

Implementation of a Real-Time Communication Library Between Smart TV Devices and Android Devices Based on WebSocket for the Development of Applications



Gian Gabriel-Caycho , Miguel Blas-Perez , and Alfredo Barrientos 

Abstract The smart device industry has grown in recent years, so the vast majority of people have a smart TV and at least one smartphone in their homes. The transformation of the digital era seeks to improve the user experience and facilitate the use of different smart devices. Due to this, it is proposed to develop a communication library that allows multiple Android devices to be connected to an LG WebOS Smart TV and thus promote the development of applications for Smart TVs, avoiding implementation from scratch and speeding up the deployment of new applications that generate great value for users. In this article, a comparison was made of the main communication protocols existing and compatible with the devices involved. During the development stage, the communication library based on the WebSocket protocol was implemented. Also, a retro videogame was implemented that used the communication library to connect the LG WebOS smart TV with the Android Smartphone. Finally, the results obtained and the opportunities for improvements found for the development of future applications will be shown.

Keywords WebOS · WebSocket · Android · Real-time communication · Smart TV

1 Introduction

At present, the vast majority of people have at least one smart device for daily use, which allows them to interact with other devices on their network or through the Internet, so the world of IoT has been growing with the passage of years. The use of Smart TVs has become very common in homes around the world because it has

G. Gabriel-Caycho (✉) · M. Blas-Perez · A. Barrientos
Universidad Peruana de Ciencias Aplicadas (UPC), Lima 15023, Peru
e-mail: u201512251@upc.edu.pe

M. Blas-Perez
e-mail: u201317356@upc.edu.pe

A. Barrientos
e-mail: pcsiabar@upc.edu.pe

become a fundamental device. The emerging Smart TV model has marked a new generation of screens due to its connection with other devices such as smartphones or tablets [1]. In addition, the growth of Smart TVs has led companies to look for new proposals that replace conventional remote control and allow these devices to be handled through smartphones [2]. In this way, we propose the development of a real-time communication library capable of connecting multiple Android devices with a Smart TV and that they can exchange information quickly. Likewise, it is proposed to use the communication library developed as a basis for future development projects and in this way developers can implement and deploy their applications in a more agile way, avoiding implementation from scratch.

2 State of the Art

2.1 *Communication Protocols*

There are different investigations where the different communication protocols that are used in IoT are compared as in [3] that compare the Web performance of messaging protocols such as Message Queue Telemetry Transport (MQTT), Advanced Message Queuing Protocol (AMQP), Extensible Messaging and Presence Protocol (XMPP), and Data Distribution Service (DDS) by measuring the latency of the arrival of messages to a sensor. Also, protocols are used within an HTML5 (HyperText Markup Language) Web site to measure latency, performance, and coding. Finally, it is mentioned that the use of the MQTT protocol and the WebSocket protocol is recommended for the implementation of Web applications that use JavaScript and HTML5. In the investigation of [4], they focused on the transmission of data through the different protocols used in the world of IoT where MQTT, CoAP, AMQP, and WebSocket were considered. A comparison of the main characteristics between the protocols was made considering security, architecture, and communication protocol as the most important points.

The authors in [5] analyze different communication protocols using different scenarios, different applications, and different devices because each protocol can be used in the best way in a specific scenario. Therefore, choosing a protocol depends on the scenario and the environment in which it will be used. When applications need to update data frequently it is recommended to use the publication/subscriber protocols. The WebSocket protocol is ideal for use in applications that use Web browsers because it provides full-duplex communication asynchronously between the server and the client allowing for a continuous session until those involved finish the need for communication.

2.2 *Applications with Bidirectional Communication*

Teamwork or collaborative work requires platforms capable of supporting multiple requests from different users in real time. In [6], an algorithm for collaborative graphics editing is developed using HTML5 Canvas, WebSocket, JQuery, and Node.js. To display a graphical user interface and perform the tests of the proposed model, a Web page is developed that allows collaborative editing of multiple users in real time because each modification is reflected for all users. Collaborative work tools allow optimizing the analysis time to improve the quality of the results. In [7], a Web site called OnGraX is developed that allows changes to be made to a graphic, which can be modified by different people in real time. In addition, a chat is included for conversations between users during the session. HTML5, JavaScript, WebSocket, and Java were used to implement the prototype. For the development of the world of IoT, it is important to establish a correct architecture that meets the requirements of the environment in which it will be used.

The work done by the authors of [8] shows a Web solution, which consists of the monitoring of actions in real time. During the tests, examples are shown that allow us to see the importance of having a flexible system for changes. In addition, it is mentioned that the proposed system was developed using the MQTT protocol and is prepared to be able to relate to other systems with which it can communicate in real time for the exchange of information.

3 **Implementation of the Solution**

According to the investigation of the different communication technologies we have selected the WebSocket protocol. Our project seeks to improve the user experience by combining the use of smartphones with Smart TVs, which is why this section has been divided into two stages: implementation of the communication library and development of a retro videogame.

3.1 *Implementation of the Communication Library*

For this stage, the work environment where the development and proof of concept will be carried out was defined. The main tools that were used were the following: Visual Studio Code, WebOS TV IDE, WebOS TV CLI, Google Chrome, Chromium, and Android Studio. To perform all the necessary tests of the development, LG WebOS Smart TV, and an Android Smartphone were used.

Technology and construction. For the development of the library, JavaScript was used for the WebOS part. To obtain the private IP of the Smart TV, an interrupted version of the WebRTC protocol was implemented, which attempts to communicate

with the network devices. The IP obtained must be cut, since we only care about the first 3 octets of the IP, which serves to identify the network in which the device is located and then travel through the 255 IP's candidates to be a WebSocket server. During the IP route, a list of servers will be filled, which will be stored temporarily and will be displayed where necessary for users. In the tests that will be carried out in the next part, the list will be shown on the main screen of the applications so that users can then choose which server they want to connect to. On the other hand, for the part of the library that will be implemented in the Android device, the JAVA programming language will be used, in which the Java_WebSocket and GSON library will be used. The library in Android can be used as a client and server so the library was divided into two classes which contain all the necessary functions for each scenario in which it is required to create a server or a WebSocket client. Next, the proofs of concept of the communication library are presented, which are divided into two parts because it must be implemented in an Android application and an LG WebOS Smart TV. So, an APK application was developed that will implement the communication library and act as the WebSocket server while the IPK application will act as the client and be placed on the LG WebOS Smart TV.

Proof of concept—Android. The Android application will have two main options: Create a server and create a client that will connect to another server. The main control has five buttons to send messages to the application on the LG WebOS Smart TV (“up,” “down,” “left,” “right,” “Send a message to TV”). The “Send a message to TV” option allows you to write a text that can be displayed on the Smart TV. The buttons will act as commands to move the object to be displayed in the IPK application.

Proof of concept—WebOS. In this section, three applications developed with JavaScript, HTML, and the communication library were developed. The first application receives the commands of the mobile application to be able to move a box in a section of the screen, and the movement is slowly done by frames. The second application shows the continuous movement of a box after receiving a command from the APK application. The third application was developed to validate the behavior of Smart TV by simulating a small game. This application shows a box that must move through obstacles.

3.2 Development of a Retro Video Game

This section will show the defined architecture and the two parts of the development where a mobile application was developed for the smartphone, and the WebOS application that will be used on the Smart TV. The video game is called “Aliens On Earth” and the theme used is that of the “Prince of Persia” of Nintendo. The graphics will be of a retro style and the control of the application will be like that of a Nintendo Nes30.

3.2.1 Architecture

Logical architecture. The logical architecture diagram is shown in Fig. 1. For the part of the LG Smart TV, the WebOS operating system is shown as the main layer, then the communication library, the Phaser Framework, the JavaScript language, the HTML language, the CSS for styles are shown and finally the video game “Aliens On Earth.” On the other hand, we have the Android Smartphone part. The main layer is the Android operating system, then the communication library, the GSON library for the JSON files, the Java WebSocket library are shown and finally the video game “Aliens On Earth” is shown.

Physical architecture. The logical architecture diagram is shown in Fig. 2, which shows the three main parts that allow interaction between devices. The diagram represents the communication of devices in a local network. Our central point is the Router, which is responsible for creating the local network. Also, LG WebOS Smart TV device and Android device are shown, which will be communicated through the WebSocket protocol.

Android application. The Android application works as the server, so the WebOS application must be connected to start the video game. The main interface of the application is shown in Fig. 3, where it can be seen that a retro style is used. This logo shows the logo, the name of the videogame and the “Iniciar sala y control” button.

After pressing the “Start room and control” button, the command will be displayed to control the character that will be shown in the video game.

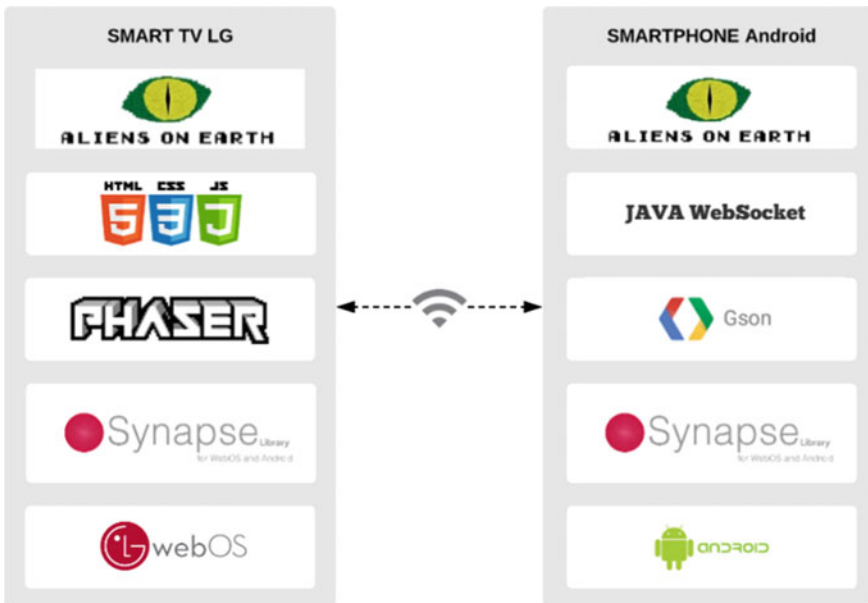


Fig. 1 Logic architecture diagram

RED LOCAL

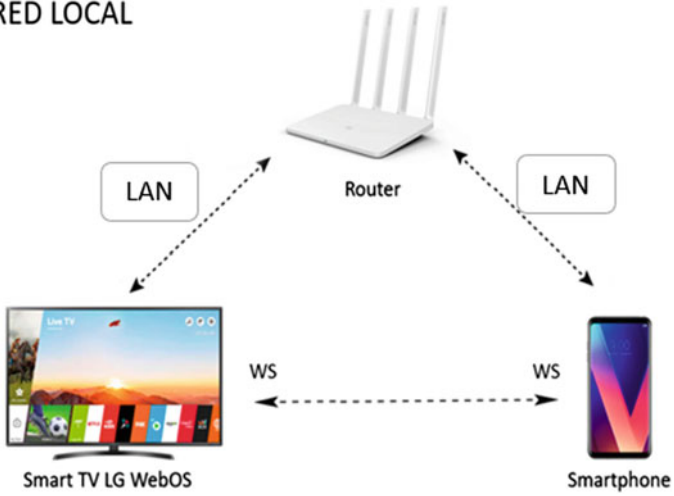


Fig. 2 Physical architecture diagram



Fig. 3 Main interface of the mobile application

WebOS application. The WebOS application works like the WebSocket client and to start the video game it is necessary to follow the steps shown in the main interface, as shown in Fig. 4. In step 1, the user must scan the QR code to be able to download the Android application from Google Play. After downloading and installing the application, the application interface will appear as shown in Fig. 3. Finally, using the Magic Remote of the LG WebOS Smart TV, you must press the “Refresh” button and the WebSocket servers available for that the user can connect. When the Smart TV manages to connect to the WebSocket server, the main videogame interface will be shown as shown in Fig. 5. In which three options will be shown. The “Play” button is responsible for starting the game and displaying Level 1. The “Instructions” button will display a menu containing the description of each button that must be used to control the character and move between levels. Finally, we have the “Exit” button, which is responsible for closing the connection to the WebSocket server and return to the main menu as shown in Fig. 5.

The video game consists of five levels as shown in Fig. 6 and the difficulty increases as it passes between each of them. In each level, there are different obstacles, enemies, potions, control points, levers, and a portal to move to the next level. The character of the video game must be controlled through the retro video game controller shown in the mobile application. To improve the game experience, a background sound and sounds that play when the character acts were added. Also, the video game can be paused when the user needs it without losing the connection to the Android device.



Fig. 4 Main interface of the WebOS application

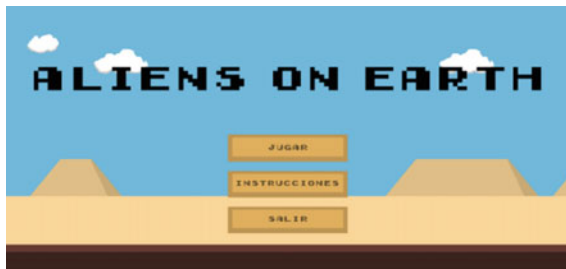


Fig. 5 Main videogame interface

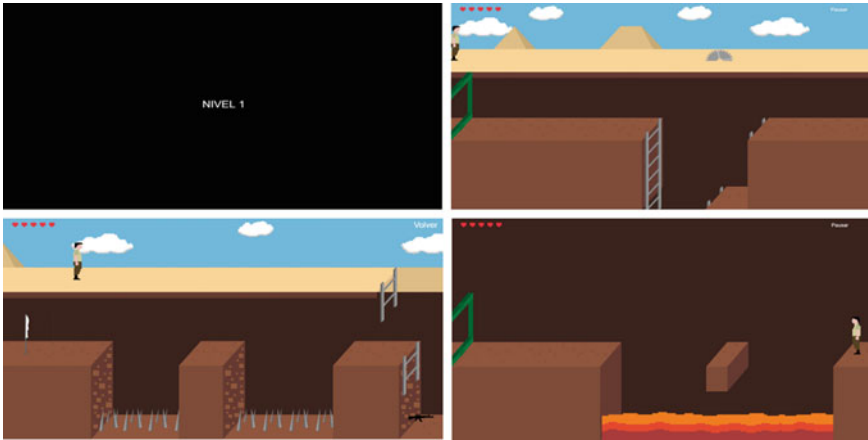


Fig. 6 Video game levels

4 Conclusions

The analysis of the different communication protocols helped us choose the WebSocket protocol, which acts correctly in the environment in which we have used it.

After carrying out the different proofs of concept we have seen that the communication library fulfilled the objective established at the beginning of the project. The communication between the devices involved worked correctly and met each of the requirements to be used in the video game developed.

5 Discussion and Future Work

Finally, we have been able to identify some improvement points that can be developed in the future projects. In the following applications that are developed with the communication library for IOS devices could be used because our research only covered Android devices. In addition, as we have seen that multiple devices can be used, applications that involve the interaction of several smartphones with a Smart TV can be developed allowing more users to interact with an application or a multiplayer video game.

References

1. Francisco, M., Jaime, C., Jhon, I., Gustavo, R., Davinia, H.: Smart TV-smartphone multiscreen interactive middleware for public displays. *Sci. World J.* **2015**, 534949 (2015)
2. Hae-Jong, J., Hwa-Young, J.: A study on the control of smart device set-top box using image based reverse mirroring technology. *Multimedia Tools Appl.* **78**(5), 5523–5534
3. Zoran, B., Jelica, P., Veljko, M.: Web performance evaluation for Internet of Things applications. *IEEE Access* **4**, 6974–6992 (2016)
4. Abdullah, O., Selvakumar M.: A brief review of messaging protocol standards for Internet of Things (IoT). *J. Cyber Secur. Mobility* **8**(1), 1–14 (2018)
5. Muneer, B., Mohammed, S., Wail, M., Dua', A.: Internet of things' business impact and its application layer protocol in embedded systems. *Int. J. Intell. Enterprise* **4**(1–2), 143–154 (2017)
6. Chunxue, W., Langfeng, L., Changwei, P., Yan, W., Naixue, X., Changhoon, L.: Design and analysis of an effective graphics collaborative editing system. *Eurasip J. Image Video Process.* (1), 50 (2019)
7. Björn, Z., Andreas, K.: OnGraX: a web-based system for the collaborative visual analysis of graphs. *J. Graph Algorithms Appl.* **21**(1), 5–27 (2017)
8. Pedro, D., Nuno, V., Luis, P.: An ideal IoT solution for real-time web monitoring. *Cluster Comput.* **20**(3), 2193–2209 (2017)