A Power Efficient Scheme for Speech Controlled IoT Applications

Article *in* International Journal of Engineering and Technical Research · January 2016

DOI: 10.17577/IJERTV5IS010382

1 author:

Nisarg Milan Vasavada
Gujarat Technological University

14 PUBLICATIONS **6** CITATIONS

SEE PROFILE

A Power Efficient Scheme for Speech Controlled IoT Applications

Nisarg M. Vasavada
GTU PG School – CDAC ACTS
Pune, India.

Swapnil Belhe
CDAC
Pune, India.

***Abstract*— Speech recognition has been a subject of research since decades. Although it has wide applications in Artificial Intelligence and modern user interfaces, when speech processing is applied to embedded systems we also need to consider the constraints which are normally faced and algorithms to overcome the same. While the applications of embedded systems are now being massively focused on Internet of Things, still the prime research concerns are power efficiency and security. Here a state of the art scheme is proposed where speech processing is applied to the constrained IoT applications and the wireless communication is made power efficient. For achieving so, the 6LoWPAN protocol is implemented and Modified Vector Algorithm for Speaker Identification (MVA-SI) is used to increase PDF of the correct interpretation input speech through predefined wake up call.**

Keywords—Speech Recognition, IoT, Power Efficiency, Speaker Identification

I. Introduction to ASR

Sound is an electromagnetic expression that is sensed, interpreted and delivered in a spectrum of frequencies and humans are the species that have learned to modulate it in many different ways. Speech is a combination of modulated sound and interpreted linguistics.[1] Initially concept of Automatic Speech Recognition (ASR) was limited to speech to text conversion which had many overheads that were resolved with time and dedicated research.[2] Today, ASR has taken an advance form where it has become one of the key pillars behind the success of Natural User Interface (NUI) and Artificial Intelligence (AI).[3]

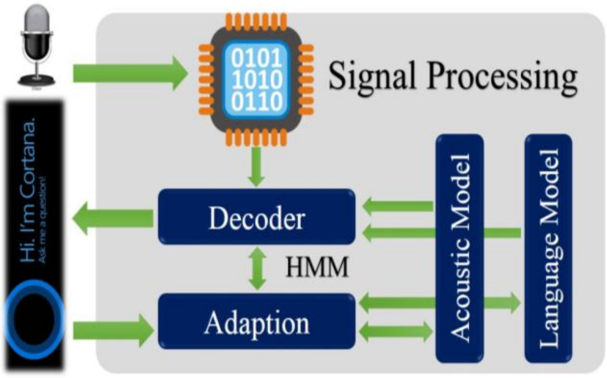

Figure 1. ASR Block Diagram [2]

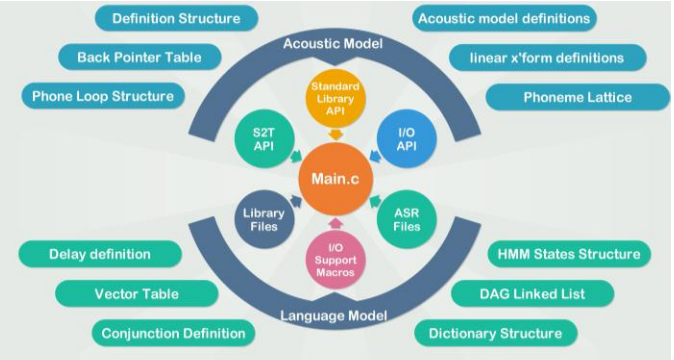

Figure 2. ASR engine file management*

Figure 1 shows the basic building blocks of a system consisting of ASR. The Microphone is used for input which is a transducer converting analog speech waves into voltage pulses that are digitized and provided to speech decoder. Considering the Linux philosophy of understanding systems, the blocks in the ASR engine are files which are interdependent and Operating System (OS) architecture dependent.[15] A file called “main” manages the order in which all the other files, functions and values defined in the files are called and used. Figure 2 describes, the Acoustic Model (AM) and the Language Model (LM) in context of files and internal contents are libraries that are referred and updated which is called training.[12] The AM recognizes phonemes, derives words from lattice and passes to LM which recognizes sentences by measuring delays and applying interpretation of conjunctions. The wider the structures and tables of AM and LM, the more they are trained which in turn makes the ASR engine more efficient.[8]

Mathematically, most of the ASR engines work on a combination of Hidden Markov Model (HMM) and Gaussian Mixture Model (GMM).[4][13] Markov Model is a statistical method of assuming the behavior of the system by the states it passes through and the values obtained from such states. The word “Hidden” focuses on the nature of systems where sequences of occurrence of states are obtained instead of values. This is the key behind word generation from phonemes. Each spoken input has a certain probability of being recognized which is calculated from its Probability Density Function (PDF).[5] The higher the probability, the finer the recognition becomes.

Note: Here it is considered that the ASR engine is written in C, if not so, the module management may differ a little but the basic concepts remain the same.[15]

II. IoT: Wireless Embedded Internet (WEI)

As the Internet of personal computers, mobile devices and high performance systems has been growing mature, one more revolution in the Internet was marching on its way–The Internet of Things (IoT).[10] The idea behind the IoT is to make small sensor driven smart devices IP enabled. In an IoT system, data is generated from multiple devices spanning various complexity which are processed in different ways. The basic IoT reference model is derived from conventional OSI network model.[10]

The newest and smallest members of Internet in IoT context are small sensors and actuators which are embedded devices by nature and do not contain scope or requirement of intelligence similar to fully fledged computing systems. Thus the TCP/IP layered approach is over-sufficient for such devices. Routing on Data Link Layer is performed based on corresponding addresses (64-bit EUI-64 or 16-bit short addresses).[19] There is one issue to resolve: as the MAC headers describe the source and destination addresses for the current layer-2 hop, in order to forward the packet to destination MAC, the node needs to know its address. Since each forwarding step overwrites the layer-2 destination address by the address of the next hop and the layer-2 source address by the address of the node doing the forwarding, this information needs to be stored somewhere else. 6LoWPAN defines the mesh header for this. Figure 3 shows layers to be implemented for using 6LoWPAN which is the official standard for wireless embedded internet defined by IETF. Figure 4 shows how the address is compressed in 6LoWPAN where LoWPAN header is larger whereas MAC address is smaller and payload is wider. Since communication is hop to hop IPv6 address is skipped. This approach is especially applicable for sensors and actuators where security is not a primary concern.

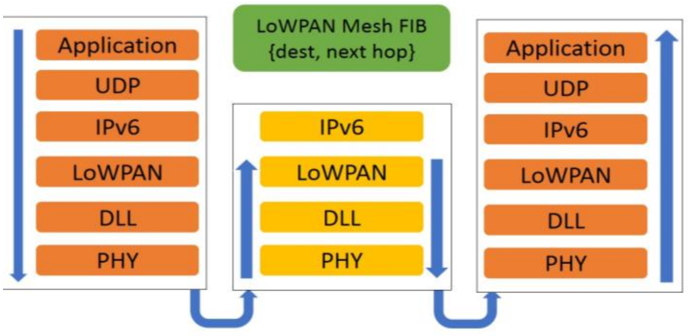

Figure 3. 6LoWPAN layers [19]

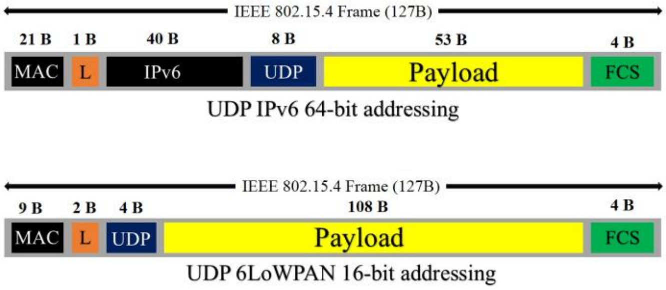
IEEE 802.15.4 Frame (127B)
21 B
1 B
40 B
8 B
53 B
4 B
MAC
L
IPv6
UDP
Payload
FCS
UDP IPv6 64-bit addressing
IEEE 802.15.4 Frame (127B)
9 B
2 B
4 B
108 B
4 B
MAC
L
UDP
Payload
FCS
UDP 6LoWPAN 16-bit addressing

Figure 4. Example of 6LoWPAN Address Compression [19]

III. Constrinted IoT Infrastructure

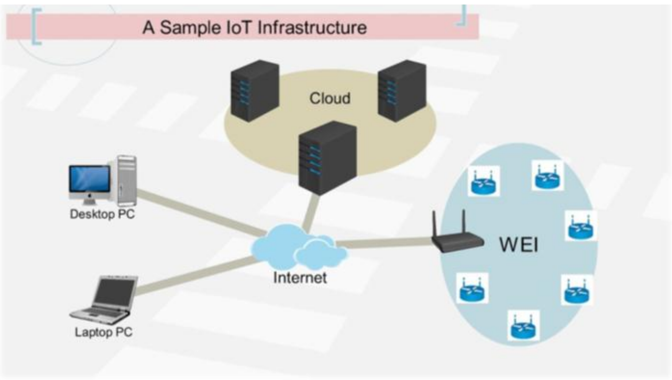

Figure 5. IoT integration with WEI

Figure 5 is a skeleton infrastructure of WEI based IoT approach where small embedded devices form a mesh/star topology based ad hoc network with a router which is connected to the internet which leads it to cloud and billions of other internet enabled devices such as PCs or smartphones.

A. Observations

Currently used IoT systems are convenient, accurate, secure and reliable but the audience for such systems is pretty limited, since the embedded domain comprises of many small low-end devices which are envisioned as the basic building blocks and source of success for the IoT. The approach for automating, monitoring and designing those devices is certainly different from regular ones and there need to be schemes and frameworks available which can act as bridges between two distinct implementations. The frameworks should be keeping following aspects into consideration.

1. The I/O interface for actuation and monitoring should be coherent and seamless.
2. The bandwidth, battery life, memory, portability and system cost should be taken into account.
3. The networking should be achieved with as less layers as possible.
4. Speaker independence in ASR should only be applied if the nature of application demands it.[18]

B. Proposed Solution

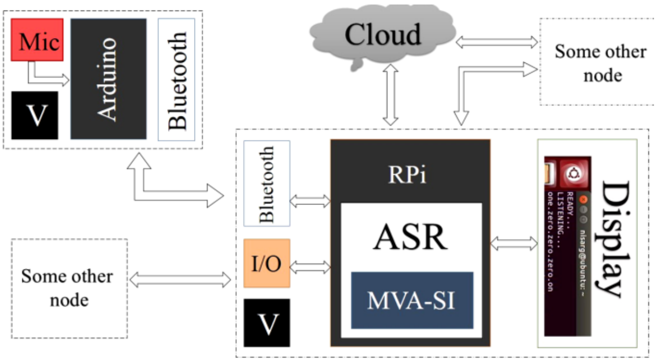

Figure 5. Proposed Scheme Prototyping Model

The proposed scheme is an IoT implementation at constrained embedded level where sensor nodes are provided local IP address while major routing nodes are provided full-fledged internet protocol support which also solves the security issue. For achieving so, the WEI tiny nodes are made independent of OS while mainstream nodes are built on top of OS which provides them full TCP/IP stack.

As the Figure 5 displays, the microphone provides speech output to the Arduino** which transmits it through Bluetooth using 6LoWPAN hopping to the major node which consists of Raspberry Pi (RPi)** and is provided full TCP/IP stack for proper internet connectivity across IPv6.[9][10][11] The I/O are connected to the RPi (either in wired manner or in wireless manner depends on the choice of the developer) which are governed by the inputs received from the voice commands sent from smaller nodes. The display is mainly for prototyping and debugging purpose. The RPi is equipped with open source modified ASR engine. The speaker identification is handled by the "Modified Vector Algorithm for Speaker Identification (MVA-SI)" algorithm which is applied in the acoustic model of the ASR engine. Then the output of AM is provided to LM. The inclusion or removal of LM depends on the nature of application. The MVA-SI actually does nothing but analyzing the phones in the speech context more precisely and giving significant eigenvalues for the vectors generated by the input speech. Phones are the starting of consonants and endings of vowels which are all distinct from each other. The algorithm focuses on the phones and the floating point values generated by them and creates an eigenvector. Such eigenvectors are compared to the stored ones and user is recognized. This is all done by one "Hello" wake up call which initiates the ASR and also recognizes the user. Since the keyword for this operation is predefined, the complexity of the algorithm decreases while the probability of correct speaker identification increases.

C. *MVA-SI Algorithm*

In the i-vector algorithm proposed by N. Dehak, a single space was created for speaker and channel.[16] This approach was later found inefficient and was modified with separate spaces.[14] In this proposed scheme where the system wakes up by only one specified word, Dehak's approach remains simple and efficient. The 512MB RAM and 700MHz ARM11 is sufficient enough to perform ASR initiation and eigenvector calculation for speaker identification thus no unexpected delay is faced.[17]

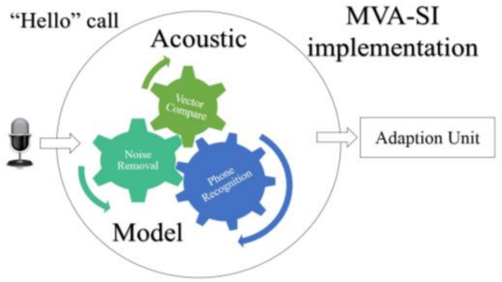

Figure 6. AM with MVA-SI [6][7][8]

The inclusion of MVA-SI as displayed in the Figure 6 fundamentally acts as pre-processing in AM which dismisses probabilities of other speakers and focuses on a defined speaker only. By doing so, the AM needs to process less phonemes, thus less operating cycles are required and the context specific interpretation of words become more accurate. The mathematics of the algorithm is out of the scope of this paper but following equation of supervector M which is used for modelling the GMM justifies the behavior of MVA-SI.

$$M = m + Tw \qquad (1)$$

In the equation 1, m is speaker independent base supervector, T is the speaker dependent factor matrix with low ranks and w notifies the count of utterance. For less trained engines, LM can be skipped if only a few tens of words need to be trained whereas in rest of the cases LM will have less eigenvectors to process and will act faster.[20]

IV. Conclusion

Parameter	T.S.	P.S.
Convenient UI		✓
Efficient ASR		✓
Power Efficiency		✓
Less code density		✓
Security	✓	
Authenticity	✓	✓

Table 1. Parameter Survey of schemes

Table 1 is a comparative survey of various parameters and their availability in both Traditional Scheme (T.S) and Proposed Scheme (P.S). Trivially it is the trade-off between constrained implementation and quality features that system developers have to make according to the requirements. The future work can consist of making the proposed scheme secure and more authentic while maintaining other parameters on constrained embedded system design domain.

References

[1] Andrew Kehler et al. "Spoken Language Processing", Prentice Hall New Jersey, ISBN: 978-0131873216.

[2] J.P. Haton, "Speech analysis for automatic speech recognition: A review," Proc. 5-th Conf. on Speech Technology and Human-Computer Dialogue, 2009, vol., no., pp. 1-5, June 2009.

[3] Ye-Yi Wang, Dong Yu, Yun-Cheng Ju, and Alex Acero, "An Introduction to Voice Search: A look at the technology, the technological challenges, and the solutions", IEEE Signal Processing Magazine, p.p 29-38, May 2008.

[4] Michelle Cutajar, Edward Gatt et al. "Comparative study of automatic speech recognition techniques", IET Signal Process., 2013, Vol. 7, Iss. 1, pp. 25–45

[5] M.J.F. Gales, "Acoustic Modelling for Speech Recognition: Hidden Markov Models and Beyond?" IEEE ASRU 2009, p.no 44.

[6] Douglas O'Shaughnessy, "Acoustic Analysis for Automatic Speech Recognition", IEEE Proceedings Vol. 101, No. 5, pp. 1038-1043, May 2013.

[7] Jinyu Li, Li Deng et al., "An Overview of Noise- Robust Automatic Speech Recognition", IEEE/ACM Transactions on Audio, Speech and Language processing, vol. 22, no. 4, pp. 745-777, April 2014.

[8] Zhen-Hua Ling, Shi-Yin Kang et al. "Deep Learning for Acoustic Modeling in Parametric Speech Generation", IEEE Signal Processing Magazine, pp. 35-52, May 2015.

***Note:* The open source Hardware platforms such as Arduino and Raspberry Pi are used for prototyping simplicity. For manufacturing purpose standalone controllers are suggested.

[9] Badamasi Y. A., "The working Principal of an Arduino", 11th International conference on Electronics, Computer and Computing, 2014.

[10] “IoT Reference Model”, A white paper by Cisco Inc, 2014.

[11] Severence C., "Eben Upton: Raspberry Pi", Computer Conversations by IEEE, p.p 14-16, October 2013.

[12] Sunyi Hu, David Mulvaney, S. Datta, “Modification of Sphinx 3 for Embedded System Implementation", International Conference on Multimedia, Signal Processing and Communication Technologies, p.p 137-140, 2011.

[13] Willie Walker, Paul Lamere et al., "Sphinx-4: A Flexible Open Source Framework for Speech Recognition" a white paper by Sun Microsystems Inc., 2004

[14] Anthony Chun, Jenny X. Chang et al., "ISIS: An Accelerator for Sphinx Speech Recognition", IEEE Symposium on Application Specific Processors, pp. 58-61, June 2011.

[15] David Huggins-Daines, Mohit Kumar et al., “Pocketsphinx: A free, Real-time continuous Speech recognition system for hand-held devices”, IEEE ICASSP, pp. 185-188, 2006.

[16] Wei Li, Tianfan Fu, Jie Zhu, "An improved i-vector extraction algorithm for speaker verification", Springer EURASIP Journal on Audio, Speech, and Music Processing, 2015.

[17] Hynek Hermansky, "Multistream Recognition of Speech: Dealing With Unknown Unknowns", IEEE Proceedings Vol. 101, no. 5, pp. 1076-1088, May 2013.

[18] Feng Deng, Chang-Chun Bao, "Speech enhancement based on Bayesian decision and spectral amplitude estimation", Springer EURASIP Journal on Audio, Speech, and Music Processing, 2015.

[19] “6LoWPAN: Wireless Embedded Internet”, Z Shelby, C Bormann, Wiley series in communication networking and distributed systems, ISBN: 978-0-470-74799-5.

[20] Javier Tejedor, Doroteo T. Toledano et al., "Spoken term detection ALBAYZIN 2014 evaluation: overview, systems, results, and discussion", Springer EURASIP Journal on Audio, Speech, and Music Processing, 2015.