SkopEdge: A Traffic-Aware Edge-Based Remote Auscultation Monitor

Pallav Kumar Deb*, *Student Member, IEEE*, Sudip Misra*, *Senior Member, IEEE*, Anandarup Mukherjee*, *Student Member, IEEE*, Abbas Jamalipour†, *Fellow, IEEE*
*Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur
Kharagpur 721302, India
†School of Electrical and Information Engineering, The University of Sydney, NSW 2006, Australia
Email: pallv.deb@gmail.com, sudip_misra@yahoo.com, anandarupmukherjee@ieee.org, abbas.jamalipour@sydney.edu.au

***Abstract*—In this paper, we develop and analyze a smart digital stethoscope – SkopEdge – to provide reliable remote e-health monitoring with a minimum delay while enhancing overall network performance. SkopEdge initially records the heart sounds from individuals and then senses the quality of the network. Depending on the network traffic, SkopEdge converts the audio clip into an appropriate format before transferring it to remote locations for estimating the number of heartbeats and storage. Towards this, we formulate the link quality along with SkopEdge's current configuration as a Markov Decision Process (MDP) with actions as conversion format selection. The remote server then returns the result, which SkopEdge displays on its screen. Real-time implementations show that SkopEdge works efficiently in all network conditions. Further, audio conversions usually degrade the quality of sound, but our proposed system does not change its primary components. Although SkopEdge exhibits an increase in energy consumption by 79% while converting to lower-quality formats, it also reduces the energy consumption by 99% while transmitting the same, which subsequently results in energy savings. Further, we provide an analysis of the estimated heartbeats in an audio clip by SkopEdge.**

***Index Terms*—Digital stethoscope, Internet of Things, Markov decision process, network traffic, auscultation sounds, Edge Processing.**

I. Introduction

The proliferation of smart ubiquitous devices has improved the quality of healthcare by bridging the gap between IoT and e-Health monitoring systems. However, they exhaust network resources, which often leads to unreliable transmissions. Suggestive healthcare devices use the same standard ISM radio bands for data transmission. Researchers have made significant efforts to deliver reliable transmission of health data over such congested networks [1]. However, healthcare systems working with multimedia data suffer from over buffering, slow servers, video/audio latency, packet drops, among others, while delivering at remote locations. Towards this, we aim to develop a network traffic-aware e-health device that helps in regular monitoring of the heart in near real-time.

In this work, we design a smart digital stethoscope (SkopEdge) for remotely monitoring an individual's heart sounds. *Skop* comes from the Greek equivalent of an instrument for viewing, and *Edge* comes from Edge Computing. Fig. 1 illustrates how the SkopEdge system works. SkopEdge consists of four phases – record and filter, link analysis and conversion (lossy/lossless), send converted data and receive reports. In the first phase, an individual records his/her heartbeats, which passes through a band-pass filter for removing noise. In the second phase, for reliable transmission of the captured sound, SkopEdge decides the format of the audio clips for sending based on the quality of the network. Such conversion of audio formats helps in reducing the size of the file, which minimizes the delay and makes the system near real-time and also saves energy. In the third phase, SkopEdge sends the converted data to a remote server for analysis and report generation. Finally, SkopEdge displays the number of heartbeats returned by the server on its screen. The results and audio samples of the individuals are stored at the remote server for review by concerned doctors. In case there is no network, SkopEdge locally computes the number of heartbeats. We also develop an Android application for computing the number of heartbeats on a smartphone. Our major contributions in this work include – 1) development of a traffic-aware smart digital stethoscope (SkopEdge), which changes audio formats depending on the network quality, 2) formulation of a Markov Decision Process (MDP) for stochastically deciding the transmission format, and 3) composition of routines that estimate the number of heartbeats from the recorded audio clips.

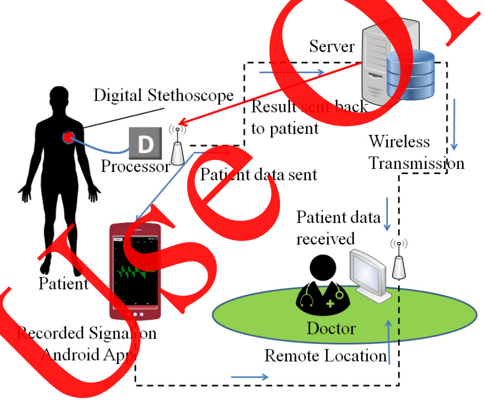


Figure 1: Overview of SkopEdge's system architecture

A. Traffic-Aware SkopEdge for Healthcare

There has been significant efforts by researchers towards reducing network traffic in IoT environments by developing new schemes for data offloading. To reduce latencies and network traffic, solutions such as caching, based on various parameters such as social interest, mobility, traffic redundancy, and others are common in literature [2]. However, healthcare data do not depend on repetition. Inferences from generated data at every instant are of paramount importance. In summary, with the increasing demand for e-health solutions, traffic-aware healthcare devices (SkopEdge and others) facilitate in delivering physiological data without fail.

B. Motivation

Suggestive healthcare devices mandate reliable transmission of data over the network in near real-time. However, with the increasing number of devices, the network gets congested. Regular checkups are of paramount importance for maintaining a healthy body, and analysis of auscultation sounds of the heart is standard preliminary practice. However, devices dealing with multimedia in such congested network traffic usually face difficulty while sending data to remote locations. This acts as a motivation for designing SkopEdge as a remote e-health monitoring device that efficiently uses network and energy resources while delivering auscultation sounds along with the number of heartbeats in near real-time.

II. Related Work

In this section, we initially present how researchers are the bridging of the gap between e-Health and the Internet of Things (IoT). We then discuss some of the existing approaches towards the development of smart stethoscopes.

Zhu *et al.* [3] highlighted the challenges faced by the Ambient Assisted Living (AAL) research community while developing various deployable systems for sensing, processing, and sending results to the end-users. For a seamless integration of diverse e-Health monitoring technologies, applications, and services, Benharref and Serhani [4] proposed a framework consisting of a Service Oriented Architecture (SOA) and the cloud. The framework contemplates the random network disconnections along with the resource-constrained nature of the mobile devices for smooth collection and communication of vital data from wearable biosensors. Apart from digital stethoscopes, Lee *et al.* [5] proposed a real-time data compression scheme for a patient's electrocardiogram (ECG) data as well as an algorithm for its transmission over the network. Researchers are developing many other e-health devices towards which Guo *et al.* [6] proposed and developed a software-based platform for facilitating interactivities among them.

Elhilali and West [7] developed a smart stethoscope for detecting pneumonia. The authors devised algorithms for the device that incorporates noise cancellation on the captured sound and then uses Artificial Intelligence (AI) techniques for differentiating abnormal behavior from the normal ones. Similarly, Perron *et al.* [8] developed a Cardio-Pulmonary Stethoscope (CPS) for measuring and analyzing fluid in the lungs. A mobile device receives the computed results for remote monitoring. The authors also presented an evaluation of the accuracy achieved by the developed CPS on volunteers (patients) [9]. Chen *et al.* [10] proposed a deep neural network

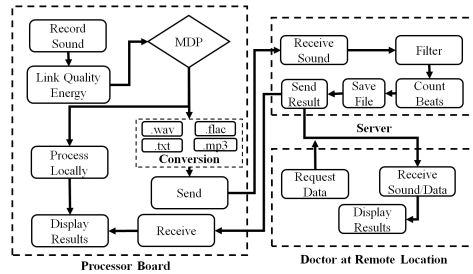


Figure 2: Block diagram of SkopEdge system

(DNN) method for identifying S1 and S2 sounds on captured heart signals. Pan *et al.* [11] assessed the Korotkoff sounds captured from a stethoscope while measuring BP using Convolutional Neural Network (CNN).

Synthesis: Although data offloading is a matured field of study, we observe that there exists a lacuna in how a device with non repetitive data behaves in congested networks. Existing schemes limit themselves to sending the data in its original form. However, converting the data format according to the network quality delivers the results in near real-time, which saves considerable energy during transmission, leading to sustainable IoT and Edge-based e-health systems.

III. System Model

In this section, we briefly discuss our system, its architecture, concepts of MDP, and how SkopEdge utilizes them to decide and take actions under different network conditions.

A. SkopEdge System

SkopEdge consists of an auxiliary cable with a Microelectromechanical System (MEMS) microphone connected on the diaphragm's end and the other end attached to a processing board. In our work, we use a Raspberry Pi as the processor board since SkopEdge requires Wi-Fi to send data to the remote server for storage and processing. SkopEdge also has an Organic Light-Emitting Diode (OLED) screen connected to the processor board for displaying notifications and results. The resolution of our captured audio has 2^{16} discrete levels as the Raspberry Pi represents each sample using 16 bits *(bits per sample, b)*. The digital value, D_t^{dgt} corresponding to the analog value captured at time instant t is $D_t^{dgt} = (ADC_{res}/V_{max}^{system}) \times V_t^{msrd}$ where, ADC_{res} is the resolution of the ADC sound card, V_{max}^{system} is the voltage obtained from the USB port (generally $5V$), and V_t^{msrd} is the acquired analog signal voltage. The 16 bit resolution enables us to record with a sampling rate ($\mathbf{S}$) of $44100Hz$, which helps in capturing detailed auscultation sounds. Thus, the bit rate $\mathcal{B}_f^r$ of the recorded file (f) is calculated as $\mathcal{B}_f^r = b \times \mathbf{S} \times n_{ch}$ where, n_{ch} is the number of channels. Finally, the size of the recorded file $\mathcal{F}_f^{size}$ (in Bytes) is: $\mathcal{F}_f^{size} = (\mathcal{B}_f^r \times t_t^{rec})/8$

B. SkopEdge System Architecture

As outlined in Fig. 1, SkopEdge records auscultation sounds from an individual/subject and sends the data to a cloud server. Since the heart sounds have a specific frequency range, we do not need to analyze the whole spectrum. Instead, we pass the recorded sound through a *Butterworth* bandpass filter and extract the relevant range. Additionally, to avoid unreliable data transmission, SkopEdge checks its residual energy and the link quality, which acts as inputs to an MDP. The decision from MDP directs the board to convert the file into one of the formats *(.wav, .flac, .mp3,* and *.txt)* or process locally. On the server-side, when the data is received, it filters the relevant spectrum and then estimates the number of beats, and sends the result back to the processor board. For counting the number of peaks, we compute the *baseline* and *centroid* of the audio clip. We then find the peaks in the specified data using the *Gaussian fitting* function, $f(x) = Ae^{-\frac{(x-\mathbf{b})^2}{2\mathbf{c}^2}}$ for constants $A, \mathbf{b}$, and non zero $\mathbf{c}$. We then enhance our resolution by again using *Gaussian fitting* and further run a centroid computation routine on neighboring peaks. This time, we perform a *Lorentzian fitting* $(g(x) = \frac{1}{1+x^2})$ to ensure precise detection of the peaks and finally scale the result. Concerned doctors can access the recorded sounds and results from the server whenever needed. In the case of poor network quality, the processor board runs the same routine as the server. SkopEdge only sends the peak locations to the server in the form of an array in a .txt file. SkopEdge follows Algorithm 1 for choosing the audio formats and Algorithm 2 for estimating the number of heartbeats.

Algorithm 1: SkopEdge Decision

Result: No. of heartbeats
Input: ψ, E_t^{res};
$\mathcal{D}$: MDP decision to convert or process locally;
$\mathcal{E}$: Conversion format;
if $\mathcal{D}$ = *convert* **then**
    convert to $\mathcal{E}$;
else
    Process locally;
    Estimate number of heartbeats (Run Algorithm 2);
end
Wait for result;
Display result;

Algorithm 2: Estimate Number of Heartbeats

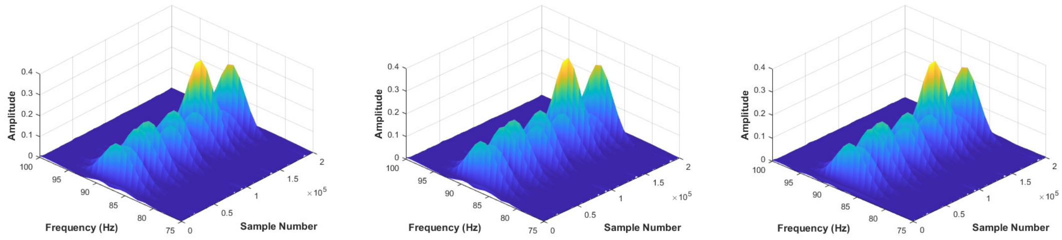
Result: No. of heartbeats
Input: Receive data from SkopEdge;
if *Input is audio file* **then**
    Filter audio;
    Run peak detection routine;
    Save file and result;
    Send result back to SkopEdge;
else
    Save received data;
end

C. Audio Conversions

Although we use command-line applications for converting the audio formats in SkopEdge, we briefly explain the audio conversion techniques. The .wav is an uncompressed audio format that needs a sizeable space. The .flac format stands for lossless conversion with almost 50% reduction in size than the original .wav. It identifies redundant data and replaces it with shorter unique symbols. However, due to the prediction of redundant data, the reconversion of .flac files to .wav with non-repeating rhythms renders inaccurate audio. However, the .mp3 is a lossy audio format that relies on perceptual noise shaping, implying that it removes parts from the audio file that are relatively silent. As conversion to .mp3 format drops significant data, it reduces file size as well as its quality.

D. Markov Decision Process for SkopEdge decisions

In this work, we consider only one device and demonstrate its characteristics. We plan to deploy a network of multiple SkopEdge devices and study their behaviors in the future. Currently, we assume that the network is already in use by other user devices and applications and we represent its quality as ψ. Let $\mathcal{S}$ represent our SkopEdge, and the link available for it be $\mathcal{L}$. The different terminologies considered in this work are: *Available Bandwidth* (BW_t^a): The rate at which SkopEdge transmits data. We calculate this using the Shannon Capacity Formula $BW_t^a = BW^{tot} log_2(1+S(t)/N(t))$, where BW^{tot} is the total bandwidth of the channel, $S(t)$ is the average received signal power at time instant t, and $N(t)$ is average power of noise and interference in the channel at time instant t. *Transmission Time* (t_t^{trans}): The time needed for SkopEdge to push data into the channel at time instant t. We calculate this as the ratio of the size of data to be sent divided by the available bandwidth at that time instant. Mathematically, $t_t^{trans} = \frac{P_t^{size}}{BW_t^a}$ where, P_t^{size} is size of data to be sent at time instant t and BW_t^a is the available bandwidth (BW_t^a) at time instant t. *Propagation Time* (t_t^{prop}): The time needed by a packet to reach the destination at time instant t. We compute this by performing a *ping test* from our processor board. We calculate t_t^{prop} from the parameter *time* in its output represented as t_t^{ping} for $P_{ping}^{size} Bytes$ of data as: $t_t^{prop} = (t_t^{prop}/P_{ping}^{size}) \times P_t^{size} \times n_{packets}$ where P_t^{size} is the size of each packet at time instant t and $n_{packets}$ is the number of packets to be sent. *Nodal Processing Delay* (t_t^{node}): The time needed by the processor board to record the auscultation sounds (t_t^{rec}) and process the data in a file *(including conversion* and *sending)* at time instant t. We calculate this as the ratio of the number of cycles required by the file divided by the computation power *(cycles per second)* of the board. Mathematically, $t_t^{node} = t_t^{rec} + (P_f^{cycles}/C_t^{cylces})$ where P_f^{cycles} is the number of cycles needed by a file f and C_t^{cylces} is the processing power of the board. *Total Delay* (t_t^{tot}): The overall time needed by a packet to get delivered at the destination at time instant t. Mathematically, $t_t^{tot} = t_t^{trans} + t_t^{prop} + t_t^{node}$. *Link Quality* (ψ_t): We define (ψ_t) as a function of t_t^{trans} and t_t^{prop} for successful delivery



(a) Format: .wav (b) Format: flac (c) Format: mp3

Figure 3: Continuous wavelet transforms for different formats after conversion

of SkopEdge's packets at time instant t. We calculate this as a weighted average $t_t^{avg} = \frac{w_1 t_t^{trans} + w_2 t_t^{prop}}{w_1 + w_2}$ and check its deviation from $\psi_t = 1 - (1/t_t^{avg})$. *Residual Energy* ($E_{t+1}^{res}$): The Residual Energy after performing the operations at at time instant t, E_{t+1}^{res} is the processor board's remaining energy after performing the operations of recording the auscultation sound, converting and sending the recorded file, and receiving the result. Mathematically, $E_{t+1}^{res} = E_t^{res} - (E_t^{rec} + E_t^{node} + E_t^{recv} + E_t^{idle})$ such that $E_t^x = E_{req}^x \times t_t^x$ where $x = [rec, node, recv, idle]$.

To optimize the usage of *Energy*, we design an MDP to take decisions on appropriate actions, such that a state opts for multiple actions but not at the same time. We use two tuples for defining our states. They are defined by *link quality* and *Residual Energy*, which is represented as $< \psi_t, E_t^{res} >$. The actions in SkopEdge is the selection of the format in which the file is to be sent. Thus, our Markov chain is a sequence of states $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, ..., \mathcal{S}_n$ abiding by the property of being *memoryless*. This is satisfied by the Markov property which is mathematically represented as $P(\mathcal{S}_{n+1} = x|\mathcal{S}_1 = x_1, \mathcal{S}_2 = x_2, ..., \mathcal{S}_n = x_n) = P(\mathcal{S}_{n+1} = x|\mathcal{S}_n = x_n)$, for $n = 0, 1, 2, ...$, and so on [12]. Similarly, the *m-step transition probabilities* is defined as $P_{ij}^m = P(\mathcal{S}_{n+m} = j|\mathcal{S}_n = i)$. In case m is very large, we define a limiting probability independent of the initial state (i) [13]. This is also termed as *steady-state probability* which is mathematically represented as: $\lim_{x\to\infty} P_{ij}^m = \eta_j > 0$ where j is state at which the system is expected to be in after large number of transitions, η_j is the steady-state probability of state j such that $\eta_j = \sum_{i=0}^m \eta_i P_{ij}$ and $\sum_{j=0}^m \eta_j = 1$.

With respect to the model defined above, the *State-Decision Probability Matrix* ($A_{t,ik}$), *we compute the State-Decision Cost Matrix* ($\delta_{t,ik}$), and *State Transition Probability Matrix* ($P_{t,ik}$) where k is the decision taken. Based on these matrices, SkopEdge takes decisions while maximizing E_t^{res}.

IV. Performance Evaluation

In this section, we discuss the results exhibited by SkopEdge. We first describe the metrics used for evaluation and then elaborate on the results.

A. Metrics

In order to analyze the performance of SkopEdge, we consider the following metrics: 1) *File Formats:* SkopEdge initially records the audio clips in .wav format, which is of high resolution and size. On the other hand, the .flac is a lossless format very similar to .mp3, which is lossy. We analyze the variations in the continuous wavelet transform (CWT) in the case of each of the formats. 2) *File Size:* With each of the formats mentioned above and their resolutions, we analyze their sizes. 3) *Traffic Pattern:* Since SkopEdge is mobile, we analyze the various network condition throughout the day and at different locations. Decisions in the MDP are made based on these conditions and energy. 4) *Energy dissipated for conversion:* SkopEdge initially records the auscultation sounds in a high-quality audio format, and based on the network quality, it converts the audio clip into a format with lower resolution. Since the conversion technique varies with the determined format, we analyze the energy required during this process. 5) *Transmission Energy:* We analyze the energy spent by SkopEdge while sending data to the server.

B. Results & Discussion

In this section, we briefly discuss about the results obtained as a consequence of developing SkopEdge.

File Formats: As mentioned earlier, SkopEdge converts its audio files into both lossless and lossy formats according to the conditions of the network traffic. In Fig. 3, we assess the CWT for an audio clip in different formats. The CWT for the original format (.wav) in Fig. [ref wave] clearly shows the heartbeats (peaks) around 87.5 Hz center frequency. We observe similar CWT results in the case of both .flac and .mp3 formats in Figs. 3(b) and 3(c) with no change in the amplitude. We also observe that the conversions do not introduce undesired noise into the audio clips. We can thus safely conclude that although the conversions affect the quality of sound, they do not affect our results in estimating the number of heartbeats.

File Size: Fig. 4 depicts the sizes of the files generated after conversion. We observe that .flac and .mp3 are much smaller in size than that of the original recorded .wav file. While .flac is a lossless conversion, the resolution is much lower than .wav files, which is why the size also reduces significantly. The .mp3 is a lossy conversion which discards irrelevant content from the original audio clip, which results in a further decrease of file size. The quality of the audio degrades in this case. We also observe that in case we send only the peak data as a .txt file, the size is minuscule. SkopEdge does not send any audio

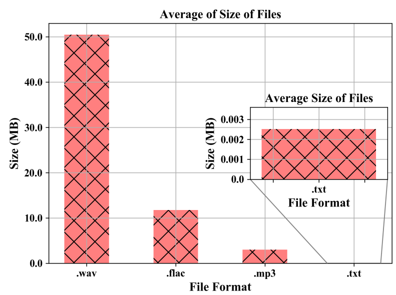


Figure 4: Size of captured audio in different formats

in the case of .txt. However, we can visualize the number of beats captured in the remote location.

Heartbeat Sample: In Fig. 5, we show a sample result from SkopEdge on an audio clip of 10 seconds. The top block is the original recorded sound. We then obtain the signal in the second layer by filtering the sound. The plot on the third layer is the peak detection. We observe that SkopEdge detects 13 beats, which approximately equals 78 beats per minute. The plot at the bottom represents the deviation in times between two consecutive captured heartbeats.

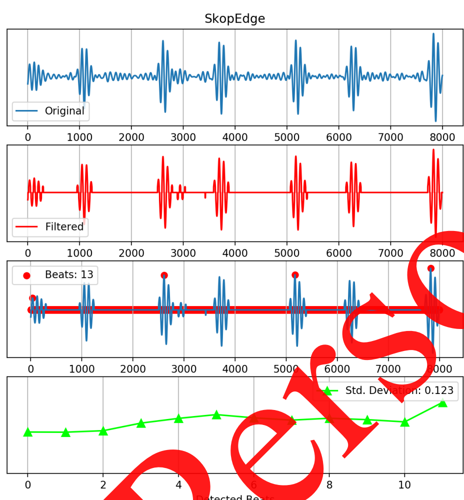


Figure 5: Sample heartbeat report of an individual

Network Traffic: We analyze the network condition between SkopEdge and the remote server from different locations throughout the day. Fig. 6 shows the time needed for a $64Byte$ packet to reach the destination in log(ms) during ping tests. This plot gives us an idea regarding the times when the network remains congested. In the figure, we observe heavy traffic at 22:30 hours.

Total Energy Consumption: The conversion from one format to the other requires energy in addition to recording and sending, especially while converting to .txt format. However, due to fairly good network conditions, we observed that

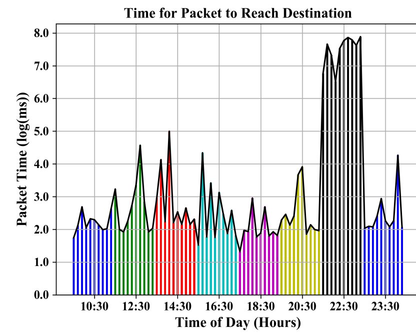


Figure 6: Network traffic observed throughout the day from different locations

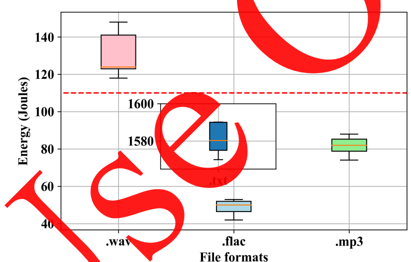
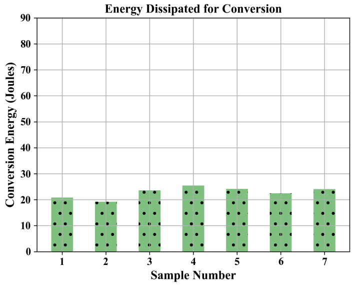


Figure 7: Total energy consumption by SkopEdge

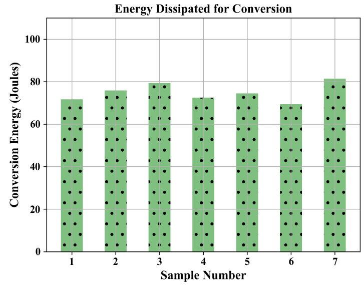
SkopEdge usually sends the data in .flac format, due to which, the average energy is in the range of 120 Joules, as shown in Fig. 7 (dashed line). Although flac does not accurately compress audio with abrupt changes, we recommend using .wav for irregular heartbeats. It may be noted that the energy consumption in the case of .wav is higher than that of .flac and that in the case of .txt is significantly higher. For further analysis, we present a breakdown of the energy consumptions.

Conversion Energy: Fig. 8 depicts the energy dissipated for converting from .wav to other formats. Since .flac files are lossless files and retain higher quality than .mp3, the energy required is much lower in Fig. 8(a) as compared to Fig. 8(b). In case of generating the .txt file, the entire process of analyzing the captured audio needs to be carried out within SkopEdge. Thus, it needs much energy, as shown in Fig. 8(c). Thus, we conclude that SkopEdge consumes more energy during conversion to lower resolution formats. There occurs as a tradeoff in order to transfer reliably to remote locations.

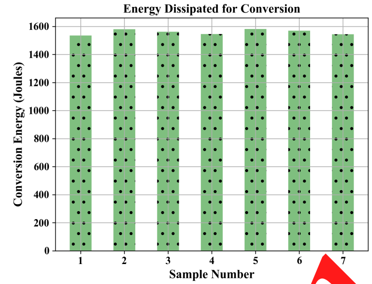
Transmission Energy: Fig. 9 depicts the energy needed for transmitting the files of different formats to a remote server. We observe from Fig. 4 how the file sizes vary in the case of each format. Correspondingly, the energy required for transmitting .wav files, as shown in Fig. 9(a) is maximum and that in the case of .txt files, as shown in Fig. 9(d) is minimum. On the other hand, as shown in Fig. 9(b) and Fig. 9(c), energy needed in case of .flac is lower than .wav but greater than in case of .mp3. The transmission energy for each format is intuitively analogous to the size of the files.



(a) .wav to .flac conversion

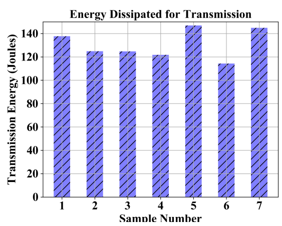


(b) .wav to .mp3 conversion

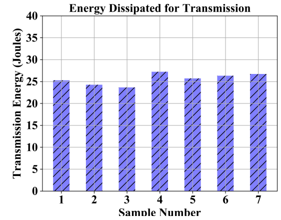


(c) .wav to conversion

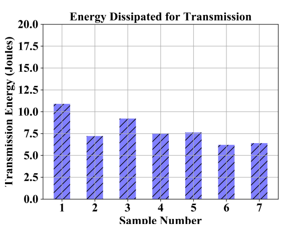
Figure 8: Comparison of energy dissipated for format conversions



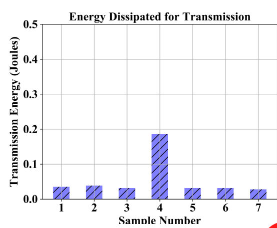
(a) Format: ".wav"



(b) Format: ".flac"



(c) Format: ".mp3"



(d) Format: ".txt"

Figure 9: Comparison of energy dissipated for transmitting data to the server

V. Conclusion

In this work, we designed and developed a traffic aware smart digital stethoscope *(SkopEdge)* as an e-health device for remotely monitoring the heart. Additionally, with the rising network congestion due to the increasing number of IoT devices, we formulated a scheme for SkopEdge, such that it automatically converts the captured high-resolution audio clips to simpler formats for generating the results in near real-time. We also exposed SkopEdge to different network environments and presented its results with detailed analysis.

In the future, we plan to deploy a network of SkopEdge devices using IoT protocols and observe its characteristics. Moreover, no sound is available in the case of .txt, and hence the results are only visualized. We plan to retransmit the recorded audio data when the network is available.

VI. Acknowledgement

The authors gratefully acknowledge the funding support received from INAE (Sanction letter no. INAE/121/AKF, Dt. 13-02-2019), and SERB/IMPRINT-II (Sanction letter no SERB/F/12680/2018-2019;IMP/2018/000451, Dt. 25-03-2019) for executing parts of this work.

References

[1] G. Markarian, L. Mihaylova, D. V. Tsitserov, and A. Zvikhachevskaya, “Video Distribution Techniques Over WiMAX Networks for m-Health Applications,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 1, pp. 24–30, Jan 2012.

[2] M. F. Tuysuz and M. E. Aydin, “QoE-based Mobility-aware Collaborative Video Streaming on the Edge of 5G,” *IEEE Transactions on Industrial Informatics*, pp. 1–1, Feb. 2020.

[3] N. Zhu, T. Diethe, M. Camplani, L. Tao, A. Burrows, N. Twomey, D. Kaleshi, M. Mirmehdi, P. Flach, and I. Craddock, “Bridging e-Health and the Internet of Things: The SPHERE Project,” *IEEE Intelligent Systems*, vol. 30, no. 4, pp. 39–46, Jul. 2015.

[4] A. Benharref and M. A. Serhani, “Novel Cloud and SOA-Based Framework for E-Health Monitoring Using Wireless Biosensors,” *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 1, pp. 46–55, Jan. 2014.

[5] S. Lee, J. Kim, and M. Lee, “A Real-Time ECG Data Compression and Transmission Algorithm for an e-Health Device,” *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 9, pp. 2448–2455, Sep. 2011.

[6] Y. Guo, Y. Yao, and G. Bai, “On Enhancement of Inter-Activity for Knowledge Sharing in e-Health,” in *Proceedings of International Conference on Communications (COMM)*, Jun. 2016, pp. 247–250.

[7] M. Elhilali and J. E. West, “The Stethoscope Gets Smart: Engineers from Johns Hopkins are giving the humble stethoscope an AI upgrade,” *in Proceedings of IEEE Spectrum*, vol. 56, pp. 36–41, 02 2019.

[8] R. R. G. Perron, M. F. Iskander, T. B. Seto, F. A. Qazi, D. A. Bibb, and E. Lim, “Multi-sensor Cardio-Pulmonary Stethoscope for quantitative lung water measurement,” in *XXXIInd General Assembly and Scientific Symposium of the International Union of Radio Science (URSI GASS)*, Aug. 2017, pp. 1–2.

[9] M. F. Iskander, T. B. Seto, R. RG Perron, E. Lim, and F. Qazi, “Cardio-Pulmonary Stethoscope: Clinical Validation With Heart Failure and Hemodialysis Patients,” *IEEE Transactions on Biomedical Engineering*, vol. PP, pp. 1–1, Aug. 2017.

[10] T. Chen, S. Yang, L. Ho, K. Tsai, Y. Chen, Y. Chang, Y. Lai, S. Wang, Y. Tsao, and C. Wu, “S1 and S2 Heart Sound Recognition Using Deep Neural Networks,” *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 2, pp. 372–380, Feb. 2017.

[11] F. Pan, P. He, C. Liu, T. Li, A. Murray, and D. Zheng, “Variation of the Korotkoff Stethoscope Sounds During Blood Pressure Measurement: Analysis Using a Convolutional Neural Network,” *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 6, pp. 1593–1598, Nov. 2017.

[12] F. S. Hillier and G. J. Lieberman, *Introduction to Operations Research: Concepts and Cases*. Tata McGraw-Hill Education, 2005.

[13] M. A. Pinsky and S. Karlin, *An Introduction to Stochastic Modeling*. Academic Press, 2010.