Adv Syst Sci Appl 2018; **03**; 123-143
Published online at http://ijassa.ipu.ru/index.php/ijassa/article/view/543

An Efficient Approach To Civil Structures Health Monitoring Using Fog Computing As Clusters Through 5G Network Environment

Divya Sinha[1], Kavish Doshi[1], M. Rajasekhara Babu[1]

[1)] *School of Computer Science and Engineering*

VIT University, Vellore, 632014, India

E-mail:

divya.sinha2014@vit.ac.in,doshikavish.nilam2014@vit.ac.in,mrajasekharababu@vit.ac.in

Abstract: Structural Health Monitoring (SHM) is a process to trace real time information about the physical health of a civil structures at any given point of time. SHM systems can also approximate the life expectancy of the structure. This constant monitoring can help during emergency situations such as earthquakes. The increasing need for SHM is the result of deterioration of several set of structures over time. A proper maintenance plan is beneficial to not just keep the structure in a favorable state throughout its lifecycle, but to additionally minimize the relative costs required by the structures through resource optimization. Various forms of SHM have been implemented over the past three decades like traditional wired systems. Recent development has resulted in Wireless Sensor Networks (WSN) being designed for the same purpose. Few publications have also included IoT framework along with WSNs. The introduction of WSN has led to easy deployment of SHM compared to the traditional wired network. But WSN's have led to significant challenges. The main challenge that has been focused in this paper is the constraint of power consumption. The power consumption is most critical factor for the sensors deployed in high-rise structures, high density traffic bridges etc., since it is difficult to reach the places to replace the battery drained sensors and also involved high installation cost. This paper proposes a concept of a framework for SHM with FOG computing which hasn't been explored in the field of SHM yet. In this model, the WSN is comprised of a piconet which transmits data through Bluetooth Low energy (LE) and the FOG transmits data to cloud by exploiting 5G network features in order to reduce the power consumption furthermore. Thus, saving power at every level possible in the SHM architecture. It also introduced a head sensor algorithm within clusters which maximizes the lifetime of the WSN. A mathematical model has been developed in support to evaluate how the power of the system can be minimized for the FOG node and cloud servers in accordance with the 5G network environment. Experiments have been carried out to validate proposed framework and demonstrate the results.

***Keywords*:** FOG computing, Civil Structural Health monitoring (SHM), Civil Structures, Wireless Sensor Networks (WSN), Internet of Things(IoT), Sensors, Cluster algorithms

1. INTRODUCTION

The Internet of Things (IoT) has rightly been claimed as the next wave of technology after the Internet of people. IoT has transformed ubiquitous computing with a plethora of applications centered around embedded and sensory systems. [1] IoT enables devices to talk to each other, sense what is going on in the environment and adjust the comfort level according to the needs of the users. The major part of the IoT network consists of communication, computing and caching components and the even distribution of these resources. This network is believed to expand to 40-50 billion objects by 2020, with each object contributing exorbitant amounts of data. This size

of data can easily outsmart today's database management systems. [2] To tackle this big data and integrate the needed resources in a wireless network architecture, cloud computing comes in the picture. Cloud computing provides numerous on-demand services such as platforms, softwares and infrastructure. (PaaS, SaaS, IaaS) [3]

While the internet isn't scalable enough to accommodate IoT big data, the cloud does provide virtual storage and processing facilities that are sufficient to IoT requirements. However, transferring such high volume of data is expensive and uses significant amount of bandwidth. [2,4] Adding to that, the need to transfer all information to the cloud will bring forward high latency rates and delay the entire decision-making procedure. [5] This will severely hamper applications where prompt actions are to be taken by IoT devices such as Healthcare. Furthermore, Users often share their personal information through IoT objects. This sensed private data needs to be kept secure which poses a great challenge in front of the vast cloud accessible by millions of devices. Another challenge faced by IoT devices is the great value of power consumption.

For example, in smart grid application, An Iot device is assigned to each home in a residential area, this smart meter collects user's electricity usage log, and reports to the control unit periodically (e.g. every 15 minutes [6]). The control unit makes real-time decisions based on this reported data and takes suitable actions. In order to make the most of the information gathered by IoT devices, the more precise and frequent the data transferred, the better real-time analysis followed by accurate decision making can be carried out. But to be able to send real-time data regularly to the cloud, it will cost humongous communication resources.

So, the problems faced by Cloud Computing are high latency, big data, power consumption and privacy.

To overcome these problems an intermediate architectural concept of "Fog Computing" was brought into picture by Cisco. This concept introduces a node between the IoT device and the cloud. This node supports the cloud computing platform in handling a part of the workload locally at the edge of the network instead of transmitting the whole workload to the cloud. [7]. The main factor that differentiates cloud computing from fog computing is the fog's localization and proximity to all of the nodes as compared to the cloud. The fog is an extension of the cloud to the network's edge. Thus, making the sensory embedded IOT devices closer to the fog than the cloud. This makes it easier and faster to access the local fog than the global cloud. In simpler terms, the fog is nothing but a descended cloud. [8] Edge computing mainly corresponds to the edge network used as mobile edge. This concept of computing is also known as fog computing, is gathering attention in order to overcome the problems of cloud computing [9]- [11]. While cloud computing requires high computational capacity, edge computing requires medium to low computational capacity. Unlike cloud servers that are very large in size and centralized, edge computing needs servers which are smaller in size and placed over many locations. While cloud computing is suitable for delay tolerant and computationally intensive applications, edge is used for applications demanding low latency and real-time operation. For the cloud computing to work, the devices need to be connected to the internet at all times but in the case of edge the information is cached in the servers and at particular intervals of time the information is sent to the cloud for storage. It is seen that cloud computing requires complicated deployment planning while in the case of edge computing there is a possibility of ad-hoc deployment with no or minimal planning. [12]

1.1 Fog computing – Architecture

The generic architecture of Fog Computing comprises of the below 3 layers
1. Cloud Computing layer, 2. Fog cluster layer, 3. Wireless terminal layer. The main focus of the architecture is the cloud computing layer. It is the center of the whole network and end responsible for storage of resources. Following the cloud computing layer are the Fog Clusters situated at the

edge. The fog clusters Communicate with mobile devices at the network's edge. Every fog cluster is responsible for a particular region's mobile devices which transmits data to the fog cluster. This is advantageous for mobile devices as they can easily achieve resources through the fog cluster without having to connect with any other mobile devices. This will in turn also decrease the power consumption. Additionally, they don't have to store resources and can simply retrieve the wanted resources from the fog cluster as per their time constraints. They also will not face any high latency issues as the fog cluster is present right at the edge. Thus, making it easy for them to exchange and transfer data with fewer hops. [13]

A Fog Micro Data Centre(MDC) also enables context-aware computing. The gateway is that device which collects information from the underlying nodes. In particular scenarios, the gateway is required to perform pre-processing- or interoperability-related tasks, the gateway cannot do these tasks on its own. Hence, the gateway has to be made better equipped with the capabilities of the fog. This gateway is commonly known as the fog smart gateway (FSG). An FSG is feedback driven and highly context aware. It transfers data only when needed. [8]

Considering a network with millions of end users, will not only contain numerous IoT devices transferring information but will also require the proper computation of the received data. So, just having fog nodes is not sufficient, we need to maintain certain standards alongside these nodes. Some new standards need to be in order when it comes to certain domains for wide acceptance of fog such as data management, security and privacy, building of unified fog-cloud platforms etc. [14]

1.2 Role of IoT Gateway

The generic architecture discussed earlier may be effective for a single sensory IoT device but it doesn't account for millions and millions of IoT devices scattered in different regions transferring big data to the fog and in turn to the cloud. This huge scalability also introduces the persistence problem of security as more and more devices are vulnerable to other devices. Additionally, sensors have constrained capabilities when it comes to their networking abilities. Adding to this is the managerial aspect, it'll be very troublesome to monitor the operation of each IoT device in the network individually. Instead, if there is a single point of access for all IoT devices. We will simply have to monitor that point of access in place of the several IoT devices. The IoT Gateway can be this point of access between all IoT devices and the fog. This flow has been shown in figure 1.

Owing to the above-mentioned issues in the generic architecture, a better architecture is suggested that comprises of the IoT Gateway. The Gateway will help in streamlining the transfer of data in a coherent manner. The gateway ensures all data passing through it follows a certain set of rules or protocols that establishes uniformity and also encourages change in protocol for certain data when required while switching between networks. While the Gateway is an intermediary between the IoT devices and the fog, it shouldn't be confused with a dumb proxy that simply transfers data from sensors to fog. It also undergoes the important task of pre-processing the data according to the requirement of cloud outlets or data centers. These pre-processing tasks can vary according to the needs of the system from message filtering to aggregation. [15]

Securing the network is probably the foremost concern today with respect to IoT. Adding a layer of security can be done relatively easily and effectively with a Gateway as compared to an architecture that lacks the gateway. A PKI infrastructure is followed to ensure security, wherein each device is assigned a unique identity, that is, a pair of cryptographic keys also referred as Digital Certificate which makes the communication secure via encryption. This can be a very tedious task to carry out with the absence of an IoT gateway. [16]

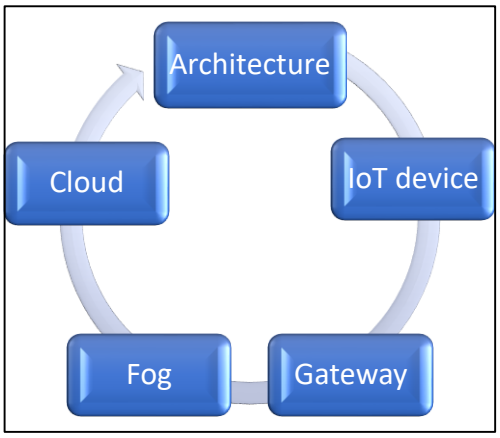


Figure 1: Cycle showcasing the flow of different components of an IoT architecture.

Now, after integration of the gateway-Viewing the architecture as a whole, it can be seen as a 4-layer model constituting of the Device layer, Hub layer, Fog layer and the Cloud layer as shown in figure 2. The lowest tier i.e. the Device layer consists of the IoT devices with sensors. The next tier- the Hub layer contains the Gateway. Above the Hub layer, lies the Fog layer which is formed by the edge and the fog nodes/clusters. The final layer is the Cloud layer which as the name suggests has the main cloud.

Device Layer: The lowermost layer of the architecture, which is used by the end users is known as the device layer. It mostly consists of different kinds of IoT devices, which are used to collect data and send it over to the next layer.

Hub Layer: This layer is sometimes called the Gateway or an intermediate layer between the device and the cloud. This layer not only helps in maintaining protocols over different networks, it also takes care of the security and makes sure that the data reaches the right destination. The GSM module attached to some of the IoT device can also be considered as a Hub layer.

Fog Layer: This additional layer between the gateway and the cloud has a series of benefits. This layer collects data from its nearby devices and then periodically sends all the data to the cloud for actual analysis and prediction. This not only helps in saving energy of the IoT devices but also helps save computational time by computing the result locally. These features have given this layer another name which is "Edge Nodes", since they are located at the edge of a network. Fog with the Hub layer makes the network more stable since, the data would follow certain protocols in a network. While, a fog layer without a hub layer with too many devices would make it difficult to cope with it but could be manageable if the number of devices are sizeable.

Cloud Layer: Since, this layer is needed for storing data from all the devices and compute the results. It becomes the most crucial layer of all.

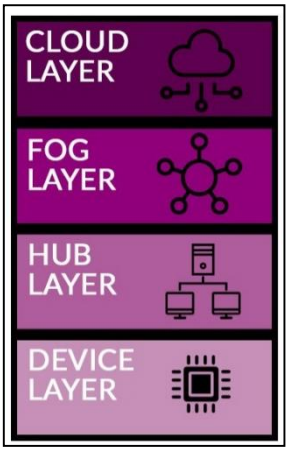


Figure 2: Fog computing Architecture

1.3 Mapping the architecture to the functional diagram of IoT

The functional diagram of IoT consists of 6 different functionalities, which are device, communication, services, management, security and the application.

Let us map our architecture to these functionalities as depicted in figure 3. The first block, the IoT devices fall under the 'DEVICE' functionality. The gateway block has the principle duty of maintaining protocols while switching networks is responsible for the 'MANAGEMENT' functionality. The primal duty of the fog is to compute data of a local area, and send the required data to the cloud in an encrypted and secure manner. The fog node comes under the functional category of 'SECURITY' and 'COMMUNICATION' along with some functionality of 'SERVICES'. Lastly, the cloud, whose main aim is to store data, compute data gathered globally and assist in proper decision making comes under 'SERVICES' and 'APPLICATION'.

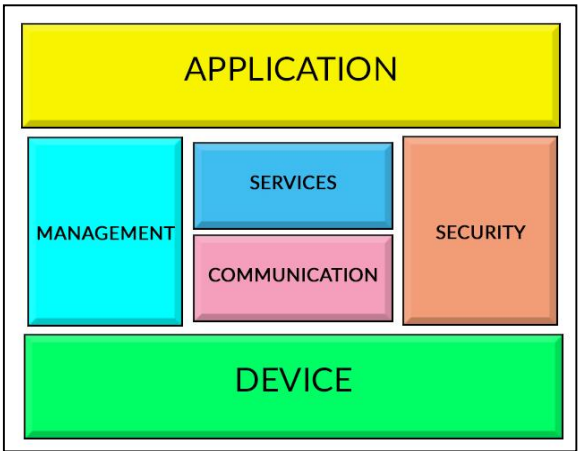


Figure 3: Mapping: IoT devices account for device functionality, Gateway accounts for the management functionality, Fog accounts for the security, services and communications functionality and the cloud accounts for the application functionality.

2. HEAD CLUSTERING ALGORITHM

Let us take up a scenario, where there are many number of IoT devices which are concentrated in a particular region. Let's say they all respond/receive signals from/to a single fog node. This situation again becomes similar to the one we were facing with the cloud computing itself but in a smaller scale. So, is it possible to reduce these computations too? Yes, it is possible. It would be

to define a set of IoT devices as a cluster/region using the pre-defined sequence of conditions and after identifying this region, a cluster head is selected based on a condition. This situation will be even more favorable if these regions that are formed are dynamic in nature based on a fixed parameter or value. So, let us assume these regions are made, and now a cluster head needs to be assigned. A cluster head is an IoT device among all IoT devices which collects all the data of the region and then transfers it to the nearest fog node. The cluster head is decided based on the battery level and power consumption. The device which appears to be most sound in these two aspects i.e. high battery level and low power consumption is chosen as the cluster head. Once, the battery of the cluster head is drained to a level below the desirable percentage, the cluster head of that region is changed to another IoT device with a higher battery level and thus saving huge amount of power consumption. But, the only assumption in this theory is that, the fog node is the node which has continuous power supply. A brief outline of the above is shown in Figure 4.

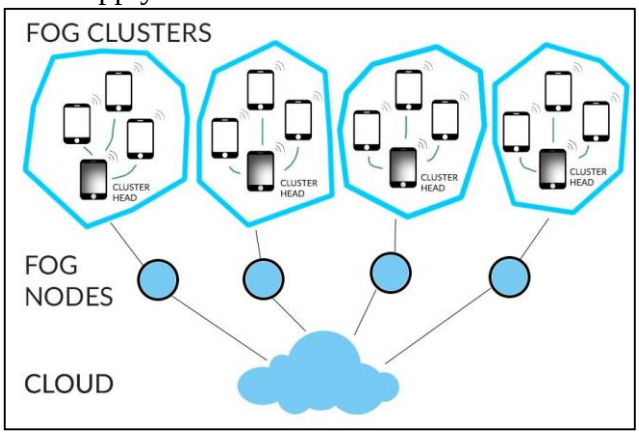


Figure 4: Each cluster signifies a separate region. The differently colored mobile device in each cluster is the cluster head which takes data from the remaining mobile devices of the respective cluster and transfers it to the fog node which transfers it to the cloud.

3. STRUCTURAL HEALTH MONITORING

The natural disasters like earthquake, floods and landslides cause tremendous amount of damage to the health of an infrastructure which in-turn puts human life in danger. Adding to this is corrosion, fatigue, scour etc. which is the outcome of organic aging of a structure. Health of a structure deteriorates with the passage of time and gets catalyzed with the onset of these natural calamities'. [17] Even though these detrimental disasters are not in our control, however the extent of damage brought on by these unfortunate happenings can be controlled by us to some extent. This can be done by carefully monitoring the health of structures thereby ensuring its robustness and stability in the wake of a climatic mishap. But that in itself isn't sufficient, this monitoring system in place needs to be fast enough to be allow consequent action to take place. It should be able to qualitatively as well as quantitatively give real time analysis of a structure's health and it's time to live. This purpose is served by a Structural Health Monitoring (SHM) model.

SHM aims to provide, a diagnosis of the materials present, of different parts, and the full assembly of parts constituting the structure as a whole at every moment during the life of a structure. SHM has other functionalities too. It also involves the integration of sensors, possibly smart materials, data transmission, computational power, and processing ability inside the structures. It therefore, makes it possible to change or reconsider the design of the structure and the full management of the structure itself and of the structure considered as a part of wider systems.

According to [18], SHM application can be seen as a four-step procedure. This four-step procedure has been illustrated in figure 6. The first step being Detection, which self explanatorily gathers data pertaining to the presence of any damage in the structure. The localization step comes after the detection, where the location of the damage detected in the previous step is estimated. Once we know that there is damage and the probable location of the damage, we carry out the assessment method which estimates the extent of this damage. The final step provides information about the remaining lifetime of the structure for maintenance purposes as well as disaster management.

Another step has been introduced into this process that lies between damage localization and assessment by the authors of [18] and [19]. As per these two publications, a new aspect i.e. the type of damage has been given importance by emphasizing that before estimating the extent of the damage, classification should be done to figure out the type of damage that exists. Combining all these detailed steps of SHM, it can be executed in the following manner in an orderly fashion :1) detection; 2) localization; 3) classification; 4) assessment; and 5) prediction. [20]

SHM application can be broken down into three components as mentioned in [19] and illustrated in figure 5. These components are:

1) Sensing and gathering data: Sensors gather data from structures under changing environmental conditions. This stage also deals with transmitting this data collected from sensors to the base station or server. This part consists of thinking about the type of sensor to deploy, where to deploy them and how many of them to be deployed and also their mode of transmission to the server. 2) Data Management: Once the data reaches the server, it needs to be evaluated and accessed. This careful assessment and data processing comes under this component. At times, the data reaching the server is overwhelming and contains lot of noise and unwanted data. Pre-processing methods need to be carried out to extract only the meaningful data out of the vast volumes of incoming data. The main processing deals with distinguishing data to tell whether building is damaged or not. There are a variety of techniques that are used to analyze the sensor network data ranging from machine learning to genetic algorithms. There are a broad number of algorithms that have been put forward but the one which has gained significant appreciation lately is novelty detection based on artificial neural networks [21]- [23]. 3) Data Access and Retrieval: This component is the final outcome of our SHM application. After data analysis and processing, results are examined for appropriate decision making.

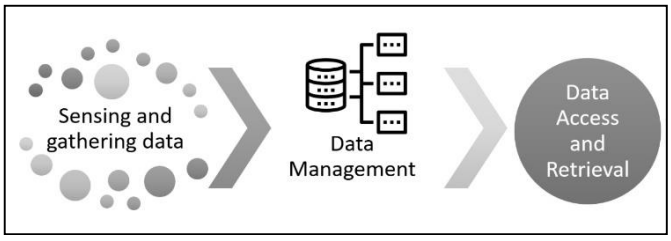


Figure 5: The 3 components of Structural Health Monitoring

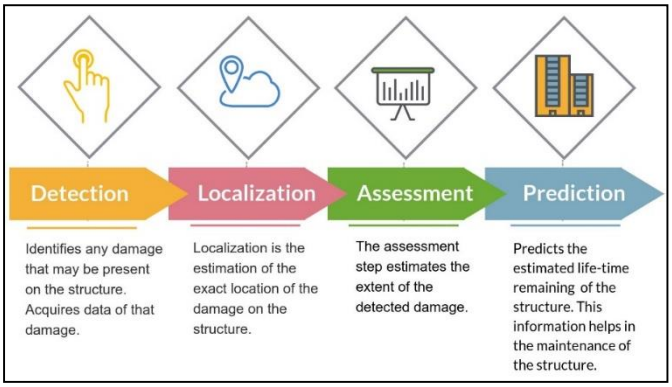


Figure 6: The four-step structural health monitoring process starting with detection of any damage on our structure and ending with the prediction phase i.e. the time remaining of a structure.

3.1 Implementation of SHM: Traditional Systems

While we've established the need of SHM and its components. The next thing which needs to be considered is how should SHM be implemented such that it's fast, reliable and actually serves the purpose of providing real time data about the health of a structure. Let's take a scenario to get a better understanding. Let's take the example of a building instead of considering any arbitrary structure. Traditionally, a wired approach for SHM was adopted. However, this traditional model increases latency due to prolonged deployment time, it is also not economical as a heavy cost goes into wire installation. [24] Apart from this, the amount of data from numerous structural data sources are very vast which brings forward the idea of bigdata that cannot be handled by traditional database management systems. [20]

3.2 WSNs within SHM

To overcome the challenges posed by the traditional wired approach, wireless sensor networks are now being adopted for SHM. WSNs have immense potential in the field of SHM that is still being explored and bettered. WSNs are far ahead of their counter traditional wired systems when it comes to SHM mainly due to its simplicity and ease of installation. In WSNs, sensors are strategically placed over the surface area of our building in consideration. The sensors are placed such that they are able to capture the building's response (like vibrations) which are caused by external factors like the wind, earthquakes or dynamic loads. [25]

While WSNs reduce cost and deployment time in comparison to wired systems but it surfaces certain problems of its own. SHM applications have a high sampling frequency (greater than 100s of Hz) which leads to a very large amount of data which in turn reaches huge values for each sensor in one round of data elicitation. This makes the entire SHM application data intensive. Also, with so much data, there is often a good amount of data that is extra and not needed which also gets transmitted. [26] WSNs also have the constraint of limited bandwidth, this low bandwidth again fails to handle the tremendous volume of data. For this, data compression techniques might help but they might also increase latency. A WSN network might contain multiple communication hops before the data is actually transmitted, this can delay the entire procedure. But the biggest limitation of WSNs is that sensors have a fixed battery life and energy supply. Additionally, If the sensor battery drains out, it is problematic to charge them over big structures such as our building. [27]

4. OUR PROPOSED MODEL

To go a step further after WSN in SHM, some of these problems can be easily tapped by the Internet of things. This will enable powerful data processing beyond the scope of WSN. The introduction of cloud will enhance storage and transmission of the big data. To go even beyond, comes the idea of the latest cloud technology i.e. fog computing. This is where our proposed model comes into play. So far, we've seen how SHM has been implemented from wired to WSN and also using Internet of things and cloud computing but the major challenge it continues to face is of battery consumption. Our model proposes to tackle this problem by maximizing the network lifetime. This can be done by integrating the fog head cluster node algorithm discussed in the previous section. Let's showcase the algorithm specific to SHM.

Different regions with buildings to monitor will be separated and identified as clusters. Each of these clusters will comprise of few buildings and these buildings will contain sensors deployed on them. Broadly speaking, a cluster will be a collection of sensors. Each cluster will have its own cluster head. This cluster head will be a sensor from all the sensors of the cluster chosen dynamically depending on its battery level. The sensor having the highest battery level compared to all other sensors in its network will be assigned as the cluster head. Initially, any one sensor will be set as the default head. Every other sensor in the network will be responsible for acquiring data from the structure and sending this data to the head sensor. Once the SHM network starts functioning, different sensors will have varied battery consumption levels. This variation will be due to different communication hops to the head node and due to the variable Euclidean distance [25] between sensors. Besides transmission, coverage and data acquiring also takes up significant amount of energy. [28] With the battery of each sensor changing erratically, there will always exist a difference between the battery level of all the sensors i.e. some sensors may be closer to the head resulting in less power consumption and a higher battery level whereas some sensors might be far away from the head leading to higher power consumption and lesser battery levels. This change in battery levels will also cause change in head of the cluster.

Furthermore, the head cluster needs to additionally forward all this data incoming ahead for data processing and analysis leading to significant power consumption of the head itself. So, to ensure no rapid change in assignment of head sensor is taking place, the assignment of one head sensor to another will only take place if the difference in battery level is significant enough. Once all the data for a cluster is collected by the head sensor. The head sensor is responsible for sending this data ahead to the fog node, the fog will act as a descended cloud where the data analysis and processing will take place leading to the decision making.

The data sent over to the fog can be categorized into two categories: first, is the real-time data which detects damage in the building. In such a case, this real-time data in the fog can warn and create an alert to vacate the building and prevent the building from collapsing. In this case, no latency can be afforded, hence the data isn't sent over to the cloud and dealt with at the fog level itself. The second category is the real-time data which doesn't detect any immediate damage in the building but can be used to predict the lifetime of the building and can be used for the maintenance of the building. This data can be stored in the cloud instead of the fog node. This data can be checked remotely from a variety of devices not depending on the location of the cluster, this will help in the restoration of buildings spread out at different locations.

Now for this to implement successfully, lets closely discuss the kind of sensors that'll make it possible. The network of sensors is known as a sensor network. Each sensor in a sensor network notes measurement in accordance to the time-stamp along with the physical measurements like heat, sound, light, pressure, or even motion. These physical measurements are then represented as data and stored into its own storage. Later when a query reaches this sensor though the sensor

network it sends the data, of what is asked in the query to the particular node in the network. The sensor itself doesn't have enough memory space to hold the continuous data it gets. Since, these types of sensors get data very frequently. Because of that the old data is compressed and most of the time erased to make space for the new data. So, the question arises that if we make one node as the head node and that too dynamic head node where is this data going to be stored?

It is possible for each sensor to be connected to an external storage device. So, this solves the problem of storage. Secondly, to save power we can turn on the power of this secondary storage device only when that sensor connected to it is made the head node. So, this head node will then send the query to all the other sensors in its cluster who will in turn send the data back to the head node. When the data is arrived from all the other sensors the data is sent to the fog node. The different sensors that are used in this are temperature sensors, strain gauges, accelerometers, anemometers, seismometers, load cells etc. [20].

This proposed mechanism tackles problems being faced by traditional systems and WSNs head on. The problem of big data is solved by introducing the fog node and forming clusters. Every sensor can handle the data within it's cluster, hence solving the problem of big data. Basically, the scalability of SHM is reducing by breaking it down into clusters. Data aggregation will happen efficiently with data being categorized at the fog node, which removes any noise and either keeps the data or sends it to the cloud where it's due processing is done. The head sensor algorithm maximizes the lifetime of the entire WSN network not completely eradicating the problem of power supply but optimizing it to an extent. [24]

Moreover, the integration of SHM, IoT, cloud and fog computing will invite ubiquitous services and provide scope for powerful data analysis techniques which can process the data streams beyond the capabilities of previous systems. The Iot architecture uses SHM data intelligently for smart monitoring and actuation with smart devices. [25] Combining all aspects, the final architecture of SHM is shown in Figure 7.

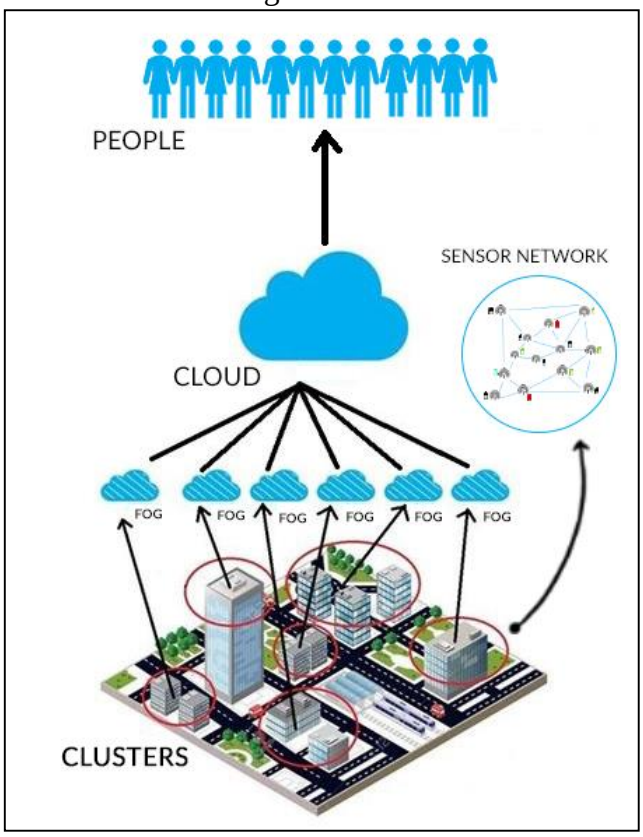


Figure 7: The architecture of our SHM proposed model

Algorithm 1 The algorithm for assigning head nodes

Input: *A predetermined located set of sensors, N. A set of buildings in which SHM has to be implemented B, A set of fog nodes located in each cluster, F → Fi is the corresponding fog node for the cluster Ci.*
Output: *A global cloud K with complete data of the structures in review from all clusters C1…. Cn covering all buildings in B*
Step 1: Define the clusters C1…Cn depending upon the area and buildings such that each building in B is covered by C AND each sensor in N is present in any C
Step 2: Divide N in sets → all sensors in a cluster form one set
Step 2: **for** *each Ci for i= 1 to n* **do**
Step 3: assign a Head Sensor H with the maximum energy
Step 4: **if** *(no of H >1)*
check Euclidian distance of each N in Ci with all H
choose min dist. (all H) and mark it as real H
end if
Step 5: define a query Q asking for sensed data
Step 6: H sends Q to all N in Ci
Step 7: All N in Ci send the required data back to H
Step 8: **if** *(H receives all the data from each N in Ci)*
send the gathered data to Fi
end if
Step 9: **when** *(Fi receives the data)*
calculate the damage D
if *(D is critical)*
issue alert
else
send the data to the cloud K
end if
end when
Step 10: K calculates the damage and the lifespan of the building. Data stored in K is accessible for maintenance purposes
end for

This algorithm is depicted with the help of flowcharts in figure 8 and Figure 9 and the change of the head sensor within a cluster is depicted in figure 10

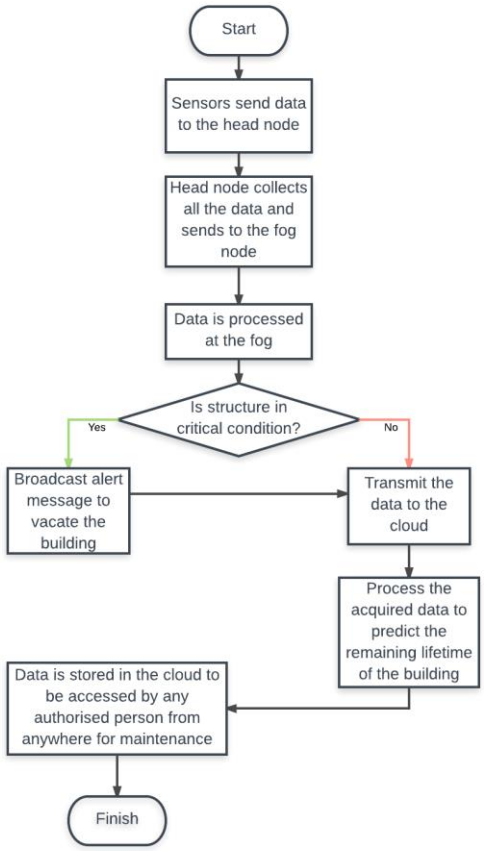


Figure 8: The flowchart depicts how the fog decides whether or not to send data to the cloud depending on the emergency of the situation and the time left on hand by processing the data sent to it from the head sensor.

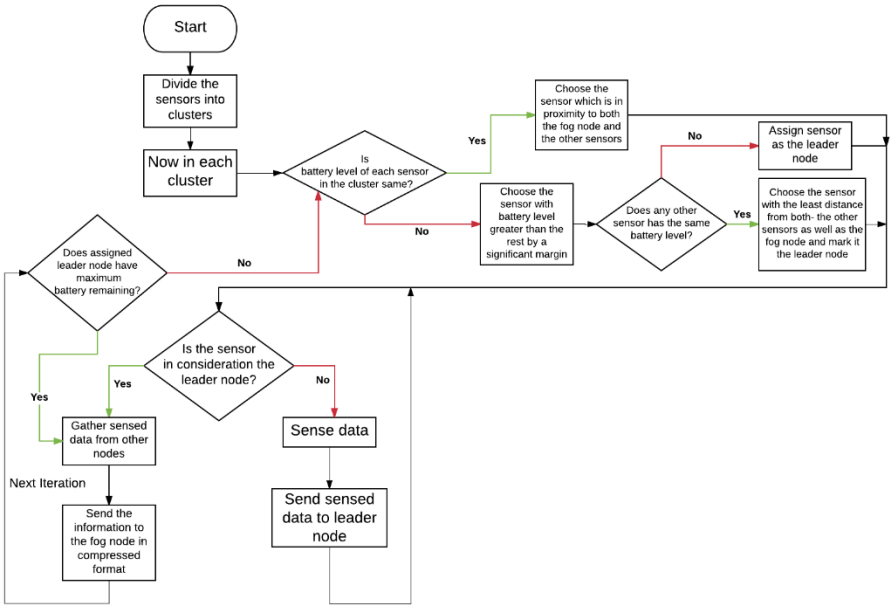
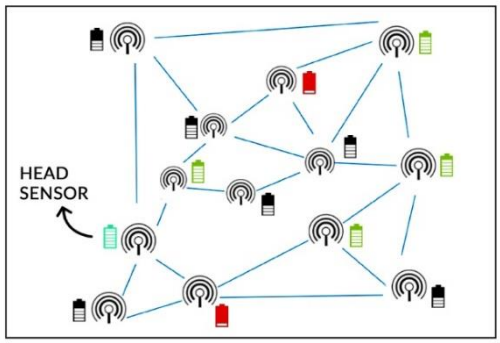
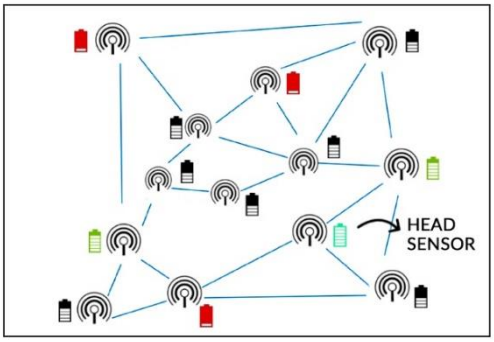


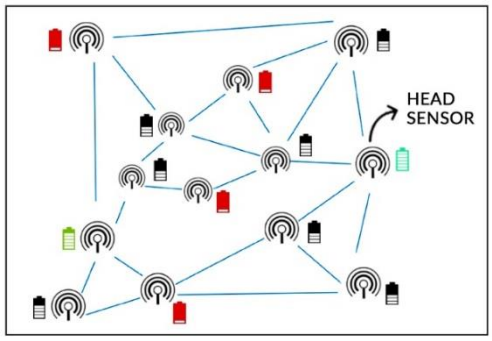
Figure 9 : Data flow of the proposed system model



(a)



(b)



(c)

Figure 10: (a) (b) and (c) depict how the battery of different sensors are changing over time which in turn is changing the assignment of the head sensor. The red batteries depict low power left while the green ones depict high power left, the remaining are having moderate power left. (a) shows the network sensor in its initial stage where most sensors have high battery and the one having highest is assigned as head sensor, (b) and (c) showcase the same network over time where the battery of sensors drop and the head sensor changes depending on the one with highest power remaining.

5. SYSTEM MODEL (A)

Our model comprises of n sensors forming a piconet and communicating with each other via bluetooth LE. The head sensor is present from one of these sensors and again uses bluetooth LE technology to transmit data to the fog. The fog then performs processing tasks on the data and sends it to the cloud through a 5G heterogenous network. The fog node can take care of delay sensitive data and pass on the rest for the cloud to process.

Let's first pay close attention to the 5G network between fog node and cloud server -

1. Power consumption of the fog node :

Our model comprises of multiple fog nodes,for a single fog node α,the power consumption covers the computation of data at the fog node as well as the transmission of data from the fog to cloud via 5G network.

There are two basic properties that the power consumption function of the fog should adher to as mentioned in [29] : a) the computational power consumption is directly propotional to the computation amount b) marginal power consumption is increasing for every fog node.

So,the power consumption function can be written as a function of the computation amount of fog node α i.e. x_α

$$P_{fog,\alpha} = ax_\alpha^2 + bx_\alpha + c \tag{1}$$

where a > 0 and b,c ≥ 0 are pre-determined parameters.

2. Transmission rate from fog to cloud :

This transmission is done via 5G network communication. Considering fog node α,it can handle the data in 2 ways. It can either do computation at the fog node or it can pass on the data to the cloud server depending on the level of latency and criticality of the situation and allow the cloud node to take care of the data. Let the data to be dealt with by fog node α be seen as a task T_α. We will use $a_{i,j,k}=1$ to denote that fog i chooses to accomplish the task T_α by transmitting to cloud server j through the channel k. On the other hand, $a_{i,j,k}$ = 0 denotes that the fog does not pass on data to cloud and does computation on it's own. j belongs to J ={1,2,3…J} which are the number of cloud servers. And i belongs to I={1,2,3…..I} which are the number of fog nodes. K belongs to K={1,2,…K}. Both the fog nodes and cloud servers operate in the same frequency band for the efficient usage of the spectrum. This spectrum is categorized into K channels of identical bandwidth W. These different channels are denoted by K. [30]

If $a_{i,j,k}$ = 1

The transmission rate from fog node α can be formulated through channel k :

$$r_{\alpha,k} = W \log_2\left(1 + p_\alpha g_\alpha / I_{\alpha,k} + \sigma^2\right)$$

p_α is the power of the fog node. g_α is the channel gain between the fog node and the cloud. $I_{\alpha,k}$ denotes the interference at the cloud on channel k, due to other fog nodes trying to transmit data to cloud. σ2 is the background noise power.

The total transmission rate from a single fog node to it's respective cloud server can be calculated as $r_i = \sum_{k=1}^{K} a_{i,2,k} r_{i,k}$

3. Power consumption of cloud server

For the task T_α , d_α is the volume or size of the input data which needs to be computed. Let c_α be the computing ability required for accomplishing this task. Unit of measurement for the computing ability is the number of CPU cycles.

The total power consumption by cloud can be formulated by

$$\text{Energy e} = b_\alpha p_\alpha d_\alpha / r_i + c_\alpha \delta^R \quad (2)$$

where δ^R is the cost of energy required by the cloud for a single CPU cycle. The cloud will always have higher computation capacity as compared to the fog,which is δ^R. The no. of channels that are utilized by fog node α to transmit data to the cloud is showcased using b_α

$$b_\alpha = \sum_{k=1}^{K} a_{i,2,k}$$

5.1. Constraints

Latency constraints are not specified as they vary depending on the data and the criticality of the damage of the civil structure. Other constraints which have been taken into account are mentioned below:

5.1.1 Fog-cloud workload balance

Both the fog nodes as well as the cloud servers have computational constraints due to which the workload or data that needs to processed should be balanced between the two. Thus,not draining any one completely. For this the size of cluster needs to be managed i.e. the number of sensors

attached to a fog node should be in check. Even if all the data that the fog node is receiving turns out to be critical and latency sensitive then all the data will have to be computed by the fog. The fog should be able to handle such a situation. The other extreme could be when all of the data needs to be processed by the cloud alone. In that case,the cloud should be well euipped to handle all the data.
The workload allotted to the fog and cloud respectively is shown by X and Y

Let L denote the total sensor input from all the sensors. The incoming data or traffic from all sensors to the fog node α can be denoted by T_α

Thus, $\mathrm{T}_\alpha = X + Y$ should hold good

5.1.2. Fog area Constraint

Every fog decive has a limit on its processing ability. Clusters should be formed in such a way that the fog node can handle the incoming data and workload from all sensors belonging to that cluster. For any fog device α, $p_{\max}$ denotes the maximum computational capacity it carries out. Also,the workload x_α on a fog node should not be more than the rate of incoming traffic t_α [25]

Therefore,we can say

$$0 \quad \leq x_\alpha \leq \min\{ p_{\max} , t_\alpha \}$$

5.1.3. Single Channel Constraint

A single channel can be allocated to atmost one fog node

$$\sum_{i=1}^{N} a_{i,2,k} = 1 , \text{ i } \in \text{N,j } \in \{2,3\}$$

5.2. Problem :

It is important and desirable to minimize the aggregated power consumption of all fog nodes and cloud servers. The complete power consumption of the 5G networked system can be represented as the combined power consumption of the fog node and the cloud server. In order to minimize the power consumption of the system,the individual power consumption of the fog and cloud needs to be minimized.

$$P^{total} = \sum_{i \in N} P_i^{fog} + \sum_{j \in M} P_j^{cloud}$$

5.3. Solution :

The equation of power consumption of a fog node as in equation (1) is a quadratic function which is monotonically increasing and a strictly convex function. [25] This function can be plotted to find the relation between the power consumption with the computation amount x_α for a fog node α. This relation is plotted in Figure 11,the minima for power consumption will be attained when $x_\alpha = -b/2a$ as showcased in the graph. This is when Power consumption of the fog node will be minimum.

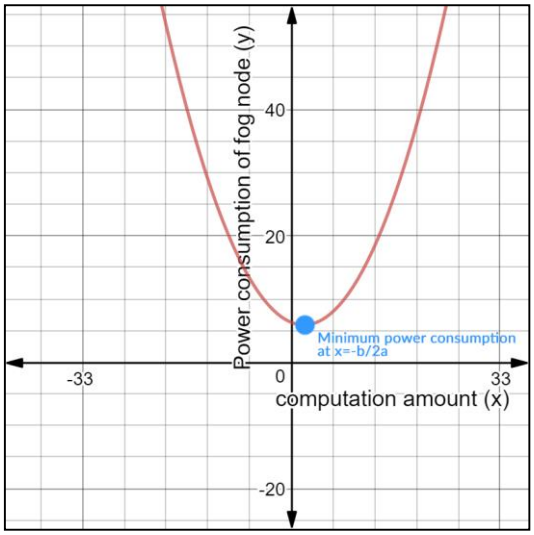


Figure 11: Graph depicted the relationship between the power consumption of the fog and the computation amount of the same fog node. The minima is obtained at a point where the x coordinate i.e. the computation amount of the fog node= -b/2a where a and b are from equation (1)

The second part is to minimize the power consumption of the cloud server as well. This is denoted by equation (2) , in order to do so reduction followed by applying EECO scheme illustrated in [30] which will get the minimum value.

6. SYSTEM MODEL (B)

In our proposed model, the sensors are divided into clusters and each cluster has one fog node. Each cluster has a head node and it dynamically keeps on changing based on the criteria that if the battery level of that sensor decreases by a significant margin. These sensors transmit data through Bluetooth and so it is really important to predict the remaining battery life of the sensor to be able to assign the head node. Here each cluster is known as "piconet" where the head node is the "master" and the other sensors are the "slaves". The following mathematical model helps in predicting the remaining battery life of the sensors.

To estimate the remaining battery life in the sensors we have taken inspiration from the method used in [31] where CC2540 Bluetooth development kit is used. Each sensor goes through the following stages in a single cycle of data flow:

Stage 0: Initial Sleep Mode-The slave component is in the sleep mode until the master sends a request to send the data

Stage 1: Wake up mode -Whenever the master sends the request for transmitting the required data there is a spike in current followed by slight decrease in the flow of current. This marks that the sensor is ready to send some data.

Stage 2: Pre- Processing Mode-Soon after the request is received by the slave from its master, the storage space for the data read by the sensor is made available and the radio is prepared to store the data in form of stack in the storage device and also to transmit the data to the master.

Stage 3: Receiving Mode-The radio receives the data from the sensor and stores it in the available space made during pre-processing.

Stage 4: Receiving mode to Transmitting Mode -The radio stops receiving data from the sensor, the data packet is made ready to be transmitted and the radio gets ready to transmit the data.

Stage 5: Transmitting Mode-The radio transmits the data packet to the master

Stage 6: Post transmitting Mode-Sleep mode protocol is put into effect and the countdown is initiated

Stage 7: Final Sleep Mode-The slave again goes back to sleep

All the sensors would be in a connected state with a 1 second connection interval and zero slave latency. This means that each sensor would be connected to the master for an interval of 1 second and during that period of time it will undergo the above-mentioned stages. Here, stage 0 and stage 7 would be known as the sleep modes of the cycle while stage 1-6 would be considered the awake modes of the cycle. This classification is necessary because maximum energy would get utilized when the sensors would be in awake mode.

As the sensor has to send different amount of data and in different quantities in each cycle so, there are possibilities that the sensors will remain in awake mode for different interval of time. The cases of different awake time occur due to the possibility of sensor taking more time to read data, process data, transmit data, or form data packets etc. Here is the distribution of amount of time spent on each stage when the awake mode interval was the shortest (2.4ms) and when the awake mode was the longest (2.8ms).

Table 1: Time distribution in shortest awake time

	Time [µs]	**Current [mA]**
State 1	400	6.0
State 2	298	7.8
State 3	369	18.4
State 4	103	17.5
State 5	109	18.4
State 6	1177	7.8

Table 2: Time distribution in longest awake time

	Time [µs]	**Current [mA]**
State 1	400	6.0
State 2	329	7.8
State 3	393	18.4
State 4	116	17.5
State 5	150	18.4
State 6	1393	7.8

Secondly, during the sleep mode of the cycle there will be 0.0009mA current flowing through the circuit. Moreover, we assume that on an average 0.0001mA of current would be lost in the printed circuit board (PCB) due to unknown resistance and thus on an average around 0.001mA of current is used while in sleep mode. Let this current be known as I_{sleep}.

The following steps would be necessary to estimate the amount of battery level in the sensor:

Step 1: Find the average current used while in awake mode. Let us denote this current as $I_{awake-avg}$.

From above tables, we know that each state in the awake mode has different time period and different current so let us denote the time taken to complete state 1 as t_1, for state 2 as t_2 and so on till state 6. Similarly, let us denote the amount of current generated during state 1 as I_1, state 2 as I_2 and so on until state 6.

So, the total awake time t_{awake} will be:

$$t_{awake} = t_1 + t_2 + ... + t_6$$

Now, let us find $I_{awake-avg}$ which will be given as:

$$I_{awake-avg} = [(t_1 * I_1) + (t_2 * I_2) + ... + (t_6 * I_6)] / t_{awake}$$

So, $I_{awake-avg}$ for table I would be:

t_{awake1} = (400 µs) + (292 µs) + (369 µs) + (103 µs) + (109µs) + (1177 µs) = (2456 µs)

$I_{awake-avg1}$ = [(400 µs) *(6 mA) + (292 µs) *(7.8 mA) + (369 µs) *(18.4 mA) + (103µs) * (17.5 mA) + (109µs) *(18.4mA) + (1177 µs) *(7.8 mA)]/ (2456 µs) = 9.9576 mA

The average current consumption during a single connection event with the shortest slot, a wake time of 2.456ms, is calculated to be approximately 9.9576 mA.

Similarly, $I_{awake-avg}$ for table II would be:

t_{awake2} = (400 µs) + (329 µs) + (393 µs) + (116 µs) + (150µs) + (1393 µs) = (2781 µs)

$I_{awake-avg2}$ = [(400 µs) *(6 mA) + (329 µs) *(7.8 mA) + (393µs) *(18.4 mA) + (116 µs) *(17.5 mA) + (150µs) *(18.4mA) + (1393 µs) *(7.8 mA)]/ (2781 µs) = 10.0154 mA

The average current consumption during a single connection event with the longest slot, a wake time of 2.781ms, is calculated to be approximately 10.0154 mA.

Step 2: Now, to find the average current over the entire connection interval

Average current (I_{avg}) is the amount of current dissipated in the entire connection interval ($t_{interval}$).

As discussed earlier $t_{interval}$ = 1000ms, and average current is given by:

$$I_{avg} = (t_{interval} - t_{awake}) * I_{sleep} + t_{awake} * I_{awake-avg}$$

Average current (I_{avg1}) for the values in table I would be:

$$I_{avg1} = (t_{interval} - t_{awake1}) * I_{sleep} + t_{awake1} * I_{awake-avg1} =$$

= [(1000 ms – 2.456 ms) *(0.001 mA) + (2.456 ms) *(9.9576mA)] / (1000 ms) = 0.0254 mA

Average current (I_{avg2})for the values in table II would be:

$$I_{avg2} = (t_{interval} - t_{awake2}) * I_{sleep} + t_{awake2} * I_{awake-avg2} =$$

= [(1000 ms – 2.781 ms) *(0.001 mA) + (2.781 ms) *(10.0154 mA)] / (1000 ms)
= 0.0279 mA

To find the average current of the sensor over any period of any awake mode interval, the number of times the shortest wake time and the number of times the longest wake time occurred would be noted and weighted average method would be used to find I_{avg} of the sensor.

Let us say that the shortest wake time occurred for 45% of the time (P_s) and the longest wake time for the other 55% (P_i). So, by calculating the weighted average based on the above values we get:

$$I_{avg} = [(P_s) * I_{avg1}] + [(P_i) * I_{avg2}]$$

Therefore, taking values from above we get

$$I_{avg} = [(0.45) * 0.0254mA] + [(0.55) * 0.0279mA] = 0.0268\text{mA}$$

Hence the average current consumption while the device is in a connected state is approximately 0.027 mA (27 μA).

Step 3: The final step would be to find out the battery life (B_{Life}) of the sensor. Battery life of a sensor can be known by dividing the available battery or the given battery capacity ($B_{Capacity}$) with average current consumption (I_{avg}).

$$B_{Life} = B_{Capacity} / I_{avg}$$

The available battery or the battery capacity of the sensor was 260mAh.
Therefore,

$$B_{Life} = (260\text{mAh}) / (0.027\text{ mA}) = 9630\text{ hrs (approximately)}$$

Thus, it can be confirmed that the battery of the sensor will last approximately another 9630 hrs which is equivalent to approximately 401 days, while continuously running in a connected environment with a 1 second connection interval and zero slave latency.

From the above steps, it can be concluded that at any given moment, the available battery life of any sensor can be calculated. As more and more time slots are noted, the available battery life becomes more accurate. Thus, this method helps in choosing the head node in the cluster dynamically.

7. CONCLUSION

This paper has introduced the concept of fog computing within structural health monitoring. The entire system model consists of a piconet among sensors. These sensors are divided into clusters where each cluster follows a head sensor clustering algorithm which enables lesser power consumption as compared to normal transmission by all the sensors of the cluster to the fog. The sensor with the maximum battery remaining is assigned as the head sensor. To be able to assign

the head sensor, we need to be able to find the battery remaining in a sensor at any given point while it is undergoing Bluetooth transmission, this has been validated in system model (B). Once the head sensor transmits data to the respective fog node of the cluster, the criticality of the data received at the fog is checked. If the data indicates immense structural damage and less time on hand. The fog does the decision-making part and issues an alert to vacate the building else the data is sent to the cloud via 5G network where it is processed and stored for structural maintenance. This transmission between the fog and node needs to be done such that minimum power consumption of the two takes place which is taken into consideration in our system model (A).

REFERENCES

[1] Ejaz, W., Anpalagan, A., Imran, M. A., Jo, M., Naeem, M., Qaisar, S. B., & Wang, W. (2016). Internet of Things (IoT) in 5G wireless communications. *IEEE Access*, 4, 10310-10314.

[2] Dastjerdi, A. V., & Buyya, R. (2016). Fog computing: Helping the Internet of Things realize its potential. *Computer*, 49(8), 112-116.

[3] Tang, J., & Quek, T. Q. (2016). The role of cloud computing in content-centric mobile networking. *IEEE Communications Magazine*, 54(8), 52-59.

[4] Sun, X., & Ansari, N. (2016). EdgeIoT: Mobile edge computing for the Internet of Things. *IEEE Communications Magazine*, 54(12), 22-29..

[5] Alippi, C., Fantacci, R., Marabissi, D., & Roveri, M. (2016). A cloud to the ground: The new frontier of intelligent and autonomous networks of things. *IEEE Communications Magazine*, 54(12), 14-20.

[6] Lu, R., Liang, X., Li, X., Lin, X., & Shen, X. (2012). Eppa: An efficient and privacy-preserving aggregation scheme for secure smart grid communications. *IEEE Transactions on Parallel and Distributed Systems*, 23(9), 1621-1631.

[7] Deng, R., Lu, R., Lai, C., Luan, T. H., & Liang, H. (2016). Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption. *IEEE Internet of Things Journal*, 3(6), 1171-1181.

[8] Aazam, M., & Huh, E. N. (2016). Fog computing: The cloud-iot\/ioe middleware paradigm. *IEEE Potentials*, 35(3), 40-44.

[9] Masip-Bruin, X., Marín-Tordera, E., Tashakor, G., Jukan, A., & Ren, G. J. (2016). Foggy clouds and cloudy fogs: a real need for coordinated management of fog-to-cloud computing systems. *IEEE Wireless Communications*, 23(5), 120-128.

[10] Vallati, C., Virdis, A., Mingozzi, E., & Stea, G. (2016). Mobile-edge computing come home connecting things in future smart homes using LTE device-to-device communications. *IEEE Consumer Electronics Magazine*, 5(4), 77-83.

[11] Satyanarayanan, M. (2017). The emergence of edge computing. *Computer*, 50(1), 30-39.

[12] Sharma, S. K., & Wang, X. (2017). Live Data Analytics With Collaborative Edge and Cloud Processing in Wireless IoT Networks. *IEEE Access*, 5(99), 4621-4635.

[13] Jingtao, S., Fuhong, L., Xianwei, Z., & Xing, L. (2015). Steiner tree based optimal resource caching scheme in fog computing. *China Communications*, 12(8), 161-168.

[14] Chiang, M., Ha, S., Chih-Lin, I., Risso, F., & Zhang, T. (2017). Clarifying fog computing and networking: 10 questions and answers. *IEEE Communications Magazine*, 55(4), 18-20.

[15] Konsek, H. (2015, August 18) The Architecture of IoT Gateways [Online]. Available: https://dzone.com/articles/iot-gateways-and-architecture

[16] Desai N. (2016, August 30) What is an IoT Gateway and How Do I Keep It Secure? [Online]. Available: https://www.globalsign.com/en/blog/what-is-an-iot-gateway-device/

[17] Bhuiyan, M. Z. A., Wang, G., Cao, J., & Wu, J. (2015). Deploying wireless sensor networks with fault-tolerance for structural health monitoring. *IEEE Transactions on Computers*, 64(2), 382-395.

[18] Rytter, A. (1993). Vibrational based inspection of civil engineering structures (Doctoral dissertation, Dept. of Building Technology and Structural Engineering, Aalborg University).

[19] Worden, K., & Dulieu-Barton, J. M. (2004). An overview of intelligent fault detection in systems and structures. *Structural Health Monitoring*, 3(1), 85-98.

[20] Tokognon, C. A., Gao, B., Tian, G. Y., & Yan, Y. (2017). Structural health monitoring framework based on Internet of Things: A survey. *IEEE Internet of Things Journal*, 4(3), 619-635.

[21] Dackermann, U., Smith, W. A., & Randall, R. B. (2014). Damage identification based on response-only measurements using cepstrum analysis and artificial neural networks. *Structural Health Monitoring*, 13(4), 430-444.

[22] Bandara, R. P., Chan, T. H., & Thambiratnam, D. P. (2014). Structural damage detection method using frequency response functions. *Structural Health Monitoring*, 13(4), 418-429.

[23] Kalafat, S., & Sause, M. G. (2015). Acoustic emission source localization by artificial neural networks. *Structural Health Monitoring*, 14(6), 633-647.

[24] Abdelgawad, A., & Yelamarthi, K. (2016, October). Structural health monitoring: Internet of things application. In *Circuits and Systems (MWSCAS), 2016 IEEE 59th International Midwest Symposium on*(pp. 1-4). IEEE.

[25] Elsersy, M., Elfouly, T. M., & Ahmed, M. H. (2016). Joint optimal placement, routing, and flow assignment in wireless sensor networks for structural health monitoring. *IEEE Sensors Journal*, 16(12), 5095-5106.

[26] Liu, X., Cao, J., Song, W. Z., Guo, P., & He, Z. (2015). Distributed sensing for high-quality structural health monitoring using WSNs. *IEEE Transactions on Parallel and Distributed Systems*, 26(3), 738-747.

[27] Aygün, B., & Cagri Gungor, V. (2011). Wireless sensor networks for structure health monitoring: recent advances and future research directions. *Sensor Review*, 31(3), 261-276.

[28] Lu, Z., Li, W. W., & Pan, M. (2015). Maximum lifetime scheduling for target coverage and data collection in wireless sensor networks. *IEEE Transactions on vehicular technology*, 64(2), 714-727.

[29] Deng, R., Lu, R., Lai, C., Luan, T. H., & Liang, H. (2016). Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption. *IEEE Internet of Things Journal*, 3(6), 1171-1181.

[30] Zhang, K., Mao, Y., Leng, S., Zhao, Q., Li, L., Peng, X. & Zhang, Y. (2016). Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks. *IEEE access*, 4, 5896-5907.

[31] Kamath, S., & Lindh, J. (2010). Measuring bluetooth low energy power consumption. Texas instruments application note AN092, Dallas.

[32] Icon credit in figures 2,10 and 5: https://thenounproject.com

