributes to the Development of Insomnia, Damaging to Childhood Development, Derails Brain Function, May Impact Fertility
Infrared Light[3] [2]	300GHz-430THz	Damages human eye lens, Premature skin ageing

are increasingly being incorporated into more and more IoT devices. Beyond mobile devices and home assistants such as Alexa and Google Assistant, we are seeing voice control being added to smart TVs, fridges, doorbells, vacuum cleaners, light bulbs, locks, and thermostats[4] ,[5]. As humans continue to communicate with IoT devices using sound, we observe millions of devices of all form factors already equipped with the required processor, speaker, and microphone for data-over-sound functionality - without requiring any physical upgrades to existing hardware. Companies always try to innovate and future-proof their services, and many are now realizing the potential of data-over-sound to provide seamless device-to-device connectivity either to nearby devices or remotely (e.g. down a phone line), using nothing but sound.

The Chirp SDK[6] provides us the flexibility to work with sound signal but it does not have any collision control mechanism which occurs in IoT enabled scenario where the nodes (connected devices) are randomly assigned a channel id. If the number of nodes exceeds the maximum number of channel then, multiple devices can be assigned same channel id which causes collision in communicating with same channel id. In this work, we have developed a collision avoidance technique to mitigate the problem of collision in multi-channel communication irrespective of the number of nodes having same channel id. Moreover, when chirp is used for communication, the coverage range is very less. We have resolved this problem by using multi-hop transmission.

The rest of the paper is organized as follow, In Section 2 we presented some previous works done in the literature. Section 3 explains the Chirp SDK and Section 4 describes the proposed methodology used in this work. In Section 5, the implementation of the proposed system has been explained briefly. Section 6 describes the case studies and then Section 7 shows the results of the work. Finally, Section 8 concludes the work with some future directions.

2 LITERATURE SURVEY

Communication through sound signal is a healthy communication medium and some studies are available in the literature. Lee et. al [5] have proposed an indoor aerial acoustic communication system using inaudible audio signal for low-rate communication. They have extended the communication range up to 25m by using Chirp signal by reducing severe frequency selectivity and random phase distortion of the indoor acoustic channel. They also proposed the chirp digital modem to demodulate the chirp signal using a combination of fast Fourier Transformation (FFT) and Hilbert transformation. As an example, they have also introduced a TV content recognition service with backend query server to resolve the low data rate of 16 bps (approximately). They have not discussed the collision problem which may occur due to the multi-path propagation property of the chirp signal. Zhang et. al [8] have proposed an inaudible attack, DolphinAttack by using modulation of voice command on ultrasonic signals. They also presented some proof-of-concept attacks by injecting some voice commands and checked the feasibility of detection of this attack using support vector machine (SVM). Similarly, Wang et. al [7] have also developed Dolphin, which is a real-time unobtrusive communication between speaker and microphone. They have proposed an embedding approach based on OFDM. They have also used channel estimation for enhancing the robustness of the dolphin and designed an orthogonal error correction mechanism for correcting small decoding errors. Chen et. al [1] introduced iChemo which enables the ability of commercial-off-the-shelf mobile devices for sensing high-frequency ultrasounds. These devices are capable of detecting the ultrasounds of maximum 24 kHz and prevents the high frequency ultrasounds. The proposed iChemo algorithm can increase the sensing frequency of the ultrasounds up to 60 kHz by customizing the co-prime sampling algorithm of the mobile devices. This algorithm improves the sound sensing for different purposes but multiple source identification is not explained here. Novak et. al [6] proposed proximity networking mechanism which uses sound waves of very high-frequency emitted and captured by mobile devices. They proposed a software based modem named "Hush" an open source library for Android. They achieved a transmission rate of 4900 bps with distance 5cm - 20 cm. In most of the aforementioned works the authors have used chirp but they have not used it's multi-path property. This property can be used for multiple nodes to communicate with a single access point. Here, multiple collisions occur due to the fact that nodes are assigned their channel id in a random fashion which may cause multiple nodes getting the same channel id. Besides there is no other mechanism exists in chirp to handle this problem. We have implemented collision avoidance mechanism for getting a collision free communication.

3 CHIRP: DATA-OVER-SOUND

Chirp is a pre-built Software Development Kit (SDK), which seamlessly transfers data over sound waves. Chirp is used to encode an array of bytes to an audio signal. Any device can transmit this signal with a speaker and receive by any device with a microphone and Chirp SDK. It is robust over distances of several meters, and in noisy, everyday environments. Chirp is configurable to use audible or inaudible near-ultrasonic frequencies. Audible frequencies are

[4]https://www.pocket-lint.com/smart-home/news/143246-the-best-new-alexa-devices-ai-powered-tvs-fridges-mirrors-and-more
[5]https://qz.com/879673/samsung-wants-to-make-its-fridges-tvs-smartphones-and-other-devices-conduits-for-controlling-your-smart-home/
[6]https://chirp.io/

Table (2) Frame format of conventional CSMA/CA protocol

Frame Control	Duration	N Receiver Address	Transmitter Address	Frame Check
2 Bytes	2 Bytes	N x 6 Bytes	6 Bytes	4 Bytes

recommended for channels which have a limited audio sample rate VoIP connections, lossy codecs, or lower-spec embedded devices. Specific protocols are available for each of these scenarios. Near-ultrasonic frequencies should be used when noise disturbance is not desired. Frequencies between 17kHz and 20kHz are supported by virtually all mobiles and computers but are generally inaudible to the human ear. There is one protocol name "ultrasonic-long-range protocol" for transferring the data. This protocol transmits up to 8 bytes in 4.2s, uses a frequency range from 18000Hz to 19800Hz with a total of 1 channel. This protocol has an advantage of data transmission up to 15.2 bps with long-range capability but with the limitation of simultaneous sending and receiving data is not possible. Chirp provides another protocol name "Ultrasonic multi-channel protocol," allowing several devices to transmit data simultaneously. This protocol transmits data up to 7 bytes in 3.24s (17.3 bps) and uses a frequency range from 17500Hz to 20090Hz with a total of 8 channels.

Though the Chirp has the capability of multi-path data transfer, it is mostly used for peer to peer communication, as it does not have any implicit collision control mechanism.

4 METHODOLOGY

In real life IoT scenario, there are different cases, especially in indoors where multiple devices connect through a single access point. One of the examples may be the smart home where each electronic appliances are connected through IoT framework. We use chirp for communication among the devices for its multipath property, but this lead to the problem of collision. Suppose, we have some nodes for communication and there are several nodes sending data to access point through ultrasonic multichannel using chirp SDK. Here nodes can send data to access point, and access point will receive their data through various channels at the same time and take any action accordingly.

Since a node gets a random channel ID, that is from 0-6 and channel ID 7 is always assigned to access point for sending the data. When there are more than 7 nodes, more than one nodes will get the same channel id. Hence, there will be data collision while transmitting it through the same channel.

Carrier-sense multiple access with collision avoidance (CSMA/CA), is a network multiple access method in which carrier sensing is used, but nodes attempt to avoid collisions by initializing the transmission only after the channel is sensed to be "idle" and at the time of transmission, nodes transmit their packet data in its entirety.

As conventional CSMA/CA is implemented with regards to every frame sent and receive so it adds up to some overhead to every frame in the network. The frame format has been shown in Figure 2 where, Frame control is for Request To Send (RTS) and Clear To Send (CTS), Duration is for TTL (time to live) and Frame check is for sequence number.

Table (3) The minimum possible frame format required in our case for conventional CSMA/CA protocol

Frame Control	Frame Check	Actual Data
2 Bytes	4 Bytes	3 Bytes

As Chirp SDK can transmit 7 bytes in 3.24s (max frame size - 7 bytes), it is not possible to implement the complete CSMA/CA on every frame, hence to reduce this overhead we have proposed a new algorithm for collision avoidance, which basically work as a part of CSMA/CA on nodes rather than every frame. So, there will not be any overhead to send and receive the frames which ultimately reduces the frame numbers and decreases the latency time and it also increases the efficiency.

As we are not transferring large amount of data, if we reduce this frame size we will have only, Frame control of 2bytes (device number) and Sequence number of 2 bytes. Hence, frame structure will be as shown in Figure. 3.

To transmit 100 bytes of data we would have to make 34 frames, but in our algorithm we will just send 1 frame [RTS and CTS] of 7 byte [device number] and after connection establishment there will only be data transfer without any overhead or extra control bytes. This ultimately adds up to be a good algorithm in our case by reducing the latency and the number of frames to be transmitted [i.e only 16 frames (1 for RTS or CTS and 100/7byte is 15 frames)].

We have tackled the problem of collision using well known concept of CTS and RTS as follows.

- A Node sleeps for a random time.
- If two Nodes gets the same Channel ID (Suppose, Node 1 and Node 3 gets the same Channel Id), the one who wakes up early sends an RTS signal to the Access point (let us assume Node 1 wakes up early and sends the RTS)
- As soon as Access point receives an RTS packet, it sends CTS packet back to the node.
- The data transmission takes place between the node and access point, meanwhile the other node (Node 3) wakes up and sends RTS after a a time duration randomly assigned, to access point, but access point is busy in receiving the data from Node 1 in channel 1, it will not send CTS packet to Node 3 as described in Figure 1.
- After completion of transmission of data from Node 1, it again sleeps for random time between 50 to 60 second.
- The periodically sending Node 3 receives a CTS packet from Access point and start transmitting the data.
- This process works in synchronisation until any external interrupt occurs.

The above mentioned steps can relieve the system from the problem of collision even if two nodes get same channel id, by using the RTS and CTS packets. Implementing the RTS and CTS may create problem of overlapping of the RTS and data packets. Consider a case when two nodes 1 and 3 got the same channel ID, and the data transmission is going on between Node 1 and access point, and the Node 3 sends RTS to access point. It is possible that the data and RTS may collide resulting in data loss. But in our implementation it is not occurring. The reason is simple as when the data is transmitted by a node, it continuously sends the data in

the packet of 7 bytes such that no two sounds of each packet gets collided and also there is no gap between two packets, which keeps the channel busy, hence the RTS packet sent by other node will not get any room to be transferred and Node 3 sleep for certain time unit, in this way the collision will be avoided. This case is depicted in T3 time stamp of Figure 1. This will reduce the collision by

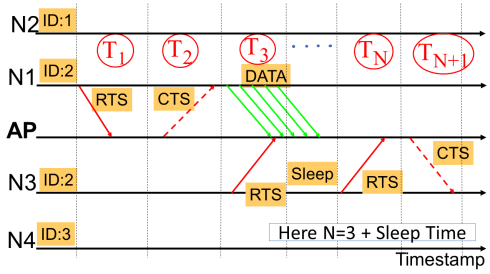

Figure (1) Flow control of data transmission when two nodes have got same channel id, in different Time stamps; Here, Node1 and Node3 have same channel id:2

We have used a sleep time of $t_s = 3.4sec$ as in Chirp SDK ultrasonic multichannel protocol can transmit up to 7 bytes in 3.24s (17.3 bps) and a gap of 3.4 sec will help in not providing the room for other data once a communication gets started. Hence, the system sleep for t_s second after sending a payload, this time is experimentally verified through running tests for multiple times. The t_s is very crucial as without correct value of t_s, either the RTS will collided with data or the converted sound for each packet will be overlapped, which results in receiving the incorrect data or even null payload by the access point. As the bandwidth of the channel is not too large so there is no probability of sending two packet or fragment simultaneously in a certain channel. That's why RTS and data will not collide at any instance of time because there is no space for RTS to go through the channel while transmitting data because of traffic of fragments in the particular channel.

The other important concern is the range of communication and we have solved it by introducing the multi-hop communication. We have used multiple access points as hops and the nodes which are far from an access point can send the data through hops.

5 IMPLEMENTATION

Initially, we declare the device number and channel ID to every node, then SDK starts on listening mode. The module 'Sensor Data Acquisition' is called to collect the data from the sensor periodically, which collects data with timestamp. After collecting data through sensor, the data is sent to module JSON string to get converted in IoT based key-value format so that user can read information in a better way. After that we convert this JSON string into a byte array for the transmission, this process is called serialization, which is done by the module 'Serialize'. When the whole string get converted into a byte array then we divide it into number of frames of size 7 bytes (as we are using Chirp SDK protocol- Ultrasonic multichannel which can send 7 Byte data at a particular time instance). Each packet or frame is sent individually using ultrasonic sound waves from node to access point. Access point save the data with a JSON file naming using device number + channel ID, because this is a unique compound name for every Node to identify node easily. After a particular time delay all the data received by access point are stored into cloud. A flow diagram depicted in Figure 2 has been explained briefly as follows,

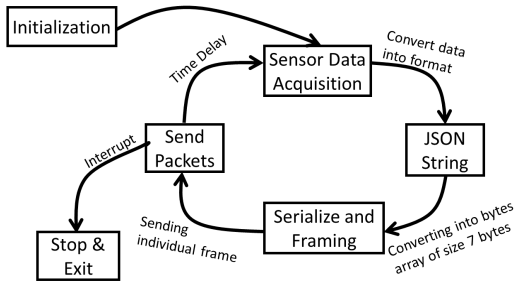

Figure (2) Implementation of the proposed protocol

(1) **Initialization:** This module defines the device number and gives every node a random channel ID from 0 to 6 and each time the channel ID 7 is assigned to access point so that access point can uniquely identify which node is going to send data through which channel.
(2) **Sensor Data Acquisition:** These modules only collect the data through different sensors in node and send it periodically to access point with a specific time stamp so that the same data can be easily identified when it was captured through sensor.
(3) **JSON string:** After data collected through sensors all data together get converted into JSON string which is basically an IoT based data string format having key-value pair for easily acquiring the information for user.
(4) **Serialization:** This is a basic method in networking to send data through any network, as not every machine have same software platform so we convert data into byte stream and this stream is then transmitted over a sound to other device or access point in our case.
(5) **Framing:** This module divides the whole byte stream into small frames, as ultrasonic multi channel chirp SDK can only send 7bytes of data at a time, so we can send packet individually from node and then assemble all the packets into a file at the access point

This process continues to run until any external interrupt occurs, after interrupt it get stop and we can run again manually the whole process. (external interrupt here referred as user interruption to stop the process).

6 CASE STUDIES

There are different possible scenarios in which we can validate our system. These case studies are mentioned in brief as follow.

Case Study-1: Multiple Nodes with Same Channel Id: Here, the three nodes got the same channel ID ie channel 1 as shown in Figure 3.a. Initially, as mentioned, all the three nodes sleep for the random time. Node 1 wakes up early and transfers the data via protocol mentioned earlier. In the meantime, suppose the node 2 sends RTS, since channel 1 is busy, node 2 will not receive CTS. Here, the RTS and data will not collide due to the concept of delay

(t_s) as mentioned in Section 4. As soon as the access point gets free, CTS will be received by the node 2 and data transmission takes place. Same situation occurs for Node 3.

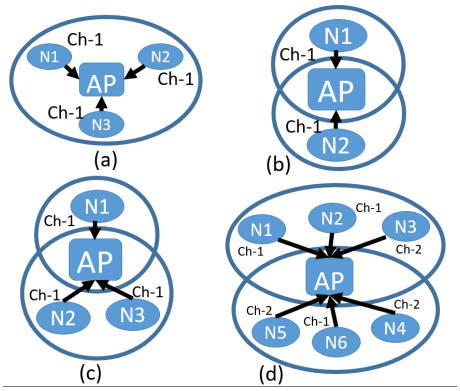

Figure (3) Different case study, (a): When the three nodes get the same channel id, (b): When the node 1 and node 2 are not in the range of each other ,(c): node 2, node 3 are in the same range and node 1 is in different range, (d): node 1, node 2 and node 3 are in same range including the access point, while node 4, node 5 and node 6 are in different range

Case Study-2: Two Nodes Not in Range: Here, node 1 and 2 are not in the range of each other, and the channel Id allotted to both of them is 1 as shown in Figure 3.b . After that, this case follows the rules of Case Study 1.

Case Study-3: Some Nodes in Range and Some are Not: In this case study, there are 3 nodes in which node 2 and node 3 are in the same range while node 1 is in different range as shown in Figure 3.c, so it cannot communicate with node 2 and node 3. Initially, all three nodes sleep for random time. After that, it mimics the aforementioned case studies to solve the collision issue.

Case Study-4: Multiple Nodes Not in Proximity: In this scenario there are total six node participating for the transmission of data. node 1, node 2 and node 3 are in same range including the access point, while node 4, node 5 and node 6 are in different range as depicted in Figure 3.d. Suppose, node 1, node 2 and node 6 are sending data through channel 1 while node 3, node 4 and node 5 are ready to send data through channel 2. This scenarios create a problem of data collision in channel 1 and channel 2. We can observe that, it mimics the aforementioned case studies with two different channel ids. Hence, the mechanism of collision avoidance will be applied for both the channels separately.

7 RESULTS AND ANALYSIS

We emulate out proposed algorithm using Android smart phone and Raspbery Pi in different scenarios and validate the performance of the communication with and without using the proposed protocol. We have taken 5 nodes for our experiment and the evaluation is made multiple times for obtaining robust result. The result of coverage range has been given in **D.** below.

A. Measurement of Data lost We have simulated the real scenario of multiple devices as different nodes. In our setup each nodes are transferring 188 bytes to a centralized access point. The nodes are assigned a random channel id and then data transfer starts. As the assignment of channel id is random, multiple channels can get the same channel id. In this scenario we have estimated the total amount of data loss in case of the existing technique and proposed technique. In this experiment the data loss is very high in case of the existing protocol and is very less (*negligible*) in case of our protocol as shown in Figure 4. The data loss in case of our protocol is very less due to the Clear to Send (CTS) and Request to Send (RTS) signals. It also infers that, the collision of data packets in case of our protocol is very less.

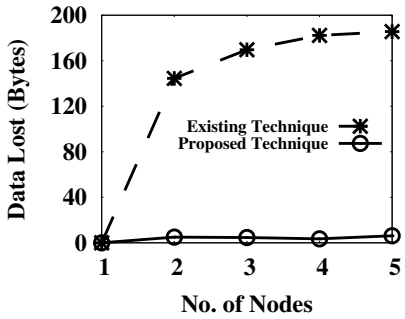

Figure (4) Data lost versus number of nodes for collision evaluation, lower the data lost, better the performance of the system

B. Delay for different data volume We have developed the scenario by incorporating some number of nodes and we are transferring different volumes of data. We then are calculating the delay for transferring for different volume of data streams. We have repeated the experiment multiple times and have taken the mean and standard deviation of the values. For analysing the collision factor, the delay for different number of nodes has been analysed as presented in Figure 4. Standard deviation has also been shown as the error bars at each point. Here, we have compared the delay in communication between different number of nodes in existing and proposed protocol. The result shows that the delay in communication in normal condition is very high with respect to the delay in our proposed protocol. Here, the delay increases due to the frequent collisions among the data packets of different nodes, which is taken care by our protocol.

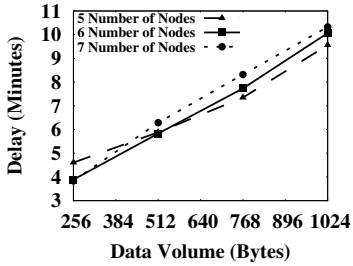

Figure (5) Delay versus data volume for different number of nodes for collision evaluation

C: Delay and Delivery probability against volume of data The delay has been observed for a fix number of nodes. Now, the

delay for multiple nodes has been analysed for different number of nodes. The result shows the variation of delay and the Delivery probability with respect to the percentage of channels with same id as shown in Figure 6. The results show that the efficiency of

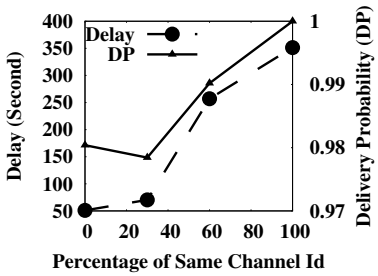

Figure (6) Variation of delay and delivery probability with respect to the percentage of channels with same id

communication is high in case of our proposed algorithm. As in our proposed algorithm collision is very less and the range of communication is high. Hence, our proposed methods can enhance the communication in IoT framework using the Sound as communication medium.

D: Extending the Range Range of communication is one of the important characteristics. The chirp SDK has a limited range of data transmission. In our work, the range of the communication has been improved by multi hopping as shown in Figure 8. In multihop communication, multiple access points can be used according to the requirement. Each access point has the responsibility to transmit the packets to its destination access point. Here, the destination access point can be selected as the nearest access point of the destination node. In Figure 8 the destination access point is AP3 for node N1. Normally, we can transfer data up to 7 meters but, if a node is present at a distance of 20 metres from the access point the normal transmission will not work. Hence, multihop communication is used and we use two hops to increase the coverage up to 21 metres. We have transferred the data of different volume in two hops. An analysis of delay and data lost is shown in Figure 7. From the figure it can be seen that the delay in multihop communication is a bit more than normal communication. In addition, the data loss is insignificant as well.

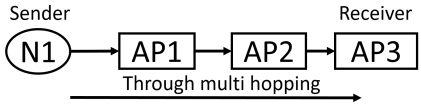

Figure (7) Improving the range of communication using Multihop system

8 CONCLUSION AND FUTURE SCOPE

The issues of hazardous radiations are one of the major challenges in the field of IoT and its several applications. In this work, we have proposed a healthy communication system using chirp SDK. We have resolved the problem of collision which is very dominating problem when chirp is used for multi device data transmission. The problem of getting the same channel id (due to the random

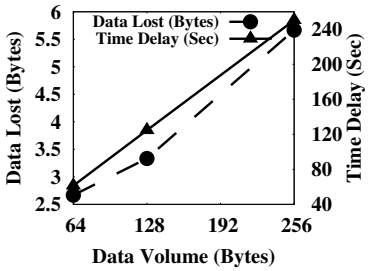

Figure (8) Variation of Delay and Data Lost in multihop communication

assignment of channel id) heavily affects the overall transmission by sever collisions. We have used RTS and CTS signals along with a time delay for controlling the collisions data interference respectively. Besides, we have tested the proposed technique for multiple cases of node positioning and we have got excellent results. We have presented the performance on the basis of Data Loss, Delay, Delivery Probability etc., in different cases. Besides, we extend the range of communication by using multi-hop. In this work, a time delay is used to protect the transmission from overlapping of data. But, in real life, incorporating the delay may turned into the problem of starvation. Moreover, the system is designed in small scale which can further be implemented in large scale to measure the performance of the proposed technique.

9 ACKNOWLEDGEMENTS:

The authors are grateful to (a) Council of Scientific & Industrial Research (CSIR), India, a premier national R&D organisation (b) Project CityProbe funded by IMPacting Research INnovation and Technology (IMPRINT) India, and (c) Project IntAirSense funded by Department of Science & Technology (DST), West Bengal, India for funding our research work in parts.

REFERENCES

[1] Yuchi Chen, Wei Gong, Jiangchuan Liu, and Yong Cui. 2018. I can hear more: Pushing the limit of ultrasound sensing on off-the-shelf mobile devices. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2015–2023.

[2] Soyun Cho, Mi Hee Shin, Yeon Kyung Kim, Jo-Eun Seo, Young Mee Lee, Chi-Hyun Park, and Jin Ho Chung. 2009. Effects of infrared radiation and heat on human skin aging in vivo. In *Journal of Investigative Dermatology Symposium Proceedings*, Vol. 14. Elsevier, 15–19.

[3] Muhammad Ali Jamshed, Fabien Heliot, and Tim Brown. 2019. A Survey on Electromagnetic Risk Assessment and Evaluation Mechanism for Future Wireless Communication Systems. *IEEE Journal of Electromagnetics, RF and Microwaves in Medicine and Biology* (2019).

[4] RSA Larik, GA Mallah, MMA Talpur, AK Suhag, and FA Larik. 2016. Effects of wireless devices on human body. *J Comput Sci Syst Biol* 9 (2016), 119–124.

[5] Hyewon Lee, Tae Hyun Kim, Jun Won Choi, and Sunghyun Choi. 2015. Chirp signal-based aerial acoustic communication for smart devices. In *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2407–2415.

[6] Ed Novak, Zhuofan Tang, and Qun Li. 2018. Ultrasound Proximity Networking on Smart Mobile Devices for IoT Applications. *IEEE Internet of Things Journal* 6, 1 (2018), 399–409.

[7] Qian Wang, Kui Ren, Man Zhou, Tao Lei, Dimitrios Koutsonikolas, and Lu Su. 2016. Messages behind the sound: real-time hidden acoustic signal capture with smartphones. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. ACM, 29–41.

[8] Guoming Zhang, Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, and Wenyuan Xu. 2017. Dolphinattack: Inaudible voice commands. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 103–117.