# Quality Preserving Voice Stream Multicast over Mobile Low Power Wireless Networks

Shneka Muthu Kumara Swamy
Department of Computer Science
Colorado School of Mines
Golden, Colorado, 80401
Email: smuthukumaraswamy@mymail.mines.edu

Qi Han
Department of Computer Science
Colorado School of Mines
Golden, Colorado, 80401
Email: qhan@mines.edu

***Abstract*—During disasters when the communication and power infrastructures are unavailable, we can deploy low-power, low-cost, portable wireless nodes and use them to facilitate multi-hop voice communication between the survivors and the rescue team. Similarly, using multi-hop, multicast, we can facilitate communication between the rescue team and multiple survivors, with whom direct communication is not possible. This paper examines a multi-layer adaptive approach to the problem, the voice data captured from the sender are compressed based on the availability of the bandwidth and contention it might cause in the network. We also perform distributed admission control to ensure that the new streams entering the network do not affect the old ones. To evaluate, we implement the ideas on a testbed of 18 Raspberry Pi equipped with Xbee radios. Our experimental results show that voice data can be multicast to at least 6 destinations with acceptable voice quality in this setup.**

***Index Terms*—Voice streaming, Low power mobile networks, Quality preserving**

## I. INTRODUCTION

A disaster such as an earthquake might lead to the destruction of pre-deployed communication and power infrastructure making communication impossible. Consider an underground mine or similar complex environments, which are prone to fires and need more effective communication for evacuation of the survivors. Evacuation path planning for people using clustering and distributed approaches has been investigated before [8]. Another commonly adopted approach is to have mobile robots to drop breadcrumb communication nodes to form an ad-hoc communication network. This network hence consists of both stationary (e.g., immobile victims, stationary communication nodes) and mobile nodes (e.g., moving robots, survivors moving around) for communication between the emergency response center and the survivors, and also among survivors. Multicasting is needed for various applications such as delivering voice evacuation instructions. While high quality voice data is desirable, it is not absolutely necessary. Instead, maintaining voice data at acceptable quality is sufficient. Figure 1 is a sample scenario motivating our work. Hence, the goal of our work is to support quality-preserving voice stream multicast over low power wireless networking where nodes might be mobile.

Most of existing work on supporting voice streaming over low bandwidth wireless networks assumes all nodes are stationary and network topology is static [13], [10], [11], so

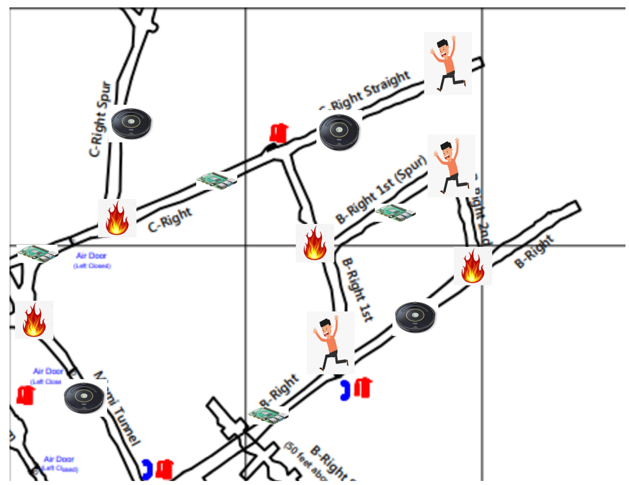



Fig. 1: A motivating scenario for voice stream multicast in an underground environment

routing paths are built in advance. However, in our scenarios, people move around. Even when people are not moving, the dynamic environment will cause the link quality to fluctuate over a wide range, so that a node many change its parent. QACM [1] is a voice streaming system that considers node mobility, it is designed for convergecast, which is different from multicast we focus on in this work.

Node mobility in low power wireless systems makes it hard to set up and maintain communication. Considerable amount of overhead in both time and communication may be incurred for identifying and maintaining a path from a source to a receiver. In addition, node mobility brings challenges to existing admission control as nodes' contention domain varies.

We tackle the challenges caused by node mobility from several perspectives: 1) we determine the path via analyzing the success probability of the path for sending the voice stream and the number of neighbours that might be affected. 2) we improve the voice codec by taking into account talk spur and silence periods inherent in human speech. 3) we conduct admission control and voice quality measurement at the intermediate nodes rather than the final sink nodes. This way, we ensure that the voice stream currently in transmission is not affected by a new node in the communication range and low quality data is not sent further.

The rest of the paper is organized as follows. Section II provides a summary of the existing work related to the work. Section III presents the details of our system. The experimental setup and results are provided in Section IV. We conclude this paper in Section V.

## II. RELATED WORK

To the best of our knowledge, no existing work has investigated how to provide satisfactory voice communication in mobile low power wireless networks. The research most relevant to our work comes from the following two areas: voice over stationary wireless sensor networks (WSNs), voice over mobile ad hoc networks (MANETs).

To support voice streaming over WSNs, the performance of a Zigbee network for voice streaming has been evaluated using a star network topology [3]. However, enforcing a star topology is not realistic in many disaster scenarios. Using wireless mesh sensor networks, a single two-way voice streaming system with a customized dual-radio hardware platform, FireFly, has been developed and deployed in the NIOSH experimental coal mine in Pennsylvania [13]. [12], [7] also investigate real time voice streaming by using analog compression methodology. Further, multiple concurrent voice streaming over WSNs has been shown to be possible in a system called QVS by dynamic adjustment of voice compression ratio and admission control [10]. ASM [11] is another voice streaming system that differs from QVS by targeting multicasting scenarios, Dynamic Source Routing (DSR) has been used for voice communication through Zigbee for unicast [14]. However, all these systems assume nodes are stationary. i.e., none of these efforts have studied how to relay voice traffic using mobile nodes with stringent resource constraints. Node mobility makes the underlying network more dynamic and voice transmission more prone to error, rendering the problem a lot more challenging.

At a high level, voice over mobile low power wireless networks bears a lot of similarity with voice over MANETs, as they share similar challenges due to multi-hop routing and high error rates. QACM [1] supports convergecast of voice streams in low power mobile wireless networks, while we focus on multicast in this work. MAODV [6] is one of the most commonly used protocol for multicast routing in MANET, which floods the network to maintain the multicast tree. Similarly, MAMR [15] maintains a reachability tree that needs to be continuously populated. However, mobile WSNs have much more stringent resource constraints than MANETs in terms of computation, communication, storage, and power resources and hence this method cannot be used. HVDB [9] and DQMRP [2] take QoS into consideration but rely on another sensor to determine the location and speed of the nodes in motion, which incurs overhead. Admission control for MANETS is also well researched. For instance, [17] uses speed to determine a probability of successful transmission which is then used to decide whether or not to admit a new source, RT-WMP [16] and DACME [4] are developed for WiFi and hence cannot be used in our case. Further, our network is heterogeneous and has intermittent connectivity; whereas, MANETs always assume homogeneous nodes and end-to-end connectivity.

## III. DESIGN OF A QUALITY-PRESERVING VOICE STREAM MULTICAST SYSTEM

In this section we explain the design of our quality preserving voice stream multicast system. We propose a system architecture as shown in Figure 2. There are three types of nodes in the system: source nodes are those who have captured voice data and need to multicast to their intended recipients; relay nodes are those who help forward data to recipients; sink nodes are the multicast recipients. We next describe in detail the specific actions taken by each type of nodes in the system.

- **Sources:** After the voice data is captured, it is encoded to reduce the amount of data for transmission. A multicast route request is then broadcasted to designated sinks. The source then waits for a period of time and processes all the replies using our probabilistic reactive routing protocol. For the sink(s) whose replies are not received the route request is broadcast again. After a pre-specified number of attempts, if the source still does not receive the route reply from the sink(s), the source declares that the particular destination(s) cannot be reached. If no reply from any destination is received it implies that the new voice stream cannot be admitted at this point of time. If a path is established, the voice data is sent.
- **Relays:** Unlike the source node, a relay has no control over the compression method or compression ratio used to encode the voice stream. Upon receiving the voice data, a relay node forwards the data for up to $K$ times to ensure reasonable voice quality at the next hop. The voice quality is estimated in the same way as done at the source.
- **Sinks:** A sink node evaluates the voice quality, which determines the usability of the voice data.

The rest of this section explains the major components in the architecture, including voice encoding, voice quality measurement, probabilistic reactive routing protocol, and admission control.

### *A. Voice Codec*

ADPCM is a traditional encoder and it has been adopted for audio data compression over low bandwidth wireless networks [13][10][11]. The wide use of ADPCM can be attributed to the fact that it is computationally less intensive than the others and it allows variable length encoding, that is, a 16-bit data can be converted into 4 bits, 3 bits and 2 bits. This availability of variable encoding makes it possible to choose an appropriate compression ratio based on the bandwidth availability, accuracy, and network congestion. *Baseline ADPCM:* ADPCM has certain inherent disadvantage. Since all data points are compressed, there is a very high dependence between the packets. A packet lost in the middle may have a significant impact on all following packets due to the way

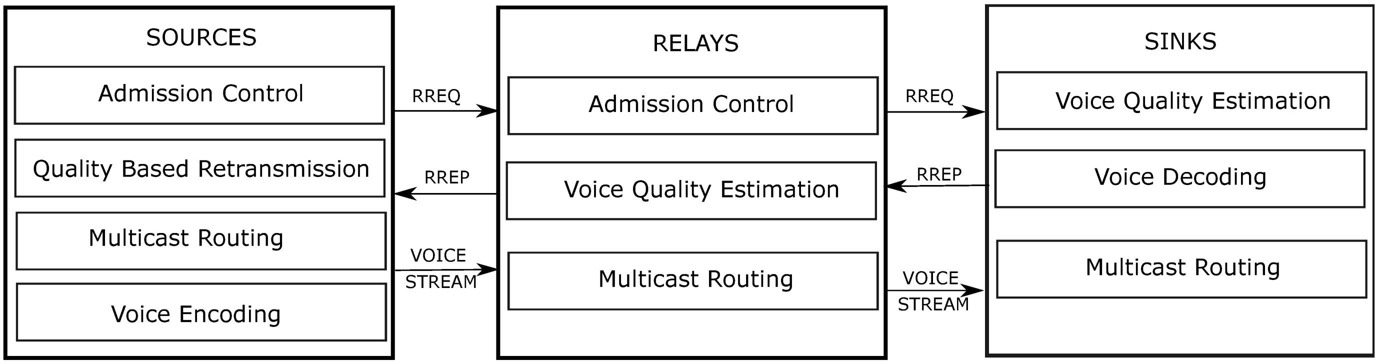



Fig. 2: System Architecture

ADPCM works. In this work, we simply send all data points $K$ times and use this approach as the baseline voice codec.

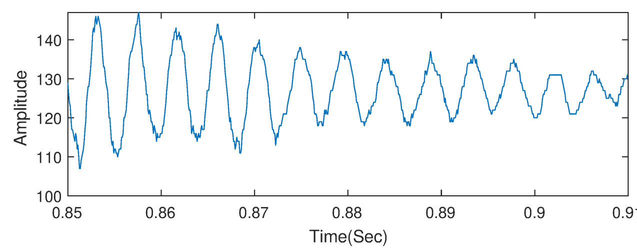
*Modified ADPCM:* We customize traditional ADPCM for voice data by taking into consideration the fact that human speech is often interspersed with voice spur and silence. Figure 3a shows the original waveform of a reference voice signal. It consists of periodic segments of peaks and troughs. Thus, instead of using all the data points, we can just use the values of the local maxima and minima, the other points in between these two values can be interpolated using a linear equation. In addition, a considerable amount of horizontal symmetry is present in the raw waveform and the curve is centered around a certain point. It is known that humans pause for a brief moment from time to time while they are talking. During this brief silence period, the microphone can pick up the noise from the environment and the noise also exhibits peaks and troughs. In original ADPCM, these points are still encoded as other voice data points. In fact, these points are not essential to understanding the voice and hence can be replaced by the central point. Figure 3b shows the voice waveform after this conversion. Our experiments show that voice quality degradation of the converted voice signal is acceptable.

To make sure the modified ADPCM works and we are able to interpolate the data points in between the maxima and minima, we must include the time stamps or the indices of the maxima and minima. The time stamps can also be encoded using ADPCM. While three compression ratios are available for ADPCM, we use 4-bit compression for the time stamps to maintain accuracy.

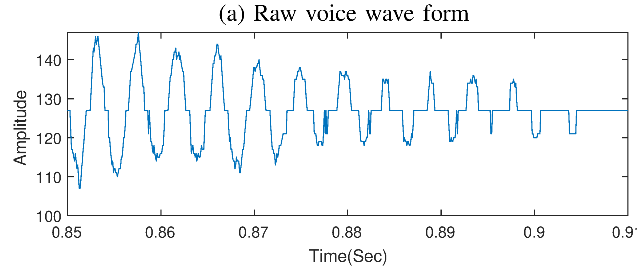
With modified ADPCM, we can send data with different compression ratios. Voice stream is sent with a default compression ratio initially. The compression ratio for future voice streams will be adjusted based on the voice quality at the next hop estimated by the source.

### B. Voice Quality Measurement

In order to measure voice quality in an online fashion, we adopt the E-model [5], which has been used in several previous voice streaming systems [10], [11], [1]. In this model, voice quality is measured using an R value as in Eqn. 1, where packet delay is considered having negligible impact on voice




(a) Raw voice wave form




(b) Converted voice wave form

Fig. 3: Justification for modified ADPCM

quality [10], hence is omitted from the calculation.

$$R = R_o - I_e \tag{1}$$

$$I_e = \alpha + \beta * ln(1 + \chi * e) \tag{2}$$

We use the same $\alpha$, $\beta$, and $\chi$ as in QACM [1]. $e$ represents data loss rate and $R_o$ is 93.2. An R value of 50 is considered acceptable voice quality.

*Voice Quality Measurement for Baseline ADPCM.* Since almost all the packets have the same number of data points in baseline ADPCM, the drop of any packet would contribute equally to the reduction of quality. The number of packets lost is simply the difference between the number of packets sent and received.

*Voice Quality Measurement for Modified ADPCM.* In contrast with baseline ADPCM, not all packets in modified ADPCM have the same number of data points. The number of data points can be calculated by using the difference between the last and the first time stamps in the packet. For the quality calculation to be fair, we need to have all the packets to have approximately the same number of data points. Therefore, we determine the smallest number of data points contained in one packet in a series of packets received and use that as a "small packet" size. Then for each packet received, we divide its total number of data points by the small packet size. The quotient of the division is considered as the number of small packets. Let $n_s$ represent the number of small packets sent, $n_r$ is the number of small packets received, and $n_{d_i}$ be the number of data points in the $i^{th}$ packet. The number of lost small packets, $n_l$, is calculated as follows.

$$\begin{aligned} n_l &= n_s - n_r \\ n_{d_{min}} &= min(n_{d_1}, n_{d_2}, \ldots, n_{d_k}) \\ n_s &= \left\lfloor \frac{n_{d_1}}{n_{d_{min}}} \right\rfloor + \left\lfloor \frac{n_{d_2}}{n_{d_{min}}} \right\rfloor + \ldots + \left\lfloor \frac{n_{d_k}}{n_{d_{min}}} \right\rfloor \end{aligned}$$

The number of small packets are appended at the end of each normal packet. The summation of all the number of small packets is used by the receiver to get the accurate value of $n_s$.

*Voice Quality Measurement for Mixed Compression Ratios.* If we send packets with different compression ratios, we propose the following equation (Eqn. 3- 7) to calculate voice quality. Let $n_s$ be the total number of packets sent, $n_1$ be the number of packets with one compression ratio, $n_2$ be the number of packets with another compression ratio, and $n_l$ be the data loss rate, then the voice quality can be derived as follows.

$$R = R_o - I_e \tag{3}$$
$$I_e = A + B * ln(1 + X * n_l) \tag{4}$$
$$A = \frac{n_1}{n_s}\alpha_1 + \frac{n_2}{n_s}\alpha_2 \tag{5}$$
$$B = \frac{n_1}{n_s}\beta_1 + \frac{n_2}{n_s}\beta_2 \tag{6}$$
$$X = \frac{n_1}{n_s}\chi_1 + \frac{n_2}{n_s}\chi_2 \tag{7}$$

We use this equation and plot the relationship between voice quality (i.e., R value) and data loss rate in Fig. 4. As the required quality must be kept above 50, the graph only shows data loss rate up to 30%. We observe that the quality can be improved by using higher-bit encoding for the dropped packets. This is more beneficial than transmitting the entire voice stream at a higher bit rate as it uses more packets and might be impacted by unreliable transmission.

All the packets have the same number of data points in baseline ADPCM, hence the length of the packets received can be used to determine whether there is packet loss at the relay node. Relay nodes do not decode data in modified ADPCM, so there is no way of knowing which packets were sent multiple times and which were not. Instead, we sum up the the number of small packets appended at the end of each packet. This value can be above the total number of small packets, in which case packet loss is assumed to be zero. Hence, this method overestimates the voice quality at the relay. At the sink node, accurate voice quality is obtained after decoding the received data.

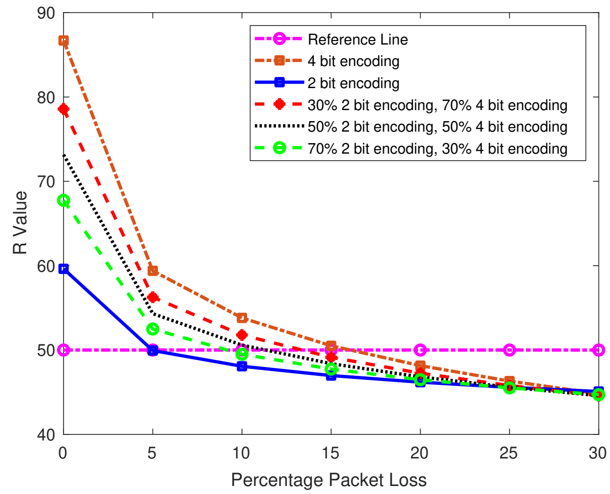



Fig. 4: Impact of packet loss rate on voice quality in modified ADPCM with multiple compression ratios

We calculate the actual voice quality at the sinks. At source and also relay nodes, we estimate the quality of voice streams received at the next hop. For this we use the default acknowledgements of packet reception sent from the receiver to the source. A lost acknowledgement need not necessarily mean a lost package, as the acknowledgment itself can be lost. Given no extra information about the packet, we assume the probability of lost packet given lost acknowledgement is 0.5. using this packet loss information, we use the voice quality equation to estimate the voice quality at the next hop.

### *C. Probabilistic Reactive Routing Protocol*

Probabilistic reactive protocol uses information of the nodes that it has communicated with before instead of all its neighbours to find the path. We incur overhead in forming the path each time, but do not maintain any tree structure to avoid constant 'hello' messages. When a voice data is received by the source. The source generates and broadcasts a route request (RREQ) to all the adjacent nodes, the relay nodes broadcast to its neighbours in the system, and so on. Then the source node waits for the route reply from the sinks. Upon receiving the first route request, the sinks start a timer and wait for more route requests. With a set of these packets it chooses a suitable path, based on probability of success, number of hops, and the node degree and sends a route reply back to the source.

- The probability of success is calculated using the RSSI value. RSSI gives the signal strength between two communicating nodes. This strength is also an indication of probability of successful data transmission between the two nodes. RSSI value remains the same if link quality does not change over a period of time. That is, neither of the nodes move very far apart, or there is no dynamic obstruction/interference in between the nodes. For instance, for Xbee radios we use in our experiments, the success probability is expressed as a function of the RSSI value. Similar relationship can be derived for other types of radios. Let $p_s$ be the success probability of one-time transmission between adjacent nodes, $n$ be the

number of attempts, and $p_{sn}$ be the success probability after n attempts, then $p_{sn} = 1 - (1 - p_s)^n$. Since the successful transmission of data on each segment of a path is independent, the probability of the entire path is the product of success probability in each path segment.
- The number of hops determines how many relay nodes must participate in routing.
- The degree of a node is the total number of nodes in its communication range. It gives us an idea about how many other nodes can get affected if the node is chosen to be on the path. Selecting a node with smaller degree implies more sources can be supported as interference will be less. This also ensures that the path at the boundary of the area have their fair share in the communication process. To reduce the communication overhead, we do not maintain neighbourhood table as there is no need for us to know accurate number of neighbours.

The source node starts with sending a route request message, which is broadcast by all the relay nodes. A sink upon receiving its first request waits for a period of time in order to get more paths to the source. The sink selects a path, as mentioned below, by taking into account all the three factors described above.

- Finds the smallest number of hops and the smallest node degree in the set of route requests obtained.
- Finds the group of route requests that are one or two hops more and one or two degrees more than the smallest value found in the previous step.
- Among the group, it chooses the path with the highest success rate.

This way of path selection ensures high probability of voice stream received are with reasonable quality and also ensures relays do not go out of the communication range in most of the scenarios.

After choosing a specific path, the sink generates a route reply and sends it back to the source. The source waits for a random amount of time before responding to the received route replies. For the replies that it gets within this period of time, the source finds a common relay node if one exists, and sends over the voice stream.

### D. Distributed Admission Control

Admission control for quality-aware system is used to determine if the path can send the data to the receiver at an acceptable quality without affecting the concurrent streams that are already using the network. Distributed admission control approaches used in QVS and ASM are not suitable for a mobile ad-hoc network as the nodes are in motion and before the receiver gets the data and replies, the path might not even exist any longer. To reduce the overhead of admission control, we conduct it at the source and the relay nodes, not the sinks.

- On a source node, before any message is sent, a route request is sent. The source waits for a finite period of time for the route reply. Failing to get one after a pre-specified number of attempts, the new stream cannot be admitted.
- On a relay node, it decides whether the voice stream is admissible or not based on voice quality. If the voice quality is not acceptable, the voice stream is not transmitted further. For decoding the data, certain information like the frame rate, the compression ratio used for encoding is required. We hence send this meta-data information in one packet. If this meta-data packet is not received, the receiver will drop the voice packet, since it cannot decode it. If only one compression ratio is used, receiving one of the meta-data packets is enough; however, when adaptive compression ratio is used, final meta-data packet has to be received. Failure to get this meta-data packets, the relay assumes that it lost connection with the source, the source will abort sending data and try for a new route, or declare itself inadmissible after a pre-determined number of attempts.

## IV. Performance Evaluation

The hardware used and some of the specifications of the software are described below. In this section we also present the results for our experiment.

### A. Experimental Setup

We implemented our system on Raspberry Pi 3B with Digi Xbee radios as shown in Fig. 5. The radio provides 250 Kbps link speed at 2.4 GHz frequency and works at 3.3 V. Some nodes are Series S1 and others are S2C. S1 uses 802.15.4(10ef) firmware, and S2C runs 802.15.4 SMT(2003) firmware version. Both types of radios are configured to communicate with each other. CSMA-CA is enabled to ensure collision avoidance.

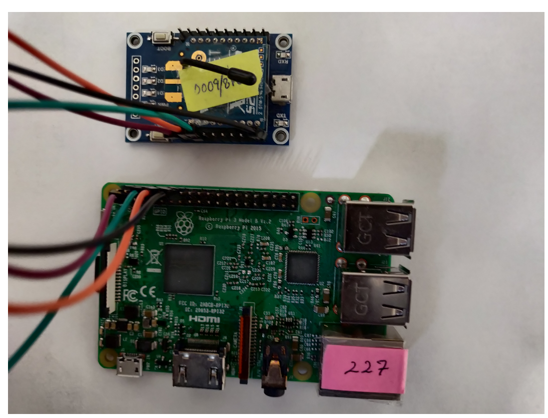
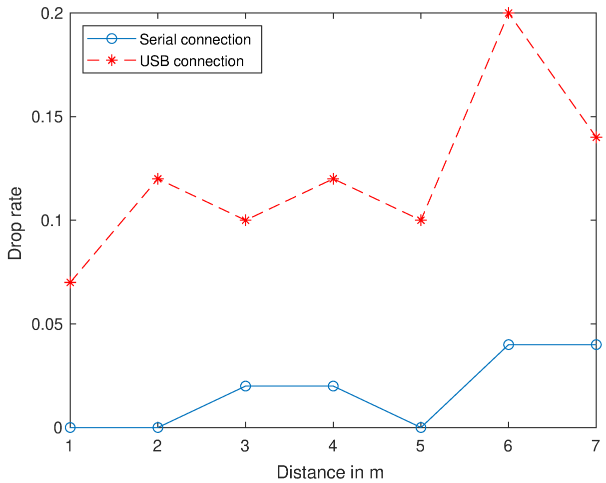
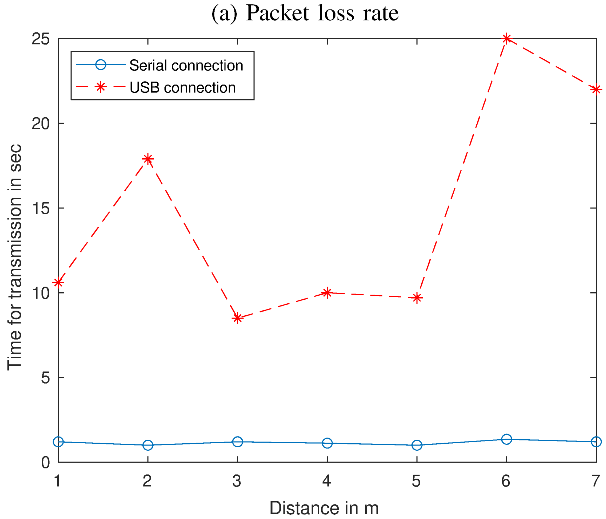



Fig. 5: Hardware used

Xbee has two types of connection with Raspberry Pi: USB connection and serial connection. In our exploratory studies, we sent a 60-bit message 50 times from one source to one sink, both being stationary. Fig. 6a shows the packet loss rate and Fig. 6b shows the packet delivery latency for both connections. These results clearly indicate the serial connection of Xbee works faster and more efficient than the USB connection, hence for our following experiments, we choose to use the serial connection of Xbee.




(a) Packet loss rate




(b) Packet delivery latency

Fig. 6: Performance comparison of two types of connection between Xbee and Raspberry Pi

To make sure that the experiments can be conducted in a small indoor environment, we place each Xbee radio in a box with layers of aluminium foil to restrict the communication range to 4m.

To make sure we can repeat the mobile scenarios, we put some nodes on robots. A node in motion is placed on the top of an iRobot Create which uses the robots command module to move at random. This setup makes repeating the experiments easier. A robot is able to detect walls and virtual walls. Virtual walls are pyramid shaped devices that emits infrared signals in the form of a cone. When a robot detects this signal, it treats it as a wall and does not go beyond it. This is used to protect the Raspberry Pi devices and does not interfere with communication. It is assumed that the robots moves in human walking speed.

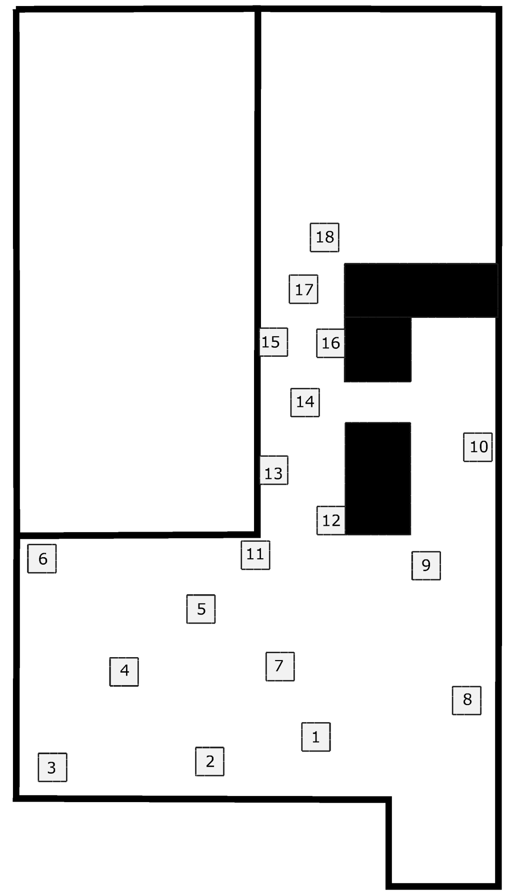



Fig. 7: Placement of the nodes in an indoor environment, Black areas represent thick walls or closet space.

An XBee radio working in API mode assumes a packet is lost when a sender does not receive an acknowledgement. However, it is possible the acknowledgement itself is lost. Without incurring complicated calculation, we can assume losing the acknowledgement is about half of the time when a packet is considered lost. This implies that the sender might overestimate the voice quality in certain scenarios and hence we increase the required R-value slightly more than that of the actual value.

We conducted the following experiments with 18 nodes laid out in an 825 SQ.FT apartment as shown in Fig. 7. Unless mentioned explicitly, node 1 is used as the source. Nodes 5, 6, 9, 13, 16 and 18 are used as the multicast sink nodes depending upon the number of destinations we want to test. We have a total of four robots and other additional mobile nodes are carried by people. The source sends voice messages to all the receivers periodically.

### B. Experimental Results

We first study the performance of modified ADPCM when compared with baseline ADPCM. This baseline is an improved version of ASM [11] which is an existing system that resembles ours the most. As described in the related work section, other systems cannot be compared with ours. Fig. 8 shows the comparison of voice quality (i.e., R-value) between baseline and modified ADPCM for 3-bit compression, with node 1 as source and nodes 5, 9, 16, 13 as stationary multicast sinks. We observe with baseline ADPCM most of the packets get dropped in the relay nodes because of low voice quality. In contrast, all the sinks receive acceptable voice data (i.e., R values above 50) when Modified ADPCM is used. For this reason we choose to use Modified ADPCM for the rest of the experiments.

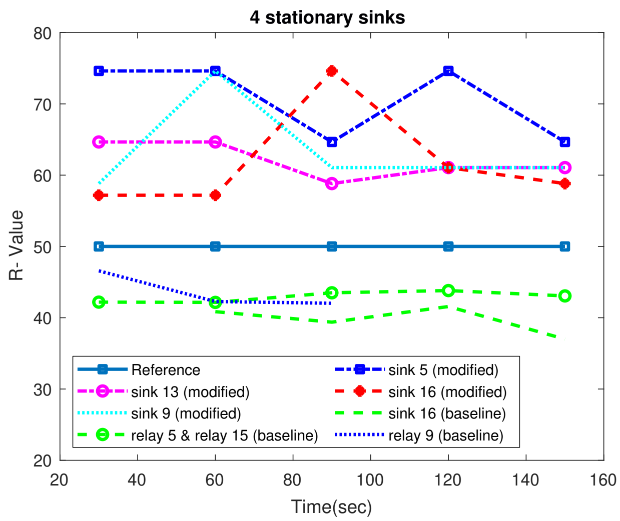



Fig. 8: Comparison between baseline and modified ADPCM for stationary multicast sinks

We next study the scalability of our system when the number of mobile sinks increases. Fig. 9 shows the voice quality of 4, 5, an 6 multicast sinks, respectively. We observe all the destinations receive voice data of reasonable quality. As we increase the number of mobile sinks, their voice quality did not decrease. This is an indication of good scalability of our system in support more mobile sinks concurrently. However, due to the limited number of mobile nodes we can provide in our experiments, we did not test more mobile sinks.

We further would like to find out if adapting compression ratio at the source will have any impact on the quality of voice data received at the sinks. Fig. 10 shows our experimental results for both stationary and mobile sinks. We observe occasionally voice data drops. This might be because the meta-data packets are lost.

We finally studied whether our system can support multicast from multiple sources. Our results are shown in Fig. 11. We considered both stationary and mobile scenarios. We used nodes 1 and 16 as sources for one experiment and nodes 1 and 13 for the other experiment. These nodes are chosen to verify the scalability of the system when the sources are far apart from each other (i.e., 3 to 4 hops between nodes 1 and 16) or close to each other (i.e., 1 or 2 hops between nodes 1 and 13). We observe most of the voice data reached the sinks with reasonable quality. The voice streams from different sources may reach a sink at the same or different times.

**Performance Summary:** Our experimental results show that quality-aware voice stream multicast is possible by using our system. In a system of 18 nodes at least 2 sources and at least 6 moving destinations can be supported, this is an improvement of 3 concurrent voice streams that can be supported in previous systems such as QACM and ASM. While our routing protocol sends more route requests than a tree structure would have, we observed no congestion in our setup. We find that modified ADPCM and adaptive ADPCM have a comparative performance and is much better than the baseline ADPCM method. By using Modified and Adaptive ADPCM most of the packets can be transferred with reasonable quality.

## V. CONCLUSION

In this paper, we present a system that allows the effective transmission of voice streams through a low power wireless network in the presence of node mobility. Our system is able to deliver voice streams with acceptable voice quality. In a physical test bed of 18 nodes, we extensively tested the system and results show that the system scales well when the number of mobile sinks increases.

## REFERENCES


[1] M. Adkins, Q. Han, and S. Pasricha, "Quality-aware voice convergecast in mobile low power wireless networks," in *International Conference on Mobile Computing, Applications, and Services*. Springer, 2019, pp. 215–229.

[2] S. Baolinr and L. L. Yuan, "Distributed qos multicast routing protocol in ad hoc networks," *Journal of Systems Engineering and Electronics*, vol. 17, no. 3, Sep. 2006.

[3] D. Brunelli, M. Maggiorotti, L. Benini, and F. L. Bellifemine, "Analysis of audio streaming capability of zigbee networks," in *Proceedings of the 5th European Conference on Wireless Sensor Networks*, ser. EWSN'08. Berlin, Heidelberg: Springer-Verlag, 2008, p. 189–204.

[4] C. Calafate, J. Oliver, J.-C. Cano, P. Manzoni, and M. Malumbres, "A distributed admission control system for manet environments supporting multipath routing protocols," *Microprocessors and Microsystems*, vol. 31, pp. 236–251, 06 2007.

[5] R. G. Cole and J. H. Rosenbluth, "Voice over ip performance monitoring," *SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 2, pp. 9–24, Apr. 2001. [Online]. Available: http://doi.acm.org/10.1145/505666.505669

[6] C. E. P. E. M. Royer, ""multicast operation of the ad-hoc on-demand distance vector routing protocol"," in *Proc. ACM MOBICOM*, Aug 1999.

[7] T. Facchinetti, M. Ghibaudi, E. Goldoni, and A. Savioli, "Real-time voice streaming over ieee 802.15.4," in *The IEEE symposium on Computers and Communications*, 2010, pp. 985–990.

[8] G. Fragkos, P. A. Apostolopoulos, and E. E. Tsiropoulou, "Escape: Evacuation strategy through clustering and autonomous operation in public safety systems," *Future Internet*, vol. 11, no. 1, p. 20, 2019.

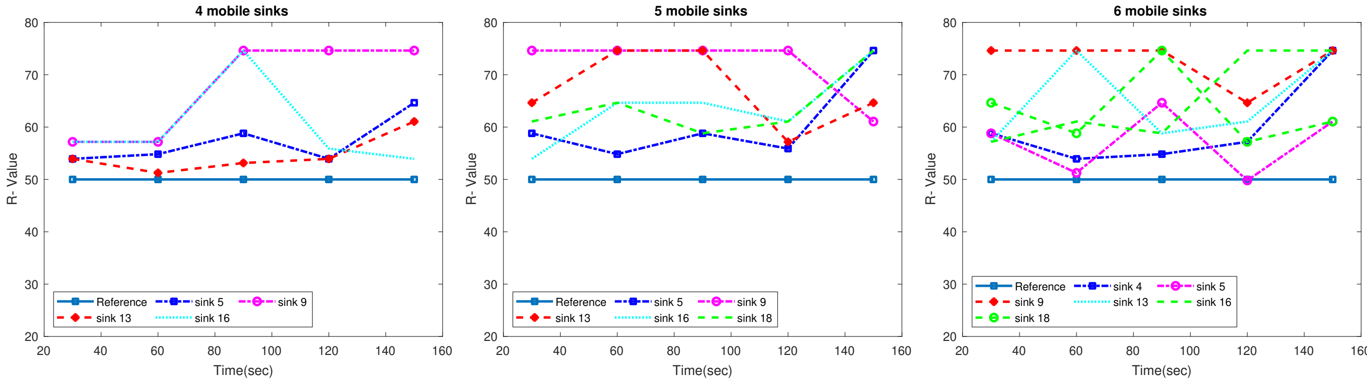
[9] G. Wang, J. Cao, L. Zhang, K. C. C. Chan, and J. Wu, "A novel qos multicast model in mobile ad hoc networks," in *Proceedings of 19th IEEE International Parallel and Distributed Processing Symposium*, April 2005.

[10] L. Li, G. Xin, L. Sun, and Y. Liu, "Qvs: Quality-aware voice streaming for wireless sensor networks," in *2009 29th IEEE International Conference on Distributed Computing Systems*, June 2009, pp. 450–457.

[11] L. Li, G. Xing, Q. Han, and L. Sun, "Asm: Adaptive voice stream multicast over low-power wireless networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 4, April 2012.

[12] M. Liu, Y. Wang, Z. Fan, and S. Zhang, "Voice communication based on zigbee wireless sensor networks," in *Proceedings of the 33rd Chinese Control Conference*, 2014, pp. 389–394.

[13] R. Mangharam, A. Rowe, R. Rajkumar, and R. Suzuki, "Voice over sensor networks," in *2006 27th IEEE International Real-Time Systems Symposium (RTSS'06)*, Dec 2006, pp. 291–302.

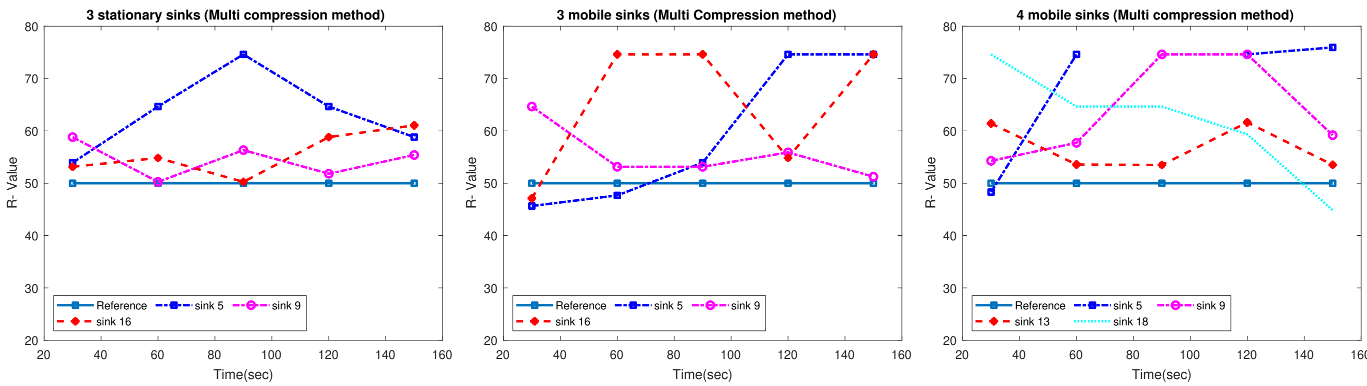



(a) 4 mobile sinks

(b) 5 mobile sinks

(c) 6 mobile sinks

Fig. 9: Performance of voice stream multicast to mobile sinks using modified ADPCM

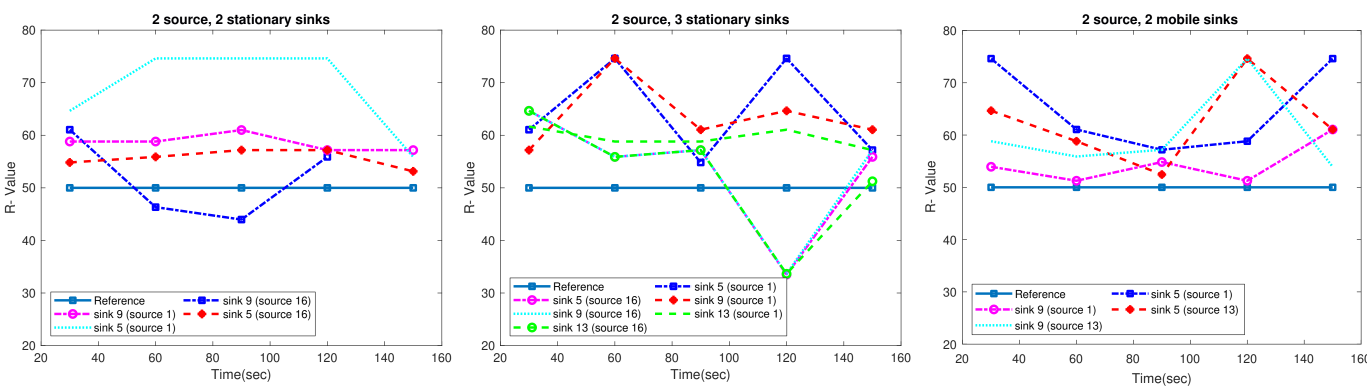



(a) 3 stationary sinks

(b) 3 mobile sinks

(c) 4 mobile sinks

Fig. 10: Performance of voice stream multicast to mobile sinks using ADPCM with multiple compression ratios




(a) 2 Sources, 2 stationary sinks

(b) 2 Source, 3 Stationary sinks

(c) 2 Source, 2 mobile sinks

Fig. 11: Performance of voice stream multicast with multiple sources and multiple sinks using modified ADPCM


[14] G. Rekha, Y. C. Adhikary, and L. Shrinivasan, "Modeling and simulation of voice communication over ad hoc network for search and rescue operations," in *2016 International Conference on Inventive Computation Technologies (ICICT)*, vol. 1, Aug 2016, pp. 1–6.

[15] H. M. P. Shekhar, M. A. A. Kumar, and K. S. Ramanatha, "Mobile agents aided multicast routing in mobile ad hoc networks," in *Proceedings of The 7th International Conference on Advanced Communication Technology (ICACT)*, Feb 2005.

[16] D. Sicignano, D. Tardioli, S. Cabrero, and J. L. Villarroel, "Real-time wireless multi-hop protocol in underground voice communication," *Ad Hoc Netw.*, vol. 11, no. 4, pp. 1484–1496, Jun. 2013. [Online]. Available: http://dx.doi.org/10.1016/j.adhoc.2011.01.017

[17] R. Zhang and I. Rubin, "Robust flow admission control and routing for mobile ad hoc networks," in *MILCOM 2006 - 2006 IEEE Military Communications conference*, Oct 2006, pp. 1–7.