

IoT-based sound-level control for audio amplifiers: mosques as a case study

Naeem Al-Oudat

Department of Communications and Computer Engineering, Tafila Technical University, Tafila, Jordan

Article Info

Article history:

Received Feb 1, 2021

Revised May 26, 2021

Accepted Jul 18, 2021

Keywords:

Azan level control

FFT for Masjids

IoT for Masjids

Nodes distance by sound

Sound-level control

ABSTRACT

When using audio-amplifiers in the open, uneven distribution of sound makes people unpleasant because it is loud or unheard. This unfortunate situation arises because audio-amplifiers volumes are set according to the guess of sound technicians. Mosques, as an example, are distributed inside wide areas and fire Azan five times a day. Due to the relatively long distances between them, speed and direction of the wind impose setting sound levels prior to each Azan such that all the area is covered and the overlap is minimized. In this paper, we propose a system based on internet of things (IoT) model to control the sound level of each mosque in the community. An IoT device (one in a mosque) sets the level of sound fired by the audio-amplifier. To do that, a synchronized series of tones is fired from each node. Once a node hears these tones, the process of sound level control starts to indicate the distances to heard nodes. As the approximate distances between nodes are known, each node can calculate its suitable sound level. Results showed that the proposed system is effective in setting sound levels for mosques audio amplifiers.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Naeem Al-Oudat

Department of Communications and Computer Engineering, Tafila Technical University

Aleis, Tafila 66110, Jordan

Email: naeemodat@ttu.edu.jo

1. INTRODUCTION

People tends to use loud speakers during festivals and large gatherings, in theaters, and inside large halls or outside in the open. However, careful placement of loud speakers and settings of sound levels are necessary for pleasant feelings of the audience especially in the open where weather conditions are not stable. Unfortunately, this is not the case, where some of the audiences are exposed to loud sound levels, from more than one speaker, while some of them barely hear the sound. For this reason, automatic and dynamic set of sound levels is necessary for people hearing health and audience overall comfort. In this paper, mosques, as a case study, are studied and a solution based on the internet of things (IoT) [1] is proposed. Mosques are everywhere in the Islamic world, but they are not connected. In Islamic world, mosques are sources of, at least, five Azan firings during a day. During a day, Azan is roughly fired every four hours, where environmental conditions witness some variations. These variations of environmental conditions impose setting the sound level of the audio amplifier at each mosque. Mosques, in this context, are things that need to be connected for the purpose of sound level control. In today's terms, IoT is a candidate for connecting these things. IoT model is used in a wide range of applications. Baucas and Spachos [2] proposed an IoT sensing framework that can be used in sound classification model. To realize sound in human machine interface, Cho *et al.* [3] used an IoT model. Many researchers used some forms of IoT models in their research [4]-[9].

Fabregat *et al.* [10] propose a system for acoustic source localization based on a two IoT receivers with microphones.

Prayers are done five times a day in predetermined and accurate times. Time of Azan (call to prayer), although it could be announced by other means like light from a light minarets [11], it should be the same in all mosques in a region for each prayer. When Azan is fired as a sound, it plays a critical rule in Muslim world as a mean of prosocial behavior [12]. Azan, which has a duration of around three minutes, should be unified to have a pleasant impact on people. Audio-amplifiers volume should be set to minimize overlap between adjacent mosques to mitigate Azan interference from different sources (mosques). The main aim of this research is proposing a system that set sound levels of audio amplifiers in a region, such that the whole region is covered and the overlapping is minimized.

The rest of this paper is organized as follows. Related work is discussed in section 2. Section 3 provides a potential way of connecting mosques in the community. Sound level measurement insight is given in section 4. In section 5, we give the details about the proposed system. A proof of concept is given in section 6. Section 7 concludes this paper.

2. RELATED WORK

Unifying Azan is an issue that were tackled in several regions by local authorities. Amman and Tafila, for instance, are two cities in Jordan, where Azan is broadcasted by a national radio station. Each mosque has a radio receiver tuned to the transmitter frequency and connected to the audio-amplifier. However, this solution is not reliable because it resembles a single point of failure where down time is large. This kind of unifying Azan is also found in Cairo (Egypt), Abu Dhabi (UAE). These examples suffer from the same single point of failure problem as well [13]. In the work of [13], authors proposed a framework for unifying call to prayers in the Islamic world. This work builds upon their work and provide an IoT design that can be used to control the volume of audio amplifiers in the mosques.

Using IoT to deal with sounds is not new. Shah *et al.* [14] proposed an IoT platform to gather sounds in homes and direct this recording to a classification server, which uses machine-learning techniques to identify several kinds of sounds such as explosion, and gunshot. By this, they claimed homes would be safe. This work lacks real-time feedback for the recorded sounds. In a patent [15], an IoT device is used as a proximity sensor that depends on the time difference between a wireless transmission and a sound transmission. However, this is only valid for indoor distances and for one IoT at a time. Zhang *et al.* [16] used an IoT platform to analyze the sound spectrum in forests to distinguish between crown and surface fires. Their work cannot be used to determine the distance of the fire. Patil [17] used an IoT platform to decide whether any car in a city crosses a noise threshold to report it to the authorities. This work lacks the distance dimension. Shah *et al.* [18] used an IoT platform along with artificial intelligence (AI) to classify noises in urban areas and log them into the cloud to be used when deciding to live in a quiet place. The authors in [19], used an IoT based system along with supervised learning algorithms to monitor noise levels in a city. However, the noise source cannot be identified using their proposed platforms.

In this research, a novel procedure for automating control of audio-amplifier sound volume is proposed. Tones with several known frequencies are monitored using fast Fourier transform (FFT) algorithm. Tones are fired repeatedly, starting with the highest-level then decreases in the subsequent firings with a preset delay between them. Each level is connected to a distance, therefore upon capturing this level the distance is known. After that, mosques are ready to fire Azan in a suitable level.

3. MOSQUES IN THE COMMUNITY

Enough number of mosques are found in a community where each one normally covers a circular area of around 500 meters radius. Several events are held in mosques besides five prayers, e.g., religious lectures by keynote speakers. Many people visit mosques around the clock to do their prayers or attend any other event. Any public announcement inside the mosque can quickly reach most of the community. The most important attribute of a mosque is being a source of a loud sound as an output of its audio-amplifier. A mosque is called a node from now on. Figure 1 shows the proposed connection of the nodes in a community. These nodes are connected by an intranet. The main part that keeps these nodes connected is an IoT device. These devices in this configuration use message queue telemetry transport (MQTT) protocol [20] to send/receive tokens from other nodes in the area. MQTT is a lightweight [21] transporting protocol on top of TCP/IP protocols that use publish/subscribe scheme for message exchange. Connections to the outside are going through the gateway. This configuration helps in limiting traffic on the network and helps in mitigating congestion during the sound level control.

To set the output sound level of a node i , the neighboring nodes monitor i 's tone level. Node i starts transmissions of its tone with a highest level, then starts to decrease this level gradually as a feedback from other nodes arrive at i [13]. This method of level control is time consuming since the decrease of node i tone level needs infinite number of steps. Therefore, a fast and practical method for level measurement is described in the next section.

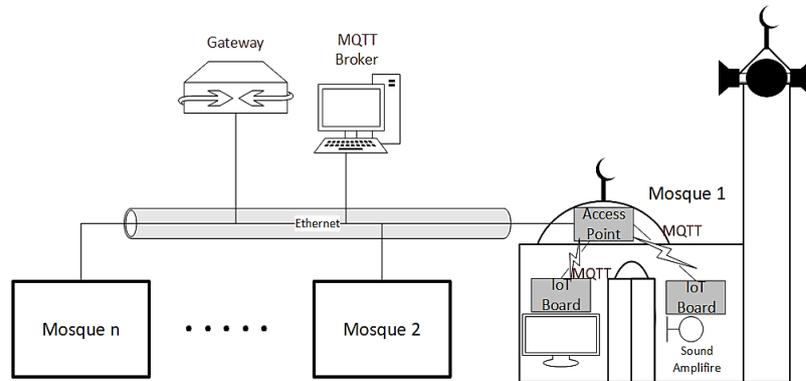


Figure 1. Mosques layout in an area, where they are connected through intranet

4. SOUND LEVEL MEASUREMENT

Each node has a known frequency that differs from others to avoid interference. Each node sends its tone in several levels until the suitable level is determined. This process happens approximately 20 seconds (actual time depends on the number of levels) prior to the actual event (Azan as an example). In the following, the details of this procedure, sound level set (SLSet), is provided, and the prototype is built, then the experimental results on the prototype is discussed.

This procedure starts, before the actual event. It takes around 20 seconds. SLSet depends on the availability of wireless connections between nodes. Once it is time to set the sound levels of the nodes, a trigger (by external source) starts this operation. The external trigger source could be the Azan timing board (found in all mosques). We assume that this trigger happens at the same exact time for all nodes. Neighbors of a node are the nodes, which help, determine its sound level. These neighbors are saved in the neighbor's table. Neighbor's table is established based on 2D Delaunay triangulation [6] according to physical locations of the nodes on ground. Under normal conditions (no wind, humidity-free air and no noise) a set of sound levels are connected to the distance to make calculations simple. Figure 2(a) shows an example of nodes under normal condition. The distance between different levels is equal and it can be connected to distance easily. In the figure, each level adds 50 m to the previous one, where the first level covers a circle of 50 m radius. As weather conditions (mainly wind direction and speed) changed, the distances between levels are also changed and hence the distances between nodes, from each node prospective, will be changed as shown in Figure 2(b) and (c).

Under normal conditions, a Fermat point [22] for the triangle of nodes can be calculated directly, then the distance from the Fermat point to each node is mapped to a suitable level. Let r_{iF} denotes the distance between node i and Fermat point F . Let each sound level covers a circle with radius that is greater than the previous one with D meters. The sound level for node i is calculated (1).

$$L_x = \left\lceil \frac{r_{iF}}{D} \right\rceil \quad (1)$$

To calculate the Fermat point, we need to find the logical distances between nodes. In this paper, we propose a system to find the distances between nodes under abnormal conditions, e.g., wind. The actual distances between nodes are around 500 m, where a sound tone needs around 1.5 seconds (speed of sound in the air is 343 m/s) to travel from one node to another. For each node to determine how far other nodes on the triangle vertices are, firing of a unique sound tone from each node is triggered at the same time.

If the sending nodes have equal firing distances and start firing tones of the same frequencies at the same time, then interference at the receiving nodes might occur. To solve this problem, nearby nodes should use different tone-frequencies. The problem of minimizing the number of used frequencies is similar to graph coloring problem [23], [24]. This problem is known to be NP-complete problem; therefore, researchers have

proposed greedy-algorithms to solve such problem. In our context, where there is a small number of nodes within the area under consideration, a manual assignment of tone frequencies could be done for each node. In particular, we force all nodes within the coverage area of any node (whether sharing a 2D Delaunay triangle [25] or not) to use different tone frequencies. Note that the frequencies used are in the audible spectrum up to 10 KHz. Then the distributed algorithm proceeds as follows:

- Before the sound firing by enough time, all nodes are triggered by an external source (Azan timing boards in mosques) to send their highest level tone.
- After a while (the time required to send and process tones), each node knows the logical distances to other nodes on the triangles vertices.
- At each node, the Fermat points (for all triangles the node is part of) are calculated. Based on that, the maximum distance between the node under consideration and Fermat points is considered its sound level.

In step 1, of the algorithm each node starts to determine the distances to the neighboring nodes. This is possible by integrating the tones firing and MQTT message passing between nodes. When time is right, each node fires its tone sound in the highest level. Sound levels are limited to some maximum number of levels. Each level is mapped to a distance. The distances are equally spaced (logarithmic spaced distances are also possible) to shorten distance-tuning time.

In step 2, the time needed to find the appropriate distance depends on several factors. It depends on the number of different levels transmitted by the source, the logical distance from the receiver (connected with physical distance and speed of wind) and the processing time on the receiver. To eliminate the interference that could happen when a node is firing its tone and receiving others tones, we restrict tones firing to be at the same time for all nodes in the area. Therefore, enough time will be available for sound propagation through the air and processing at the receiving nodes.

In step 3, as the previous two steps are completed, a simple calculations of the Fermat point for each triangle (formed from current node and other two nodes) is conducted. Hence, a maximum distance from current node to the Fermat points is connected to current node's sound-level. Figure 3 shows the timings of tone firing, recording of other tones and processing at one of the nodes.

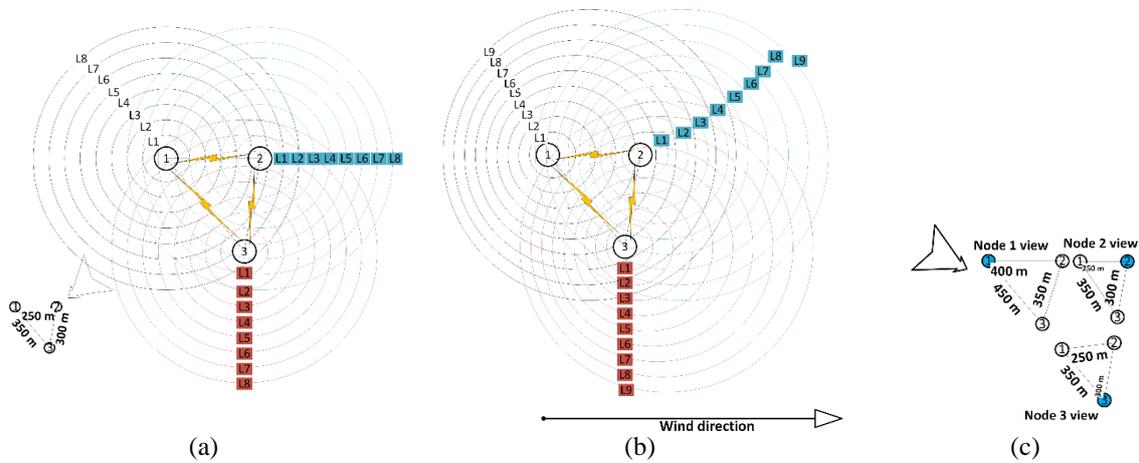


Figure 2. Sound levels for each node and their actual locations on ground: (a) when wind is calm (b) when there is a wind of direction indicated by the arrow, and (c) logical distances as seen by each node

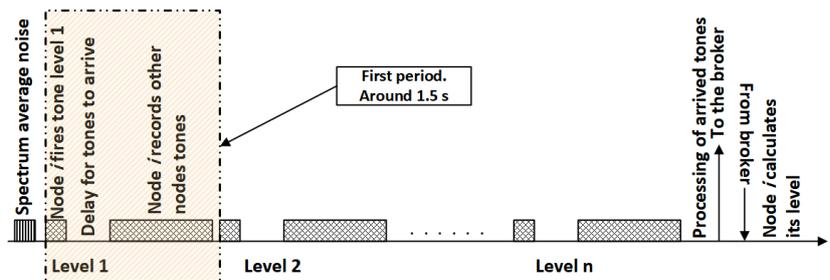


Figure 3. Timing diagram of sound tones received at node *i* from other nodes

5. SYSTEM DESIGN

The main component of the system is the IoT-sound-level-set device. These components are connected to each other's through MQTT protocol and using the Internet via a gateway to other parts. The software (procedure that sets the sound level) is also an important part of the system.

5.1. System architecture

Figure 4 shows the IoT-device connections to various parts available inside the node and the rest of the system. A source that triggers the start of sound-level-control process is the Azan timing board. This board is found in each mosque. Audio-amplifier is the second component in this system, which is also found in each mosque. It is assumed that the volume control of the audio-amplifier can be done through a signal from the IoTdevice. IoT-device is connected to the MQTT broker via Internet connection. MQTT broker is a central server shared between all mosques. Each node is a client (publisher and subscriber) in this network architecture.

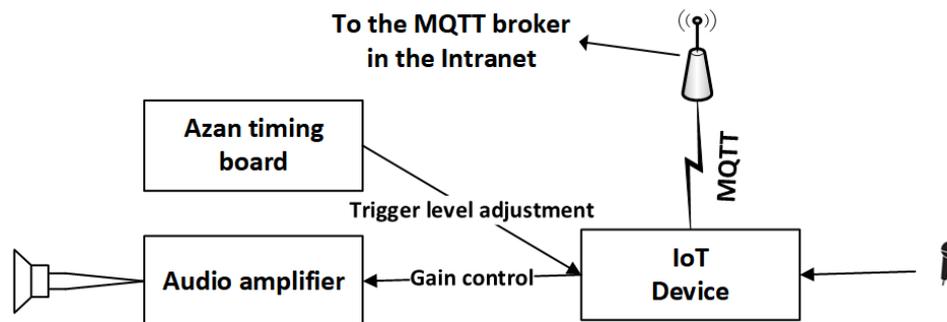


Figure 4. Hardware functional diagram

5.2. IoT-device

The device is mainly a controller that has a Wi-Fi connectivity. An ESP32 module (ESP32-WROOM-32) is used for its low price and a suitable computation power [26]. A microphone (INMP441 MEMS with I2S output [27]) is connected via an I2S bus. It outputs the volume control signal to the audio-amplifier. Azan timing board has an output that triggers the start of sound level set for each node.

5.3. System software

On each node, the same algorithm is running. This procedure depicted by the algorithm waits for appropriate event to start. Algorithm 1 represents the SLSet procedure, which runs about 20 seconds before Azan time. It initiates the sound level setup at each node. Each node has an array of size equal to the number of its neighbors that is initially filled with a maximum sound level. This array, by the end of the algorithm, will be filled with levels heard by the neighbor nodes. These levels represent the distances of the current node from its neighbors.

In line 4 of the algorithm, *tone_noise()* records the level of noise on each frequency of the neighbors. When actual sound level is captured for the neighbors, the logged level is subtracted from it to know the actual level of the captured sound. The core of this function is FFT [28], [29]. Lines 7-15 are repeated number-of-levels times. The block starts by firing the tone of this node in its highest level. A delay that is long enough to allow tones from neighboring nodes to arrive at the node under consideration. The *record_samples()* function records around one second length of samples for later processing. In lines 10-13, a loop over the recorded samples, taking a window at a time, starts to look for tones of the neighbors. In each iteration the array of levels are updated. In line 17, the array is published on the MQTT broker for the subscribers. In line 18, the messages (tones arrays) that a current node is subscribed to, are received from the broker once available (where each node publish its array of tones). Based on the received messages, which represent the levels of current node tones at their sides, this node starts to calculate the appropriate sound level. We encourage interested reader to refer to [13] for a complete discussion of this method.

Algorithm 1: Sound-Level-Set (SLSet) procedure, which runs at each node i to find its suitable sound level

Input : Adjacent nodes freq. $Af = \{f_0, f_1, \dots, f_n\}$
 Adjacent nodes IDs $Ai = \{ID_0, ID_1, \dots, ID_n\}$
Output: Current node sound level

```

Ln ← |Ai|;
level ← max_level;
As = {level, level, ...};
tone_noise();
t0 ← current_time();
repeat
    fire(level);
    delay();
    record_samples();
    While There are samples do
        find_tones();
        Update the corresponding level
        in As;
    end
    level --;
    t ← current_time();
until t > t0 + 1.5 × Ln
publish(As);
check_topics_update();
calculate_level();
    
```

6. PROOF OF CONCEPT

The most important component in this system is the IoT-device, Figure 5. This prototype is designed using off-the-shelf components. In this section, the design of the IoT-device is shown and some results that prove the concept are discussed.

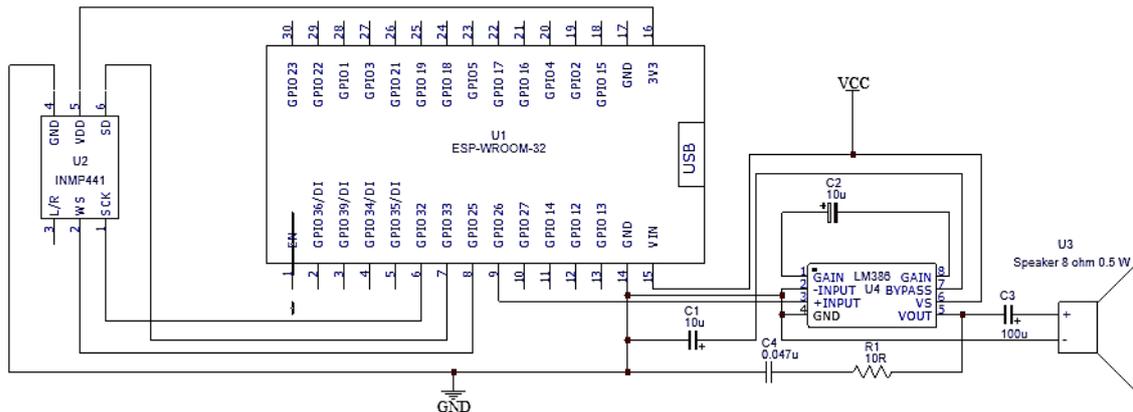


Figure 5. IoT-sound-set-device prototype [30]

6.1. Test structure

To prove that, the proposed system is working as expected, we have deployed two nodes, a mosquitto broker [31] and a monitor node. Nodes are identical except that, the monitor node is only a subscriber to other nodes topics. When the monitor node got the messages, it displayed them, using the serial monitor, on Arduino IDE. To emulate the timing trigger, which is generated by Azan timing board in the actual system, we used a push button. Once this button is pressed, both nodes start control procedure simultaneously. The test structure is shown in Figure 6. The software used for testing is on <https://github.com/DrNaeemOdat/IoT> for Masjids

6.2. Results

Sound levels can be changed by two ways; by varying the volume of the audio amplifiers or by moving the sources themselves apart from each other's. These nodes use simple audio amplifiers, as an emulation of the actual audio amplifiers in the mosques. A simple audio amplifier in this configuration is a single-sound-level source. Therefore, we varied the distances between nodes to have the effect of different levels with respect to each node. Two distances were considered; 3 meters and 9 meters, Figure 7. The highest level, level 1, is represented by the 3-meter distance. The lowest level, level 5, is represented by the 9-meter distance.

The tone (8305 Hz) is transmitted five times by node ID0. Another tone with different frequency is transmitted by node ID1. The length of the tone is approximately 140 ms. This is repeated every second for a length of five seconds. Then the one-second length spectrum is analyzed using the FFT algorithm. The analysis continues, taking "200 ms"-chunks at a time. The sound threshold that we used is 20000 (=20000X3.3e-6 V).

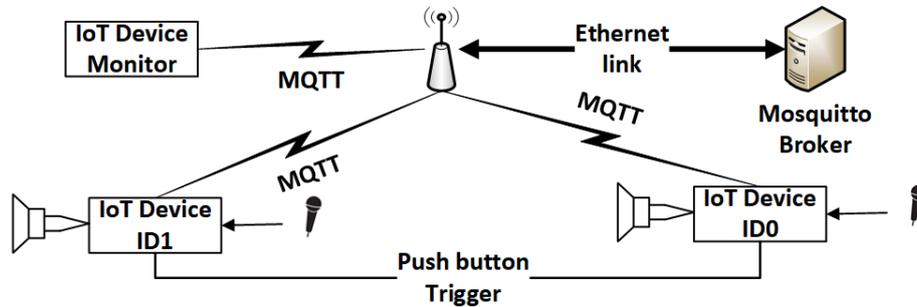


Figure 6. System test structure of two nodes, mosquito broker and monitor node

For the 3-meter-distance the sound levels captured at one of the nodes are shown in Figure 7(a). One bar is above the threshold, which means that the sound level is heard at this node. The monitor node printed out the levels at both nodes:

	ID0	ID1
1) Message arrived on topic: ID0. Message:	1	1
2) Message arrived on topic: ID1. Message:	1	1

For the 9-meter-distance the sound levels captured were lower, Figure 7(b). The levels are below 20000. Therefore the recorded levels by the monitor node are:

	ID0	ID1
1) Message arrived on topic: ID0. Message:	1	5
2) Message arrived on topic: ID1. Message:	5	1

Note that, each node hears its sound with a highest level (level 1). From these results, we conclude that this procedure can help in setting the sound levels of the nodes under variable weather conditions. Weather conditions (wind direction and speed) weaken or strengthen the sound heard at the receiver. Hence, brings the impression of sound source distance (near or far away) at the receiver that is not the actual physical distance. Therefore, the results are correct because it depends on the heard sound to produce the distance from the source.

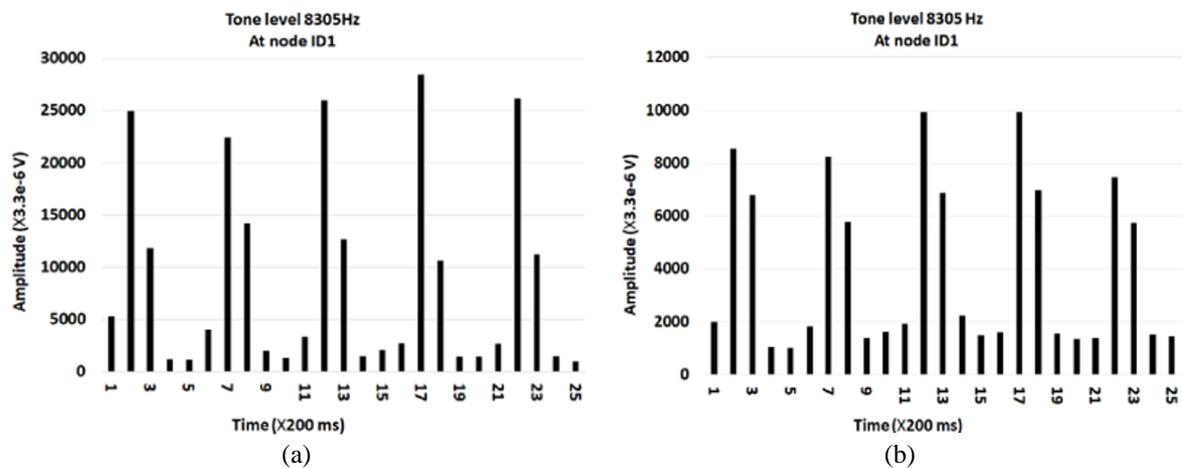


Figure 7. Sound levels of node ID0 at node ID1: (a) 3 m apart, (b) 9 m apart

7. CONCLUSION

In this paper, an IoT-device to capture tones of known frequencies is designed. Then a novel procedure to approximate the distance between mosques is proposed and built on this IoT-device. This system was tested in a test structure built for this purpose. Results showed that it is possible to set the sound level of mosque audio amplifiers for the sake of community comfort. For this system to be effective, the audio-amplifier in mosques should support several sound levels. Further, this solution is not immune to sound-interference, which has the same frequencies of the nodes during the sound-level-setup procedure. As a future work, we intend to integrate the proposed work with Azan streaming to bring about a complete solution for unified call to prayers.

REFERENCES

- [1] K. Shafique, B. A. Khawaja, F. Sabir, S. Qazi, and M. Mustaqim, "Internet of things (IoT) for nextgeneration smart systems: A review of current challenges, future trends and prospects for emerging 5G-IoT scenarios," *IEEE Access*, vol. 8, pp. 23022-23040, 2020, doi: 10.1109/ACCESS.2020.2970118.
- [2] M. J. Baucas and P. Spachos, "A scalable iot-fog framework for urban sound sensing," *Computer Communications*, vol. 153, pp. 302-310, 2020, doi: 10.1016/j.comcom.2020.02.012.
- [3] S. Cho *et al.*, "Highly stretchable sound-indisply electronics based on strain-insensitive metallic nanonetworks," *Advanced Science*, vol. 8, no. 1, 2021, Art. no. 2001647, doi: 10.1002/adv.202001647.
- [4] H. S. Parihar *et al.*, "IoT-based ambiance monitoring system," *Advances in Mechanical Engineering*, pp. 793-798, 2021, doi: 10.1007/978-981-15-3639-7_95.
- [5] K. Sanghavi and A. Rajurkar, "Early detection of grape stem borer using IoT," *Next Generation Information Processing System*, vol. 1162, pp. 203-212, 2021, doi: 10.1007/978-981-15-4851-2_22.
- [6] S. Uma and R. Eswari, "Accident prevention and safety assistance using iot and machine learning," *Journal of Reliable Intelligent Environments*, pp. 1-25, 2021, doi: 10.1007/s40860-021-00136-3.
- [7] A. Abdulkareem, T. E. Somefun, O. K. Chinedum, and F. Agbetuyi, "Design and implementation of speech recognition system integrated with internet of things," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 2, pp. 1796-1803, 2021, doi: 10.11591/ijece.v11i2.pp1796-1803.
- [8] S. Priyadharshini, C. Subramani, and J. P. Roselyn, "An IoT based smart metering development for energy management system," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 4, pp. 3041-3050, 2019, doi: 10.11591/ijece.v9i4.pp3041-3050.
- [9] K. Bhagchandani and D. P. Augustine, "IoT based heart monitoring and alerting system with cloud computing and managing the traffic for an ambulance in india," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 6, pp. 5068-5074, 2019, doi: 10.11591/ijece.v9i6.pp5068-5074.
- [10] G. Fabregat, J. A. Belloch, J. M. Bad'ia, and M. Cobos, "Design and implementation of acoustic source localization on a low-cost IoT edge platform," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 12, pp. 3547-3551, 2020, doi: 10.1109/TCSII.2020.2986296.
- [11] P. T. Arab, "A minaret of light": transducing the islamic call to prayer?," *Material Religion*, vol. 11, no. 2, pp. 136-163, 2015, doi: 10.1080/17432200.2015.1059126.
- [12] E. P. Duhaime, "Is the call to prayer a call to cooperate? a field experiment on the impact of religious salience on prosocial behavior," *Judgment and Decision making*, vol. 10, no. 6, pp. 593-596, 2015.
- [13] N. Al-Oudat and A. I. Alshbatat, "A unified call-to-prayer framework in muslim world," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 4, no. 3, pp. 5758-3766, 2014, doi: 10.11591/ijece.v4i3.5758.
- [14] S. K. Shah, Z. Tariq, and Y. Lee, "Audio iot analytics for home automation safety," *2018 IEEE International Conference on Big Data (Big Data)*, 2018, pp. 5181-5186, doi: 10.1109/BigData.2018.8622587.
- [15] G. Burns, M. Lioy, and E. Rongo, "Proximity detection of internet of things (IoT) devices using sound chirps," *uS Patent 9,438,440*, 2016.
- [16] S. Zhang, D. Gao, H. Lin, and Q. Sun, "Wildfire detection using sound spectrum analysis based on the internet of things," *Sensors*, vol. 19, no. 23, 2019, Art. no. 5093, doi: 10.3390/s19235093.
- [17] P. Patil, "Smart iot based system for vehicle noise and pollution monitoring," *2017 International Conference on Trends in Electronics and Informatics (ICEI)*, 2017, pp. 322-326, doi: 10.1109/ICOEI.2017.8300941.
- [18] S. K. Shah, Z. Tariq, and Y. Lee, "IoT based urban noise monitoring in deep learning using historical reports," *2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 4179-4184, doi: 10.1109/BigData47090.2019.9006176.
- [19] L. Monti, M. Vincenzi, S. Mirri, G. Pau, and P. Salomoni, "Raveguard: A noise monitoring platform using low-end microphones and machine learning," *Sensors*, vol. 20, no. 19, 2020, Art. no. 5583, doi: 10.3390/s20195583.
- [20] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, "MQTT-S-A publish/subscribe protocol for wireless sensor networks," *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE'08)*, 2008, pp. 791-798, doi: 10.1109/COMSWA.2008.4554519.
- [21] N. Naik, "Choice of effective messaging protocols for IoT systems: MQTT, COAP, AMQP AND HTTP," *2017 IEEE international systems engineering symposium (ISSE)*, 2017, pp. 1-7, doi: 10.1109/SysEng.2017.8088251.
- [22] M. Hajja, "An advanced calculus approach to finding the fermat point," *Mathematics Magazine*, vol. 67, no. 1, pp. 29-34, 1994, doi: 10.2307/2690551.
- [23] D. W. Matula, G. Marble, and J. D. Isaacson, "Graph coloring algorithms," *Graph theory and computing*, pp. 109-122, 1972, doi: 10.1016/B978-1-4832-3187-7.50015-5.
- [24] F. T. Leighton, "A graph coloring algorithm for large scheduling problems," *Journal of research of the national Bureau of standards*, vol. 84, no. 6, pp. 489-506, 1979.
- [25] J. R. Shewchuk, "Triangle: Engineering a 2d quality mesh generator and delaunay triangulator," *Workshop on Applied Computational Geometry*, vol. 1148, 2005, pp. 203-222, doi: 10.1007/BFb0014497.
- [26] "esp32-wroom-32_datasheet_en," *Espressif.com*. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf (accessed Jan. 27, 2021).

- [27] “Omnidirectional Microphone with Bottom Port and I2S Digital Output,” *INMP441, InvenSense*, 2014. [Online]. Available <https://invensense.tdk.com/wp-content/uploads/2015/02/INMP441.pdf> (accessed Jan. 27, 2021).
- [28] H. J. Nussbaumer, “The fast fourier transform and Convolution Algorithms,” *Springer-Verlag Berlin Heidelberg*, 1981, pp. 80-111, doi: 10.1007/978-3-642-81897-4.
- [29] C. Rader and N. Brenner, “A new principle for fast fourier transformation,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 3, pp. 264-266, 1976, doi: 10.1109/TASSP.1976.1162805.
- [30] *EasyEDA(standard) - a simple and powerful electronic circuit design tool*. 2021. Easyeda.STD. accessed Jan 27, 2021. [Online]. Available: <https://easyeda.com/editor>
- [31] R. A. Light, “Mosquitto: server and client implementation of the mqtt protocol,” *Journal of Open Source Software*, vol. 2, no. 13, pp. 265-266, 2017, doi: 10.21105/joss.00265.

BIOGRAPHY OF AUTHOR



Naeem Al-Oudat    is an assistant professor at Communications and Computer Engineering Department, Tafila Technical University, Jordan. He obtained Bachelor Degree in Communications Engineering from Muta University (Jordan) in 1992. He obtained Master Degree in Computer Engineering from Yarmouk University (Jordan) in 2004. He obtained PhD Degree in Computer Engineering from Iowa State University (USA) in 2012. His research interests are in the fields of real-time distributed systems, educational operating system design, AI, and IoT. He teaches several bachelor-level courses for electrical and computer engineering students. He can be contacted at email: naeemodat@ttu.edu.jo.