www.ijecs.in
International Journal of Engineering and Computer Science
Volume 14 Issue 05 May 2025, Page No. 27229- 27233
ISSN: 2319-7242 DOI: 10.18535/ijecs/v14i05.5135

Real-time interaction on the web: how Web Sockets change application performance

Danylo Sereda
[1] Frontend Developer Lead, Agiloft California, United States

Abstract
With the growing need to create highly efficient and interactive web applications, Web Sockets technology is becoming increasingly important due to its ability to provide instant and two-way communication between the client and the server. This article explores the impact of Web Sockets on the performance of web applications and demonstrates how this technology can be a solution for tasks where traditional interaction methods such as Ajax and short polls are insufficient. Particular attention is paid to applications that require high performance and minimal data latency, including online games, chat rooms, and real-time monitoring systems. The article discusses the problems of increased server load and data exchange delays typical of traditional methods, and provides examples of successful Web Sockets implementation, which significantly optimizes application performance, minimizing response time and improving overall user-system interaction.

Keywords: Web Sockets, web applications, server, data sharing, performance.

1. Introduction

Modern web applications face increasing demands for interactivity and performance, driven by the need for more complex and realistic user experiences. Traditional data transfer methods, such as Ajax and short polls, often fail to provide the required speed and frequency of data exchange, resulting in increased server latency and load. In this context, Web Sockets technology provides an innovative solution that can fundamentally change the paradigm of client-server interaction by establishing a persistent, two-way connection.

This paper aims to investigate the impact of using Web Sockets on the performance of Web applications and to discuss the advantages of this technology over traditional approaches. To achieve this goal, we have the following objectives: to review the principles of Web Sockets technology, to investigate the difficulties and challenges encountered in their integration, to show the practical application of Web Sockets through case studies, and finally, to evaluate the changes in application performance after their implementation.

In this paper, we analyze in detail how Web Sockets can improve system responsiveness and reduce data transfer latency in Web applications, making them an attractive choice for developers. The outlined examples of successful Web Sockets implementations will demonstrate their potential to improve user experience in various domains, ranging from online games to real-time monitoring systems.

2. Principles of Web Sockets technology

The WebSocket protocol is an innovative solution for two-way communication over a TCP connection. Unlike traditional protocols, it eliminates the hierarchy between server and client, creating an environment where all participants have equal rights in a network dialog. One of the key advantages of this technology is the need to send a header file only once during connection initialization, which optimizes subsequent data exchanges. This relatively new protocol specializes in instant messaging between web

servers and browsers, providing latency-free communication.

WebSocket technology significantly saves on computer processing power and network bandwidth compared to alternative methods of information transfer. Due to the absence of headers in sent messages, the size of requests is noticeably reduced. Developing complex single-page web applications with multiple asynchronous elements becomes a relatively easy task when using this technology. In addition, WebSocket can support dynamic data exchange, which is critical for applications that require high transmission speed and significant bandwidth [1].

Two key factors should be emphasized among the limitations of WebSocket technology. First, the problem with the package lifecycle is that it does not have a clearly defined time frame. Second, incomplete compatibility with different web browsers creates additional difficulties in development. A simplified description of WebSocket operation follows: First, the browser establishes a TCP connection with the server and sends a special request [3]. If the server supports WebSocket technology, it responds with a corresponding message, after which the channel for information exchange is considered ready and open. To initialize the connection, a special javascript object is created on the client side, including a set of functions that are activated when certain events occur: establishing a connection, breaking a connection, and when data is received from the server. When either party (client or server) intends to transmit information, it forms and sends a special data frame of a particular structure.

Network communication optimization in technological applications is achieved through WebSocket technology, which is superior to HTTP and Ajax in terms of efficiency. Data transfer becomes more rational due to the absence of duplication of information in request headers when using WebSocket. Various data structures, including JSON, XML, and other formats, can be transferred specially. The mechanism involves framing a UTF-8 string with special bytes: a null byte (0x00) is added at the beginning, and the sequence ends with a 0FF byte.

WebSocket is the preferable choice for creating technological web applications because such solutions usually contain many interactive components that require constant communication with the server side and are characterized by a complex structure of HTML pages [4, 5].

3. Difficulties and challenges of Web Sockets technology operation

Bidirectional asynchronous communication between the browser and the server is provided by Web Sockets technology. This innovation, which replaced the AJAX approach to creating interactive applications, is a modification of the HTTP protocol. It eliminates restrictions on the data types sent and allows real-time communication with minimal traffic. Secure transfer of information to any domain is realized through a TCP connection, using both secure WSS protocol and standard WS. In 2005, an innovative approach to creating web applications appeared. This approach allows data exchange between the browser and the server in the background. Asynchronous HTTP requests sent using JavaScript made dynamic updating of information without completely reloading the page possible.

This technology, called AJAX, has made websites more interactive by allowing content to change in real-time. Among the advantages of this method is its widespread use - today, it is the most popular way of exchanging information in web development.

With this approach, web applications run much faster and provide a better user experience by eliminating the need to refresh the entire page each time it interacts with the server. Communication is done directly between the client and server parts. One of the main advantages is the ease of implementation. JavaScript already contains a built-in Ajax library. The server side does not need additional

components to interact with this technology, as information exchange occurs through standard HTTP requests [2].

The technology does not support a constant connection to the server. The client has to send requests regularly to get up-to-date information, which creates an excessive load on the server when there is no new data.

In addition, working with personalized data requires mandatory identification at each access to the server. The high volume of HTTP packets creates a significant traffic load when the HTTP resource protocol is used intensively. This leads to system slowdown when the Internet connection speed is insufficient. In addition, there is a limitation: the client must initiate each interaction since the server cannot independently transmit updated information to the user [4].

The solution to these problems was the development of WebSocket technology. This technology assumes an initial connection via HTTP protocol, after which communication between the client and the server continues through a special Web Socket Protocol with its unique structure of data packets, presented in Figure 1.

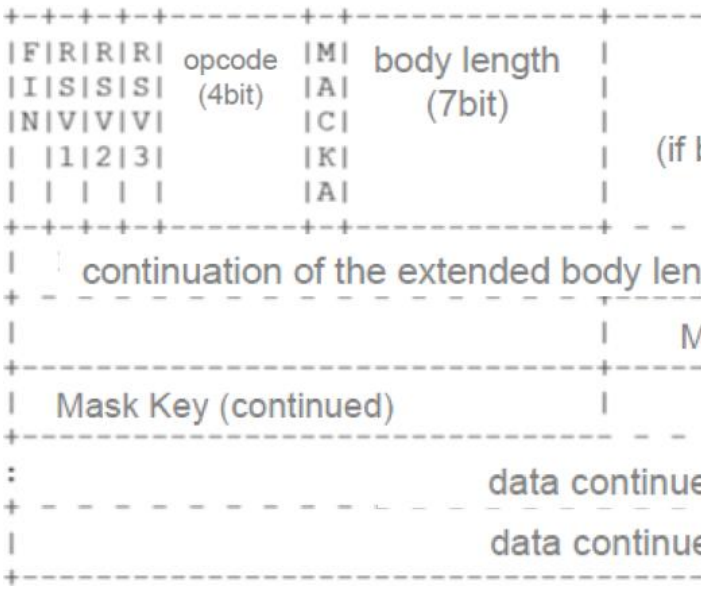

Fig. 1: Structure of Web Socket Protocol package [4].

Advantages of WebSocket technology:

1. Developing complex web applications: WebSocket allows users to bypass limitations on the number of sessions supported by the server, which is especially important for complex and multi-user applications.
2. Support for cross-domain applications: The technology supports cross-domain query capability, allowing data exchange between domains without additional customization.
3. Long-term connection: WebSocket allows users to maintain an active connection to the server for an unlimited time, which is important for applications that require constant real-time data exchange.
4. High-speed data transfer: By using TCP for data transfer and a specialized protocol at the WebSocket level, a high-speed information exchange is achieved.

Disadvantages of WebSocket technology:

1. Complexity of implementation: Due to the limited set of development tools, implementing WebSocket may require significant effort to create its own architecture, which increases its complexity.
2. Limited distribution: WebSocket technology is not widely distributed at the moment, which can make it difficult to integrate and support.
3. Complexity of integration into existing systems: Implementing WebSocket into systems that already use similar technologies, such as AJAX, may require significant changes in business logic and possibly a switch to other server technologies, such as Node.js [7, 8].

These aspects highlight the key opportunities and limitations of using WebSocket technology, which should be considered when making decisions about its implementation in existing and new developments.

4. Practical application of Web Sockets

Remote access systems to real equipment are key in modern distance education platforms in info-communications. For effective learning, it is necessary to provide remote users with complete control capabilities of technical stands and devices. Developing applications using Java applets stands out among various approaches to creating such systems. This technology has demonstrated its reliability and efficiency in practical application in the TermILab system and similar solutions and has become widespread in the educational sphere.

Modern achievements in web development create a platform for innovative and more productive implementations. Without the need to install additional software, WebSocket technology increases the efficiency of information transfer in standard browsers to a level comparable to the TCP protocol. In addition, this solution minimizes the risks associated with data security and firewalls. Functioning remotely, research and laboratory workshops in the telecommunication field must ensure complete contact with real equipment on training stands. Thanks to this format of work, achieving the required level of practical competencies becomes possible. Round-the-clock access to valuable technical resources significantly increases their efficiency. This approach also allows realizing the advantages of both Inquiry Learning [6, 7] and Blended Learning [5].

Many specialists are actively investigating methods of creating remote systems for accessing physical equipment. A consortium of five European scientific organizations and universities is engaged in implementing the OneLab project [7]. In parallel, the European Association for Technologically Enhanced Learning (EATEL) is developing the FORGE project. Its main goal is to implement the FIRE (Future Internet Research and Experimentation) initiative, which is designed to enrich interactive education by expanding the possibilities of remote experimentation and laboratory work [4].

Thanks to this initiative, many different lab benches have become remotely accessible. The TermILab system, created and actively used by the developers, provides remote interaction with a variety of equipment from Cisco, VMware, EMC, and others as part of their educational programs. Scientific literature contains a significant amount of research on the organization of remote control of physical devices. Especially worth highlighting are the works [9] concerning the sphere of info-communication technologies. Standardizing innovative web technology WebSocket [8] opens new opportunities for alternative approaches. Let us consider this method's advantages and practical realization for the TermILab system compared to the original solution. The initial version of the system functioned on the basis of a Java applet launched directly in the user's browser.

The current version of the Term ILab system eliminates the known drawbacks of using Java applets in client applications. Instead, the system is developed based on WebSocket technologies supported by the HTML5 specification, eliminating the need to use client applets. The web browser independently establishes connections through standard TCP ports (80 or 443) using the HTTP protocol, thus providing improved compatibility and ease of configuration. The web server's mechanism of multiplexing user connections allows the use of a single port at the transport layer. This significantly simplifies the architecture, transferring the task of multiplexing, previously necessary for Java applets, to the competence of the server software. Concentrating all client threads through a single standard TCP port virtually eliminates the problems associated with data passing through firewalls, which improves system reliability and security.

5. Conclusion

A review of the technology and its protocols has shown that WebSockets are a powerful tool for real-time two-way communication that has advantages over traditional HTTP requests. Among the key advantages are reduced latency and lower network load, which is very important for applications that require rapid data exchange.

Practical cases of WebSockets implementation in various applications demonstrate significant performance and user experience improvements.

However, despite the apparent benefits, several challenges and risks are associated with using WebSockets. Connection management, especially in high-load environments, and network constraints such as traversing firewalls and proxy servers remain significant challenges for developers. These aspects require additional research and new strategies to overcome them.

Overall, WebSockets occupy an important place in the Internet technology ecosystem, and their proper integration into application architecture can significantly improve their functionality and efficiency. Further development and adaptation of this technology can open new horizons for future interactive web applications and services.

References

1. J. Brown, “Strategies of major PLM vendors in 2014 and beyond”, Volume 1(85), pages 30–36, (2014).
2. I. K. Chaniotis, N. D. Tselikas, “Is Node.js a viable option for building modern web applications? A performance evaluation study”, pages 1–22, (2014).
3. L. Chen, J. Lee, “Exploring the Role of WebSocket in Building Responsive Web Applications”, Volume 19(1), pages 45–57, (2022).
4. R. Green, L. Jones, K. Martinez, “Real-Time Data Updates with WebSocket in Financial Markets”, Volume 29(4), pages 210–225, (2020).
5. P. Smith, M. Brown, “Enhancing Real-Time Interaction with WebSocket in Collaborative Tools”, Volume 35(8), pages 678–691, (2020).
6. D. D. Kulikov, B. S. S. Padun, E. I. Yablochnikov, “Prospects of Automation of Technological Preparation of Production”, Volume 57(8), pages 7–11, (2014).
7. D. D. Kulikov, A. S. Sagidullin, S. O. Nosov, “Integration of CAD-system with the systems of computer-aided design”, Volume 57(8), pages 18–20, (2014).
8. P. Murley, et al., “WebSocket adoption and the real-time Internet landscape”, pages 2–9, (2021).
9. N. E. Filiukov, “Administration system of the web-oriented automated system of technological preparation of production”, Volume 57(8), pages 15–17, (2014).