Advances in Intelligent Systems Research, volume 172
Sriwijaya International Conference on Information Technology and Its Applications (SICONIAN 2019)

A Proposed Design of Realtime Patient Monitoring System Using Websocket as a Basis of Telemedicine

NEHRU[1] and Andreo YUDERTHA[2*]

[1,2]*Departement of Electrical Engineering, Faculty of Engineering, Universitas Jambi, Jambi, Indonesia*
**Corresponding author: andreo.y@unja.ac.id*

ABSTRACT
Telemedicine provides service which can monitoring patient remotely. Some requirements must be provided by telemedicine system, such system is able handling real-time communication, managing account, and medical record. WebSocket is one of the protocols that can we utilize for bidirectional communication. In this design, we recommended using socket.io library to implement WebSocket in a server, client and device. We propose a scenario for authenticating users and monitoring device. Data communication transaction can be designed for security issues by using a hashing method to generate a token and then authorized it every data transaction. Realization of a real-time system, WebSocket that implemented by socket-io is good enough to handle 25 connections per second which have less than one-second latency.
Keywords: *telemedicine, IoT, WebSocket*

INTRODUCTION

Several diseases can change the patient's condition drastically and suddenly. Patients with this condition need treatment as early as possible to avoid the condition become worse. On the other hand, patients cannot always be on doctor's observing, so the handling of diseases patients can't be on time. One example is a heart disease patient, who has a heart attack can occur suddenly and can cause death if not treated as early as possible. Even more, handling outpatients, that is difficult to monitor their condition. Especially for remote areas in Indonesia, where the distance between hospitals and residents is quite far. The quality of care for patients will decrease if the patient's condition cannot be monitored at any time, so patient handling will be late. The late of handling the patient can lead to death and will worsen the patient's condition.

The solution to monitor a patient's condition any time is to use the Internet. The patient's condition is monitored by using sensors and sending it via Internet so that both nurses and doctors can monitor the patient's condition wherever and whenever. The challenge in implementing the monitoring system is that the system has capability catching patient condition with some parameters, sending data and having an application that able to manage data and communicate patients with healthcare providers is needed. Telemedicine is a technology that connecting the patient with the healthcare provider without time and geographical gap. There are four elements which closely related to telemedicine, provide clinical support, overcome the geographical gap, connecting users who are not in the same physical location, using various types of ICT, and improve health outcomes [1].

Telemedicine can be broadly defined as using information technology to provide medical information and services. Information technology including telephone, facsimile, and distance learning, telemedicine is increasingly being used as an abbreviation for electronic remote clinical consultation. Interesting in telemedicine increased dramatically in the 1990s. State and federal US allocations for telemedicine and related technology are likely to exceed $100 million in fiscal 1994-1995 [2]. At present, the development of telemedicine systems is increasing by adapting current existing ICT technologies.

An intelligent embedded device has been developed to monitor a patient's condition. The device was developed to reduce the critical situation between shift interval of the doctor's arrival where the doctor is not around the patient so that cause the patient can lose his life [3]. The device was developed using Arduino Uno as a microprocessor and GSM as a data transmission medium. Data are sent to the doctor via SMS by the system obtained from the device wear on the patient's body.

The development of a remote patient monitoring system has also been developed in several studies. The developed system involves information about vital signs that can help health care providers more easily send assistance to patients when their health is in an emergency [4]. This research presents a solution in establishing a web-based patient monitoring system, and patients do not have to be directly in a certain location to get help but allow patients and emergency response units to interact beyond cellular networks.

Using web services in implementing remote patient monitoring has been developed by Dogan et.al [5]. The research is based on web services to improve telemedicine services. The system offers Electronic Health Recording (EHR) consisting of biological signals and physiological parameters. The system is built based on a client-server consisting of patient monitor software, web services and interfaces that allow remote healthcare to access web services.

Telepathology systems (TPS) were established by Petrolini et.al [6]. That system can improve and accelerate the process of diagnosis, particularly in those complex cases where the second expert/specialized opinion is required. It allows pathologists to report pathology cases employing a collaborative environment with audio/video conferencing and that simulates virtual multiheaded microscopy. That system scored good results, especially regarding its effectiveness, learnability, and comfortability.

Repu et.al proposed advanced communication technology for collaborative learning in Telemedicine and telecare [7]. They use video conferencing platforms, high-speed communication networks, web portal, and live streaming to implement telemedicine. Collaborative telemedicine process includes Registration, Creation of Room ID for the event, Scheduling event for recording and live streaming, updating of information in the portal for sharing content, telemedicine events, etc and sharing of standard operating protocol or manual for new participants. The service can access from any platform, like Desktop, H.323 device, Android or IOS smartphone. Communication is supported by various known protocols like NAT, ICMP, TCP/IP, and UDP.

Design and implementation technique of Microcontroller-based Real-time WebSocket Server for monitoring and control applications have done by Santi Nuratch [8]. All proposed algorithms are designed and implemented as a microcontroller platform-independent. To evaluate the proposed algorithms, they develop Web-based applications working as 4-channel oscilloscope and 4-channel PWM signal designer to monitor and control the system. The experimental results show that the proposed algorithms can be used for real-time application as expected.

Websocket was implemented to develop a realtime data sensor acquisition. Sensors management on a framework would base on a designed model. The model is developed in the RESTful API form. Model development considers the security aspect, sensor heterogeneity, and privilege. The framework design is done by using the REST API to send data from the sensor to the server. Problems that arise when using REST API is that data cannot be sent to the client in real-time. The use of the WebSocket in this framework can be one solution so that sensor data can be sent in real-time. This framework testing is done by developing a client application using Angular 5. The test results show that the framework can be used to transmit data from sensors in realtime, and client applications that are built can manage various types of sensors [9].

Analysis and Design System

In the application of telemedicine, three main entities are included in the system, like a patient, doctor or nurse, and administrator. A patient is a person who is the object being monitored, the doctor or nurse acts as an observer who monitors the condition of the patient, as well as the administrator in charge of managing the patient and observer. Patients will be given an account and sensor devices, such as temperature and heart rate sensors that are used to measure patient conditions that can be monitored by observers. Patients are also given the facility to make calls and video calls to the observer to consult their conditions. Figure 1 shows the use-case of the proposed system.

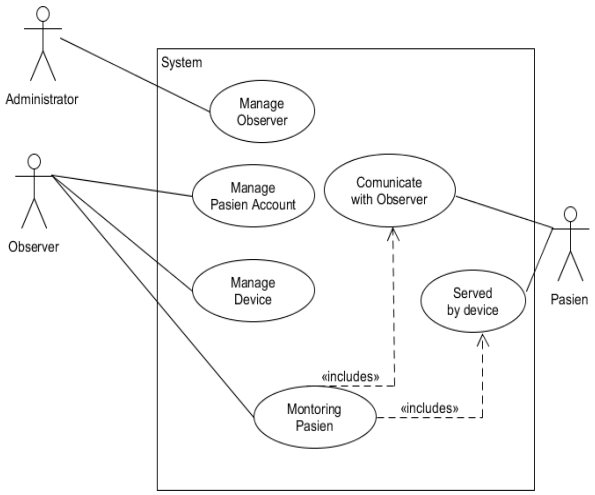

Figure 1. Use case diagram

Telemedicine provides service which can monitoring patient remotely. Some requirements must be provided by telemedicine system, such system is able handling real-time communication, managing account, and medical record. Communication between patients and healthcare providers is needed. A good system must have the ability to transaction data in real-time. Websocket is one of the protocols that can we utilize for bidirectional communication. In this design, we recommended using the socket.io library to implement WebSocket in a server, client and device. Socket io has characteristics to realize realtime and bidirectional communication.

Socket io has two core methods such as sending and receiving event which used to realize bidirectional communication [10]. Script below, show how sending and receiving an event with socket io. When a client connected, the server generates socket objects, so every connection has information about connection attributes include id code for connection.

```
socket.emit('event-name','data-that-will-be send')
socket.on('event-name', callbackfunction())
```

Emit method is used to send data to the socket that listens according to event name. Both methods are the main transaction between a device, patient, and observer (healthcare provider). State diagram showing the behavior of the device after connected to the internet will follow to Figure 2. After connecting to the Internet, a device sending a register to the server and server checked the hashing ID, if ID is authenticated, the server will send a token for the next data transaction. At the same time, the server will get observer ID from databases for broadcast data to the connected observer. To recognize a device, we use identity for every device. The identity will generate by the system and hashing it with the JSON web token. Device stores that identity to get code as a token for authorized that connection. Figure 3 and Figure 4 Show sequence diagram of the system.

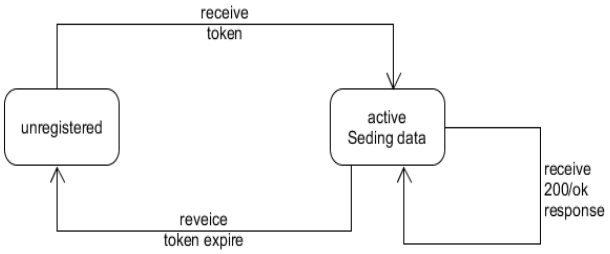

Figure 2. State Diagram Device

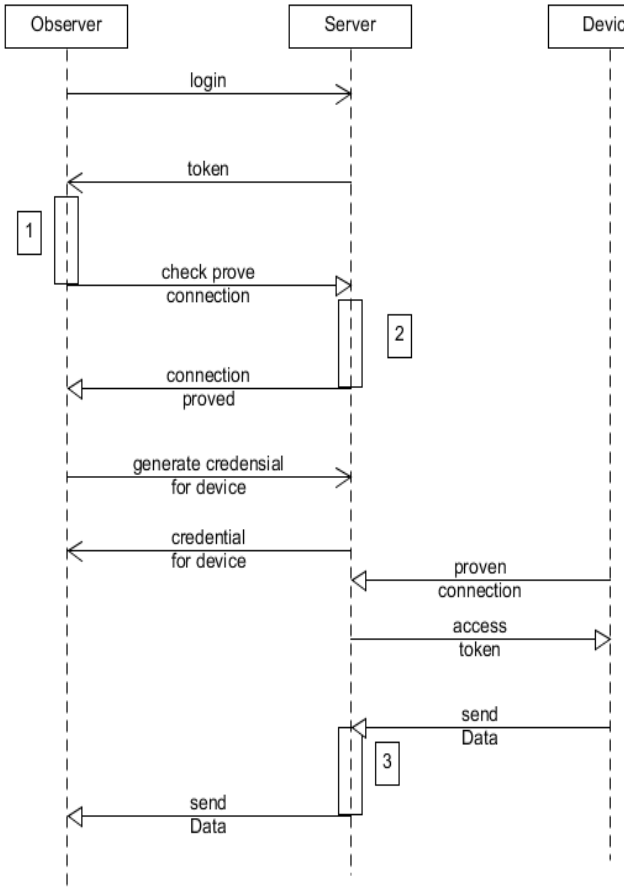

Figure 3 Sequence Diagram Device and Observer

1 = save token, and including token in socket information object
2 = check token for suitable socket is valid
3 = check receiver (observer)

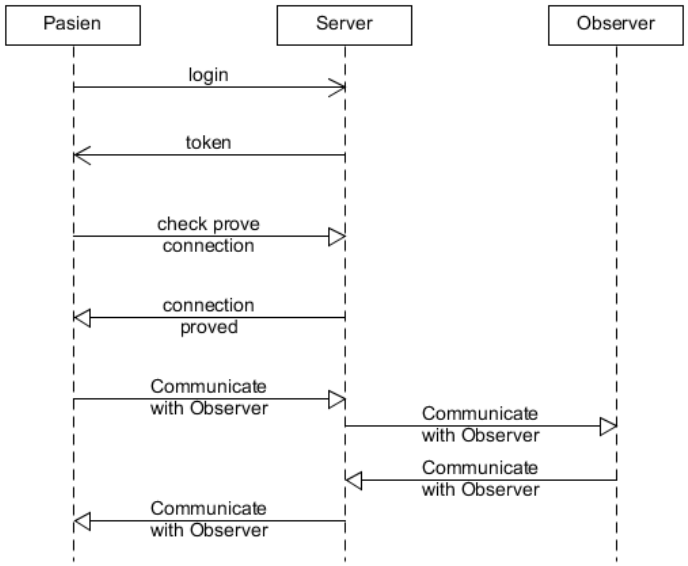

Figure 4 Sequence Diagram Patient with Observer

There are three main entities in this system, that is a device, observer, and patient. The table of device and observer have a relation with a patient. Each patient can relate with more than one device and observer. The device is used to capture and record patient conditions. In other hand, relation with observers is useful to know who is receiving data and monitoring patients. In practicals, selection the observer from the patient table and then seeing what is connection object in a socket and sent data from observer socket.

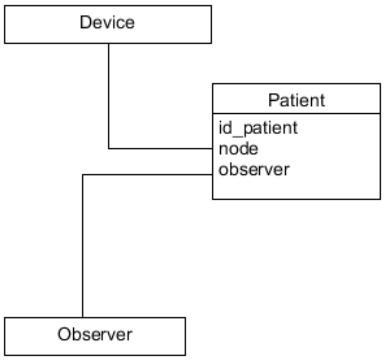

Figure 5. Data Object Relation

For an accepted heterogenous data sensor, a standardized data format is needed. JSON (Javascript Standard Object Notation) is used as a data format. Data consist of two attributes, it is a type of sensor or data, and data its self. Following script show data format.

{type:*sensor type*, data: *data from sensor*}

IMPLEMENTATION

The model was tried to implementation with MEAN (MongoDB, Express, Angular, Nodejs) Stack. In the server-side, there is two separated data communication model, that is synchronizing communication with socket io and HTTP with REST API model for authenticating the observer to access system and generate an ID for a device.

Some main services using rest model and with socket transaction, see table 1 and table 2.

Table.1 Main API for observer

Method	URL	Response (data)	Description
POST	/login	Token	For authenticate Observer
POST	/node	device id	Create new device
POST	/patient	patient data	Create patient and assign device to pasien, observer saved

Table 2. Main socket IO event

Event	Sender	Receiver	Data
observer.register	Observer	Observer	obserser description
device.register	Device	Device	token
device.data	Device	Observer	Data

Psedocode for handling event in socket io:

```
Socket.on('observer.register',function(data){
        Decoded data
        If id exist
                socket.emit(‘observer.register’,observer decription)
                add observer in online list
});
Socket.on(‘device.register’, function(data){
        Decoded data
        If id exist
                Add device in online list
                Generate access token
                socket.emit(‘node.register’, access token)
});
Socket.on(‘device.data’, function(data){
        Decoded data
        If id exist
                Check observer from device table
                Find observer in online list
```

```
                For each observer
                        Socket.emit('device.data', data)
});
```

Node MCU can be base to develop a device to capture patient conditions, for example, temperature and heartbeat. NodeMCU is a microcontroller that has embedded wifi. NodeMCU is used by uploading code, including a program to manage communication with WebSocket protocol. The proposed scenario can be programmed to NodeMCU, there some socket io libraries that can be utilized to implement that design. Figure 6 shows an example circuit for creating a device that using node MCU, thermometer and heartbeat sensor.

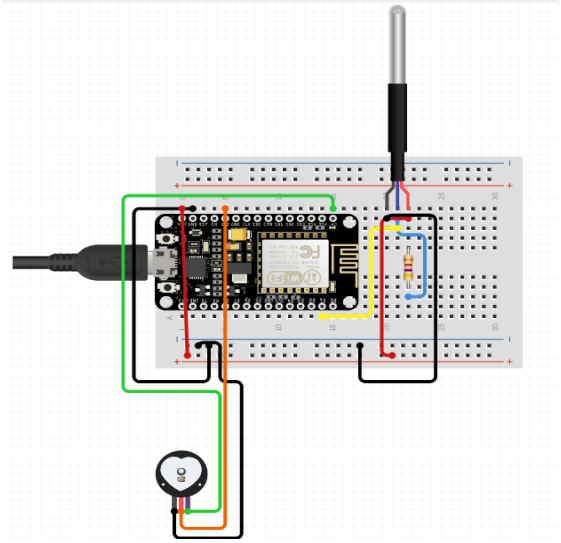

Figure 6. Circuit Example for device

TESTING

We measure latency in one way data transactions with socket.io. Data was sent and server response according to request data. We measured time interval between request and response from the server. Figure 7 shows the testing scenario.

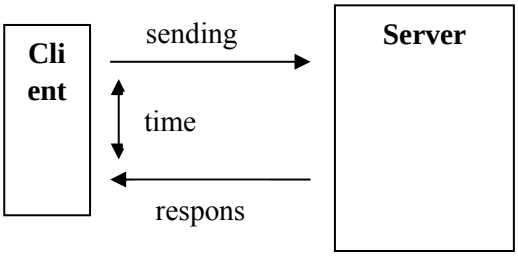

Figure 7. Testing scenario

Performance is measured by a stress test. The latency of connection is a parameter to assess the performance of the system. We try to make a connection to a server for every second and count how long the response comes. The system runs on the server with 2.4Ghz CPU and 1 Gb Memory. The scenario was tested with 5 connections to 100 connections every second and run until 10 seconds. Figure 8 shows latency data communication with socket io. Three data captured, like minimal, maximal and median of latency in 10 seconds.

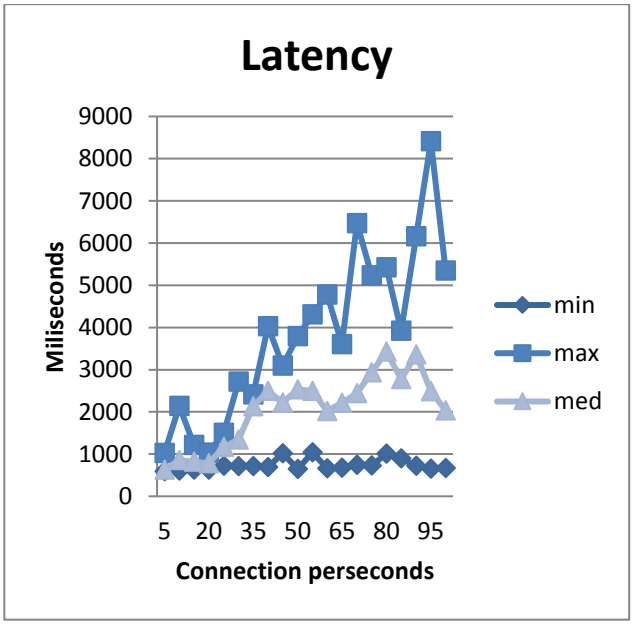

Figure 8. Latency data communication socket io

The result shows medians of latency until 25 connections in a second indicate that communication stands on below one second. Maximum latency every scenario is very fluctuating, and minimum latency of at least one second. This test runs on very minimum server specifications and standard internet connection speed in Indonesia. So, using socket io as a realtime communication is not to bad for a standard number of connections (average 25 connection per second) but for a large system that needs to handle much entity to connect require others scenario and architecture design to be tested such parallel server implemented.
Minimum latency is shown stagnant latency. It indicates that the system scenario is not affecting the latency. The latency increase when number connections increase. To improve the performance of the system, we can improve server specifications, like memory and CPU speed. Maximum latency is very fluctuating, but still shown an upward trend. It is caused by the concurrency of socket connection processing.

SUMMARY

Websocket can be implemented as a communication media to realize a real-time monitoring system. To create a monitoring system for the patient, we can include three entities such as device, observer, and patient itself and there has a relation in the database side. The device must be designed, regulated and authenticated the connection with

the server to identified. Data communication transactions can be designed for security issues by using a hashing method to generate token and then authorized it every data transaction. Realizing real-time system, WebSocket that implemented by socket io is good enough to handle 25 connections per second which have less than one-second latency.

ACKNOWLEDGMENT

This research was financially supported by LP2M Universitas Jambi.

REFERENCES

[1] "WHO | Global Observatory for eHealth series - Volume 2." [Online]. Available: https://www.who.int/goe/publications/ehealth_series_vol2/en/. [Accessed: 08-Okt-2019].

[2] D. A. Perednia and A. Allen, "Telemedicine Technology and Clinical Applications," JAMA, vol. 273, no. 6, pp. 483–488, Feb. 1995.

[3] P. W. Digarse and S. L. Patil, "Arduino UNO and GSM based wireless health monitoring system for patients," in 2017 International Conference on Intelligent Computing and Control Systems (ICICCS), 2017, pp. 583–588.

[4] M. Omoogun, V. Ramsurrun, S. Guness, P. Seeam, X. Bellekens, and A. Seeam, "Critical patient eHealth monitoring system using wearable sensors," in 2017 1st International Conference on Next Generation Computing Applications (NextComp), 2017, pp. 169–174.

[5] R. Ö. Doğan and T. Kayıkçıoğlu, "Remote patient monitoring and Electronic Health Record system based on web services," in 2016 24th Signal Processing and Communication Application Conference (SIU), 2016, pp. 1785–1788.

[6] V. A. Petrolini, E. Beckhauser, A. Savaris, M. I. Meurer, A. von Wangenheim, dan D. Krechel, "Collaborative Telepathology in a Statewide Telemedicine Environment - First Tests in the Context of the Brazilian Public Healthcare System," dalam *2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS)*, 2019, hlm. 684–689.

[7] R. D. Chand, A. Kumar, A. Kumar, P. Tiwari, R. Rajnish, dan S. K. Mishra, "Advanced Communication Technologies for Collaborative Learning in Telemedicine and Tele-care," dalam *2019 9th International Conference on Cloud Computing, Data Science Engineering (Confluence)*, 2019, hlm. 601–605.

[8] S. Nuratch, "Design and implementation of microcontroller-based platform-independent Real-time WebSocket Server for monitoring and control applications," dalam *2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 2017, hlm. 624–627.

[9] Nehru, A. Yudertha, dan Y. R. Hais, "A Design Framework for Real-time Heterogeneous Sensor Data Acquisition As a Basis of IoT Ecosystem Development," dalam *2018 International Conference on Information Technology Systems and Innovation (ICITSI)*, 2018, hlm. 205–208.

[10] "Socket.IO — Docs," *Socket.IO*. [Online]. Available: https://socket.io/docs/index.html. [Accessed: 17-Okt-2019].