

Smart Dog Tracking Collar

Ashwin Vallaban, Anika Rai, Sonya Nadar, Akansha Patnaik

Project Guide Prof. Sneha Revankar

Sneha Revankar is a professor in the Department of Electronics and Telecommunications at Fr. C. Rodrigues Institute of Technology, Vashi, Maharashtra, India

Ashwin Vallaban is currently pursuing his bachelors in Electronics and Telecommunication from Fr. C. Rodrigues Institute of Technology, Vashi, Maharashtra, India

Sonya Nadar is currently pursuing her bachelors in Electronics and Telecommunication from Fr. C. Rodrigues Institute of Technology, Vashi, Maharashtra, India

Anika Rai is currently pursuing her bachelors in Electronics and Telecommunication from Fr. C. Rodrigues Institute of Technology, Vashi, Maharashtra, India

Akansha Patnaik is currently pursuing her bachelors in Electronics and Telecommunication from Fr. C. Rodrigues Institute of Technology, Vashi, Maharashtra, India

ABSTRACT

A new idea on Animal Tracking technology is proposed wherein the location of the Animal can be traced and Accidents can be detected and notified to the Guardian. The technology uses mostly off-the-shelf available devices. This technology can be easily deployed on the animals using a collar/harness. Care has been taken to make the whole Tracker smaller and compact with maximum battery life so that nuisance to the animal because of the tracker can be minimized. Use cases are also present wherein this technology can be installed on a group of animals (livestock) and each tracker can adapt their roles as the main tracking device (which reports to the guardian) or secondary tracking device (reports to the main tracking device) dynamically.

A database is created where the information like the animal's location in latitude and longitude along with the acceleration data is sent by the tracker. The Guardian can view the location of the animal easily by accessing a tracking website hosted on a NodeJS server. In case the acceleration of dog exceeds the pre-defined threshold then the guardian immediately gets an accident alert message.

Keyword: Node, Root Node, Tracker, Accident Detection, NMEA coordinates.

1. INTRODUCTION

Tracking the animal can be for various reasons, right from a pet owner trying to find their lost pet to a farmer who wants to track their livestock movement across the field. In India livestock are sometimes left in the open field. This can cause accidents due to vehicles because of a nearby road or trains because of the near railway tracks. Tracking thus becomes important especially for the farmers to alert if the animals are moving off the set range of radius. Tracking devices also include accident detection which can also help to alert the farmer if the animal is hit and further prevention of the accident to other animals can thus be avoided.

The tracker uses mostly cheap off-the-shelf components which can be found in any country easily. A GSM/GPRS module (A9G) is required for sending the data to the Guardian or a predefined server. GPS module (A9G) is used for getting the location of the tracker. For managing the battery a battery load management chip is used (TP4056). Wi-Fi module (ESP8266) is used for communication between the trackers. The figure shows the basic block diagram of all the important components in the tracker.

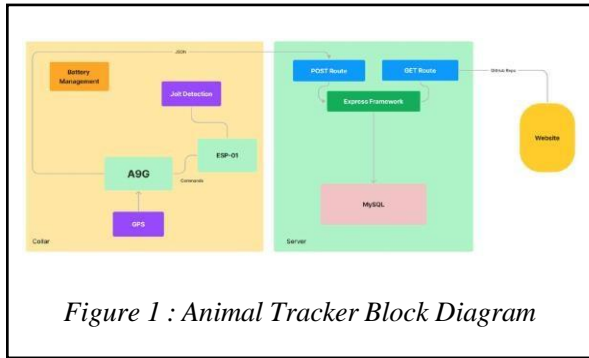


Figure 1 : Animal Tracker Block Diagram

The project aims to use smart dog tracking collar to track dogs/livestock. ESP8266-01 is used for communication with the A9G module and getting the current location. The ESP8266-01 and A9G communicates with each other using UART protocol. ESP8266-01E will send the acquired latitudes and longitudes from A9G and POST and information to a database server via a NodeJS server. The software component is responsible for handling the data gathered by the hardware, storing it in a MySQL database, and making it available through a web interface built using HTML, CSS, and JavaScript. The server for the web interface is implemented using the Express framework, which provides a simple and efficient way to manage HTTP requests and database access.

2. HARDWARE IMPLEMENTATION

ESP8266-01 communicates with all the other peripherals and acts as the main microcontroller. MPU-6050 Acceleration Gyroscope sensor sends data to ESP8266-01 which is analysed and proper decision is made if an alert and location information is to be sent or not. A9G module is used for connecting to the GPRS network and getting the GPS location information. The tracker can either use a GPRS network or a WiFi network for sending the information to the database. ESP8266-01 reads the GPS information and updates the animal's position on the database. ESP8266-01 and A9G communicates using the UART protocol. There is an LED on the A9G module that indicates the status of the 'Position Fix'. It will blink at different rates depending on which state it is in:

- No blinking – it is searching for satellites.

- Blink every 1s – Position Fix is found (the module can see enough satellites)

The raw data from the GPS module is sent by the ESP8266-01 and parsed by the server and sent to the database.. The co-ordinates returned by A9G module is in NMEA format. This format includes various types like GNRMC, GNGGA, GPGGA, GPGSV etc. Out of all these formats, only GPGGA is selected and sent to the server where the format is converted into actual coordinates.

The collar is also used for accident detection using the MPU-6050 sensor. MPU-6050 is a 6-axis Motion tracking device. It has Gyroscope and Accelerometer with Digital Motion Processor. For detecting the accidents, the MPU-6050 acceleration data is monitored. Figure 2 shows an example of acceleration on only x-axis.

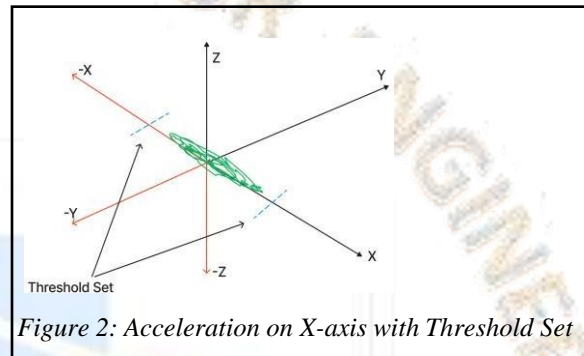


Figure 2: Acceleration on X-axis with Threshold Set

By setting a threshold level, an alert can be sent when the acceleration level crosses the Threshold. The threshold can be found by trial and error method. For monitoring the acceleration in all the 3 axis, RMS method can be used. It is therefore given by the equation $\sqrt{(X^2 + Y^2 + Z^2)}$ which gives approx magnitude of acceleration in all the axis. [1]

Figure 3 shows the accident detection algorithm flowchart ran by all the nodes (including the Root node) in a cluster.

Detection is done continuously. In order to prevent false alarm successive approximation can be done. This approximation can be done on next n datasets obtained from the MPU-6050 and confirm whether threshold is crossed atleast n/2 times. When an accident is detected, alert is sent to the Root Node (discussed in the Use Cases) or the NodeJS server via /alert route using HTTP commands.

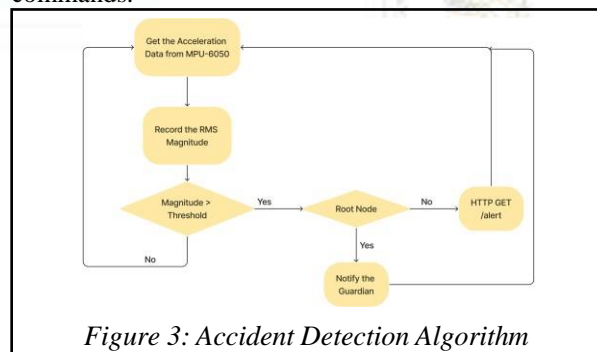


Figure 3: Accident Detection Algorithm

3. SOFTWARE IMPLEMENTATION

Libraries are included, such as the ESP8266WiFi library to connect the ESP8266 to a WiFi network, the ESPAsyncTCP library for asynchronous TCP connections, the ESPAsyncWebServer library to create a web server, and the WebSerial library for debugging purposes.

The data received from the ESP-01 one is in .JSON object format. Below is the HTTP requests that updates the MySQL database and retrieves the same to display it on the webpage. The first route, `app.get("/s", ...)` listens for GET requests to the /s endpoint. When a request is made, it extracts the collarId query parameter from the request object. It then constructs a SQL query to select all data from the collars table that matches the specified collarId. If an error occurs while querying the database, it sends back an error message. Otherwise, it sends the query result as a response back to the client.

The second route, `app.post("/s", ...)` listens for POST requests to the /s endpoint. When a request is made, it extracts the nmeaCoords and collarId fields from the request body. It then uses the `convertNMEaToDecimal` function to convert the nmeaCoords to latitude and longitude coordinates in decimal format. Afterward, it formats the decimal coordinates to six decimal places using `Number.parseFloat` and `toFixed`. Finally, it constructs a SQL query to update the lat and lng fields of the collars table where the collarId matches the specified value. If an error occurs while querying the database, it sends back an error message. Otherwise, it sends a "Success" message back to the client. Overall, this is a basic implementation of a RESTful API that is used to retrieve and update data in a MySQL database. It uses the Express.js library to define the routes and handle incoming requests, and the MySQL library to interact with the database. The code JavaScript function named `convertNMEaToDecimal` is a code that takes a string parameter representing an NMEA coordinate and converts it to decimal degrees. The NMEA coordinate string is expected to be in the format of the GPRMC sentence, which is a commonly used sentence in NMEA data. The sentence includes information such as time, latitude, longitude, speed, and course.[5]

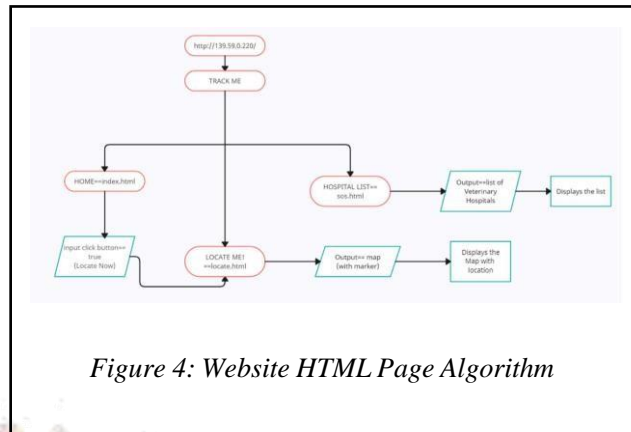


Figure 4: Website HTML Page Algorithm

Figure 4 shows how pages in the website are linked and what outcome they produce.

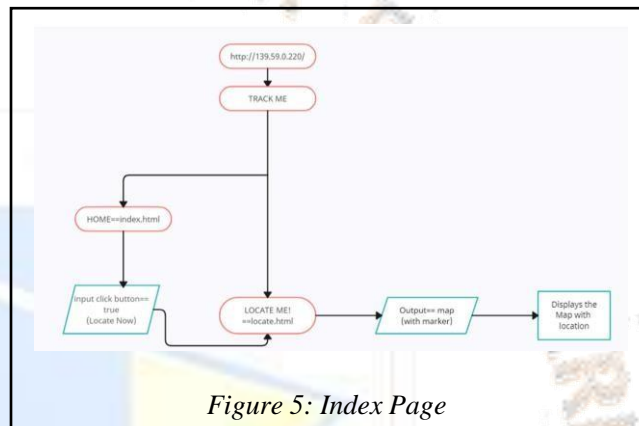


Figure 5: Index Page

Figure 5 is the algorithm for the website pages that creates a basic web page layout. The head section contains the title of the web page and a reference to an external CSS file, which will be used to style the web page. The body section contains a banner, which is a container that contains a navigation bar and a content section. The navigation bar contains an image logo and a list of links that direct the user to different pages within the web application. The content section contains a title, a paragraph, and a button. The title is the main heading of the page, and the paragraph is a short description of the page's purpose. The button is a call-to-action that links to the "Locate Me!" page of the web application. The 'div' element is used to create containers for different sections of the web page. The 'ul' and 'li' elements are used to create lists, and the 'img' element is used to embed images. The 'class' attribute is used to apply styles to elements using CSS. Overall, the code creates a basic algorithm for web page layout with a navigation bar and a call-to-action button that links to another page within the web application.

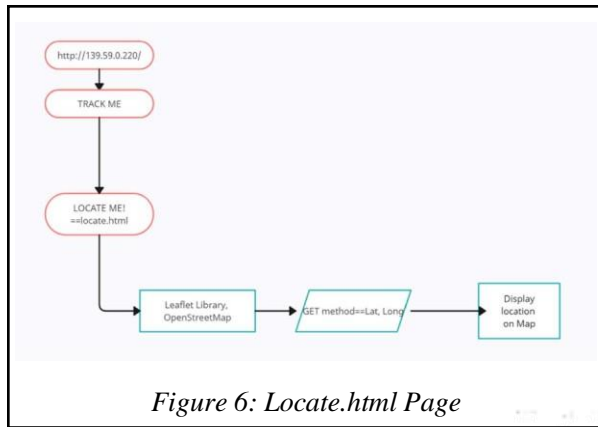


Figure 6: Locate.html Page

Locate.html is a HTML webpage that displays a map using the Leaflet library. The webpage includes a header with a title, a CSS stylesheet, and a reference to the Leaflet library. The body of the webpage contains a banner and navbar at the top, with a logo and links to different pages. Below the navbar, there is a div element with an id of "map" which will display the Leaflet map. In the script section of the code, the Leaflet library is used to create a map centered on the coordinates received from the MySQL database with a zoom level of 11. The map is then set to display tiles from the OpenStreetMap server. The fetch function is used to retrieve data from a server at the URL "/" using the GET method. The returned data is in JSON format and is then used to set the latitude and longitude of a marker on the map. The marker is created with the L.marker function and the riseOnHover property is set to true. The marker is then added to the map using the addTo method. Overall, Locate.html sets up a basic webpage that displays a Leaflet map and adds a marker to the map using data retrieved from a server.

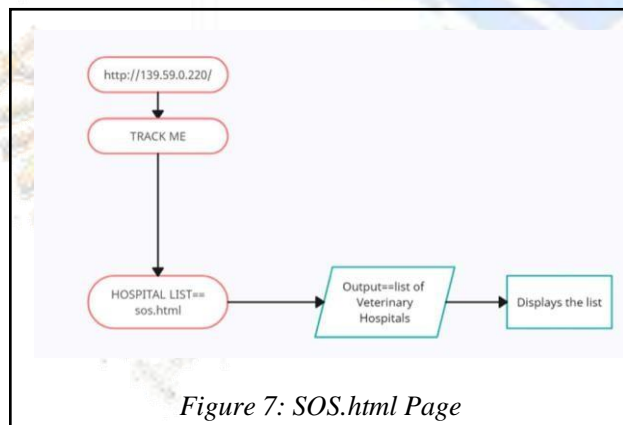


Figure 7: SOS.html Page

The HTML code for the above algorithm includes a banner section with a logo and a navigation bar. The navigation bar has links to the various pages of the website. The main content of the sos.html webpage is a table that lists various veterinary hospitals. The table has columns for the hospital ID, hospital name, contact number, location, and customer rating out of 5. Overall the HTML code creates a simple and functional webpage.

4. RESULTS

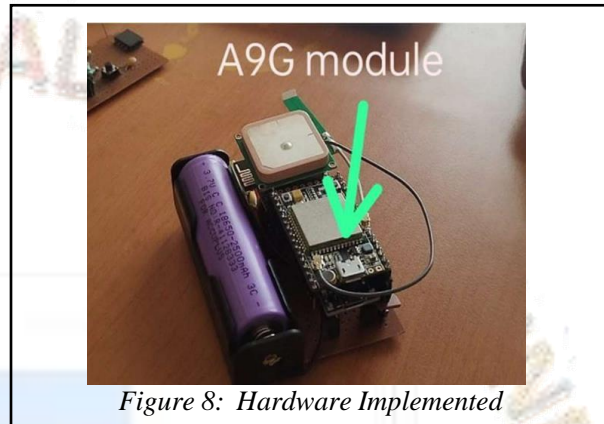


Figure 8: Hardware Implemented

The above image shows the final hardware made.

ESP8266-01 incorporates ESP8266 chip within used to connect to WiFi. The SSID or the WiFi name to connect to and its password is included in the flash file of ESP8266-01. After the flashing is complete, the module tries connecting to the mentioned WiFi SSID. More than 1 WiFi can be configured and the chip will connect to the one with the strongest signal. The console window of Arduino IDE is used here to read the data serially from the ESP8266-01. The diagram below shows the ESP8266-01 chip connecting to the WiFi named 'HOME'. The router then allots an IP address to the ESP8266-01 microcontroller.

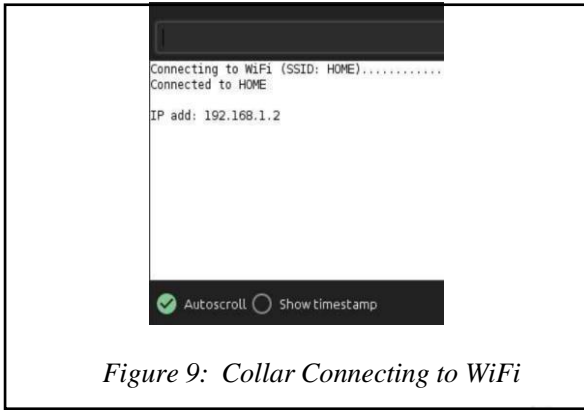


Figure 9: Collar Connecting to WiFi

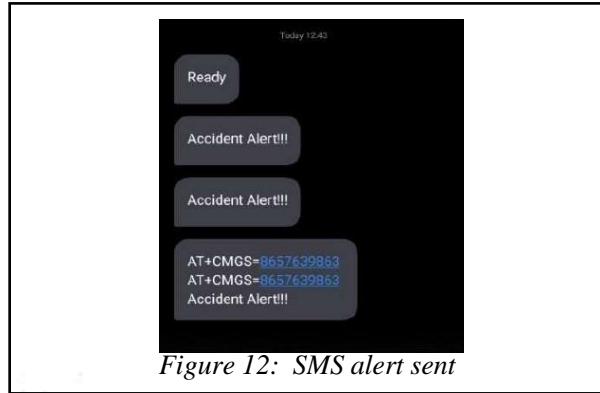


Figure 12: SMS alert sent

Web Serial is used for getting the logs generated by ESP8266

The above image shows how the alert SMS is sent when the A9G is violently shaken.

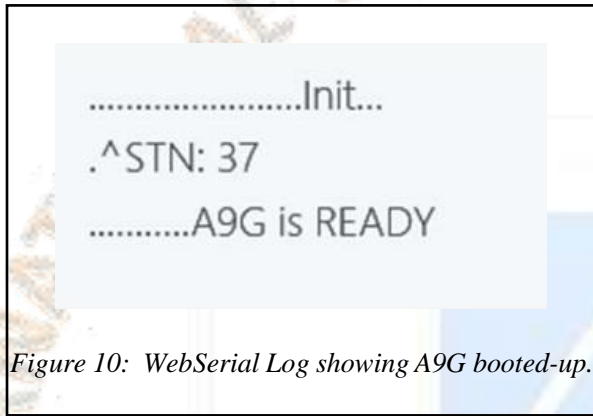


Figure 10: WebSerial Log showing A9G booted-up.

The above log was taken from the Web Serial window. It shows A9G booting up and connecting to a network.

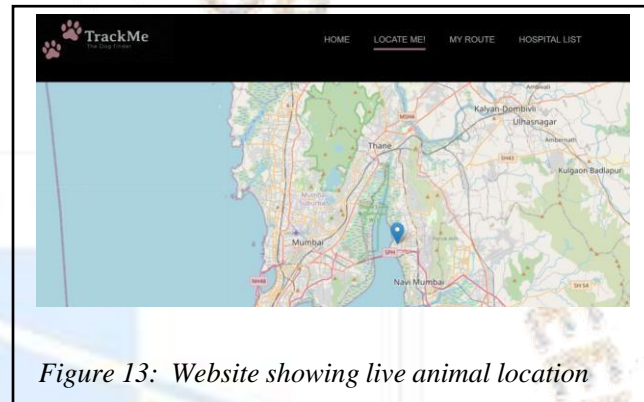


Figure 13: Website showing live animal location

The website above shows the tracking of the animal.



Figure 11: WebSerial Log showing GPS data

The image above shows how the co-ordinates are obtained in the A9G module. These co-ordinates are then posted to the server where it is converted into latitude and longitude

5. FUTURE SCOPE

a. Adaptable Peer Network

When a group of tracker are deployed on the animals in the same area, the trackers can adapt with each other and start following a role. Group of nodes together form a cluster. There are two roles which can be undertaken by the tracker; Root Node and Node. Root Node is the main node in the cluster to which other nodes can communicate. Root node turns on the A9G Module for GSM/GPRS and GPS data. Other nodes only have ESP8266 module turned on which acts as a main controller in the project and collects the data from the MPU-6050 sensor. [6]

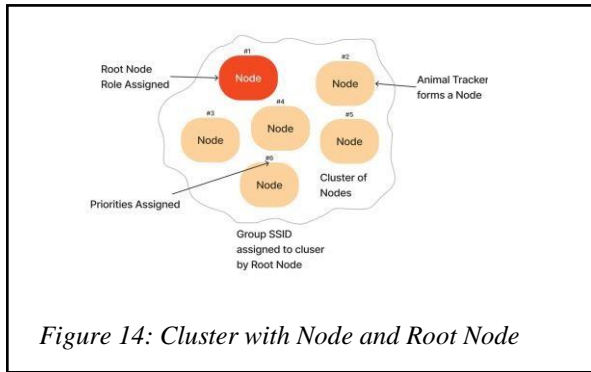


Figure 14: Cluster with Node and Root Node

Each cluster has a unique Group SSID. On Power-Up the node scans for this Group SSID and connects to it if available. If the SSID is not available the Node takes up the role of the Root Node and updates its location to the Guardian or the Main Server.

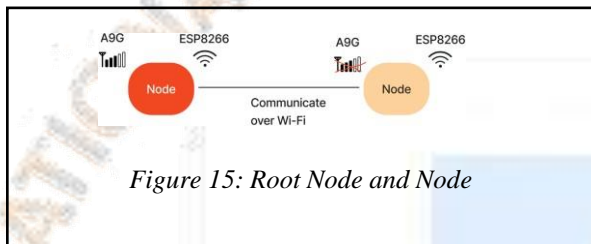


Figure 15: Root Node and Node

Figure 15 shows how Root Node and Node communicates with each other. Root Node turns on it's A9G and ESP8266 module. Other Nodes in the cluster communicate with the Root Node over the Wi-Fi. The communication is uni-directional with data only being sent to the Root Node. HTTP protocol is used for sending the data to the Root Node. HTTP routes are defined in the Root Node which can be used to give GET command. GET is done with the parameters like nodepriority which is the Priority ID assigned to each Node in the cluster and battery voltage level. The Root Node then keeps a track of all the other nodes in the cluster. When guardian requests the location of a Node (Identified by it's priority ID), the Root Node replies with the cluster's location as identified by the Root Node and the last time the other node updated it's presence with the Root Node.

Figure 16 shows the Algorithm of Animal Tracker performing different tasks as a Root Node and a Normal Node.

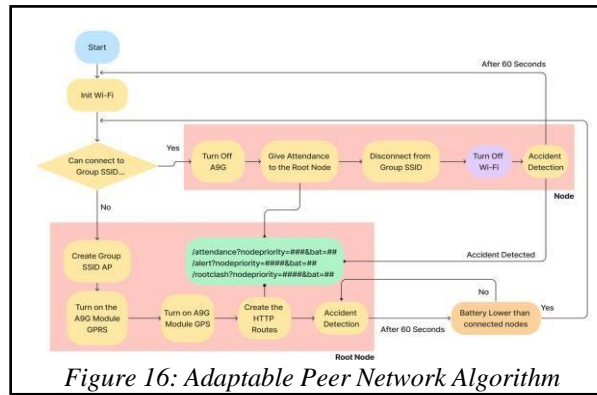


Figure 16: Adaptable Peer Network Algorithm

As shown in figure 16, the algorithm makes sure that the roles taken up by the Nodes are dynamic in nature. If more than one Group SSIDs are found in the same cluster, the normal Nodes sends a rootclash notification to the root node it can connect to. The Root node then accordingly delegates it's root role to the highest priority Root Node in the cluster. The algorithm also makes sure that if a root node's battery is below the threshold, it will delegate it's Root Role to another node with better battery gauge and become a normal node by itself.

b. HTTP Routes in Adaptable Peer Network

HTTP Routes are defined in the Root Node so that other nodes can send their data to the Root node using HTTP GET with params defining their priorityid and battery voltage.

Figure 17 shows the HTTP routes defined in the Root Node.

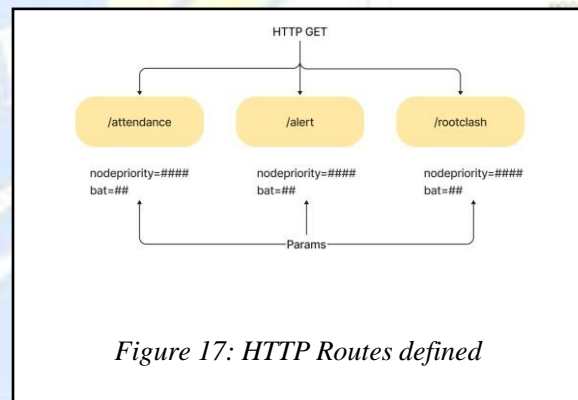


Figure 17: HTTP Routes defined

When a node powers-up for the first time it checks whether the Group SSID is found in it's cluster. Once it has connected to the Group SSID, the node sends it's presence to the Root Node via /attendance route with it's parameters nodepriority and battery voltage. The Root Node keeps a track of this information in a form of a data structure so that later on the information

can be retrieved with minimal complexity. The node then monitors the MPU-6050 data and performs Accident detection. If an accident is detected an alert is sent to the Root Node via /alert route. The Root Node then records the quering node's ID and sends an alert to the Guardian. Another route which is defines in the Root Node is the /rootclash route. If the nodes in the cluster capture more than one Group SSID in a cluster, a notification is sent to one of the root nodes via the /rootclash route. The root node then checks with the other root node and confirms it's priority and accordingly the Root Node with the highest priority is delegated the Root Role while the other root nodes take up the role of normal node. This ensures that only single Root Node is present in the cluster and it can save the battery power of the other nodes.

c. Deployment cases of Adaptable Peer Network

There can be several ways this technology can be deployed. It depends on the need of the Guardian. If the animal is a single Pet then a Internet connection would be required so that the Guardian comes to know the location of the animal. If a livestock is to be tracked then Adaptable Peer Network can be used for tracking all the animals in a cluster. Figure 18 shows a typical deployment case.

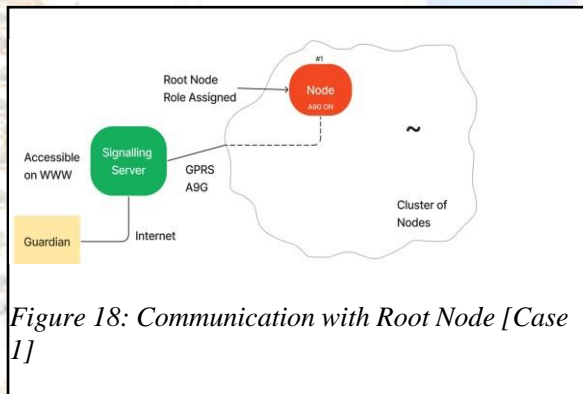


Figure 18: Communication with Root Node [Case 1]

A signalling server is present in this case. The role of this server is to signal the Guardian's request to the appropriate Root Node in the Internet. The Root Node communicates with the signalling server using GPRS on A9G module. This case also requires the Guardian to have an internet connection for sending the request to the Signalling server which can be accessed from WWW. For the Node to send an alert back to the Guardian, the signalling server forwards the alert message from the Root Node to the Guardian's device. Instead of a signalling server a dedicated server can also be used which can maintain the database of the nodes in

the cluster. The database can include location of the animal and last attendance time of the node. Database can then also be used to track the movement of the animal.

Another way of deploying this technology is given in Figure 19.

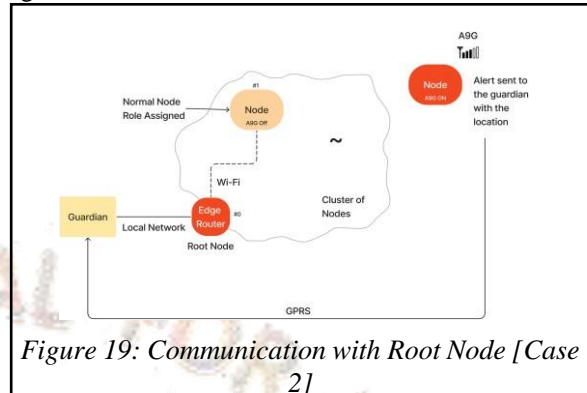


Figure 19: Communication with Root Node [Case 2]

In this case an Edge Router is used in the cluster. The edge is assigned priority 0 (Highest Priority) in the cluster. All the nodes in the cluster communicate to the Edge Router using the Wi-Fi. All the other nodes have their A9G module turned off which increases the battery efficiency. Edge Router can be powered with an external supply. Here the Guardian can communicate with the Edge Router locally without an Internet Connection. Provision can also be made wherein if a node goes out of the cluster an alert can be sent to the Guardian.

d. Issues to be tackled while deployment

One of the issued that needs to be tackled is the assignment of the Root Node. This becomes critical because if an edge node is made a root node then the cluster area changes and false alert might be sent even though the animal is in the desirable cluster according thetheGuardian. ESP8266 mostly has a omnidirectional radiation pattern therefore choosing a node which is in the middle of the desirable radius of the cluster is required. This approximation can be done by approximating the location of each animal in the cluster and choosing the node which is closest to this approximation. This will of-course increase the computation complexity but will help in better assignment of the Root Node.

Another issue to be tackled is the Power Requirement of the whole project. ESP8266 chip along with A9G module can consume very high power. This can become a problem especially in the scenario where only a single Pet is being tracked. By increasing the idle time threshold before turning on the A9G and ESP8266 module the battery life can be increased though it also increases the time taken to identify the animal. Also the location which might be saved on the server can be old and querying the server will result in sending the old location back to the Guardian. In this situation, the Adaptable Peer Network can be dropped and only A9G tracking can be done along with a simple 8-bit microcontroller removing ESP8266. This gives an advantage of increasing the battery life by just making use of a simple microcontroller for accident detection and then report the data to the Guardian using the A9G GSM/GPRS module.

ACKNOWLEDGMENT

We would like to express our sincere gratitude to all those who have contributed to the successful completion of this research paper on the topic of smart dog tracking collars.

Firstly, we would like to extend our thanks to our supervisor Mrs. Sneha Revankar, whose invaluable guidance, feedback, and suggestions have been instrumental in shaping this paper. We are grateful for her expertise and support throughout the research process.

We are indebted to the manufacturers and developers of the smart dog tracking collars for creating innovative products that have the potential to improve the lives of dogs and their owners. Their willingness to share their insights and knowledge has been invaluable in informing this research.

6. REFERENCE

- [1] Anumol Mariyam Shaji, Gayathri S, Sharon Koshy Varghese, Sreejith Chandran, Anju Rachel Oommen - "Implementation on an arduino based accident identification", May 2019 JETIR Volume 6, Issue 5
- [2] Prenern Reddy, "Trackable Pet and Livestock Collar," 2020 Electrical and electronic engineering science at the university of johannesburg,(201105802)
- [3] Zhengming Tang¹ , Harlan Hile² , Sushil Bajracharya¹ , Raja Jurdak¹, "PetTracker – Pet Tracking System Using Motes" 2014 Information and Computer Science University of California, Irvine, Computer Science and Engineering University of Washington.
- [4]Dr. Sasi Kumar, Soumyalatha, Shruti G Hegde(2017)" IoT Approach to Save Life using GPS for the Traveller during Accident" 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI).
- [5]Neha Mangla, Sivananda.G, Aishwarya Kashyap, Vinutha (2017) "A GPS-GSM Predicated Vehicle Tracking System, Monitored in A Mobile App based on Google Maps". 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS).
- [6] Aravind, D & Anupriya, S & Aarthi, A & Chandrasekaran, Saravanakumar. (2017). An Automatic Wildlife Tracking System using GPS and Wireless Sensor Networks.

