Financial stock application using websocket in real time application

Yinka-Banjo Chika, Ogundeyi Kehinde Esther
Department of Computer science, University of Lagos, Nigeria

Article Info

Article history:

Received Oct 10, 2019
Revised Nov 13, 2019
Accepted Nov 23, 2019

Keywords:

Comet
HTTP polling
Real time
Serversent event
Websocket

ABSTRACT

Real time web application involves clients which is also known as a browser getting updates from the server as they happen. The finance world is moving fast, and human brains cannot sustain processing data at that speed, so algorithms are used with regards to the limit observed in old-style real-time communication which include long polling, polling server-sent events and comets, this paper uses WebSocket technology when dealing with real time applications. The Web Socket offers improved result as compared to the conventional approaches that are considered to be good solution of providing real time information and lessens overhead acquired while communicating over the internet and offers stateful, efficient communication among web servers and clients. When there is need to track companies worth/stock using a dashboard immediately and not some seconds ago, WebSocket can be used to stream data without delays.

Corresponding Author:

Yinka-Banjo Chika,
Department of Computer Science
University of Lagos,
Akoka Rd, Yaba, Lagos, Nigeria.
Email: chikabanjo@unilag.edu.ng

1. INTRODUCTION

Real time web application involves clients which is also known as a browser getting updates from the server immediately as they occur [1]. An example of real time web application is google doc in which documents can be edited by a user and at the same time see what is been done by others simultaneously.

Hypertext Transfer Protocol (HTTP), an application layer protocol (stateless and request-response based) is the basis for the communication between servers and browsers. A request is initiated by an HTTP client, thereby establishing a Transmission Control Protocol (TCP) connection [2]. After the client's request has been received, a response is sent back by the server and thereafter the connection is terminated. Using this model, real time data cannot be sent by servers to clients. Consequently, Technologies like Ajax long polling and comet have been used to accomplish real time communication between server and client. Still, real time communication cannot be accomplished with these technologies because some of them needs plug-in to be installed on browser which put heavy burden on the server. Real time data communication was achieved with the advent of Web socket protocol and HTML 5.

WebSocket description depicts an Application Programming Interface which allow the use of web socket protocol by a web page for two-way communication alongside a remote host [3]. Thus, the HTTP solution can simulate the real time communication with high tag in price, increased network traffic and increased latency. The results to act out the full duplex connection were unscalable polling and long polling methods. WebSocket offers the incredible reduction in the quantity of latency and network traffic in communication system. The WebSocket is quicker than these old-fashioned solutions.

It is an established fact that data are received by the server when requests are made from client's side when such data are received by the server, it becomes its responsibility to accomplish the task. This has been the standard used for a long time until the advent of AJAX in 2005 which lead to the discovery of bidirectional connections between server and client. There has been a lot of increase in the consumption of data in web application, but the major drawback to the improvement of this is the traditional HTTP model of client-initiated transactions been used. To overcome this, there are different strategies used to push data to the client from the server and one of the popular strategies used is long polling which keeps HTTP connection open until data has been push from the server to the client own to the client.

The flaw in the strategies introduced is that they transmit overhead of HTTP and large cookie data and headers are usually transferred alongside an HTTP request which increases the size of the data which is to be sent thereby increasing latency.

To develop a financial stock application application that create a persistent, low latency connection that can support transactions initiated by either the client or server, WebSocket has been used to demonstrate an effectual Realtime web. WebSocket provide a persistent connection between a client and server that both parties can use to start sending data at any time. The client establishes a WebSocket connection through a process known as the WebSocket handshake. This process starts with the client sending a regular HTTP request to the server.

The Web started out as a simple document sharing service to make the lives of researchers simpler. The transition into the Web of today has been made possible by the fact that it became capable of real-time updates. The real-time web enables a website to update itself dynamically when new content is available.

The following two examples illustrate the meaning of the term real-time:

- A stock price application where several clients connect to a server. A stock application is a symbol which is used to uniquely identify company stock. This stock price server subscribes to a broker backend for stock price updates. When the stock price server receives price updates from the backend, it broadcasts these to all clients.
- A chat room application where several clients connect to a centralized server. The server listens for client messages and as soon as a client sends a message to the server, the server immediately broadcasts the message to all other connected clients. Both examples show that data is handled and distributed immediately after it is received, which is what is meant by the term real-time in this research. The first example requires unidirectional messaging, with stock prices only going server-to-client. The second example is bidirectional, requiring both server-to-client and client-to-server messaging.

Even though real-time capabilities were not considered when the Web was designed, there are several ways to use traditional HTTP in order to achieve near real-time updates, such as the techniques Long Polling and Streaming. Recently, with the new protocol WebSocket and the new HTTP-based technology Server-Sent Events, a true real-time web is possible. This research will therefore focus on WebSocket, Server-Sent Events, and HTTP Long Polling. Server-Sent Events and Long Polling are both unidirectional (server-to-client) while WebSocket is bidirectional (server-to-client and client-to-server). For simplicity's sake, this paper will refer to the real-time delivering technologies as transports. It is believed that WebSocket performs better than HTTP.

The problem with all of old-style real-time communication is that they carry the overhead of HTTP. Every time you make an HTTP request a bunch of headers; cookie data are transferred to the server. This can add up to a reasonably large amount of data that needs to be transferred, which in turn increases latency when developing a real time application, reducing latency is crucial to keeping things running smoothly. The worst part of this is that a lot of these headers and cookies are not actually needed to fulfil the client's request.

For developing a financial stock application that create a persistent, low latency connection that can support transactions initiated by either the client or server, WebSocket will be applied to illustrate efficient realtime web. WebSocket provide a persistent connection between a client and server that both parties can use to start sending data at any time. The client establishes a WebSocket connection through a process known as the WebSocket handshake. This process starts with the client sending a regular HTTP request to the server.

WebSocket Protocol, similar to Socket whose communication is based on TCP/IP is a full duplex communication protocol [4]. The major advantage of WebSocket is that information can be sent between client and server at any time. Ajax communication does not restrain WebSocket because in Ajax technology, it is required that the client initiate the request. Contemporary browser fully supports WebSocket protocol so WebSocket client can push data to the server [5].

The establishment of WebSocket connection is in this manner, the browser first sends a WebSocket connection request, in which the browser responds to the request sent by the client, the process described above is known as 'handshake'. The data transmission at this phase is based on text and use ASCII encoding and is compatible with existing HTTP/1.1 protocols [2]

WebSocket is connection oriented which implies that if a broadcast message is sent from the server, they are instantly dispersed to all client. This feature makes websocket containing only one component for both client nd server communication. [6]

The aim of this paper is to create a financial stock application using WebSocket. The detailed objectives are:
- Creating a web-based system financial stock application
- Integration of WebSocket framework system in financial stock application system

This project has focused on real-time computing from a software and web application perspective in implementing web socket protocol with a financial application system for presenting real time market data. This project is limited due to the fact that Web Socket is not fully supported by current browser, and different browser support different portion of the Web Socket specification. For this drive to make real time applications capable of running anywhere, the client side should comprise some Web Socket simulation framework such as SignalR, which provides C# library

The methodologies used for implementing the financial stock application project is WebSocket. The web socket server will also be implemented using C# Library. The code will be written using standard libraries available in both Windows and Linux for maximum portability. Third party libraries must only be used if the licensing is permissive. Software development will be conducted in an iterative way. Work will follow a basic plan and allow for changes during the implementation if additional features are required or if performance issues with the selected design surface.

2. RESEARCH METHOD

Synchronous communication system also known as real-time communication system always includes a large number of passive recipients [7].

Web clients is able to poll server-side events because of the Light comfort provided by AJAX to the HTTP communication model. Comet presented a better approach by deviating from HTTP model of Communication and permitting communication over HTTP with push-style.

With the advent of HTML 5, web socket brings about a higher development than Ajax and Comet. The condition of HTML 5 WebSocket defines a single socket full duplex (bi-directional) connection which is used to pull and push data between a client (browser) and a server, thereby delivering a better result than comet and Ajax polling [3].

2.1. Prior methods for real time data communication

HTTP can simulate real time communication environment. The HTTP long polling and HTTP polling can be considered as the good solution for delivering real time information [4]:

2.1.1. HTTP polling

A common feature of HTTP Polling is its sequence in request response messages. A request is sent by the client to the server, when the server receive the message, it returns a new message if there is a message to send otherwise an empty response is returned if there is no message available. After a small intermission say t, called as polling interval, the server is polled again by client to realize if there is a vacant new message. Examples of application which use HTTP Polling are chat rooms and games in Figure 1.

For HTTP Polling to be adequate in providing real time details, the interval of delivering the message must be known which implies that the rate of transmitting the data must be stable. Thus, the client will only be synchronized by the application developer to request for data when it is certain to be available. However, latency is accumulated when the rate of data request grows which is caused due to the constant repetition of important header details by the overhead intrinsic to HTTP polling. Due to the complexity of real-time HTTP Web applications, it has been proposed in early research that HTTP was not conceived for real time full duplex communication. Thus, the HTTP solution can simulate the real time communication with high tag in price, increased network traffic and increased latency.

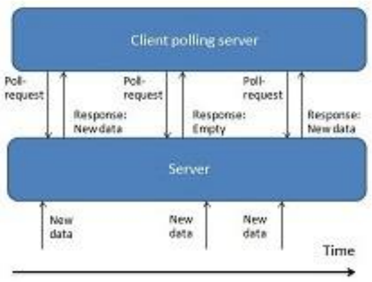


Figure 1. Client using HTTP polling

2.1.2. HTTP long polling

A flaw in polling is the large amount of futile request it sent to the server when it does not have new message for a client. Therefore, the details which are pushed form servers to client are well handled by HTTP Long polling which is a variation of polling [8]. With HTTP long polling as shown as Figure 2, when there is no new message to be sent to the client, the server does reply with an empty message but instead keep the request until a fresh message is available to be sent to the client or timeout occurs thereby decreasing the number of pointless request to the server but still HTTP Long polling did not solve the problem in traditional polling as there is still increased latency and overhead.

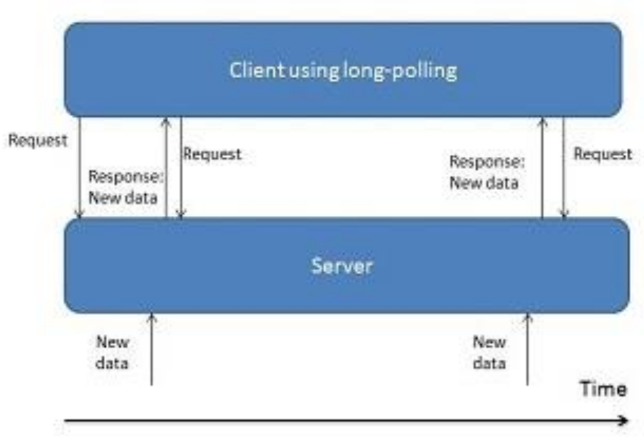
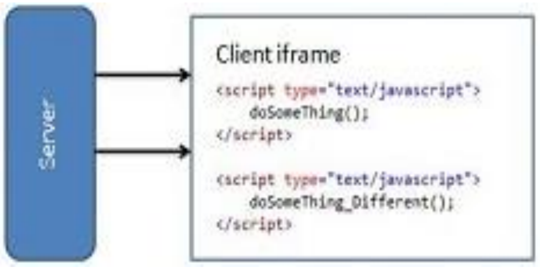


Figure 2. Client using long polling

2.1.3. HTTP streaming

HTTP Streaming which was first made known to the web by Netscape in 1992 originates in two kinds namely page streaming and service streaming. The content of the server streaming in a long-lived HTTP Connection is contained in page streaming as well as service streaming [9]. With HTTP Streaming, the server does not close the connection but keeps it open throughout a client session by running a long loop. A change in state is detected by the server script using some methods like event registration. Once a change is detected, the new data is streamed and flushed while the connection remains open. For the meantime, the new data must be displayed in the user interface while awaiting the server's response [9]. A difference between the forms of streaming is that service streaming is initialized by client's request which is known as XHR-streaming while page streaming uses the request of the initial page to stream data which provide flexibility in the duration of connection [4]. An example of HTTP Streaming implementation is Forever frame (an iframe which obtains script tags from a server in an everlasting response) which is shown in Figure 3. Codes are implemented in script tags when it is been read by a browser, therefore the data received by a client from the server are enclosed as JavaScript functions [5]. Comparison between HTTP long polling and HTTP streaming protocols as show in Table 1.



```
<script type="text/javascript">
    doSomeThing();
</script>

<script type="text/javascript">
    doSomeThing_Different();
</script>
```

Figure 3. HTTP streaming

Table 1. Comparison between HTTP long polling and HTTP streaming protocols

HTTP long polling	HTTP streaming
With HTTP long polling when there is no new message to be sent to the client.	With HTTP Streaming, the server does not close the connection but keeps it open throughout a client session by running a long loop.
The server does reply with an empty message but instead keep the request until a fresh message is available to be sent to the client or timeout occurs thereby decreasing the number of pointless requests to the server	A change in state is detected by the server script using some methods like event registration. Once a change is detected, the new data is streamed and flushed while the connection remains open

2.1.4. Comet

Comet scheme which exist due to the persistence attribute of HTTP/1.1 uses the methods practiced by long polling to accomplish real time behavior [10]. Before the advent of persistence connection, different URLS are been fetched by different TCP connection which cause congestion on the internet because of the weight being put on the server but with persistence connection the connection between the client and the browser are always open till it is closed by either one of the party sending a precise close connection message or a time/network error take place. As soon as the connection is closed, the server cannot start it back to gain a connection to the browser [5].

With persistence connection, CPU time is saved in both hosts and routers, fewer TCP connections are opened and closed and the memory use in the control blocks of the TCP Protocol can be saved in hosts.

2.1.5. Server-sent events

Server sent Event uses the text/event stream feature of HTTP/1.1 content type to send message to the browser (client). The channel in which server-sent event use to communicate to the client from the server is a one-way communication channel. However, the client has to subscribe to the channel by first connecting, then when there is a fresh information available the server post events. The connection is always open until it is closed by a proxy or the client itself. The connection can also be configured to close after a duration of time, same as the client reconnection time [3]. As shown in Figure 4, Server sent Events behaves quite like long-polling. Server-sent events can be used by developers to access APIs. Event Source interface receive access from the API thereby providing a direct JavaScript code, allowing events to be triggered by the server to the browser and updating the content on the client [10]. When an ID is set on messages sent, the client reconnects and continue where it stops after a look up is been done by the server on its ID. This feature makes server sent event robust.

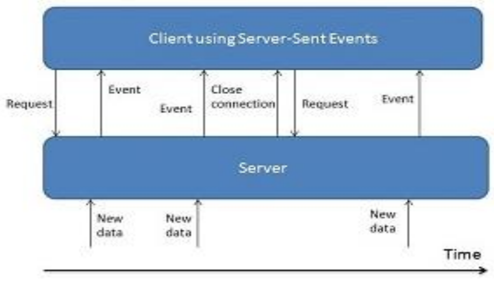


Figure 4. Client using server sent events

The model in which client-side events are transmitted to the server (i.e. a user clicks on a link to request a new page from the server) can be referred to as “client-sent events”. Server-sent events permits servers to pass information to the client as it becomes accessible, without client polling. To use Server Sent Events, an application implements an Event Source API in the browser and pushes data from the server with Server Sent Event’s event stream data format.

2.2. Drawbacks of real time web with HTTP

- **Overhead**

Achieving real time with HTTP is a naïve approach because HTTP headers consume excess data which is in most case not useful for real time application while WebSocket protocol is intended for it because it makes available a bidirectional, full-duplex communication channel for use which is carry out with a single socket leading to a scalable real time Web application.

- **Unidirectional**

HTTP is unidirectional so when using real time application with HTTP, several TCP connection is used to achieve a replicated bi-directional communication [11], even with server sent event, a connection will still be needed to push events from the server to the client and also to send message to the server but WebSocket protocol which bring improvement to real time applications is purposely for the aim of bi-directional communication which real time application is all about.

2.3. Web socket

Communication overhead can be intensified by the use of continuous polling in an application. The WebSocket protocol allows for a robust real time web application because it provides a full-duplex bi-directional communication channel which is done with a single socket [7].

2.3.1. Web socket architecture

Figure 5 depicts the architecture of WebSocket. WebSocket protocol consist of two categories; the first part is the Handshake which contains the handshake response from the server and the message from the client, the second of the category is data transfer. The Web socket also provide a socket that is innate to the browser which eliminate most of the problems with using HTTP thereby building a scalable real time application.

For a WebSocket connection to be established, both the server and client has to be upgraded from HTTP protocol to web socket protocol at the early stage of handshake. When the connection has been established, WebSocket data, text and binary frame can be sent at the same time in any direction in full duplex mode. Replicating the WebSocket bi-directional communication with HTTP comes with high price to pay in slow CPU performance, increased latency and network traffic and can be very difficult and is error prone because of the complexity associated with real time application [12]. Another benefit of WebSocket is that it has the ability to transverse proxy and firewall which is an area difficult for many applications. WebSocket pass by proxy by spontaneously setting up a tunnel when it detects the presence of a proxy server.

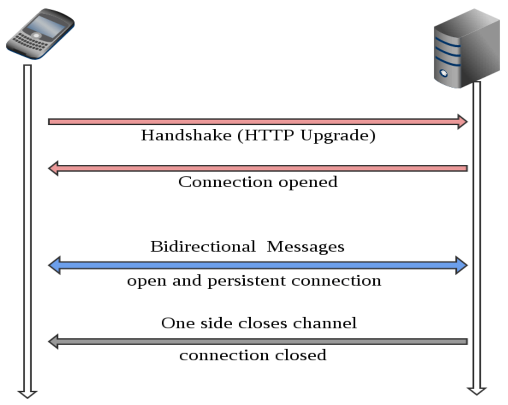


Figure 5. Web socket architecture

3. RESULTS AND DISCUSSION

3.1. Features of the system

The user interface of the application includes the financial application dashboard, this page contains an easy access link to some functional page such as the company holding module, the financial stock application list module among others. The list of functionalities and features that are used in this application are discussed below.

3.1.1. Financial stock application dashboard

This financial stock application dashboard contains the short cut menu for most of the module in the application. The modules that exists in this dashboard are listed and explained below:

- There is the dashboard icon which contains hyperlink for the dashboard and thus redirects back.
- The dashboard also contains an icon which redirect to the modify Stock application module when clicked on.

- There is also an icon for My holding page which list the company stock.
- The index page which contains all the company and their stock icon is also included in the dashboard.
- The dashboard has the side bar which contain the following action links:
- Dashboard link
- My holdings
- Stock Application List
- Modify stock
- There is the Search input field located in the top part of the side bar which allows searches. When a word is inputted in the input field for the purpose of searching, the result should be able to be filtered according to description of the user search.

3.1.2. Modify stock module

In this module, the previous stock of a company can be updated, the page contains the company name, the current stock price, the company initial opening price, the symbol which is used in identifying the company, the last previous price of the company, the net change and the percentage change.

The company details and necessary information are fetching when the user log in with the credentials in to a form input field that has matching fields with the database. To update the company's new stock, the user enters the new price in the stock field section, the net change and percentage change are updated immediately.

The module shows the stock and company dependency and users should be able to manage their stock effectively. After the users enter the price, the new price is immediately captured by the PrevClose (Previous close) field and calculations are made with both values.

Each field (Net change and % change) are gotten from the calculations of the new stock value and the PrevClose value. The Net change and % change shows a monitor as to the positive or negative change of the company stock, color red which indicate low stock and green denotes positive. When the user saves the new values, the financial stock index page is updated immediately.

3.2. Technology overview

Microsoft's .NET framework (Microsoft, 2017) was used to implement the financial stock system and the choice of language was C#. The platform.NET framework main target is Microsoft windows but other platform also includes Linux, Mono which is free and an open source implementation of .NET framework (Xamarin, 2013), with Mono, different simulator component can be processed on different platforms. For this project, the simulator used between the financial stock simulator and the agents is SignalR WebSocket implementation (SignalR, 2017) and this foster the core communication.

- **Microsoft .NET framework**

Microsoft .NET framework is known as a software that aids development and thereby used its environment, many languages such as C3, C++ and visual basic uses it because it supports many languages. It is divided into run time component and compile time components.

- **WebSocket**

WebSocket protocol consist of two categories; the first part is the Handshake which contains the handshake response from the server and the message from the client, the second of the category is data transfer. The Web socket also provide a socket that is innate to the browser which eliminate most of the problems with using HTTP thereby building a scalable real time application.

- **SignalR**

For Microsoft web/ASP.NET technology, SignalR is used in carrying out WebSocket operations. An OWIN Project which is known as a container is being implemented on a webserver which is essential when using SignalR. The communication pattern used in WebSocket has already been described in chapter two; The browser first sends a WebSocket connection request, in which the browser responds to the request sent by the client, the process described above is known as 'handshake'. The data transmission at this phase is based on text and use ASCII encoding and is compatible with existing HTTP/1.1 protocols.

3.3. RESULTS

3.3.1. Financial stock application dashboard

The Stock Application dashboard shows a brief layout of the project and also include links to some module and was not developed to have any functionality. There are four icons/columns in the Financial Stock Application dashboard module as shown in Figure 6, the first column shows a hyperlink which links to the financial stock application page where all the company and their stock can be seen. The second column contain a link to modify a company previous application which includes the unit price, quantity, previous close, net change percentage change.

The third icon contain a hyperlink to the details module which shows the details of a registered company and its stock while the fourth column shows an icon which is a link to the view detail module.

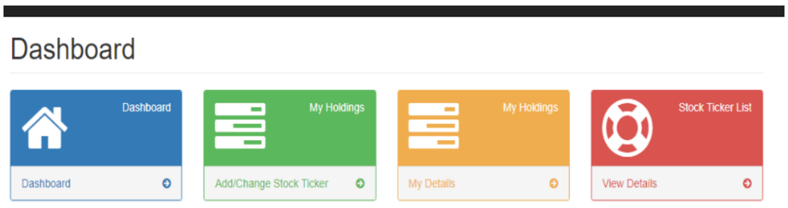


Figure 6. Dashboard

3.3.2. Login page

Figure 7 shows the login page where the user credentials are entered to access the application.

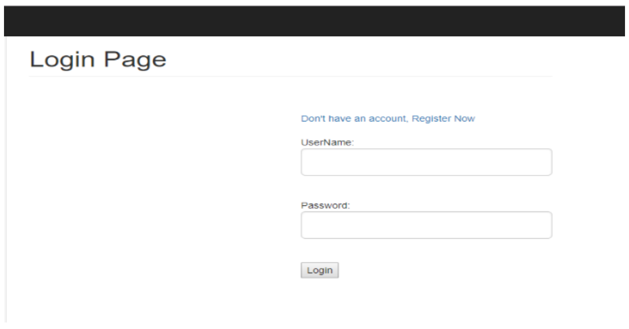


Figure 7. Login page

3.3.3. Register company application module

This module as shown in Figure 8 contains the registration of users, there are three roles contained in this project, which are listed and explained below:

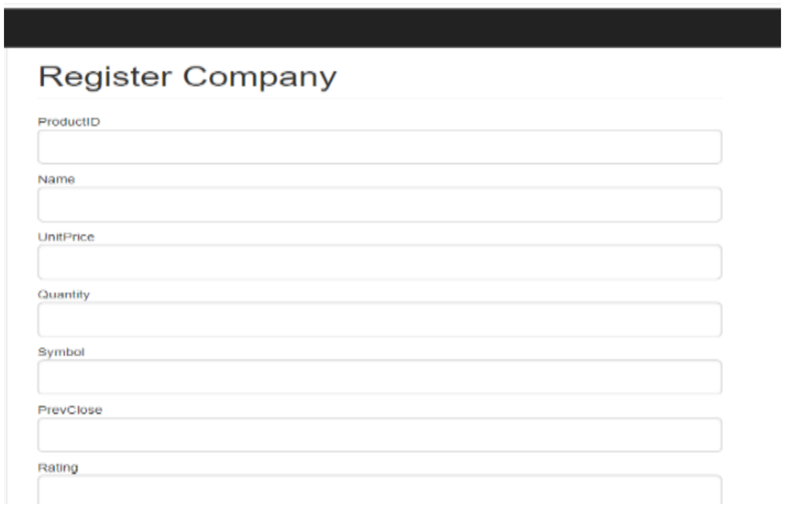
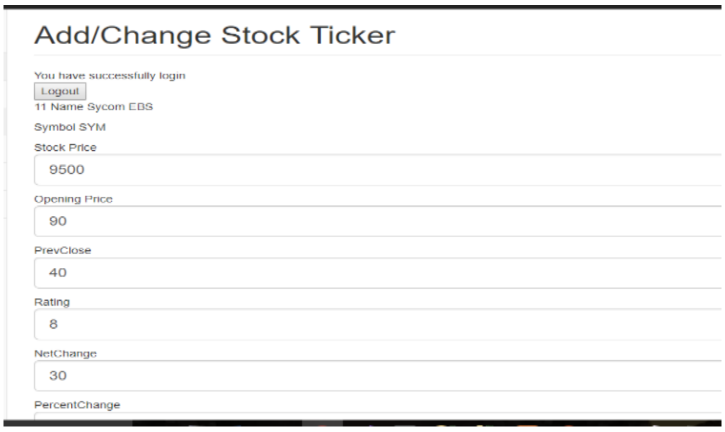


Figure 8. Register company

3.3.4. My stock module

As for the ‘My Stock dashboard, it consists of the modify Stock module and detail module. So, when a company changes its stock on My holdings module which is been displayed in Figure 9, it reflects immediately in the stock application list module without the users having to refresh their pages. This Module is arranged following the presumed user work flow. Right beside the Application symbol, there is the NetChange and PercentageChange column reflecting the change in value and in percentage in the stock price of the company.


Add/Change Stock Ticker

You have successfully login
Logout
11 Name Sycom EBS
Symbol SYM
Stock Price
9500
Opening Price
90
PrevClose
40
Rating
8
NetChange
30
PercentChange

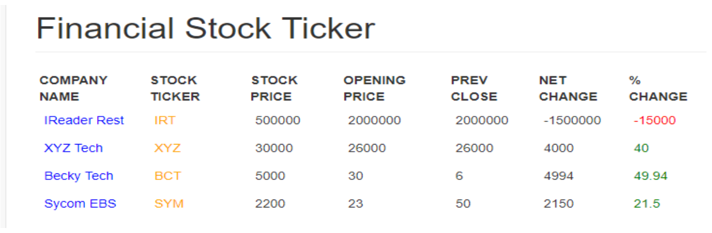
Figure 9. Add /change stock application

3.3.5. Stock application list module

The stock application list shows the latest stock in dynamic manner, Figure 10 illustrates the stock price server waiting for available socket hereafter Information of the registered company’s stock are gotten in real time. This module contains all company and their stock that has been set live by the admin. This module also has the highest functionality of the system and it is in this module, WebSocket real time feature is applied. This stock price server subscribes to a broker backend for stock price updates. When the stock price server receives price updates from the backend, it broadcasts these to all clients. The fields shown in Figure 11 are included in the stock application list.

Waiting for available socket...

Figure 10. Waiting for available Socket


Financial Stock Ticker

COMPANY NAME	STOCK TICKER	STOCK PRICE	OPENING PRICE	PREV CLOSE	NET CHANGE	% CHANGE
IReader Rest	IRT	500000	2000000	2000000	-1500000	-15000
XYZ Tech	XYZ	30000	26000	26000	4000	40
Becky Tech	BCT	5000	30	6	4994	49.94
Sycom EBS	SYM	2200	23	50	2150	21.5

Figure 11. Stock application list

a) Company name

The company field is an input field that accept value from the user in text format, on the creation of the company, the user would be allowed to input the name with the given field provided but on updating the stock the company input field is not made available since it has been previously entered but read to user in the module, the company name field on the database is stored as a varchar character which allow larger length to accommodate a possibility of a long user name thereby allowing for better user experience.

b) Symbol

This input field contains the symbol the user chooses as the application for its company, on registration of the company, the input field is also made blank for the user to put in its application choice but on log in, the input field is not editable but displayed on the user page, the property of the symbol on the database is also of type varchar to allow for a larger field.

c) Stock price

This field contains the stock price of the company, the field property on the database is of type varchar. On registering the company is allowed to input the price of the stock, the field is also made editable on updating of the stock price.

d) Previous close

The previous close field is an input field that contains the previous price the stock of a company was sold, on registering the field is made in-editable as the stock price will be used as the previous close price of the company, on updating of stock, the previous close field is also made in-editable, the property used on the database for this field is varchar.

e) Net change

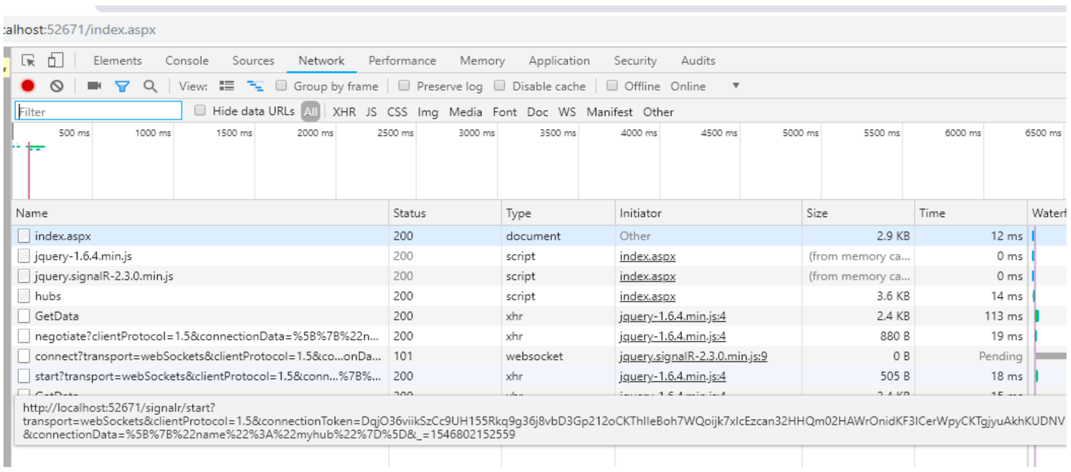
The NetChange field on the database is of property varchar, this is the calculation of the company current stock price and the previous close, on registering, this field is made in-editable as the calculation is made by system based on the input field of the current stock price and the previous close, same method is applied on the update stock module.

f) Percentage change

This is the calculation in percentage of the change in the previous price and current stock price. This is stored in varchar in the database and also made in-editable and calculated by the system both on registering of a new company and on updating of the stock.

3.4. Network test

Figure 12 shows the network test displayed in the client side, it shows that WebSocket protocol was used at the transport layer of the Stock application application.


:alhost:52671/index.aspx
Elements Console Sources Network Performance Memory Application Security Audits
View: Group by frame Preserve log Disable cache Offline Online
Filter Hide data URLs All XHR JS CSS Img Media Font Doc WS Manifest Other
500 ms 1000 ms 1500 ms 2000 ms 2500 ms 3000 ms 3500 ms 4000 ms 4500 ms 5000 ms 5500 ms 6000 ms 6500 ms

Name	Status	Type	Initiator	Size	Time
index.aspx	200	document	Other	2.9 KB	12 ms
jquery-1.6.4.min.js	200	script	index.aspx	(from memory ca...	0 ms
jquery.signalR-2.3.0.min.js	200	script	index.aspx	(from memory ca...	0 ms
hubs	200	script	index.aspx	3.6 KB	14 ms
GetData	200	xhr	jquery-1.6.4.min.js:4	2.4 KB	113 ms
negotiate?clientProtocol=1.5&connectionData=%5B%7B%22n...	200	xhr	jquery-1.6.4.min.js:4	880 B	19 ms
connect?transport=webSockets&clientProtocol=1.5&co...onDa...	101	websocket	jquery.signalR-2.3.0.min.js:9	0 B	Pending
start?transport=webSockets&clientProtocol=1.5&conn...%7B%...	200	xhr	jquery-1.6.4.min.js:4	505 B	18 ms

```
http://localhost:52671/signalr/start?
transport=webSockets&clientProtocol=1.5&connectionToken=DqjO36viikSzCc9UH155Rkq9g36j8vbD3Gp212oCKThlleBoh7WQoijk7xlcEzcan32HHQm02HAWrOnidKF3lCerWpyCKTgjyuAkhKUDNV
&connectionData=%5B%7B%22name%22%3A%22myhub%22%7D%5D&_=1546802152559
```

Figure 12. Network test

3.5. Comparative result analysis of websocket with other real time methods

3.5.1. Less overhead

An issue associated with HTTP methods is the huge volume of header data accumulated when sending mesages but due to the fact that Websocket is stateful, messags can be sent in small sizes. In the financial stock, Updating the application price is denoted with a 58 bytes JSON object which has seven attributres (company name, stock ticker, stock price, opening price, prev close, net change, % change), The below shows the difference with other methos are compared with WebSocket

- HTTP Long Polling

Apart from the 58 Byte that will be sent, an addition of 221 byte is used by HTTP headers, which sums up to 279 byte per updating the application stock price which is four times the message that would be sent.

For HTTP Polling, to update the application stock price requires 542 bytes: A total of Response body and headers (279) and request header (263).

- Server-Sent Events

Server-Sent Event is connection orient which means HTTP-like request is not required once a connection has been made resulting in a less wasted header. The message can be in the below format:

id: 1
data: {“companyname”:”XYZ Company“,”stockticker”:”XYZ”,“ stockprice”:“24.45”,” openingprice”:”20.90”,“ prevclose”:“20”,” netchange”:”0.1”,” %change”:”10”}

With Server-sent, only 74 byte is used with 16 bytes as the header data which is an improvement over HTTP.

- WebSocket

With the 58 bytes long message that will be sent, WebSocket requires only 3 bytes which means with WebSocket, the application price update requires only 61 bytes.

This shows that WebSocket lessens complexity when developing a real time application by eliminating overhead unlike polling and HTTP long polling which HTTP headers consume excess data which is in most case not useful for real time application

3.5.2. CPU load during broadcast

In Figure 13, Long Polling attained maximum CPU Utilization with only 100 clients during broadcast and at point, the server is already stressed but WebSocket and server-sent events reached maximum CPU utilization at client 200 and 150. This result shows that Websocket is a better choice when compared with other methods

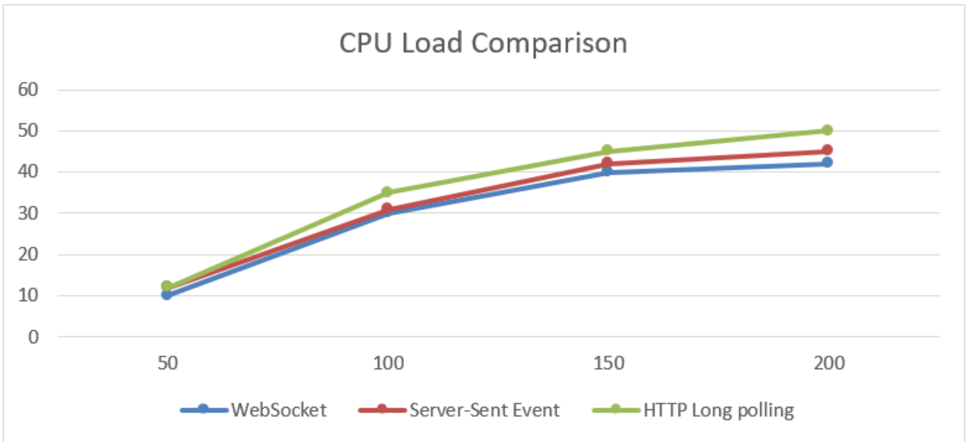


Figure 13. CPU load during broadcast

3.5.3. Memory footprint

Figure 14 shows that HTTP Long polling has the highest memory consumption, This is connected to the CPU utilization in Figure 13, as long as the CPU utilization is high, memory usage rises linearly as the number of client increases. The result still shows that WebSocket is extraordinary.

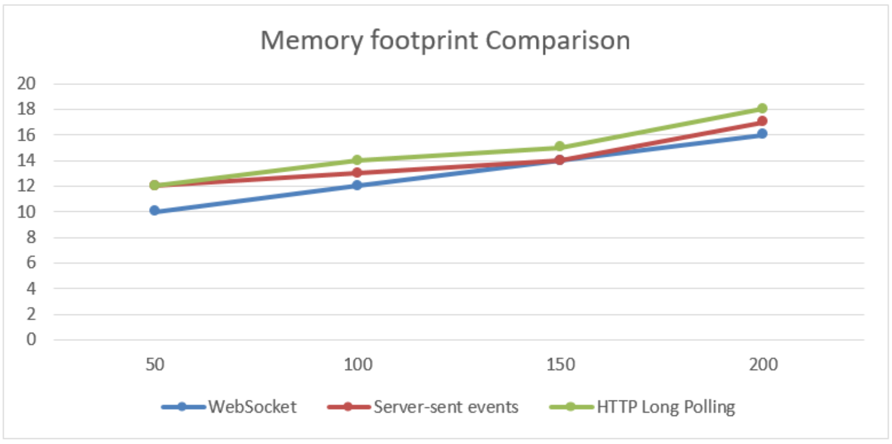


Figure 14. Memory footprint

3.5.4. Response time during broadcast

When CPU utilization has been reached for HTTP polling, the response time increases rapidly which constantly increases stress on the server. The Server- sent response time is is still very low as compared to HTTP Polling but in Figure 15, it shows that the response for Websocket from client 1-150 stays very low. Real time communication with WebSocket becomes proficient as resources such as the CPU and bandwidth are well utilized which in return improves performance unlike server-sent event which consumes more CPU Power and Bandwidth.

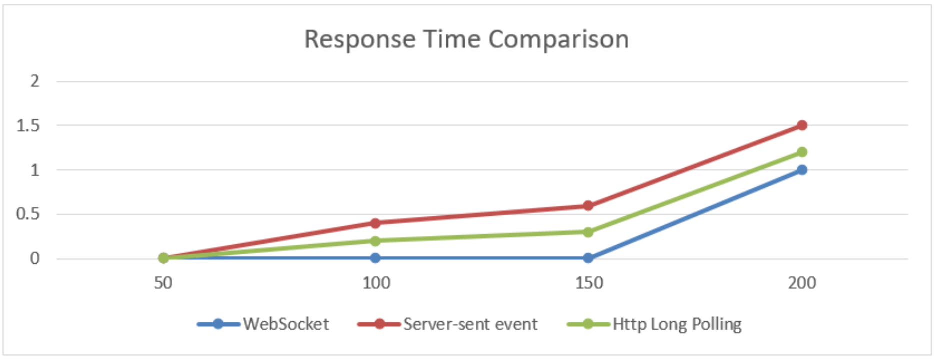


Figure 15. Response time during broadcast

4. CONCLUSION

The values of real time application using WebSocket was carried out and explained using a financial stock application. This project showed that access to real time updates are advantaged to company and users. Firstly, the company stock is showed immediately as they registered and also as the stock is updated, this allow company to have their stocks easily accessible to the users. Secondly, users to have up-to-date information about all registered company stocks such as the recent price, the previous stock price, the opening stock price of the company, the Net price and the percentage price.

Even though HTTP has a higher percentage in being used for real time applications, it still cannot achieve full duplex communication which WebSocket offers, because it causes heavy load on the server and increased network overhead. Thus, this research provides a detail description of various methods that are used for real time data communication, discussed various issues involved in designing the application using traditional methods like polling and long polling. Also, it gives the comparative analysis of methods like polling, long polling with the WebSocket while considering various performance measurement parameters like network overhead and latency. WebSocket protocol offers the tremendous reduction in network overhead, efficient low-latency and high-throughput transport and latency when real time data communication is concerned. If you want low-latency, high-throughput messaging back to the server, WebSocket can do it, Super easy API and now supported in all modern browsers.

There are still some aspects that still need to be added in this research to improve the functionality of the stock application project using WebSocket such as implementing on a mobile application. One of the major concerns of WebSocket is that some browser is not compatible with the protocol but there has been improvement as modern browser are being built to support recent technologies such as WebSocket. An important functionality that could be added to this research to improve its performance is the ability to allow users buy stock made available from the company applications, which should also be done in Realtime using WebSocket, the choice of users might be incited through the net change and percentage, this would also be beneficial to the company and also the user. It would be beneficial to the company because there would be more profitability as client's patronage would increase. It would be beneficial to the user because best decisions would be taken before investing in company. Other functionality that could be implemented in this project work is the inclusion of a rating service which allows the user to rate their performance while using the company's stock. In order to get good rating, company will tend to improve their services, this functionality will also provide insight on choice of company stock to interested clients.

More functionality that can be implemented is the addition of an indicator which should also be implemented using WebSocket and would standardizes and improves real time communication, the indicator should work as marker that increases or decreases as the Net change and percentage change changes. This would allow more statistics in the calculations of each company stock.

REFERENCES

[1] Kristian J., "Real time web applications, comparing frameworks and transport mechanisms," *UIO Department of Informatics*, vol. 1(1), p. 15, 2014.

[2] Xue L. and Liu Z.,. "Network real-time communication based on websocket," *Computer & Digital Engineering*, vol. 1(1), p. 5, 2014.

[3] Lerner R., "At the forge: Communication in HTML5," *Linux Journal*, vol. 11(7), p. 2, 2011.

[4] Fu-Hau H., Chia-Hao L., Cheng-Yu T., Wei-Tai C., Yan-Ling H. and KaiWei C., "TRAP: A three-way handshake server for TCP connection establishment," *Applied science journal*, pp. 3-4, 2016.

[5] Bozdag E., "Push solutions for AJAX software," *Engineering Research Group Department of Software Technology Faculty EEMCS, Delft University of Technology*, pp. 22, 2011.

[6] I. Fette and A. Melnikov, "The websocket protocol," 2011. https://tools.ietf.org/html/rfc6455

[7] Anton D., Yang Y. A., Bajcsy R. And Kurillo, "Platform for augmented telemedicine for real time remote medical consultation," *In Proceedings of the International Conference on Multimedia Modeling*, pp. 77-89, 2017.

[8] Pan R., Zhao H.,Wang J., Liu D. and P. Cai, "The design and implement of TCP/IP protocol cluster on AVR single chip," *School of Information Science and Engineering, Northeastern University*, pp. 763-764, 2010.

[9] Pereira R. and Pereira E.G., "Dynamic adaptive streaming over http and progressive download: Comparative considerations," *In Proceedings of the IEEE International Conference on Advanced Information Networking and Applications Workshops*, pp. 95-99, 2014.

[10] Puranik G., Feiock C and H. Hill, "Real-time monitoring using AJAX and websocket," *20th IEEE International Conference and Workshops on Engineering of Computer Based Systems (ECBS)*, pp. 7, 2013.

[11] Pimentel V. and Nickerson G., "Communicating and displaying real-time data with WebSocket," *IEEE Internet Computing*, vol. 16(4), p. 6, 2012.

[12] Rakhunde S., "Using websocket for real time data communication in full duplex network," *IOSR Journal of Computer Science Conference (IOSR-JCE)*, e-ISSN: 2278-0661, pp. 17-18, 2014