Adaptive Rule Engine Based IoT Enabled Remote Health Care Data Acquisition and Smart Transmission System

M. P. R. Sai Kiran, P. Rajalakshmi, Krishna Bharadwaj, Amit Acharyya
Department of Electrical Engineering
Indian Institute of Technology Hyderabad, India
Email: ee12m1021, raji, krishnabharadwaj, amit_acharyya@iith.ac.in

***Abstract*—In the remote health care monitoring applications, the collected medical data from bio-medical sensors should be transmitted to the nearest gateway for further processing. Transmission of data contributes to a significant amount of power consumption by the transmitter and increase in the network traffic. In this paper we propose a low complex rule engine based health care data acquisition and smart transmission system architecture, which uses IEEE 802.15.4 standard for transferring data to the gateway. The power consumed and the network traffic generated by the device can be reduced by event based transmission rather than continuous transmission of data. We developed two different rule engines: static rule engine and adaptive rule engine, which decides whether to transmit the collected data based on the important features extracted from the data, thereby achieving power saving. In this paper, ECG data acquisition and transmission architecture is considered. The metrics used for performance analysis are the amount of power saving and reduction in network traffic. It is shown that the proposed rule engine gives a significant reduction in energy consumption and network traffic generated.**

***Keywords*—Adaptive rule engine, IEEE 802.15.4, data rate, energy consumption, ECG.**

I. Introduction

In remote health care monitoring applications, the Body Area Networks (BAN) provide a new paradigm for the WSNs in monitoring the bio-medical sensors. The data collected by the sensor nodes play a crucial role in further diagnosis. For further diagnosis on the data collected, it has to be transmitted to the central node or gateway node for further processing and storage. In general ZigBee devices which uses the same IEEE 802.15.4 PHY and MAC standard are used for wireless transmission to the central node [1]. In every remote monitoring application, one of the main limitations is power. The sensor nodes that are used to collect data are generally battery powered devices and frequent battery changes are also difficult. In this kind of applications the power consumption by the nodes should be reduced.

In the IoT enabled remote health care monitoring applications, the data collected from the sensors should be accessible anytime and anywhere, which requires constant network connectivity. If the remote health care monitoring application, transmits the data continuously, the amount of data generated will be huge. This also contributes to the hyper connectivity scenario. In hyper connectivity each device which has an ability to connect to the network will be connected to the network. According to the predictions made by GSMA, the total number of devices connected will be 15 billion by around 2015 and 24 billion by the year 2020 [4], [5]. In remote health care monitoring application we cannot make use of the available bandwidth effectively, if we use the traditional mode of transmitting the data continuously. It even leads to loss of data due to delay and buffer overloading, which is not acceptable particularly in the health care applications. An analysis on the delay and the data loss that occur in the WSNs based on ZigBee technology for transmission due to channel overlapping when the number of nodes that transmit data increase, has been made in [6]. The ZigBee uses only limited number of channels for the transmission. Whenever a ZigBee node has to transfer the data it first performs Clear Channel Assessment (CCA). If the channel is free then the ZigBee node is free to transmit the data to the destination, else the node has to wait for some backoff time which is decided by parameters like Maximum Backoff Number (NB) and Minimum Backoff exponent (BE). A detailed working of the CCA and CSMA-CA in IEEE 802.15.4 standard is given in [1]. As the amount of data to be transferred increases due to increase in the number of devices, the delay in transmission and losses during the transmission increases.

In order to prevent this scenario, one solution is to reduce the amount of data that is to be transmitted. In remote ECG monitoring applications the data need not be transferred continuously which will increase load on the network. In the existing architectures for data acquisition and transmission architectures [3], the traditional continuous transmission of data was used, which leads to higher power consumption and increase in the network traffic. In this paper, we propose an intelligent rule engine based transmission mechanism through which we can reduce the data losses due to delay in channel access and buffer overloading at transmitter and achieve power saving at the node. In this paper, we considered the remote ECG data acquisition and smart transmission system, which makes use of the rule engine controlled transmitter. The architecture proposed is shown in Fig. 1. The rule engine plays a prominent role in achieving the power saving and preventing losses. From the ECG data collected by the sensors, some key features like PR interval, QRS interval and QT interval are calculated using the features extracted by the ECG feature extraction block and are fed to the rule engine, which then decides whether the data has to be transmitted or not. The proposed architecture works on the 12 Lead ECG data and for

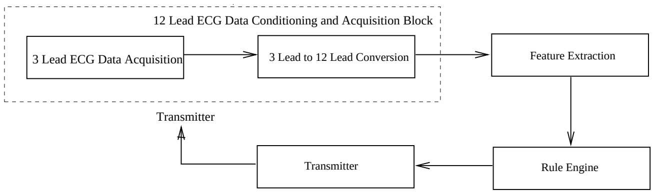

Fig. 1: 12 Lead ECG Data Acquisition and transmission Architecture

any other health care monitoring applications, where power is a critical limitation.

The rest of the paper is organized as follows. Section II discusses the proposed remote health care data monitoring and smart transmission architecture system. Section III discusses the performance of the proposed rule engine compared to the traditional continuous transmission mechanism. Section IV concludes the paper.

II. Proposed Adaptive Rule Engine Based Data Acquisition and Smart Transmission Architecture

In this section we discuss the architecture of the proposed adaptive rule engine based data acquisition and smart transmission architecture shown in Fig. 1. The proposed architecture consists of five functional units, which are briefly discussed in the following sections.

A. 12 Lead ECG data conditioning and acquisition system

For performance analysis of the proposed rule engine, a prototype of the ECG data acquisition and signal conditioning system shown in Fig. 2, is developed in IIT Hyderabad. The system is used to collect the ECG data from patient using signal processing techniques for removal of the noise generally generated from electrodes contact, body movements and power line. The architecture of the acquisition system is shown in Fig. 3. It contains various filtering and amplifying stages. The standard 12 lead ECG acquisition system uses 10 electrodes to collect the 12 Lead ECG data [7]. Recent developments in the ECG data acquisition methodologies made it possible to extract the 12 lead ECG data from 3 lead ECG data [8]. The construction of the 3 lead ECG data acquisition system is easy rather than the standard 12 lead ECG data acquisition system due to the number of electrodes that are to be connected to the body for monitoring of the data. The 3 lead ECG data acquisition system only requires 4 electrodes that are to be connected at 4 different parts of the body. Later the 12 lead ECG data is extracted from the 3 lead ECG data collected. The architecture for the 3 lead ECG data acquisition system is shown in Fig. 2. The 3 lead ECG data is collected by using 4 electrodes which are placed on 4 different locations on the body (RA (Right arm), LA (Left arm), LL (Left leg), RL (Right leg)). Each lead measures potential difference between two electrodes. For details about the lead's measurement refer to the TABLE I. The electrode placed in the RL (Right Leg) location is used as the common node for the measurement of the 3 leads.

Case	Lead	Potential diff. between	Common node
1	Lead I	RA & LA	RL
2	Lead II	RA & LL	RL
3	Lead III	LA & LL	RL

TABLE I: Lead details in 3 Lead ECG data acquisition system

Fig. 2: Developed ECG data acquisition system

The processing architecture shown in Fig. 3 contains an instrumentation amplifier at the beginning and followed by filtering and amplification stages. The lower cut-off frequency is 0.5 Hz and the upper cut-off frequency is 100 Hz. Additionally a notch filter has also been used for removing the power line frequency. Finally for storing the digital data, ADC on the Spartan-3E FPGA is used with a sampling rate of 1000 Hz. A PQRST complex of the lead I from the collected data is shown in Fig. 4. The complete analysis followed in this paper is made using the lead I ECG data collected from the 9 patients of several age groups collected using the in house developed ECG data acquisition and conditioning system. The same can be applied to the other leads with slight modifications in the architecture.

B. ECG Feature Extraction

The present world is equipped technologically providing an automated health prognosis. Usually signals from ECG (Electrocardiogram) are analyzed based on the important features like P, Q, R, S, T as shown in Fig. 5, through which a medical professional annotates to classify the condition of a patient. The P-wave in ECG signal represents atrial depolarization. The QRS depicts the ventricular depolarization. The T-wave gives the atrial and ventricular repolarization [9]. Earlier identifying of these points were based on heuristics. Later on a notion to automate this process, laid foundation for several signal processing algorithms and finally has taken its shape to monitor health of the patient remotely. Identifying these important points by means of some automated algorithms is feature extraction. In recent times there are several feature extraction

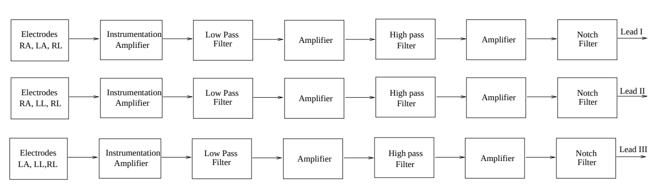

Fig. 3: 3 Lead ECG Data acquisition architecture

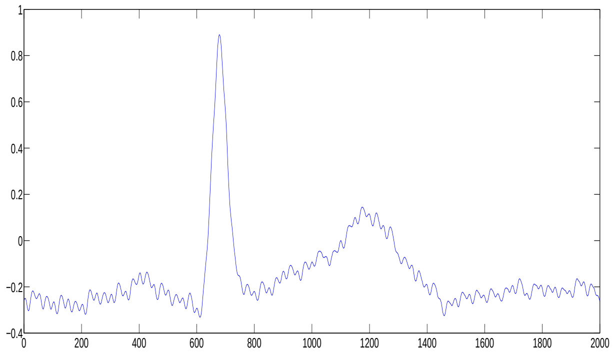

Fig. 4: PQRST complex from the collected Lead I Digital ECG data

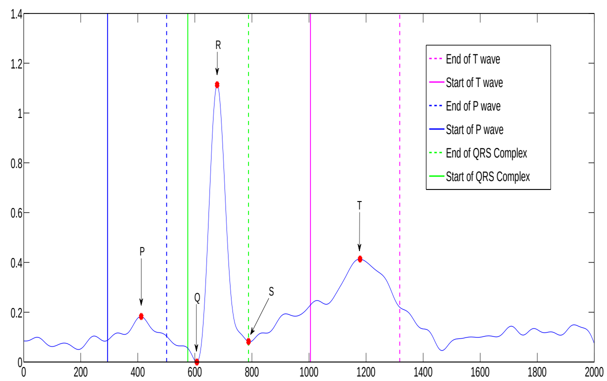

Fig. 5: Extracted features from Lead-I ECG Data

algorithms available [10]-[11]. The state of the art delineation algorithms exploited only R-peak to start with [12]. Then on several other algorithms based on frequency analysis of the ECG signal came into existence with which other points like P, Q, S, T have been eventually identified. It is also essential to have a track on algorithmic accuracy and computational complexity, as in the case of remote health care monitoring the power resources are limited. So further research evolved in developing the low-complex delineation (feature extraction) algorithms [13]-[14] which can be taken as a basis to make classification of the ECG signal. The feature extraction that is used here is based on the wavelet transform with a cascaded filter bank structure. For more detailed working of the feature extraction block, please refer to [13].

The data collected by the acquisition system will be fed to the ECG feature extraction block, which gives us the important features (P, Q, R, S, T) in the data. The features extracted from the data are shown in Fig. 4. Here one can observe the P wave, T wave, QRS Complex. We make use of the intervals calculated from the extracted features in the rule engine to classify the data as normal or abnormal.

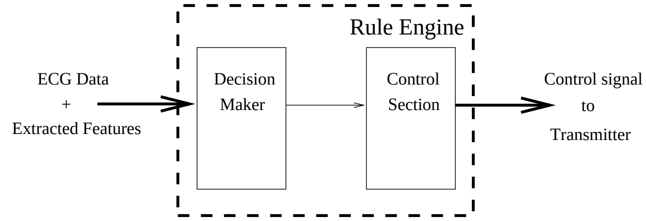

Fig. 6: Proposed Rule engine

C. Proposed Rule Engine

The rule engine is the key component in this architecture, which aids for the low power consumption and the low network traffic generation. It basically consists of two sections namely, *decision making* and *transmitter control* shown in Fig. 6. The decision making section makes use of the features extracted by the feature extraction block. In this paper, two types of rule engines are developed and their performances are evaluated based on the data rate they generate and energy consumption.

1) Static Rule engine: The static rule engine consists of *"decision making"* section and *"transmitter control"* section. Aim of the decision making section is to analyze the features extracted from the collected data and to decide whether the data is normal or abnormal. The decision made by the decision maker will then be made use by the transmitter control section for controlling the transmitter. The key features that doctors use to classify the data are listed in the TABLE II. The values listed in the TABLE II are normal ranges of the ECG data for a healthy patient [2] and are used as the *hard threshold*, which is the bounding limit of perfectly healthy ECG data. If the values of the parameters are in the range listed in the table, the data is then classified as a normal data else it is classified as an abnormal data. Whenever the data is classified as the abnormal data, the rule engine switches on the transmitter and the data samples are transmitted to the gateway. At the same time the samples that are already processed ahead of the current abnormal data and the samples that will processed after the current data samples are then stored from buffer in to the local storage which resides on the node. The advantage achieved by storing the data is, whenever the doctor is alarmed with the abnormality, the doctor can query and monitor the data that is stored in the local storage during the abnormality. By using this kind of storage mechanism, the accuracy of the diagnosis can be maintained by eliminating false alarms caused sometimes due to the improper contact of the electrodes. The steps involved in the static rule engine in order to classify the data are given in Algorithm. 1.

The features (P, Q, R, S, T) extracted from the lead I ECG data by the feature extraction block are fed to the decision maker section. It then calculates the PR, QRS, QT intervals. The intervals calculated are then compared with the $HardThreshold$ values shown in TABLE. II and makes a decision. If any one of the three intervals calculated exceeds the threshold value, the data is classified as an abnormal data and triggers the control section to switch on the transmitter. Then the control section takes the control signal given by the decision making section and switches on the transmitter.

Algorithm 1 Static Rule Engine

Initial: Set $HardThreshold$ values

procedure DECISION MAKER($ExtractedFeatures$)
  Comment: Calculate PR, QRS, QT intervals.
  Calculate $Data.PR\_interval$;
  Calculate $Data.QRS\_interval$;
  Calculate $Data.QT\_interval$;
  if $Data > HardThreshold$ **then**
    Decide the patient is abnormal;
    CONTROL SECTION(on);
    Transmit the data;
    Store the data samples in the local storage;
  else
    Decide the patient is normal;
    Do not transmit the data;
  end if
end procedure
procedure CONTROL SECTION($ControlSignal$)
  if $ControlSignal$ == on **then**
    Switch on the transmitter;
    Wait for the data to be transmitted;
    Switch off the transmitter;
  else
    Maintain transmitter in off state;
  end if
end procedure

Case	Parameter	Normal Threshold
1	PR interval	0.12 - 0.20 Sec
2	QRS interval	$\leq$ 0.12 Sec
3	QT interval	$\leq$ 0.42 Sec

TABLE II: Threshold values of the intervals

Algorithm 2 Adaptive Rule Engine

Initial: Set $HardThreshold$ values
Set $SoftThreshold$ = $HardThreshold$
Set $abnormal\_count$=0 and start timer T;

procedure DECISION MAKER($ExtractedFeatures$)
  Comment: Calculate PR, QRS, QT intervals.
  Calculate $Data.PR\_interval$;
  Calculate $Data.QRS\_interval$;
  Calculate $Data.QT\_interval$;
  if T expires **then**
    Reset $SoftThreshold$; Restart timer T;
  end if
  Decide the data is abnormal;
  if $Data > HardThreshold$ **then**
    Decide the data is abnormal;
    Store the data in local storage;
    if $Data > SoftThreshold$ **then**
      CONTROL SECTION(on);
      Transmit the data;
      if abnormal $Data.PR\_interval$ **then**
        $SoftThreshold.PR\_interval$= $Data.PR\_interval$;
      else if abnormal $Data.QRS\_interval$ **then**
        $SoftThreshold.QRS\_interval$= $Data.QRS\_interval$;
      else if abnormal $Data.QT_interval$ **then**
        $SoftThreshold.QT\_interval$= $Data.QT\_interval$;
      end if
      Set $abnormal\_count$=0;
    else
      Do not change $SoftThreshold$ parameters;
      $abnormal\_count$ = $abnormal\_count$+1;
    end if
  else
    Decide the patient is normal;
    Do not transmit the data;
  end if
end procedure
procedure CONTROL SECTION($ControlSignal$)
  if $ControlSignal$ == on **then**
    Switch on the transmitter;
    Wait for the data to be transmitted;
    Switch off the transmitter;
  else
    Maintain transmitter in off state;
  end if
end procedure

2) Adaptive Rule Engine: The static rule engine discussed above uses only a single *hard threshold*, with which it compares the extracted features from the data. It gives a good performance in some situations, but in some situations it performs similar to the traditional continuous transmission architecture. For a patient suffering from first degree atrioventricular block, the PR interval always exceed 0.20 seconds, in this case the data exceeds *hard threshold* always and leads to continuous transmission of the data. It can be better optimized using this adaptive rule engine. Doctors need not be informed every time, the patient crosses the *hard threshold*. In the adaptive rule engine, we make use of two thresholds, *hard threshold* and *soft threshold*. The *hard threshold* is similar to the threshold used in the static rule engine scenario and uses the same threshold values as shown in TABLE. II. The *soft threshold* is an internal variable, which is initialized to *hard threshold* and whenever the sensed value exceeds the current *soft threshold*, the sensed value is assigned to the *soft threshold*. Steps involved in the adaptive rule engine are shown in ALGORITHM 2.

The input to the adaptive rule engine is same as the static rule engine i.e. the features extracted from the ECG data. Initially the $SoftThresold$ value is same as the $HardThreshold$. Later the values of the $SoftThreshold$ parameters are changed based on the observed parameters of the data. In the first iteration, the parameters observed from the data are compared with the $HardThreshold$ values. If any of the parameter exceeds, it is classified as an abnormal data and it is again compared with the $SoftThreshold$. In the first iteration the values of the parameters in $SoftThreshold$ and $HardThreshold$ are same. Hence the parameters also exceed $SoftThreshold$, if they exceed $HardThreshold$. Then the parameter values in which the data exceeded are assigned to the $SoftThreshold$ parameters i.e. if in a case the data of a patient has a QRS interval of 0.14 seconds, it will be classified as an abnormal data by the static rule engine. The same will be the case in the first iteration of the adaptive rule engine. Now in the adaptive rule engine, the parameter QRS interval value is changed to 0.14 seconds. In the second iteration, if the same case is repeated, the data will be classified as the abnormal data and the value of the $abnormal\_count$ value is incremented but the data will not be transmitted. The value $abnormal\_count$ is used to determine the number of times the patient has crossed the $HardThreshold$ but is within the particular $SoftThreshold$. If in the third iteration the data exceeds the QT interval threshold, again the parameter QT interval in the *Soft Threshold* values are adjusted accordingly. The parameter T indicates the time to reset the *soft threshold*, which can be defined by the doctor. After every T duration the *soft threshold* is reset to the *hard threshold* value. If T is set to zero, the adaptive rule engine works similar to the static rule engine.

III. Performance Analysis

The performance analysis of the proposed rule engine is done on the lead I ECG data collected from 9 patients for a duration of 30 seconds at a sampling rate of 1000 Hz, using the in house developed data acquisition system at IIT Hyderabad shown in Fig. 2. Performance metrics that are used in evaluating are energy consumption and the data rate generated. The performance of the proposed rule engine is discussed briefly in the following sections.

A. Analysis on Energy Consumption

Analytic models for energy consumption of the sensor nodes are done in [15], [18]. In the analysis, they have considered energy consumed by the processor, transceiver and sensors. Here in this paper we are considering only the energy consumed by the transmitter as the other parameters remain constant for the analysis. For the modeling of energy consumption, the transmitter is assumed to operate only in two states, on and off state. The energy consumed by the transmitter can be modeled as the energy consumed in a particular state and for the state transitions.

$$E_{cons} = E_{state} + E_{trans} \quad (1)$$

$$E_{state} = \sum_{i} \frac{P_{TX} L_i}{R} + P_{Off} T_{Off} \quad (2)$$

$$E_{trans} = \sum_{j=1}^{n} P_{on-off} T_{on-off} + \sum_{k=1}^{n} P_{off-on} T_{off-on} \quad (3)$$

For the purpose of analysis, the transmitter is assumed to operate at at 3.3 volts, consume 17 mA current in the transmitting state and 0.02 μA in the off state. The E_{cons} in (1) shows the over all power consumption by the transmitter. It is the sum of power consumed in the two states (on, off) and energy consumed for state transitions (on to off, off to on) by the transmitter. The P_{TX} and P_{off} indicates the power consumption in transmitting and off state respectively. T_{off} is the total time the transmitter spent in the off state. L_i indicates the packet length to be transmitted and i indicates the packet number. P_{on-off} and P_{off-on} indicates the power consumed during the state transition from on to off and off to on respectively and the corresponding transition times are T_{on-off} and T_{off-on}. Fig. 7 plots Energy consumed by a patient node for transmitting Vs. patient id in three scenarios - burst/continuous transmission, with static rule engine and with adaptive rule engine. In the Fig. 7, one can observe the energy consumed using continuous transmission for patient 1 is 0.085 J which is constant for all patients. The power consumed for patient 1, in the static rule engine and adaptive rule engine based transmissions are 0.032 J and 0.008 J respectively. The performance of the adaptive rule engine based transmission is good compared to the other two scenarios. In the case of patient 2, the energy consumed in the static rule engine is 0.078 J, which is nearly same as in the continuous transmission. The reason is, patient 2 is abnormal which leads his ECG data to exceed the hard threshold in most of the cases. In the case of patient 2 the adaptive rule engine performs well by consuming an energy of 0.018 J in 30 seconds. In the case of patient 9, the ECG data observed is normal all the time, which leads static rule engine and adaptive rule engine behave in the same way. From Fig. 7, we can observe that the proposed adaptive rule engine yields significant energy saving compared to the other two scenarios.

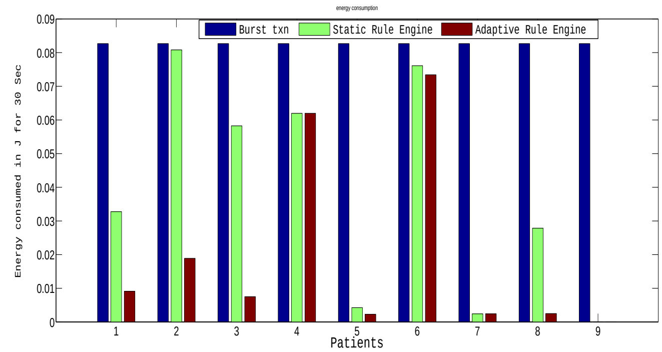

Fig. 7: Energy consumption in all the three scenarios

B. Analysis on Data Rate Generated / Network Load

Fig. 8 plots the data rate generated per patient Vs. the patient id. The data rate generated depends on the number of abnormal samples, that have to be transmitted. Each sample is encoded using 12 bits symbol. From Fig. 8, it is observed that the continuous transmission transmission for patient 1, leads to a data rate of 12 K bps and remains constant for all the patients, whereas the adaptive rule engine generates a data rate of 1.6 K bps. The expected delay and losses in the burst transmission scenario will be high compared to the static rule engine and adaptive rule engine scenarios. Thus by using the adaptive rule engine based transmission, the network traffic can be significantly reduced compared to the other two scenarios.

C. Estimated Battery Lifetime

For the estimation of the battery life time, a 3.3 volt, 2300 mAh battery is considered. Fig. 9 plots the estimated life time of the battery Vs. patient id. For patient 9 the energy consumed is very less compared to other 8 patients, since he is a normal person there is no much data to be transmitted. The battery life time of the patient 9 is very high. For scaling purposes the patient 9 is not included in the figure. For estimating battery

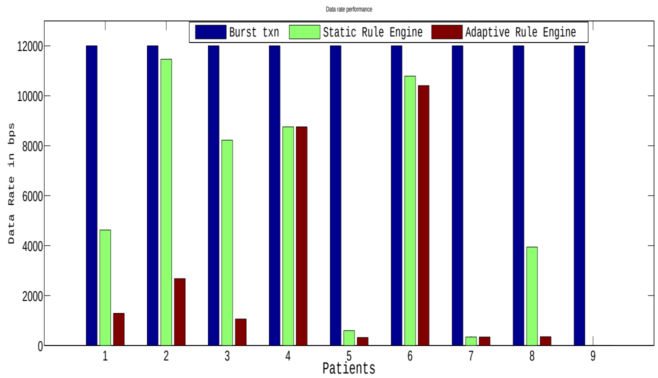

Fig. 8: Data rate generated in all the three scenarios

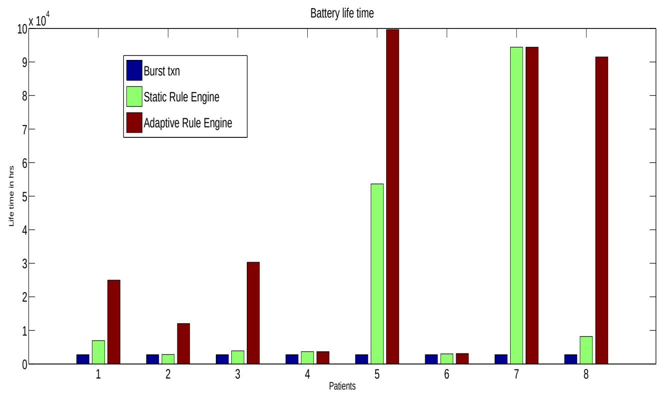

Fig. 9: Estimated battery life time in all the three scenarios

life, power consumed by the patient in the 30 seconds interval is considered as an average power consumption. In the Fig. 9, one can observe that the battery life time in the adaptive rule engine based transmission scenario is far better than the burst scenario in most of the cases. In the case of patient 5 the battery can last for upto 11.4 years in adaptive rule engine based transmission scenario compared to 0.314 years in the burst transmission scenario.

The analysis shown above is also performed using the ECG data from "The PTB Diagnostic ECG Database" data base [16]-[17], which also yielded the similar performance.

IV. Conclusion

In this paper, we proposed an adaptive rule engine based remote health care data acquisition and smart storage system. Two kinds of rule engines: static rule engine and adaptive rule engine were proposed and their performance is evaluated. For the performance evaluation, ECG data of different patients of different age groups were considered. The analysis show that the adaptive rule engine based transmission mechanism gives better performance by achieving very good energy savings and significant reduction in the network traffic generated. The adaptive rule engine based health care data acquisition and smart transmission architecture can aid, low power and low data rate networks, which is an important aspect of IoT enabled health care systems.

References

[1] IEEE Standard for Local and Metropolitan Area Networks-Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs), IEEE Computer Society Std.

[2] www.meds.queensu.ca/central/modules/ECG/normal_ecg.html

[3] Borromeo, S.; Rodriguez-Sanchez, C.; Machado, F.; Hernandez-Tamames, J.A.; de La Prieta, R., "A Reconfigurable, Wearable, Wireless ECG System," *Engineering in Medicine and Biology Society*, 2007. EMBS 2007. 29th Annual International Conference of the IEEE , vol., no., pp.1659,1662, 22-26 Aug. 2007

[4] http://gigaom.com/2011/10/13/internet-of-things-will-have-24-billion-devices-by-2020/

[5] The Mobile Economy, GSMA, 2013.

[6] Sahandi, R.; Yuanlong Liu, "Channel Overlap Problems of ZigBee Networks for Remote Patient Monitoring on General Hospital Wards," *Communications and Mobile Computing (CMC), 2010 International Conference on*, vol.3, no., pp.259,263, 12-14 April 2010

[7] Abdul Qayoon Bhat, Vineet kumar, Sunil Kumar, "Design of ECG Data Acquisition System," *International Journal of Advanced Research in Computer Science and Software Engineering"*, vol. 3, issue 4, April 2013.

[8] S. Maheshwari, A. Acharyya, P. Rajalakshmi, P. E. Puddu and M. Schiariti (2013), "Accurate and Reliable 3-Lead to 12-Lead ECG Reconstruction Methodology for Remote Health Monitoring Applications," *IEEE International Conference on e-Health Networking, Applications and Services (Healthcom, 2013)*, 9-12 October, 2013, Portugal, 2013

[9] GALEN S. WAGNER, Henry J. L. Marriot, "Marriott's Practical Electrocaardiography'.

[10] Cuiwei Li; Chongxun Zheng; Changfeng Tai, "Detection of ECG characteristic points using wavelet transforms," *Biomedical Engineering, IEEE Transactions on* , vol.42, no.1, pp.21,28, Jan. 1995

[11] L. Y. Di Marco, Lorenzo Chiari, "A wavelet-based ECG delineation algorithm for 32-bit integer online processing," *Biomedical Engineering online,* vol. 10, no. 2, article 23, 2011.

[12] Pan, Jiapu; Tompkins, Willis J., "A Real-Time QRS Detection Algorithm," *Biomedical Engineering, IEEE Transactions on* , vol.BME-32, no.3, pp.230,236, March 1985.

[13] Mazomenos, E.B.; Biswas, D.; Acharyya, A.; Chen, T.; Maharatna, K.; Rosengarten, J.; Morgan, J.; Curzen, N., "A Low-Complexity ECG Feature Extraction Algorithm for Mobile Healthcare Applications," *Biomedical and Health Informatics, IEEE Journal of* , vol.17, no.2, pp.459,469, March 2013

[14] Mazomenos, E.B.; Chen, T.; Acharyya, A.; Bhattacharya, A.; Rosengarten, J.; Maharatna, K., "A Time-Domain Morphology and Gradient based algorithm for ECG feature extraction," *Industrial Technology (ICIT), 2012 IEEE International Conference on* , vol., no., pp.117,122, 19-21 March 2012

[15] H. Zhou, D. Luo, Y. Gao and D. Zuo, "Modeling of Node Energy Consumption for Wireless Sensor Networks," *Wireless Sensor Network*, Vol. 3 No. 1, 2011, pp. 18-23. doi: 10.4236/wsn.2011.31003.

[16] Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCh, Mark RG, Mietus JE, Moody GB, Peng C-K, Stanley HE. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. Circulation 101(23):e215-e220 [Circulation Electronic Pages; http://circ.ahajournals.org/cgi/content/full/101/23/e215]; 2000 (June 13).

[17] Bousseljot R, Kreiseler D, Schnabel, A. Nutzung der EKG-Signaldatenbank CARDIODAT der PTB ber das Internet. Biomedizinische Technik, Band 40, Ergnzungsband 1 (1995) S 317

[18] Ian F. Akyildiz, Mehmet Can Vuran, "Wireless Sensor Networks"