F4PW: Fog Computer for Pregnant Women

Amy Sene[1(✉)], Ibrahima Niang[1], Alassane Diop[2], and Assane Gueye[1]

[1] Cheikh Anta Diop University, Dakar, Senegal
amy.sene@uadb.edu.sn
[2] Alioune Diop University, Bambey, Senegal

Abstract. Maternal health is very important for healthy and productive populations. Women need access to high quality care during pregnancy. With the significant advance of technology, the use of IoT (Internet of Things) can be a mean to provide a medical quality system. Our paper presents a new architecture called F4PW (Fog for Pregnant Woman). F4PW is inspired from Fog Computing and uses the Wireless Body Area Networks (WBANs). The purpose of this new architecture is to provide an end-to-end real-time remote monitoring to pregnant women during the pregnancy period. Medical applications cannot tolerate high latency and should be reliable at any time. F4PW provides functionality for quickly processing, analyzing and transmission of collected data.

Keywords: Network · Biosensors · Health · F4PW · Fog Computing · WBAN

1 Introduction

Maternal health is paramount for healthy and productive populations. Women need access to high quality care during pregnancy. Unfortunately, most women who are in remote areas and poorer communities struggle to access and to receive adequate care during pregnancy. Some countries fall [13] below the WHO (World Health Organization) standards about the minimal threshold of doctors, midwives and nurses per population. According to the WHO [3], maternal mortality is higher in developing countries, and women living in rural areas or in poorer communities are especially vulnerable. The majority of maternal deaths can be prevented with adequate access to health care before, during, and after pregnancy.

To improve the quality maternal health, and avoid predictable complications during pregnancy we propose a real-time and remote system monitoring during the pregnancy periodical. For this, we have implemented a new communication and computing architecture F4PW (Fog For Pregnant Women). F4PW is based on a 3 layers architecture inspired from Fog Computing which [1] is an extension of the cloud computing paradigm from the core of network to the edge of the network that provides computation, storage, and networking services between end devices and traditional cloud servers. Fog Computing can analyze data collected from IoT devices with minimal overhead in real-time. While the combination of IoT and cloud can solve many challenges; additional challenges are anticipated due to the integration of these two technologies.

A. Seeam et al. (Eds.): InterSol 2023, LNICST 541, pp. 391–402, 2024.
https://doi.org/10.1007/978-3-031-51849-2_25

To collect medical data from pregnant women, F4PW uses the Wireless Body Area Network (WBAN) which [2] is a type of wireless sensor network that requires a number of nodes to be worn on the body or implanted within the human body to collect the vital information from the body. The biosensors are limited resources (such as battery, central processing units (CPUs), memory and storage). Therefore for permanent storage, the generated data must be sent and stored permanently in the cloud (using the Internet), for examination by a staff medical.

Medical applications cannot tolerate high latency and should be reliable at any time. They require [4] real-time monitoring, availability, immediate response and guarantees. Therefor; data collected by the biosensors need to be send in the cloud with very low latency.

Taking account of all these requirements, F4PW permits the analysis of data collected by biosensors and minimizes latency for transmission in the Cloud. To achieve this performance, F4PW determines dynamically the best way with low latency communication and data processing before sending measured data in the Cloud.

F4PW proposal defines measurement frequencies for biosensors. These measurement frequencies indicate the interval time between 2 successive measurements and are adapted according to the state of health of the pregnant woman to be monitored.

This paper is organized as follow. Section 2 offers an analysis of fog computing including fundamental principles, fog computing architecture. A literature review is presented in Sect. 3. Section 4 and 5 give respectively a detailed description and the result performances evaluation of F4PW. Finally, Sect. 6 concludes the paper.

2 Fog Computing

The generated data continues to grow with the proliferation of recent technologies, such as the Internet of Things (IoT) and the Intelligence Artificial. The volume of global real-time data is expected to expand tenfold from 5 to 51 zettabytes [5] between 2018 and 2025, and global data creation is projected to grow to more than 180 zettabytes by 2025. As data is increasingly generated therefore, an essential structure that can effectively process data near IoT, is needed. Fog computing is a layered model for enabling ubiquitous access to a shared continuum of scalable computing resources. The model facilitates the deployment of distributed, latency-aware applications and services.

Fog computing architecture consists of three layers named Cloud layer, Fog layer and the Terminal layer.

- The terminal layer is responsible for extending cloud computing services for the end devices and is located near the end-consumers physical environment. It comprises several IoT-based smart devices, such as mobile phones, sensors, smart vehicles, readers, and smart cards, which devices are used to sense the data to process and transmit the sensed data to the upper layer.
- The fog layer is composed of a large number of distributed fog nodes, where gateways, access points, routers, switches, fog services, and base stations are merged. This layer is located at the network edge.
- The cloud layer comprises several servers and storage devices that offer high-performance computing resources and services to various applications, such as smart

transportation, smart home, and smart factories. It has powerful computing resources to perform extensive computation analysis and provides storage to permanently store a huge amount of data.

3 Related Works

Various research on the use of Iot in Fog Computing has been done.

This paper [6] presents a solution for using Fog Computing in healthcare. To identify the data that required low latency time and prioritize, they used a new layer after the end user, to classify the data according to the application. In addition, they use a Fog Computing Point which receive the traffic collected from the sensors and then validate the traffic to relay in the remote Cloud and the part to process locally. The authors [7] proposed A Multilevel Mobile Fog Computing that offers a multilayer architecture whose processing begins with the lowest level. If the latter is unable to process the data for example for lack of resources, it sends the information back to the upper layer and so on until it reaches the last layer. [8] Proposed a solution for ends users by adopting a 3-layer architecture. Each node is attached to the fog server within its range. For any required information, the end node sends a request to its attached server (that is on second level). The result will be send to the node if the server finds the answers, otherwise it asks the third level that contains all the information. Due to the large number of requests and data received in the cloud, [9] they classifies the requests from ends devices. Thus, a weight is assigned to each type of traffic. This ensures through their forwarding policy the transmission of traffic to the appropriate cloud requests. In [10], after the end layer, it adds an additional module layer. This layer is used to retrieve data from the end devices. After this recovery, the data is grouped according to the types of applications. The GKS (Greedy Knapsack-based Scheduling) algorithm is used for allocating appropriately resources to modules. [11] Proposals a Fog clusters on layer 2. Each fog cluster is composed of several fog nodes. This layer is responsible for recovering data from layer 1 and transferring in cloud. Each fog cluster has a load balancer that allows load balancing between clusters. All load balancers are managed by a main load balancer which ensures load balancing according to the predefined states associated with each cluster. The authors [12] of this article have proposed a solution to improve the monitoring of patients suffering from chronic diseases and requiring remote intensive care by referring to fog computing. For this they adopt an algorithm called TCVC (Tasks Classification and Virtual Machines Categorization). The algorithm allows to classify the tasks in 3 categories (from the most important to the least). Thus each type of task is allocated to a type of VM whose resources are dynamically allocated as needed.

Most of these studies focus on proposals allowing to identify the type of traffic to be processed locally and the part to send to the cloud, which level is used to process a request or attach a node to a server.

Through our proposal we minimize latency while sending all traffic in the cloud by taking into account mobility.

4 F4PW Operation

F4PW is inspired from Fog Computing and provides a regular and real-time monitoring to women during pregnancy. F4PW uses biosensors.

The biosensors are limited resources (such as battery, central processing units (CPUs), memory and storage). Therefore for permanent storage, the generated data must be stored permanently in the cloud and accessible to medical staff.

Real-time processing and event response are essential in Healthcare, data must be immediately available to enable staff medical to make decisions based on evidence. However, the mobility independence and usual living situation for pregnant women must not be affected by this requirement. Similarly, the staff medical must be able to access to measured data regardless of their geographical location. Taking into consideration all these constraints and important factors, we propose a new architecture F4PW inspired from Fog Computing and providing end-to-end services. F4PW is a 3 layers architecture for receiving, analyzing and storing data coming from biosensors worn by pregnant women. F4PW minimizes latency, supports mobility and dynamically adapts according to the mobility of pregnant women and medical staff. Figure 1 depicts the F4PW architecture with the three different layers namely: F4PW-T1, F4PW-T2 and F4PW-T3.

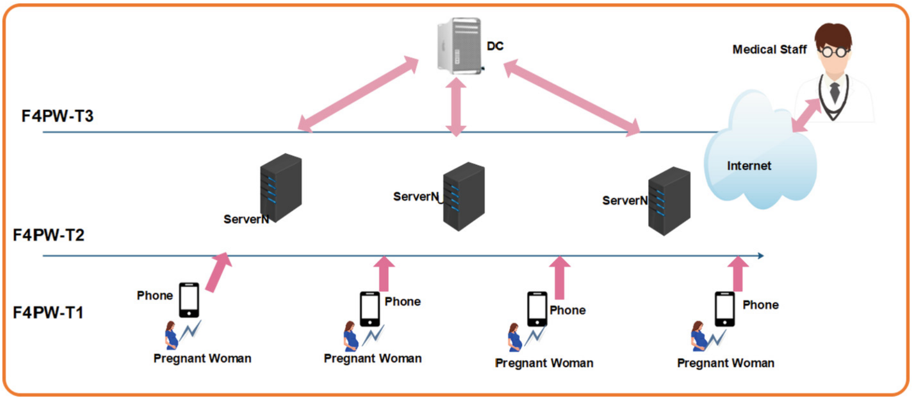

Fig. 1. F4PW Architecture

4.1 F4PW-T1

The F4PW-T1 is the lower layer. F4PW-T1 is composed with different biosensors worn by pregnant women. These biosensors are used to generate and collect healthcare data from the pregnant woman. The phone acts as a gateway between F4PW-T1 and F4PW-T2. Biosensors and the phone communicate through wireless Wi-Fi or Bluetooth.

In order to determine the health status of the pregnant woman, we must have some information. For that, we use blood pressure, blood glucose, temperature, pulse biosensors. The biosensors allow the measurement of health-related physiology. For each, thresholds are defined to classify data (Table 1).

After each measuring data, the following algorithm is used to determine the health status (Fig. 2):

For each state, a measuring periodicity validated by a doctor is predefined. This frequency (F) indicates the interval time between 2 successive measurements. We can

Table 1. Data Classification

Type	Class 1	Class 2	Class 3	Class 4
blood pressure	$7 \leq D \leq 8$ and $12 \leq S \leq 13$	$6 \leq D < 7$ or $13 < S \leq 14$	$5 \leq D < 6$ or $14 < S \leq 15$	$5 < D$ or $S > 15$
blood glucose (G) g/l	$0{,}7 \leq G \leq 1$	$1 < G \leq 1{,}1$	$1 < G \leq 1{,}25$	$G > 1{,}25$ or $G < 0{,}7$
Pulse (P) beats/min	$80 \leq P \leq 120$	$120 < P \leq 140$	$140 < P \leq 160$	$P > 160$
Temperature	$37.6° \leq T \leq 38$	$38 < T \leq 39$	$39 < T \leq 40$	$T > 40$

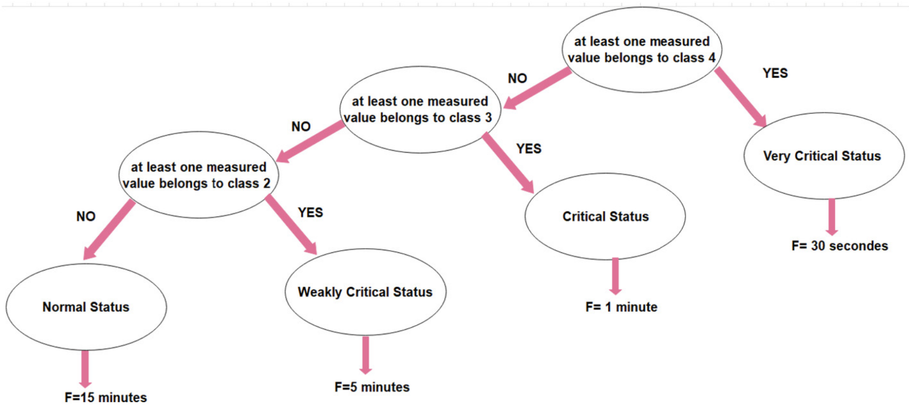

Fig. 2. Algorithm for Health Care Status

see through the algorithm that, more the health status becomes critical, more the higher the frequencies.

4.2 F4PW-T2

The second layer is the F4PW-T2 layer. F4PW-T2 supports collaborations between the F4PW-T1 and F4PW-T3. This layer has the responsibility to receive data generated in FW4P-T1 and their transmission to FW4P-T3. In others words, this layer consisted as the communication between the phone and the F4PW-T3 layers. Health data are delay-sensitive and real-time. To ensure that measured data is delivered to the cloud with very low delay, F4PW minimizes the latency for transmitting data to F4PW-T3.

Through the defined frequency, the phone knows the duration of the next measurement. Therefore just before measurement by biosensors the following steps are performed by the phone:

- The phone needs to look up the network latency with each serverN to determine which offers the low latency connectivity services. This latency refers to the time it takes for data to travel from the phone to a serverN. Obviously latency can depend on geographical distance to the destination. But sometimes it can happen that a remote

server offers lower latency than a near server. Hence the importance of determining before each measurement the latency time to reach the different servers. To get this metric the phone sends a ping request to each serverN. The ping result between the phone and each server N is represented by P.
- To obtain the second metric, a same process is executed regularly on each server N to determine the local processing time (Tlp) of the serverN. The execution of this process locally on each server makes possible to have the time it takes by each serverN to process a request. This procedure allows to know, the local performance of the serverN.

At this step, the phone has the required information (latency and local processing time) for each serverN.

In order to identify the optimal serverN, the following algorithm is executed:

– n defined the number of ServerN
– Each ServerN is characterized by 2 parameters P and L where: P is the latency to reach the Server and L is local processing time.

```
Int array[n-1]
Pi= Result ping from phone to ServerN0
Li= Result local processing time of  ServerN0
Ti= Pi+Li
Array[0]=Ti
b=0
B=Ti
FOR (i=1 to i=n-1)
        Pi= Result ping from phone to ServerNi
        Li= Result local processing time of  ServerNi
       Ti= Pi+Li
       Array[i]=Ti
IF array[i]<B
                     B=array[i]
                     b=i
      END IF
END FOR
```

The serverN(i) with the lowest B is considered to be the best. Therefore, the phone sends the data to this server. The same procedure is repeated before each measurement. With this technique, F4PW can independently on the geographical position of the pregnant woman, dynamically determine the server that offers the best performances, unlike to solutions that associate a user to a nearest server. Whereas, a remote server can be more available than a nearest.

This level also allows notifications. Instead to waiting for arrival and analyzing data in the F4PW-T3 level before notifying the medical staff can increase medical intervention times. Notifications are generated according to the health status of the pregnant woman. The phone sends a mail and SMS to medical staff in critical and very critical.

The phone takes into the Biofeedback: that can help to improve the woman's health during pregnancy. Through this layer, the phone has the possibility to generate audio messages in anomaly situations to alert the pregnant woman.

4.3 F4PW-T3

For providing high-availability and improving data accessibility we set up multiple serversN connected to the cloud DC. Therefore, even when a serverN fails, the other serversN continue working and hence the failure has no major consequences. These servers ensure the role of permanent storage and in-depth analysis of all global data. They are accessible to medical staff in charge of patient follow-up. For accessing to serverN, the medical staff uses the same scenario described in F4PW-T2.

We have the same database in all serversN and Cloud DC to ensure data conformity. For this, F4PW uses automatic replication data between databases. The DC is the central element for the automatic replication.

We have a master slave relation between the DC and each ServerN. As shown in the following figure; any change operating in a ServerN is automatically duplicated on to the Cloud DC and after the change is instantly applied from the cloud DC to others serversN (Fig. 3).

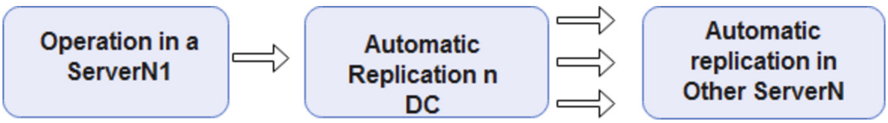

Fig. 3. Replication

Thus, the staff medical handle the same information regardless the serverN they are connected to.

5 Performance Evaluation

In this section, the experimental performance of F4PW is evaluated. To demonstrate the efficiency of F4PW, we have deployed 4 ServerN in dispersed geographical locations. We have also 4 persons distributed geographically as follows:

- PW1 is living in the same geographical area than ServerN1
- PW2 is living in the same geographical area than ServerN2
- PW3 is living in the same geographical area than ServerN3
- PW4 does not live in any of these areas

The following table gives the distance between the ServerN (Table 2).

After several sequences made by each person residing in these different areas, the following results (latency + local processing time), are obtained (Fig. 4, 5, 6 and 7).

Table 2. Distance between ServerN

	ServerN1	ServerN2	ServerN3	ServerN4
PW1	near	53,28 km	230 km	4088 km
PW2	53,28 km	near	260 km	4094 km
PW3	230 km	260 km	near	4285 km
PW4	107 km	54,22 km	274 km	4124 km

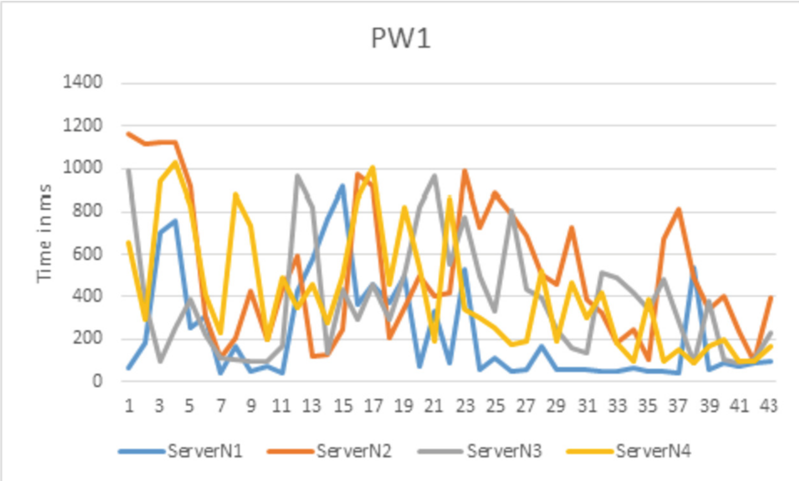

Fig. 4. PW1 Result

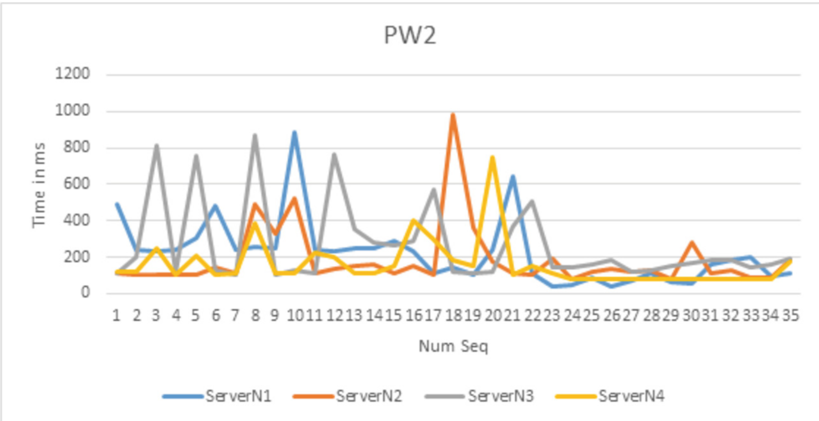

Fig. 5. PW2 Result

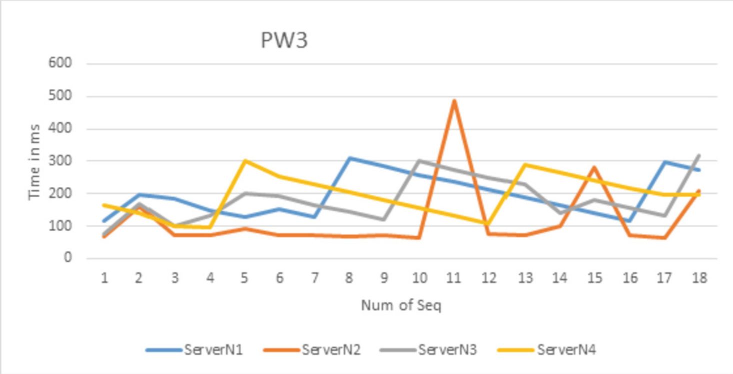

Fig. 6. PW3 Result

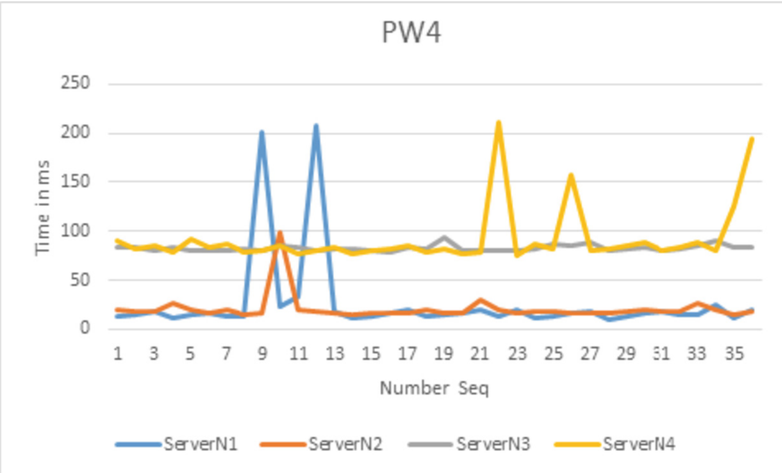

Fig. 7. PW4 Result

An in-depth analysis of these figures confirms that a geographically remote ServerN can offer higher performance than a near ServerN. Hence the importance of determining for each ServerN the B value before each measurement to allow the phone to check on which server it should send the measured data.

The following figures give for each PW the percentage of use per server. If we take the PW3 example, for all his sequences the serverN associated with its area never provided the best performance, which is why he used the other serversN for transmitting data (Fig. 8, 9, 10 and 11).

Through the results, we can see that PW4 which lives in the capital obtains the lowest delta values compared to the others which obtain higher values. These values confirm the significant gap in household connectivity between large cities and rural areas. Thus pregnant women residing in remote areas should not be left out. To address

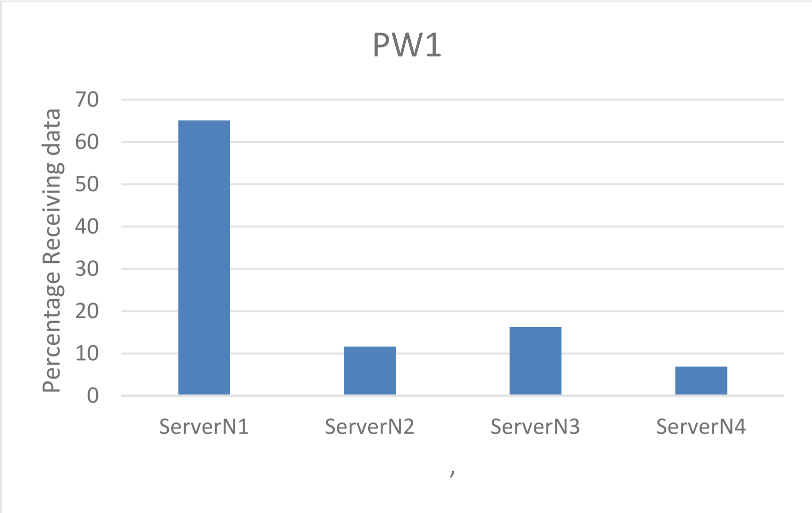

Fig. 8. PW1 Percentage using ServerN

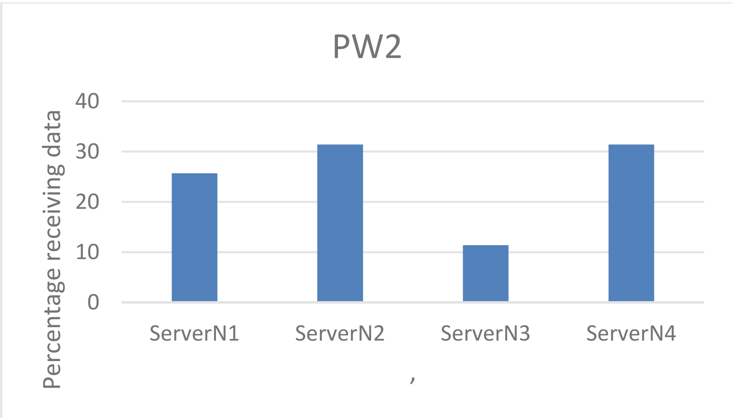

Fig. 9. PW2 Percentage using ServerN

these disparities, F4PW provides a solution which minimizes the time needed to send the data of these pregnant women to the cloud.

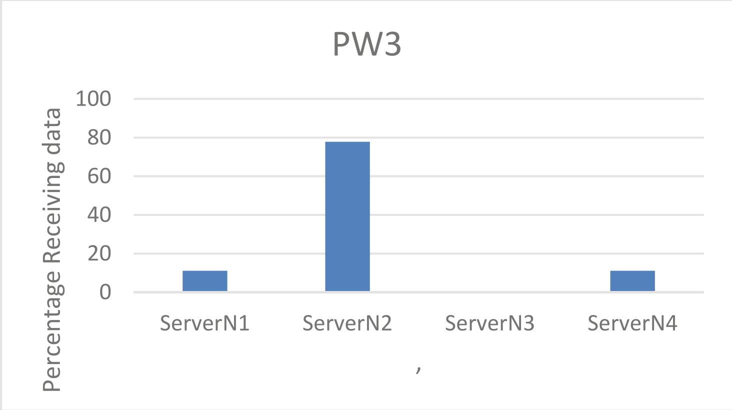

Fig. 10. PW3 Percentage using ServerN

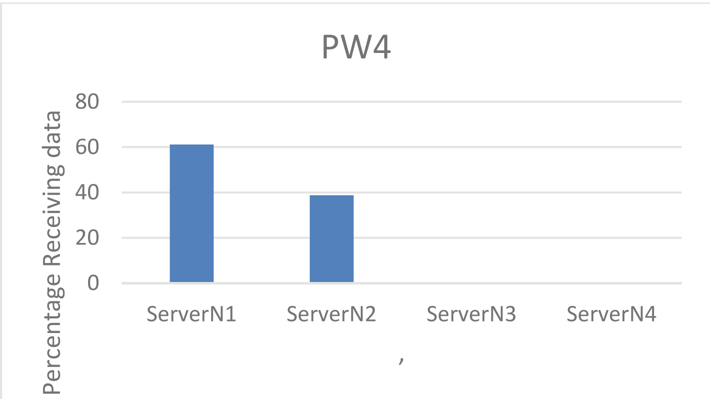

Fig. 11. PW4 Percentage using ServerN

6 Conclusion

The use of Fog Computing in IoT can improve our quality of life especially in healthcare and particularly for pregnant women. In this paper, we proposed a new architecture called F4PW inspired from Fog Computing. The purpose of F4PW is to provide better care for pregnant women especially those residing in rural areas and help achieve health equity. F4PW provides accurate real-time data and facilitates remote monitoring for pregnant women. Low latency is required in healthcare. F4PW provides functionality for quickly processing, analyzing and transmission of data collected from biosensors. The results obtained through our simulation prove that F4PW provides very minimal latency whatever the geographical position of the pregnant woman as well as the medical staff.

The results obtained through our proposal are satisfactory and open many research perspectives. For improvement, we plan in our future work to have a cluster at the third level.

References

1. Yi, S., Li, C., Li, Q.: A survey of fog computing: concepts, applications and issues
2. Pushpan, S., Velusamy, B.: Fuzzy-based dynamic time slot allocation for wireless body area networks (2019)
3. https://www.who.int/news-room/fact-sheets/detail/maternal-mortality
4. Pareek, K., Tiwari, P.K., Bhatnagar, V.: Fog computing in healthcare: a review (2020)
5. Yang, M., Chen, X., Tan, L., Lan, X., Luo, Y.: Listen carefully to experts when you classify data: a generic data classification ontology encoded from regulations (2022)
6. Lee, Y.-H., Lin, F.J.: Enabling IoT/M2M System Scalability with Fog Computing. In: Li, B., Zheng, J., Fang, Y., Yang, M., Yan, Z. (eds.) IoTaaS 2019. LNICSSITE, vol. 316, pp. 489–502. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-44751-9_41
7. Chen, Z., Xiao, N., Han, D.: A multilevel mobile fog computing offloading model based on UAV-assisted and heterogeneous network (2020)
8. Sheltami, T.R., Shahra, E.Q., Shakshuki, E.M.: Fog computing: data streaming services for mobile end-users (2018)
9. Al Masarweh, M., Alwada'n, T., Afandi, W.: Fog computing, cloud computing and IoT environment: advanced broker management system (2022)
10. Dadmehr, R., Nickray, M.: Low-latency and energy-efficient scheduling in fog-based IoT applications. Turkish J. Electr. Eng. Comput. Sci. **27**(2), 1406–1427 (2019)
11. Singh, P., et al.: A fog-cluster based load-balancing technique (2022)
12. Aladwani, T.: Scheduling IoT healthcare tasks in fog computing based on their importance. Procedia Comput. Sci. (2019)
13. Health Workforce Thresholds for Supporting Attainment of Universal Health Coverage. In: The African Region: World Health Organisation (2021)