

DOI: <http://dx.doi.org/10.21123/bsj.2021.18.4.1303>

## Improvement of the Fault Tolerance in IoT Based Positioning Systems by Applying for Redundancy in the Controller Layer

*Nawzad K. Al-Salihi*

Department of Computer Science and Engineering, College of Engineering, University of Kurdistan Hewlêr (UKH), Erbil 44001, Iraq.

E-mail: [n.al-salihi@ukh.edu.krd](mailto:n.al-salihi@ukh.edu.krd)

\*ORCID ID: <https://orcid.org/0000-0002-4180-799X>

Received 3/3/2020, Accepted 2/9/2020, Published Online First 30/4/2021, Published 1/12/2021



This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).

### Abstract:

In recent years, the positioning applications of Internet-of-Things (IoT) based systems have grown increasingly popular, and are found to be useful in tracking the daily activities of children, the elderly and vehicle tracking. It can be argued that the data obtained from GPS based systems may contain error, hence taking these factors into account, the proposed method for this study is based on the application of IoT-based positioning and the replacement of using IoT instead of GPS. This cannot, however, be a reason for not using the GPS, and in order to enhance the reliability, a parallel combination of the modern system and traditional methods simultaneously can be applied. Although GPS signals can only be accessed in open spaces, GPS devices are error-prone primarily when the receiver is located in an urban-canyons area, due to congestion and the possible interference. The outcome presents a redundancy-based model for improving the fault tolerance of IoT-based positioning systems. The simulation results show a 22.5% improvement in the fault tolerance of the IoT-based positioning system after applying the proposed validation mechanism, and a 77.4% improvement in this tolerance after applying for a more expensive module redundancy.

**Keywords:** Fault Tolerance, Global Positioning System (GPS), Internet of Things (IoT), Redundancy, Reliability.

### Introduction:

The significant wave of research into the Global Positioning System (GPS) navigation began in the late 1990s. Since then, GPS devices have undergone considerable development. Most GPS devices, however, still operate in static navigation mode, where the optimal path refers only to the shortest distance to a geographic position. With the high rate of population growth, industrialization, and urbanization, along with the resulting increase in the number of vehicles, the existing urban transportation systems are undergoing pressure to utilize technologies such as GPS for improved efficiency (1). An example of the Internet of Things (IoT)-enabled environment is an integrated transport system that can be dynamically routed and reorganized in response to changing traffic needs and conditions (2). IoT-based applications in today's life have developed considerably. These applications are positioning software, tracking children and elders, monitoring people with

criminal records, or tracking vehicles, which has grown increasingly popular among users. As a result, GPS technology has found extensive use in traffic monitoring and vehicle navigation systems. In the future, this technology will synergize with emerging technologies such as the IoT, which are expected to dramatically increase the volume of positional data flows in the coming years. Approximately 28 billion devices, including more than 15 billion machine-to-machine (M2M) and consumer electronic devices, are expected to be communicating over short-range radio technologies such as Wi-Fi and Bluetooth, as well as over wide area networks (WANs) based on cellular technology by 2021 (3). With the advent of various IoT protocols like Bluetooth, Wi-Fi, etc., connectivity among various devices has increased, making systems more autonomous and integrated (4). The analysis of such data flows can have many benefits

for smart decision making on a variety of applications (5).

On the other hand, various factors can cause faults in such systems and exert a negative impact on the data obtained from GPS systems. In addition, the use of the GPS system can only be exercised in the areas covered with satellite vision. These two factors lead us to look for an IoT-based and fault-tolerant tracking system, which was the purpose of this study. To increase reliability, the parallel combination of the modern system and traditional tracking techniques in parallel can be used.

The existing structures for the integration of navigation systems with an IoT platform can be divided into three categories:

#### A. Sensor networks

These networks are responsible for the real-time collection of traffic data and sending them to communication platforms.

#### B. IoT communication platform

This platform provides a set of ports for receiving positional data, which will then be forwarded through a controller toward the application layer. Examples of wireless IoT communication platforms are illustrated in Fig. 1.

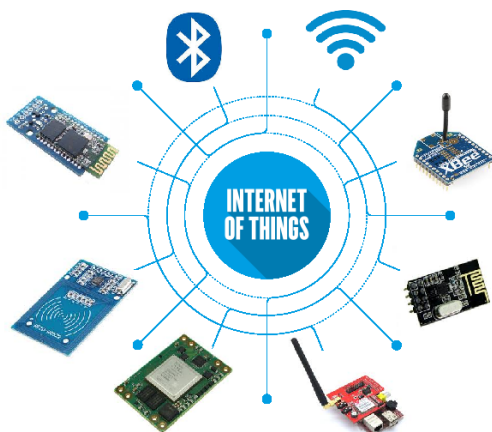


Figure 1. Wireless technologies for IoT

#### C. Navigation monitoring system

This system downloads positional data via an application programming interface (API) through Arduino IDE software and marks the position on a map based on the received coordinates. The proposed system is based on the three-layer architecture of the Internet of Things. In this architecture, the three constituent layers of the system include:

- Sensor layer
- IoT Controller Layer
- Communication and application layer

In the subsequent studies, this architecture is also introduced:

The rigid communication architecture of the Internet is one of the challenging issues in IoT. Network virtualization enables the Internet to retain its communication architecture while enlarging and transforming as IoT (6). In the (7-10), this architecture is also introduced. In this article, in the base of sensors the position relative to the Internet server stations in the layout of the calculator controller is done to determine the position. Furthermore, in the application layer, the user accesses this information through various communication platforms, such as the Web.2.

#### Related work:

Among the studies related to IoT tracking (11) and fault tolerance discussion, can be referred to (11) and (12-14), respectively.

Timely processing is increasingly required for IoT smart devices, which will result in the direct implementation of information processing on IoT devices to release the bandwidth and guarantee privacy. Specifically, like the position of sensitive people, including smart homes and medical monitoring programs, continuously tracking the signals has become a common trend for a variety of IoT devices for proactive processing. In these systems, however, the possibility of fault, due to limited resources, is inevitable. Some solutions have been proposed in (15) for the timely processing of efficient systems to trace signals with limited memory space. Still, this study only focused on the use of certain branches of smart applications and was not able to classify the data related to many applications. In this study, like our study, the Arduino controller has been used to check such limitations as low memory volume. In (16), researchers proposed a mobile phone-based positioning technique that uses GPS and cellular network infrastructure for position tracking. This technique contributes to the performance of the Patient Tracking Location System (PTLS) in assisting family members in tracking a patient under emergency conditions.

In (17), the relationship between the mobility of patients suffering from Parkinson's disease and the illness level was investigated using a GPS-based system. The assessment of the living environment was performed based on the criteria derived from the GPS data. In this research, participants were asked to carry their smartphones throughout the day and allow their positional data to be recorded. This study found that the patient's Unified Parkinson's Disease Rating Scale (UPDRS) had an inverse relationship with the patient's mobility.

Statistics have shown a rising rate of mortality due to heart disease and high blood pressure in recent years. It is widely known that blood pressure plays a vital role in heart disease, hence, taking preventive measures against high blood pressure is of immense importance. In (18), researchers presented a new architecture for providing smart health services based on the Global System for Mobile communication (GSM) and GPS. The goal of this work was to provide real-time healthcare tracking services to patients. The system was designed to analyze the patient's vital signs and the collected health data and send the physician a message containing the patient's name, body temperature, heart rate, and exact location whenever analysis results are indicative of a risky health condition.

At present, positioning services rely on satellites to operate, in the sense that they calculate the position on the Earth's surface based on the signal response time and distance from the satellite. This type of positioning system has several drawbacks, including the chance of signal weakness, vulnerability to errors caused by natural factors, and exposure to hackers. The future of GPS technology seems to be very different from its condition today. The Internet of Things (IoT) is a relatively new branch of communications technology that aims to integrate all devices around us which can be linked to the Internet. These include smartphones, voice assistants, smart TVs, and any other devices that is used on a daily basis. These devices can also feature GPS-assisted positional tracking capability, which has significant application in preventing the loss or theft of devices and locating individuals and objects. The integration of GPS and IoT can benefit a wide range of applications (Fig. 2), such as the tracing of manufactured products from stock to retail shelves. Moreover, the integration of GPS with IoT allows us to replace conventional position tracking devices with sensor-to-sensor communications to reduce energy consumption and make the traditional GPS services faster and cheaper (19).

Current GPS-based and Wi-Fi-based positioning systems suffer from problems such as limited bandwidth, high energy demand, and expensiveness.



Figure 2. Integration of IoT with GPS

### The Proposed method, models, and definitions:

Depending on the country or the city where a GPS reading is accomplished, the latitude and longitude readings have a certain amount of validity range. A reading that is outside the expected range is a clear sign of error in the system. For example, for the Kurdistan Region, and Iraq, which is the country used in the simulation of this work, the readings should remain between N 29° and N 36° 30° degrees north latitude and E 38° and E 48° degrees east longitude as shown in Fig. 3.



Figure 3. Permissible latitude and longitude values in the area of simulation

### Validity assessment of the positional data read from the GPS module

Before forwarding positional data, their values will be compared with the validity range and any mismatch will be checked. Error detection scenarios considered in this work are explained below.

```

Lt, Lg, H=read from GPS (latitude, longitude, height) at time=t
If (latitude<29 or latitude>36.3) or (longitude<38 or longitude>48) then
Fault_Detect=True
Else
Fault_Detect=False;
    
```

At the program, variables H, Lg and Lt Respectively correspond to latitude, longitude and elevation values used to determine the position.

#### Assessment of the rate of change

Depending on the position of individuals and whether they are standing, walking, running, or using a vehicle, the degree of change between each two reading instances cannot exceed a maximum value. If the amount of observed change in data is improbable, the probability of an error in the system will be checked. The following pseudo-code is used for error detection in the receiver module:

```
Lt1, Lg1, H1=read from GPS(latitude, longitude, height) at time=t1
Lt2, Lg2, H2=read from GPS(latitude, longitude, height) at time=t2
ΔLatitude_Position=Lt2- Lt1;
Δ Longitude _Position= Lg2- Lg1;
Δ Height _Position= H2- H1;
If {(Δ Latitude _Position>Threshold_ latitude)
or (Δ Longitude _Position>Threshold_ latitude)
or (Δ Height _Position>Threshold_ latitude)}
Then
Fault_Detect=True
Else
Fault_Detect=False;
```

#### Reliability

Reliability is defined as the probability of the system operating correctly for a given period of time, provided that it operates properly from the start. The reliability of a GPS-based positioning system depends on the following components:

#### Transmitter module reliability

This refers to the reliability of a system module that can be identified as an individual subsystem. This module may be a part of a ground station, air station, or other major systems.

#### Receiver reliability

This refers to the reliability of the devices, such as aircraft navigation equipment. Such equipment may be configured to enjoy single, double or triple redundancy.

#### Station reliability

This refers to the reliability of transmission units or stations. In many cases, a single station is not enough to ensure reliable navigation or positioning.

#### Use of hardware redundancy for error detection and correction

In a system with simple redundancy, a backup module added to the system works in parallel with the main module. Such redundancy is commonly referred to as Dual Modular Redundancy

(DMR). The general schematic diagram of a DMR system is shown in Fig. 4.

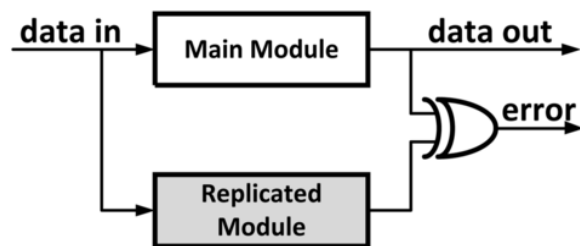


Figure 4. Schematic diagram of a DMR system

In many critical security, commercial, and monitoring applications, it is essential to have a degree of fault tolerance. Tolerance describes the ability to contain the effects of faults so that a system can avoid failures and continue working properly (20). Critical security applications refer to those systems that are responsible for people's lives, such as air control systems, computer-assisted radiation control mechanisms, cardiac guidance systems, and military systems. Critical commercial applications are those that facilitate business activities, such as automated transactions, trading, inventory, and banking systems (21).

The existing applications for fault-tolerant systems can be divided into four major categories: long-life applications, critical computations, complicated repair and maintenance operations and, finally, applications that require high availability. Each of these applications has its own requirements and therefore requires its own techniques.

#### Fault model

Faults can be classified based on their duration, nature, and size. But considering the objectives of this study, the faults are classified according to:

- Fault duration
- Fault layer in the IoT model

From the duration perspective, faults can be divided into three categories: permanent, intermittent, and transient.

- Transient or soft faults are the faults caused by temporary malfunctions that last for a limited period of time and will go away after a while, even if not addressed.
- Permanent or hard faults are the faults that will not go away by themselves. These faults result from physical damage or permanent malfunction in components or design.
- Intermittent faults are the faults that cause the system to alternate between proper and defective functioning. Intermittent faults are usually caused by marginal factors.

Intermittent and transient faults are generally more frequent than permanent faults and are also more difficult to trace as they may simply disappear after a while.

The following Figs. (5-7) show the diagrams of three types of fault appearing at  $t = 1s$ :

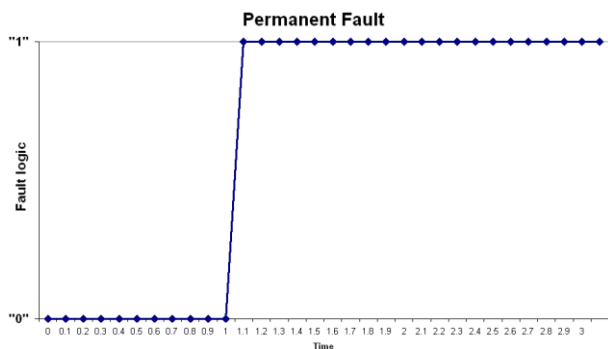


Figure 5. An example of a permanent fault

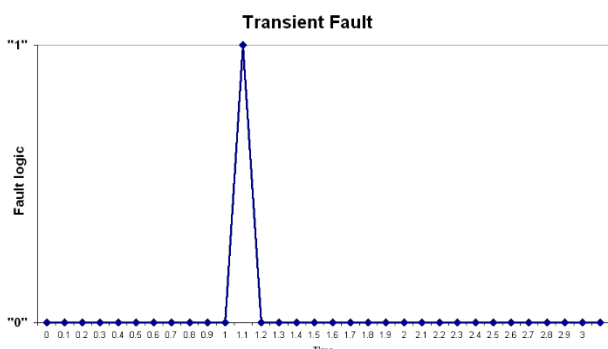


Figure 6. An example of a transient fault

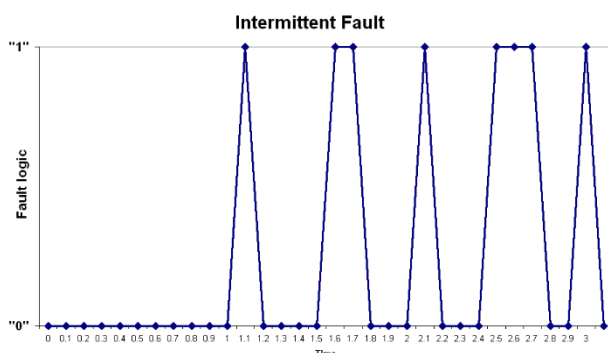


Figure 7. An example of an intermittent fault

The size of a fault is determined by the breadth of affected areas. While local faults affect individual components, global faults are likely to involve multiple components. Because of cost constraints, many fault tolerance and test strategies can only respond to single faults. Naturally, numerous fault solutions require more expensive fault models and global fault tolerance strategies. In this study, the fault model is based on the single fault assumption Fig. 8.

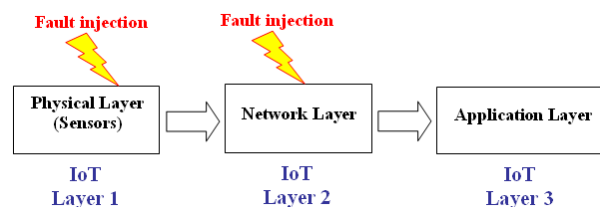


Figure 8. Fault injection areas

Fault tolerance approaches are necessary as they aid in detecting and handling faults in the system that may occur either due to hardware (H/W) failure or software (S/W) faults (22-24).

One way to increase the fault-tolerance of the IoT architecture is to add redundancy to all layers:

- Hardware redundancy of data generation modules and sensors in the physical layer
- Hardware redundancy in network layer communications modules
- Software redundancy in the application layer to control information validity

Fault coverage is one of the primary solutions for stabilizing or improving the system behavior in fault-prone environments (25, 26).

A system is considered fault-tolerant if it is able to perform appropriately even in the event of a hardware malfunction or software error (27).

Digital systems that carry out more critical tasks require higher degrees of reliability. Often, the use of high-quality components and design techniques is not enough to guarantee an acceptably low failure probability. Thus, the systems should be made fault-tolerant (28).

### Analysis:

To evaluate and analyze the proposed mechanism, the followings are examined, the components of the proposed architecture and their reliability in the various positions of standard architecture and after the proposed changes made by this study.

The reliability of a module indicates the possibility that the system will work properly and without any repair until the time  $t$ . System reliability is defined based on the failure rate of the component; that is, the rate of failure based on the separate components (29):

$$R(t) = e^{-\lambda t} = \exp(-\lambda t)$$

Given that this article uses three-layer architecture for the IoT, the reliability of the whole system depends on the correct and simultaneous performance of three layers:

$$R_{IoT}(t) = R_{Sens}(t) \cdot R_{cont}(t) \cdot R_{Mon}(t)$$

In the above equation,  $R_{Sens}(t)$  represents the reliability of the sensor layer in the IoT architecture.  $R_{Cont}(t)$  denotes the reliability of the IoT controller and network layer, and finally, the last part,  $R_{Mon}(t)$ , represents the reliability of the monitoring layer. Since the first two layers include hardware and software, the third layer only involves the application software, based on the defined error model, the error injection are carried out in the first two layers and assume that:

$$R_{Mon}(t)=1$$

Hence, the reliability of the whole system will be simplified as follows:

$$R_{IoT}(t)= R_{Sens}(t) \cdot R_{Cont}(t)$$

Assuming that denotation of the fault rate in the sensor and controller with the parameters  $\lambda_S$  and  $\lambda_C$ , respectively are:

$$R_{Sens}(t)= \exp(-\lambda_S \cdot t)$$

$$R_{Cont}(t)= \exp(-\lambda_C \cdot t)$$

$$R_{IoT}(t)=\exp(-\lambda_S \cdot t) \cdot \exp(-\lambda_C \cdot t)$$

In this study, the software techniques for checking the permitted data range and their combination with hardware techniques can be employed, while the second time hardware redundancy can be used in the sensor and controller layers. This means that instead of using one sensor in the first layer, there are two main sensors and one plugin sensor (Redundant), and also utilize two main controllers and plugin controllers in the controller layer. The reliability of each of these two layers will be calculated as follows:

$$R_{proposed-Sens}(t)= R_{Sens1}(t)+(1- R_{Sens1}(t)) \cdot R_{Sens1}(t)$$

$$R_{proposed-Cont}(t)= R_{Cont1}(t)+(1- R_{Cont1}(t)) \cdot R_{Cont2}(t)$$

Therefore, by applying the fault-tolerant mechanisms and using plugin modules in the sensor and controller layers, the reliability of the proposed IoT-based system will be calculated as follows:

$$R_{proposed-IoT}(t)= R_{proposed-Sens}(t) \cdot R_{proposed-Cont}(t)$$

$$R_{proposed-IoT}(t)= ( R_{Sens1}(t)+(1- R_{Sens1}(t)) \cdot R_{Sens1}(t) ) \cdot ( R_{Cont1}(t)+(1- R_{Cont1}(t)) \cdot R_{Cont2}(t) )$$

To compare the reliability of the proposed system for different values of the failure rate in the sensor and controller layers, check the reliability after different time periods. To efficiently and readably compare, the following is assumed:

$$\lambda_S=\lambda_C=\lambda \rightarrow R_{IoT}(t)=\exp(-\lambda_S \cdot t) \cdot \exp(-\lambda_C \cdot t)= \exp(-\lambda \cdot t) \cdot \exp(-\lambda \cdot t)= \exp(-2\lambda \cdot t)$$

$$\lambda_{S1}=\lambda_{S2}=\lambda_{C1}=\lambda_{C2}=\lambda \rightarrow$$

$$R_{proposed-IoT}(t)=$$

$$\begin{aligned} & (\exp(-\lambda_S \cdot t)+(1-\exp(-\lambda_S \cdot t)) \cdot \exp(-\lambda_S \cdot t)) \cdot \\ & (\exp(-\lambda_C \cdot t)+(1-\exp(-\lambda_C \cdot t)) \cdot \exp(-\lambda_C \cdot t)) \\ & =(\exp(-\lambda t)+(1-\exp(-\lambda t)) \cdot \exp(-\lambda t)) \cdot (\exp(-\lambda t)+(1- \\ & \exp(-\lambda t)) \cdot \exp(-\lambda t)) \\ & =(\exp(-\lambda t)+(1-\exp(-\lambda t)) \cdot \exp(-\lambda t))^2 \end{aligned}$$

The results for component reliability are shown in the graph below, Fig. 9.

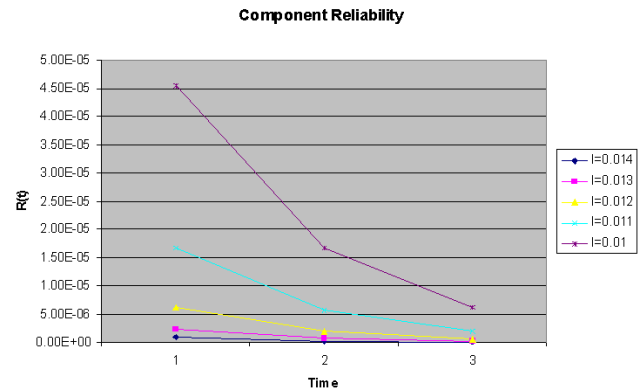


Figure 9. Comparison of evaluation for different fault rates

### Simulation details:

In this study, an IoT-based tracking system is implemented, the coordinate data of the target is sent through the Internet to a local server, where they can be accessed on a computer or mobile phone via a webpage. In accordance with the fault model section, considering the simultaneous coverage of permanent, transient, and intermittent faults as well as fault injection in different IoT layers, no similar work has previously been conducted to be used as a comparison to this study. It is for this reason, according to the evaluation of the proposed mechanism in the paper analysis section, that the simulation results before the fault injection as well as after, which are in line with the scenarios presented in the paper, were compared based on the fault model, from which the final results were obtained.



Figure 10. Components of the proposed system for simulation

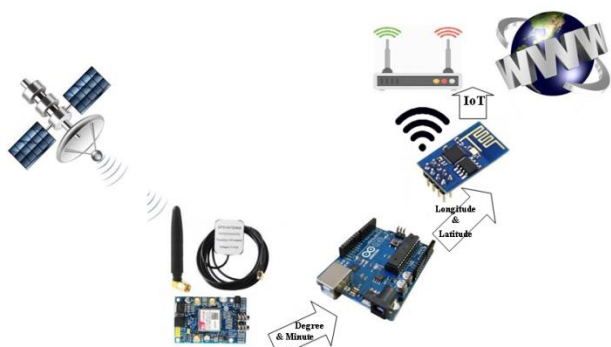
According to the above structure, the proposed mechanism consists of three parts, shown in Fig. 10:

- a) Positional data generator and data transmitter (via the Internet)
- b) Internet infrastructure for ubiquitous access to information
- c) IoT receiver, which includes a web page for viewing positional data on a mobile phone or computer

The positional data transmitter, which operates based on the IoT infrastructure, consists of the following modules:

- GPS module: to obtain coordinates
- Wi-Fi Module: to send positional data to a modem connected to the Internet
- Arduino controller: for configuring modules and linking Wi-Fi to GPS (via the Internet)

The relationship between these modules is illustrated in Fig. 11.



**Figure 11. The relationship between the modules of the proposed system**

In the proposed method, it receives geolocation through the GPS module that has serial communication capabilities, the data is sent through the serial port to the Internet controller objects and after applying error tolerance algorithm, error-free data is sent via the Internet.

The IoT controller is responsible for performing the following tasks:

- 1- Obtaining coordinates from GPS
- 2- Converting GPS coordinates from degree-minute format to decimal format
- 3- Issuing commands to the Wi-Fi module for configuration and connection to the web page via the Internet

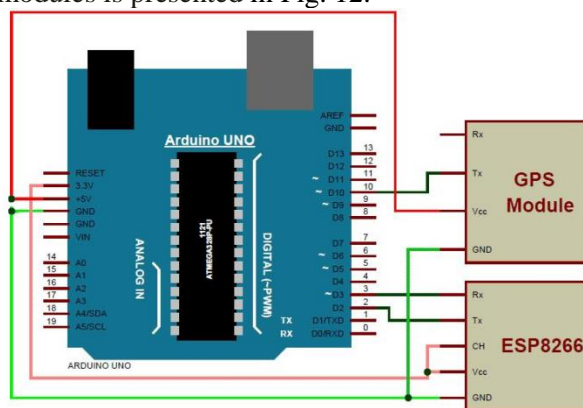
The first step taken in the controller is to convert the GPS coordinates from degree-minute format to

decimal format. The latitude coordinates received from the GPS module consist of two parts, degrees and minutes, which must be converted through division by 60. The same conversion process must also be performed for longitude.

```
float minut= lat_minut.toFloat();
minut=minut/60;
float degree=lat_degree.toFloat();
latitude=degree+minut;
minut= long_minut.toFloat();
minut=minut/60;
degree=long_degree.toFloat();
longitude=degree+minut;
```

In the second step, the controller sends the information to a specific IP (the web page) via the Wi-Fi module.

The schematic diagram of the links between the modules is presented in Fig. 12.



**Figure 12. Schematic diagram of the modules of the IoT-based transmitter**

In the extracted data, positive and negative signs are interpreted as follows:

- + for east and north
- for west and south

### Simulation results

In this section, simulation results are examined.

#### Simulation results in normal mode (without fault)

The positional data of the traveled path in terms of latitude, longitude (in degrees) is provided in Fig. 13.

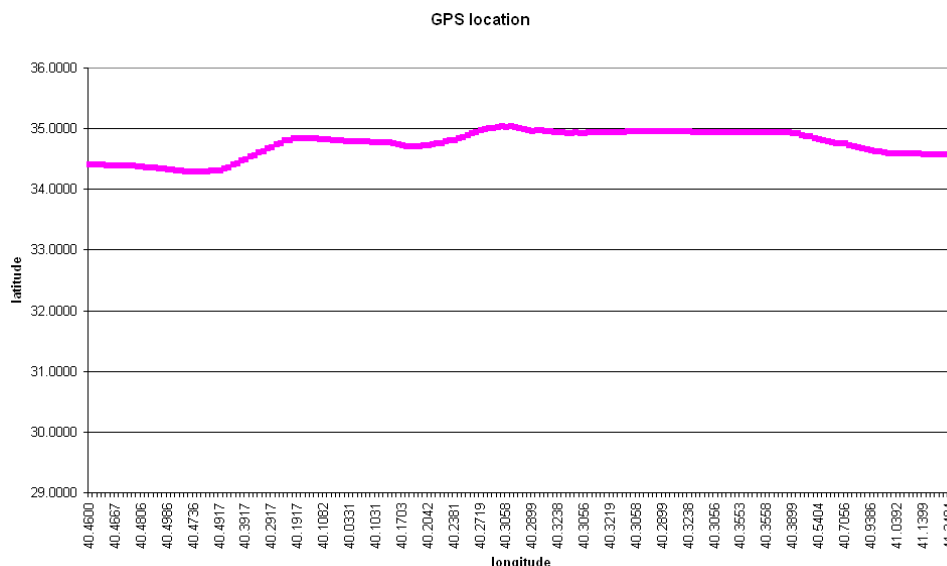


Figure 13. Positional data in the normal state (without fault)

For a closer analysis, the positional data is divided into four parts. (See Table.1)

Table 1: Divided positional data

Positional data for the first quarter of the path	Positional data for two quarters of the path
Positional data for three quarters of the path	Positional data for the entire path

**Simulation results in fault-injection state**

The faults were injected into three modes:

- a) Transient fault
- b) Intermittent fault
- c) Permanent fault



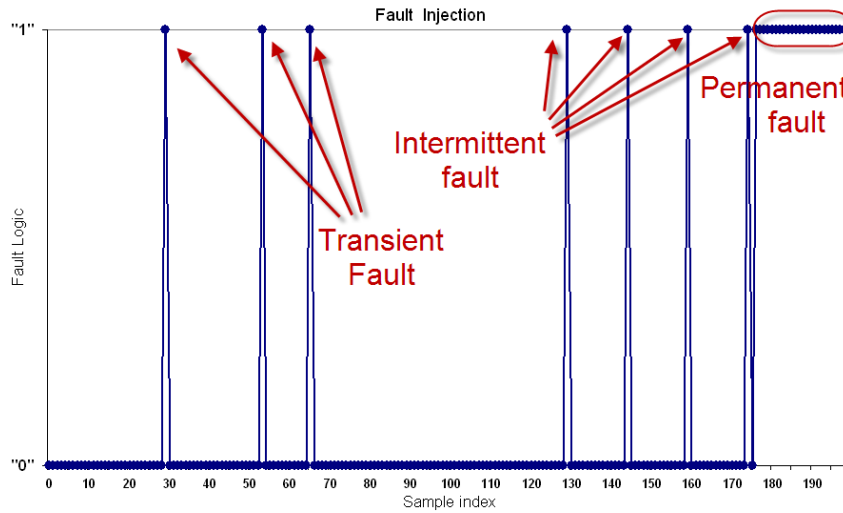


Figure 14. Fault injection diagram

The faults injected based on the ID number of the extracted data samples from a total of 200 samples obtained from the system are demonstrated in Fig. 14.  
TransintFaults={30,54,66}

Intermittent Faults={130,145,160,175} &period=15 samples  
Permanent Fault={178..200}  
After the injection of the above faults, the positional data were read as follows: (Fig. 15)

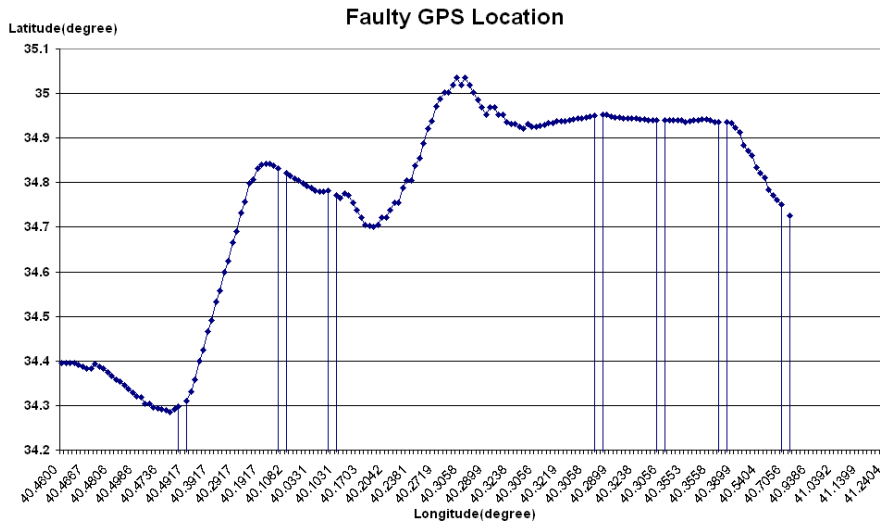


Figure 15. Positional data obtained after fault injection

Transient and intermittent faults are detected based on relative changes in data. The

diagram of these changes before and after fault-injection is provided in Fig. 16.

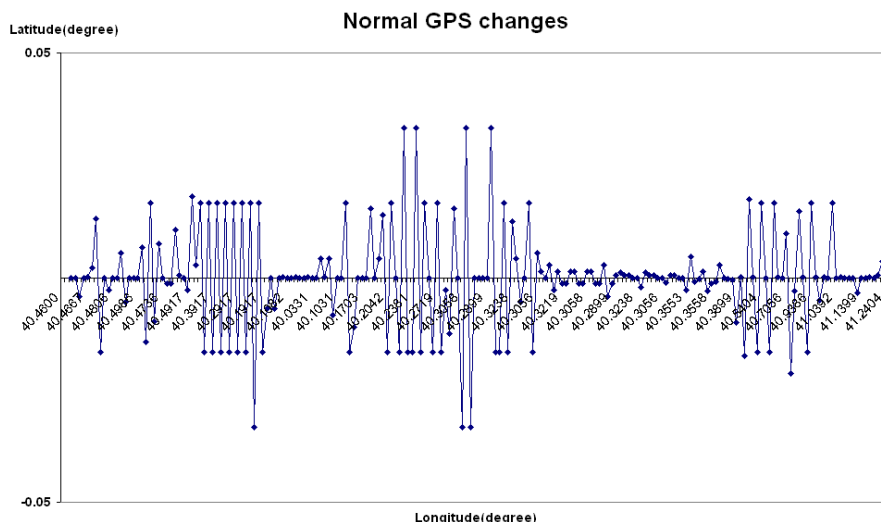


Figure 16. Relative changes in positional data in the normal state

As shown in Fig. 16, the amount of permissible change depends on the speed of the object being monitored. According to this diagram, which expresses the changes based on the geographic data derived from satellite information,

these changes are within the permissible range. According to the above diagram, the changes in the normal state (before fault injection) are less than 0.05 degrees. The degree of change in positional data in the fault-injection state is plotted in Fig. 17.

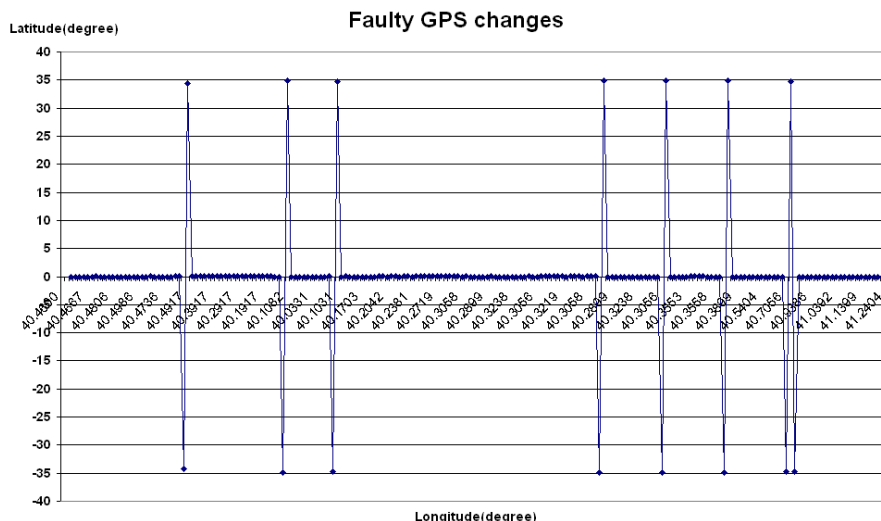
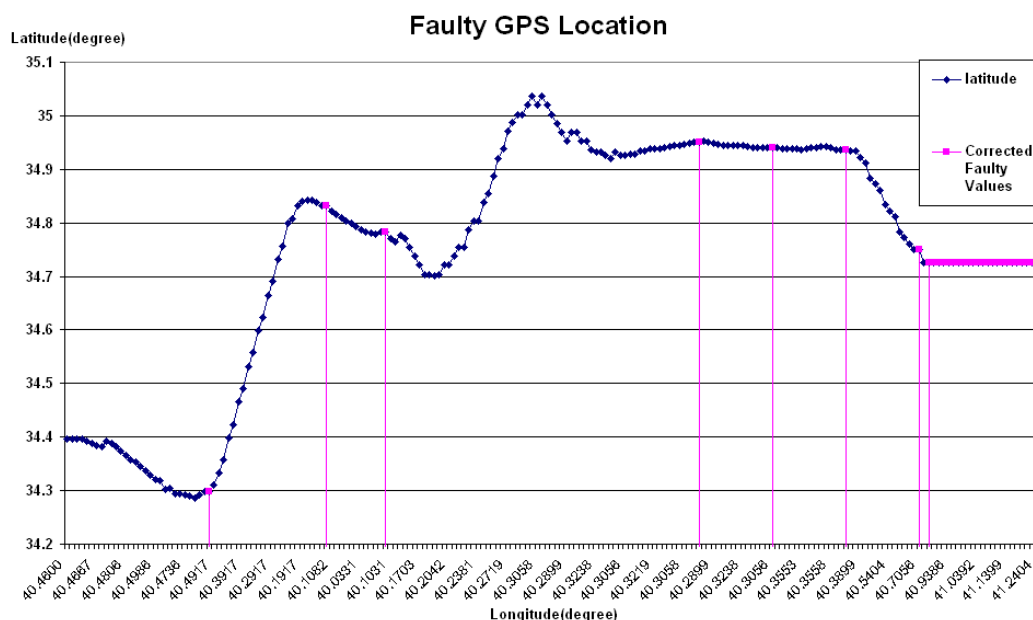


Figure 17. Relative changes in positional data after fault injection

Fig. 17 shows some deviations from the permissible range, which depends on the latitude and longitude of the area of simulation. Here, transient and intermittent faults can be detected by tracking sudden changes. To achieve system stability at these short intervals, the last position read by the positioning system is defined as the system output.

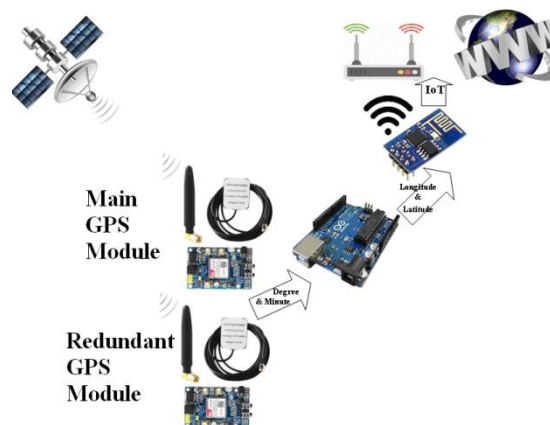
As shown in Fig. 18, the logical faults, which describe the invalid data and the exit of data from the permitted range, have been identified. To this end, the rate of changes and the permitted range of data are checked in the controller. In this method, after detecting the fault, the average of the data preceding and following the fault is used to correct it.



**Figure 18. System output after the correction of intermittent and transient faults**

As can be seen, the above mechanism increases the tolerance to transient and intermittent faults. The proposed mechanism for permanent faults, however, is to add redundancy to the GPS module of the transmitter, so that all three types of fault are covered. For critical applications, the system has to properly cover the transient faults

rather than eliminating the discontinuity by using the previous values. This feature is provided in the hardware redundancy mechanism. The schematic diagram of the proposed circuit for the second mechanism is presented in Fig. 19.



**Figure 19. Schematic diagram of the proposed circuit for all three types of fault**

The three-layer IoT architecture is used for positioning, using position sensors receiving latitude and longitude data. These data are sent to the IoT controller for analysis and positioning, and the user's position is sent via the controller for the

wireless communication module to be sent to the Internet infrastructure. Using the proposed circuit, the system output in the presence of faults will be as shown in Fig. 20.

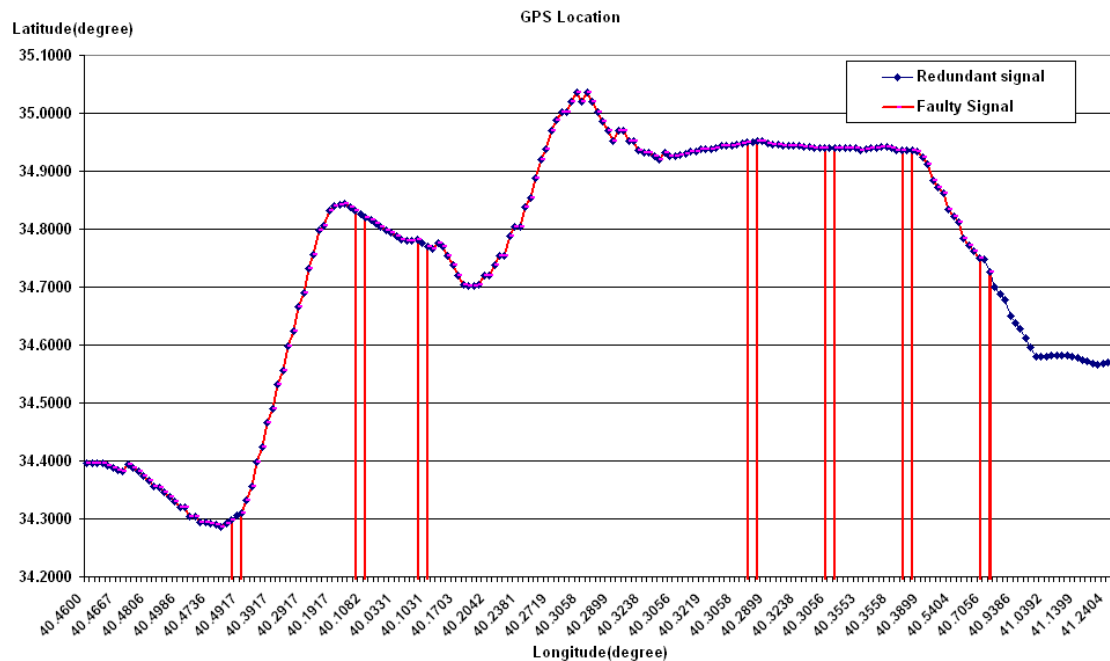


Figure 20. Signals received from the redundancy modules

### Conclusion:

Considering the importance of positioning-based applications and the development of the new Internet of Things (IoT) platforms, this paper presented a mechanism for object tracking, based on the instigation of IoT and GPS modules. Given the inevitability of permanent, intermittent, and transient faults, and the resulting need for fault tolerance, the system was equipped with mechanisms for dealing with these faults. The mechanism proposed in this paper utilizes data validation in the IoT controller to ensure tolerance to transient and intermittent faults and add hardware redundancy to the GPS module to achieve tolerance to permanent faults. The simulation of the proposed mechanisms showed a 22.5% improvement in the fault tolerance of the system after applying the validation mechanism, and a 77.4% improvement in this tolerance after applying the module redundancy, which will be more expensive and suitable for critical applications.

### Author's declaration:

- Conflicts of Interest: None.
- I hereby confirm that all the Figures and Tables in the manuscript are mine. Besides, the Figures and images, which are not mine, have been given the permission for re-publication attached with the manuscript.
- Ethical Clearance: The project was approved by the local ethical committee in University of Kurdistan Hewlêr.

### References:

1. Rodríguez-Puente R, Lazo-Cortés MS. Algorithm for shortest path search in Geographic Information Systems by using reduced graphs. SpringerPlus 2 [Internet]. 2013 Jul [cited 2020 Jan 12];2(291). Available from: <https://rdcu.be/b5zIz> DOI: 10.1186/2193-1801-2-291.
2. Pianini D, Salvaneschi G. Proceedings First Workshop on Architectures, Languages and Paradigms for IoT. arXiv preprint arXiv:1802.00976. 2018 Feb 3; 264:17-1.
3. Qadir QM, Rashid TA, Al-Salihi NK, Ismael B, Kist AA, Zhang A. Low Power Wide Area Networks: A Survey of Enabling Technologies, Applications and Interoperability Needs, IEEE Access [Internet]. 2018 Nov [cited 2020 Jun 19];6:77454-77473. Available from: <https://ieeexplore.ieee.org/abstract/document/8550663> DOI:10.1109/ACCESS.2018.2883151.
4. Nath RK, Bajpai R, Thapliyal H. IoT based indoor location detection system for smart home environment. IEEE International Conference on Consumer Electronics (ICCE). 2018 Jan: 3-1.
5. Ellis B. Real - Time Analytics: Techniques to analyze and visualize streaming data. Wiley Publishing; 2014. 432 p.
6. Kaiwartya O, Abdullah A, Cao Y, Lloret J, Kumar S, Shah R, et al. Virtualization in Wireless Sensor Networks: Fault Tolerant Embedding for Internet of Things. IEEE Internet of Things Journal. 2018 Apr;5(2):580-571.
7. Ngu AH, Gutierrez M, Metsis V, Nepal S, Sheng QZ. IoT Middleware: A Survey on Issues and Enabling Technologies, IEEE Internet of Things Journal. 2017 Feb;4(1):20-1.
8. Silva BN, Khan M, Han K. Internet of Things: A Comprehensive Review of Enabling Technologies,

- Architecture, and Challenges. IEEE Technical Review; 2017 Feb;35(2):220-205.
9. Zhao K, Ge L. A survey on the Internet of things security. Computational Intelligence and Security (CIS). 9th International Conference. 2013 Dec. 667-663.
  10. Mukherjee M, Adhikary I, Mondal S, Mondal AK, Pundir M, Chowdary V. A vision of IoT: Applications, Challenges, and Opportunities with Dehradun Perspective. Proceeding of International Conference on Intelligent Communication. Springer. 2017 Sep;479:559-553.
  11. Saini HS, Srinivas T, Vinod Kumar DM, Chandragupta Mauryan KS. (Eds.). Innovations in Electrical and Electronics Engineering. Springer; 2020. 865.
  12. Casado-Vara R, De la Prieta F, Rodriguez S, Sitton I, Calvo-Rolle JL, Venayagamoorthy GK, et al. Adaptive Fault-Tolerant Tracking Control Algorithm for IoT Systems: Smart Building Case Study. Springer. 2019 May;950:490-481.
  13. Lu X, Liu J, Zhao H. Collaborative Target Tracking of IoT Heterogeneous Nodes. Measurement, Elsevier. 2019 Dec;147.
  14. Song F, Zhu M, Zhou Y, You I, Zhang H. Smart Collaborative Tracking for Ubiquitous Power IoT in Edge-Cloud Interplay Domain. IEEE Internet of Things Journal. 2019 Jul;7(7):6055-6046.
  15. Aftab M, Chau SCK, Shenoy P. Efficient Online Classification and Tracking on Resource-constrained IoT Devices. ACM Transaction of Internet of Things. 2020 Apr.
  16. Chew SH, Chong PA, Gunawan E, Goh KW, Kim Y, Soh CBA. Hybrid Mobile-based Patient Location Tracking System for Personal Healthcare Applications. Conf Proc IEEE Eng Med Biol Soc. 2006:5191-5188.
  17. Ireland D, McBride S, Liddle J, Chenery H. Towards Quantifying the Impact of Parkinson's Disease Using GPS and Lifespace Assessment. 6th International Conference on Biomedical Engineering and Informatics. 2013 Dec: 569-564.
  18. Aziz K, Tarapiah S, Ismail SH, Atalla S. Smart Real-Time Healthcare Monitoring and Tracking System using GSM/GPS Technologies. ICBDS. 2016 Apr:7-1.
  19. Nelson VP. For IoT, alternative location services are better than GPS. Network World, 2018. [cited, Sept 30] Available from: <https://www.networkworld.com/article/3278592/internet-of-things/for-iot-alternative-location-services-are-better-than-gps.html>
  20. Rullo A, Serra E, Lobo J. Redundancy as a Measure of Fault-Tolerance for the Internet of Things: A Review. Springer International Publishing; 2019. Chapter 2, Policies and Autonomy in Federated and Distributed Environments; p.206-202.
  21. Vial J, Bosio A, GirardP, Landraut C, Pravossoudovitch S, Virazel A. Using TMR Architectures for Yield Improvement. International Symposium on Defect and Fault Tolerance of VLSI Systems. IEEE Computer Society, 2008 Oct:15-7.
  22. Kumari P, Kaur P. A survey of fault tolerance in cloud computing. Journal of King Saud University. Computer and Information Sciences [Internet]. 2018 Oct [cited 2020 Jun 17];30(4):560-431. Available from: <https://www.sciencedirect.com/science/article/pii/S1319157818306438> DOI:10.1016/j.jksuci.2018.09.021
  23. Gokhroo MK, Govil MC, Pilli ES. Detecting and mitigating faults in cloud computing environment. IEEE International Conference. 2017 Jul:9-1.
  24. Cheraghlou NM, Khadem-Zadeh A, Haghparast M. A survey of fault tolerance architecture in cloud computing. J. Netw. Comput. Appl. 2016 Feb; 61:p.91-81.
  25. Johnson BW. Design and Analysis of Fault Tolerant Digital Systems. New York: Addison Wesley; 1998. 640 p.
  26. Laprie JC. Dependable computing and fault tolerance: concepts and terminology. Digest of Papers FTCS-15. IEEE. 1985 Jun: 11-2.
  27. Fu H, Cai M, FangL, Liu P, Dong J. Research on RTOS-Integrated TMR for Fault Tolerant Systems. 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing. IEEE. 2007 Aug: 755-750.
  28. Nelson VP. Fault Tolerant Computing: Fundamental Concepts. Computer IEEE. 1990 Jul;23(7):25-19.
  29. Shamshiri RR, Kalantari F, Ting KC, Thorp KR, Hameed IA, Weltzien C, et al. Advances in greenhouse automation and controlled environment agriculture: A transition to plant factories and urban agriculture. Int J Agric & Biol Eng, 2018 Jan; 11(1): 1-22.

## تحسين تحمل الخطأ في أنظمة تحديد المواقع القائمة على إنترنت الأشياء عن طريق التقدم بطلب للتكرار في طبقة جهاز التحكم

نوزاد كامران الصالحي

قسم علوم وهندسة الكمبيوتر، جامعة كردستان هولير (UKH)، أربيل، العراق.

### الخلاصة:

في السنوات الأخيرة ، ازداد انتشار تطبيقات تحديد المواقع للأنظمة القائمة على إنترنت الأشياء (IoT) بشكل متزايد ، ووجدت استخدامات في تتبع الأنشطة اليومية للأطفال والمسنين وتتبع المركبات. من وجهة نظر واحدة ، قد تحتوي البيانات التي تم الحصول عليها من الأنظمة القائمة على نظام تحديد المواقع العالمي (GPS) على خطأ ، مع مراعاة هذه العوامل ، تعتمد الطريقة المقترحة لهذه الدراسة على تطبيق تحديد المواقع القائم على إنترنت الأشياء واستبدال استخدام إنترنت الأشياء بدلاً من نظام تحديد المواقع العالمي (GPS). ومع ذلك ، لا يمكن أن يكون هذا سبباً لعدم استخدام نظام تحديد المواقع العالمي (GPS) ، ولتعزيز الموثوقية ، يمكننا تطبيق مجموعة متوازنة من النظام الحديث والأساليب التقليدية في وقت واحد. على الرغم من أنه يمكن الوصول إلى إشارات نظام تحديد المواقع العالمي (GPS) فقط في الأماكن المفتوحة ، فإن أجهزة نظام تحديد المواقع العالمي (GPS) معرضة للخطأ في المقام الأول عندما يكون جهاز الاستقبال موجوداً في منطقة مدنية ، بسبب الازدحام والتداخل المحتمل. تقدم النتيجة نموذجاً قائماً على التكرار لتحسين تحمل الأخطاء لأنظمة تحديد المواقع القائمة على إنترنت الأشياء. تُظهر نتائج المحاكاة تحسناً بنسبة 22.5% في تحمل الأخطاء لنظام تحديد المواقع القائم على إنترنت الأشياء بعد تطبيق آلية التحقق المقترحة وتحسين 77.4% في هذا التسامح بعد التقدم للحصول على تكرار أكثر تكلفة للوحدة النمطية.

**الكلمات المفتاحية:** التسامح مع الخطأ؛ نظام تحديد المواقع العالمي (GPS) ؛ إنترنت الأشياء (IoT) ؛ وفرة؛ الموثوقية