

1-1-2019

Performance evaluation of WebRTC-based online consultation platform

ERGÜN ALPERAY TARIM

HÜSEYİN CUMHUR TEKİN

Follow this and additional works at: <https://journals.tubitak.gov.tr/elektrik>



Part of the [Computer Engineering Commons](#), [Computer Sciences Commons](#), and the [Electrical and Computer Engineering Commons](#)

Recommended Citation

TARIM, ERGÜN ALPERAY and TEKİN, HÜSEYİN CUMHUR (2019) "Performance evaluation of WebRTC-based online consultation platform," *Turkish Journal of Electrical Engineering and Computer Sciences*: Vol. 27: No. 6, Article 19. <https://doi.org/10.3906/elk-1903-44>
Available at: <https://journals.tubitak.gov.tr/elektrik/vol27/iss6/19>

This Article is brought to you for free and open access by TÜBİTAK Academic Journals. It has been accepted for inclusion in Turkish Journal of Electrical Engineering and Computer Sciences by an authorized editor of TÜBİTAK Academic Journals. For more information, please contact academic.publications@tubitak.gov.tr.

Performance evaluation of WebRTC-based online consultation platform

E. Alperay TARIM¹, H. Cumhuri TEKİN*¹

Department of Bioengineering, Faculty of Engineering, İzmir Institute of Technology, İzmir, Turkey

Received: 07.03.2019

Accepted/Published Online: 08.07.2019

Final Version: 26.11.2019

Abstract: Information technologies give patients the opportunity to communicate with medical professionals remotely. Telemedicine uses these technologies to provide advanced healthcare and medical services. We present a medical online consultation application based on Web Real-Time Communications (WebRTC) technology enabling chat, audio, and video calls. Communication architecture and protocols of the application are explained in detail. Additionally, the user interface of the application is shown via performed calls. The application is tested and evaluated on different network connections (3G, 4G, local, and DSL) and different browsers and mobile operating systems (Android, Chrome, Firefox, Internet Explorer, iOS, Opera, Safari). During calls, communication quality parameters such as round-trip time (RTT) and packet loss, obtained via the WebRTC application programming interface, are analyzed. 3G, 4G, and local connections show low packet losses (<1%). Packet losses are high (>1%) in Android, Chrome, iOS, Opera, and Safari for DSL connection, but RTT values are low (<100 ms) in all different conditions excluding iOS. In the presented application, RTT and packet loss remain lower than 100 ms and 1%, respectively, in various scenarios, indicating good communication quality. RTT and packet loss are related to total time and hang time parameters, which describe the necessary time to establish and to end a call. It is shown that communication quality of the application can simply be measured by analyzing the total time parameter. This enables predictable information for communication quality for WebRTC-based applications without continuously monitoring RTT and packet loss for the first time.

Key words: WebRTC, online consultation, video call, cross platform, telemedicine

1. Introduction

Healthcare services have become more innovative and patient-centric with the help of information technology. The exchange of information between patients and doctors has become easier with communication technologies and that improves healthcare services and technologies [1–5]. Patients can readily reach doctors and healthcare service providers through communication tools [6]. Moreover, it is possible to share medical data with/among healthcare service providers [7]. As a result of these services, a new medical field called telemedicine has emerged, whereby healthcare providers can remotely diagnose, monitor, and treat diseases via telecommunication tools [8]. Telemedicine aims to provide remote clinical support, overcome geographical barriers, and improve health outcomes by using various types of information and communication technologies [9]. Telemedicine has many uses in the medical field and is mostly referred to as teleradiology [10], telepathology [11], tele-emergency [12], teletrauma [13], telecardiology [14], teledermatology [15], teleophthalmology [16], telepsychology [17], and telesurgery [18]. The use of information technology in medicine is enhancing day-by-day and telemedicine has become a necessity for today's healthcare services [19].

*Correspondence: cumhurtekin@iyte.edu.tr

There are many tools available in the telemedicine field and online consultation is one of the key elements in this field that allows to reduce nursing costs, to overcome shortages of the healthcare work force, to maintain better home healthcare, and also to enhance public health, health education, and patient–doctor and interprofessional relationships [20–22]. Web Real-Time Communication (WebRTC), an open source project that provides real-time communications to web browsers, is one of the important tools in this regard [23]. WebRTC offers secure peer-to-peer media and direct file sharing between browsers without the need of mediating servers or third-party software [24]. The WebRTC application programming interface (API) is used for developing communication applications with data, audio, and video channels. WebRTC API has three main elements: (i) PeerConnection, (ii) DataChannel, and (iii) MediaStream [25]. PeerConnection establishes direct communication between peers. DataChannel is a data transport service between peers via bidirectional peer-to-peer connection. MediaStream creates audio and video streams and manages their content.

WebRTC has been used for online medical consultation, enabling treatment and rehabilitation [26], and also for diagnosis by transmitting radiology images. For instance, ultrasound images of organs were sent over the WebRTC framework for remote diagnostics [27]. In some cases, WebRTC has been utilized for monitoring and examination of elderly people and physically or mentally challenged patients [22, 28–30]. The WebRTC-based architecture was designed for remote and continuous monitoring of seniors, remote medical examination, and emergency intervention [28]. In this architecture, wearable devices were planned to supply necessary information remotely to healthcare personnel via WebRTC DataChannel while online consultation took place. A WebRTC-based application was developed for coordination of paramedics with an incident command center in disaster scenarios [29]. This application works via Google Glass, allowing paramedics hands-free communication. A video conferencing system based on WebRTC was designed for a tele-home monitoring project conducted in different states of Austria [30]. EasyRTC, which is an online toolkit available to implement WebRTC applications, was used for conferencing. WebRTC was used for physical therapy and exercise over a telepresence application [31]. A telepresence exercise platform based on WebRTC technology was developed for elderly women with a high risk of falling. Moreover, real-time communication technology could be integrated with virtual and augmented reality for healthcare services [32].

Here, we present an online consultation application based on native WebRTC technology, and in this work, corresponding communication protocols of the application are explained in detail. The coverage of the application has been tested for different networks, browsers, and mobile systems by evaluating communication quality parameters. In this way, WebRTC performance in different usage scenarios has been revealed for the first time. Moreover, necessary times to establish and to end a call are studied to reveal their effects on communication quality.

2. Method

2.1. WebRTC communication protocols

The WebRTC communication architecture used in our application is shown in Figure 1. In this architecture, both peers need to send necessary protocol information through a socket.io server to establish a connection. Interactive Connectivity Establishment (ICE) candidates are initiated and identified through the protocol to locate peers [33]. For this purpose, first a Session Traversal Utilities for Network Address Translator (STUN) server is used to identify peer addresses. If the STUN server is successful in identifying addresses, a direct media connection is established between peers. In some cases, the STUN server cannot provide a direct connection to

the peer. Then a Traversal Using Relays around Network Address Translator (TURN) server allows obtaining a public address of the peers and transferring media between peers.

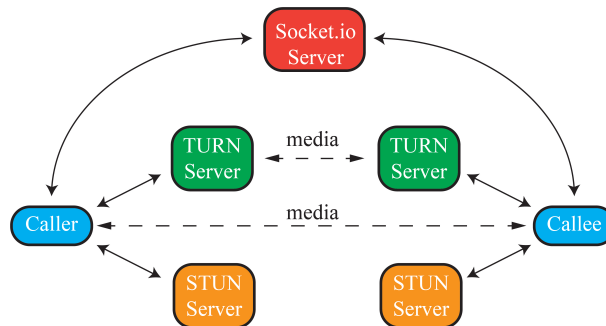


Figure 1. The WebRTC communication architecture. This diagram shows the communication between peers (caller or callee) in the presence of TURN, STUN, and Socket.io servers.

Figure 2 shows the communication protocol between peers to establish a call procedure composed of initiation, handshake, ICE negotiation, and hang-up stages. In the initiation stage, first peers log in to the application and they are assigned to different rooms by the socket.io server. These rooms are used to send the communication status of peers. In this way, users are in an Idle step, where they can receive or initiate calls. For the calling process, a peer (caller) starts a call and registers with the room of the second peer (callee) receiving a call. If callee's room is full (i.e. more than two peers are connected) or empty, the calling process is terminated. If not, the callee receives the incoming call. If the callee accepts the call, the WebRTC communication procedure will be started.

The WebRTC library is used to send/receive media to/from peers*. First, peers start their media streams (video and/or audio). When the streams are ready, the callee creates a peer connection, adds its local stream, creates an offer, and sends this offer to the caller. For chat communications, the proposed application will follow the same procedure by taking users' media streams in ready states without starting the media and the application will start the data channel for transferring text messages. When a caller receives this offer, the caller also creates a peer connection, sets a remote description, adds its local stream, creates the answer, and sends this answer to the callee. When the answer is received, the callee sets the remote descriptions. While a peer connection is set, ICE candidates are initiated. If a local ICE candidate is created, the protocol checks the created offer and adds the remote ICE candidate. If one of these checked items is available, the local ICE candidate is sent to the remote user and the remote user adds this candidate as a remote ICE candidate.

When the remote and local streams are ready, ICE connection states are controlled. If the ICE state is connected or completed, the call will be established and communication screens for chat/audio/video calls will be shown to the peers. When the ICE connection is in a checking or new state, the protocol will wait for a while and check the ICE connection state again. If the ICE connection is in another state (disconnected, failed, or closed), the connection will be restarted by closing the peer connection.

If the callee or caller hangs up on the call, the peer connection and data channel will be closed and media streams will be terminated. Furthermore, the caller will exit the callee's room and register with its own room. Hence, the hang-up procedure (Hang Up Stage) will be completed and peers will be ready to receive or initiate a new call.

*WebRTC (2011). WebRTC Architecture [online]. Website <https://webrtc.org/> [accessed 22.07.2019].

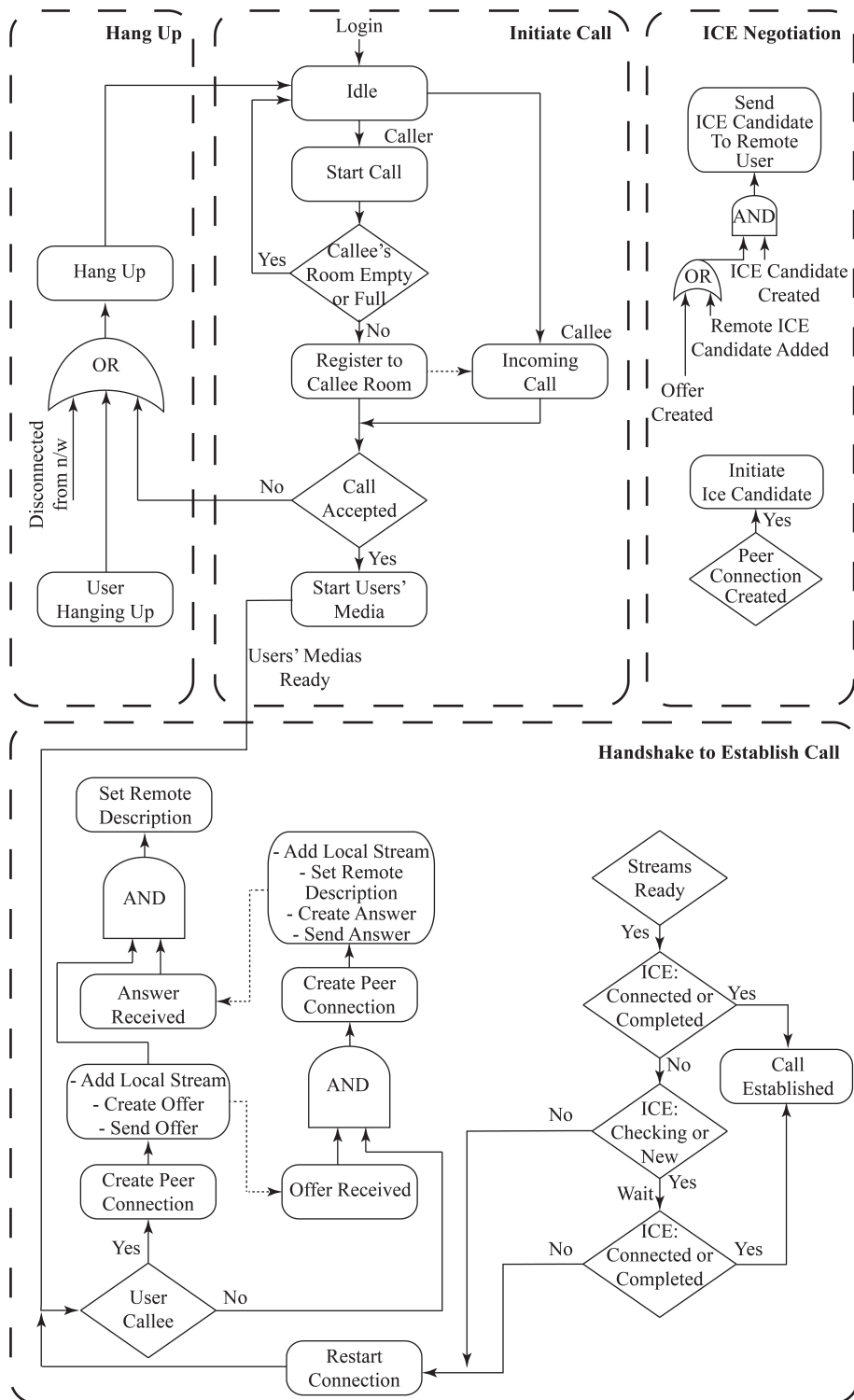


Figure 2. Communication protocol between peers. The state machine diagram is given for a call procedure composed of initiation, handshake, ICE negotiation, and hang-up stages.

2.2. Development of WebRTC application

The user interface of our application is given in Figure 3. After users log in to the application, they can search corresponding users to make chat, audio, and video calls (Figure 4). We used HTML, CSS, PHP, and JavaScript to develop our online consultation application. WebRTC API adapter.js (version v0.14.0) was utilized to realize WebRTC communication protocols.

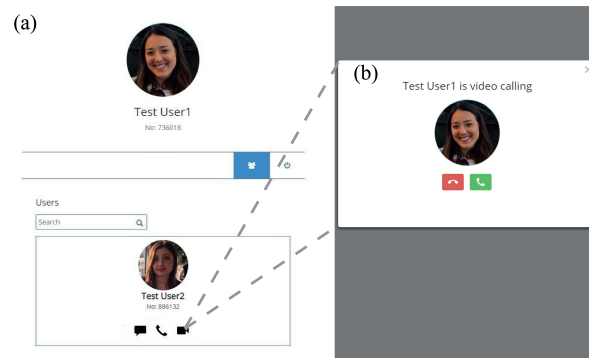


Figure 3. Application visuals. (a) Main window of the application after a user logs in to the application. The user (Test User1) can search and see registered users and can send chat, audio, and video communication requests to these users by clicking on corresponding icons under the search results. (b) Incoming communication request. The users are informed about the communication request. If the user who receives the call accepts the incoming communication request with the green earphone button, the communication will be started.

After developing the web-based application, the code was built for mobile systems (Android and iOS) by using the Apache Cordova development framework. The Cordova iOSRTC plugin enables WebRTC API on iOS. In this way, the same code was utilized for both web and mobile applications. Web application can run on browsers that support WebRTC, such as Chrome, Firefox, or Opera. Moreover, the Temasys Browser Plugin was used to provide WebRTC support for other browsers, such as Safari and Internet Explorer.

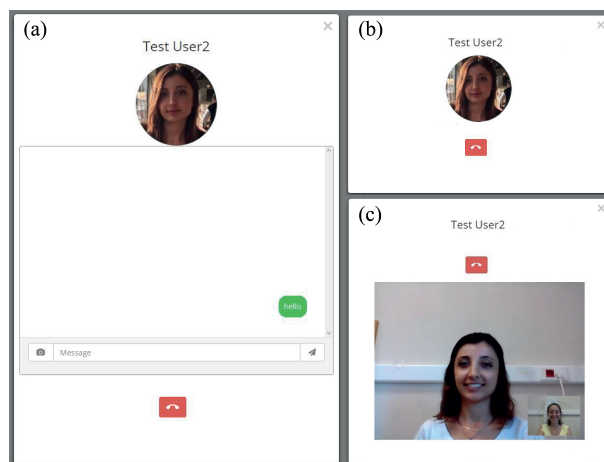


Figure 4. Visuals of (a) chat, (b) audio, and (c) video communication interfaces. The users can finish the communication by pressing the red hang-up button.

2.3. Performance analysis

WebRTC-based real-time video, audio, and chat communications were tested for different connection types (3G, 4G, DSL, local) and for different browsers (Chrome version 59.0.3071.86, Firefox version 53.0.3, Internet Explorer version 11.0.9600.18321, Safari version 9.1.1, Opera version 45.0.2552.888) and mobile operating systems (Android 5.0, iOS 8.1.3).

The performance data of the communication were obtained via GetStats API, which gives access to WebRTC data on Chrome browsers. Using this API, we analyzed (i) round-trip time (RTT), which is the measured elapsed time in milliseconds from the browser's request sending time to the browser's server response receiving time [34]; (ii) packets received, which is the total number of packets received for this connection over the network; (iii) packets lost, which is the total number of packets lost for this connection [35]; and (iv) frame rate (frames per second, fps), only for video communication. We also analyzed packet loss, which is the percentage of packets lost with respect to total received packets. Moreover, total connection time and hang time were measured. Total connection time indicates the time elapsed between the start of users' media and call established states (Figure 2). Hang time is the time to complete the hang-up stage (Figure 2).

We tested the performances of different communication types (chat, audio, video) with six different calls. Each call was 5 min long and during the call performance data were collected at intervals of 5 s. For chat calls, users were sending 20-character-long messages in each second. For video calls, a 640×480 sized video at 30 fps was set to be streamed. After 5 min, the call was ended automatically. The Chrome browser was utilized to analyze different connection performances of our application. For the local connection type, the caller and callee were in the same local network. For the other connection types, the caller's connection was changing but the callee connection stayed in DSL. For testing the performance of different browsers and mobile operating systems, the caller's browser or operating system was changed accordingly and, in the meantime, the callee was using the Chrome browser. The DSL connection was utilized for testing different browsers and mobile operating systems. The callee's data were used for all tests and the test data were compared with the one-way ANOVA (analysis of variance) Tukey test. All performance data are available in the supplementary tables and figures.

3. Results and discussion

The performance data of the application were analyzed for different types of connections, which are 3G, 4G, DSL, and local (Figure 5). 4G technology has recently gained momentum and has begun to replace 3G technology. In 2018, 29% and 45% of mobile subscribers uses 3G and 4G networks, respectively.* Although the use of 4G technology is increasing and even 5G technology is being introduced, it is predicted that by 2024, 3G technology will still account for 17% of mobile subscribers globally. Meanwhile, 4G is expected to reach 56% of all mobile subscribers. Each tested connection type has different network architecture features such as data transfer rate, security, access, and transmission terminology [36]. Moreover, radio access network can change for different network architectures. For example, 3G and 4G are mobile networks that are based on Universal Mobile Telecommunication Service (UMTS) and Long Term Evolution (LTE) systems, respectively.

Hence, 3G and 4G have different frequency bands, channel bandwidths, protocols, channel types, etc. Also, different networks have different Internet connection speeds, which were measured as 8.97 ± 1.59 Mbps, 26.07 ± 3.45 Mbps, and 13.18 ± 2.55 Mbps for 3G, 4G, and DSL connections, respectively. Since local connection tests were conducted in the same local area network, connection speeds were expected to be higher

*Ericsson (2019). Ericsson Mobility Report June 2019 [online]. Website <https://www.ericsson.com/en/mobility-report/reports/june-2019> [accessed 22.07.2019].

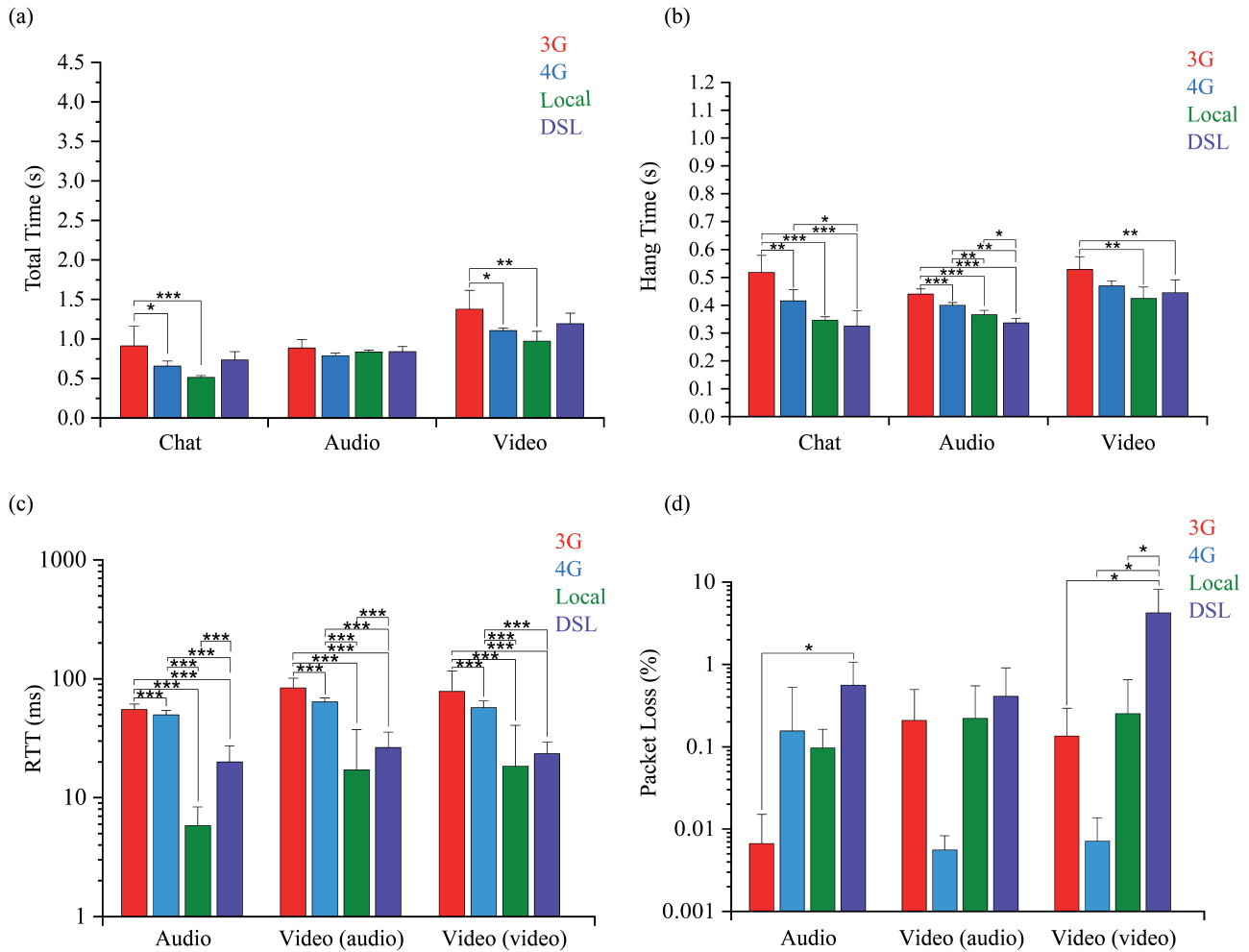


Figure 5. Communication quality parameters of different network connections: (a) total connection time, (b) hang time, (c) RTT, and (d) packet losses shown for different call types in 3G, 4G, local, and DSL. *, **, and *** are used to indicate $P < 0.05$, $P < 0.01$, and $P < 0.001$, respectively.

than the other connection types. As shown in Figure 5a, 3G resulted in longer total connection time and it was significantly different from 4G and local networks for chat and video calls. However, for audio calls, total connection time was not significantly different for different connection types. Total connection time reached 1.38 ± 0.24 s for video calls in 3G. On the other hand, this connection time was reduced to 0.97 ± 0.12 s for the local connection. Since there was a huge video media stream in video calls, these calls showed longer total connection times than chat and audio calls. Although 3G showed the longest hang times for each call type (Figure 5b), hang times remained under 0.53 s for different call types. Hang time values of 3G were significantly different from other networks for chat and audio calls, and they were also significantly different from local and DSL networks for video calls. DSL and local networks had smaller hang times than 3G and 4G networks. RTT values had big differences in all networks for all call types (Figure 5c). In audio calls, the 3G network reached 78.42 ± 5.76 ms RTT values, whereas the local network remained at 11.86 ± 2.48 ms. For audio packets of video calls, the 3G network reached 55.04 ± 6.41 ms RTT values and the local network remained at 5.82 ± 2.54 ms. For video packets of video calls, the 3G network reached 77.98 ± 37.99 ms RTT values and the local

network remained at 18.31 ± 22.42 ms. In the meantime, the local network showed the smallest RTT values for audio call packets, audio packets of video calls, and video packets of video calls, respectively. RTT had maximum values for 3G connections and it was 77.98 ± 38.00 ms for video packets of video calls. This value is well below 100 ms, which is a measure for optimal media quality.**

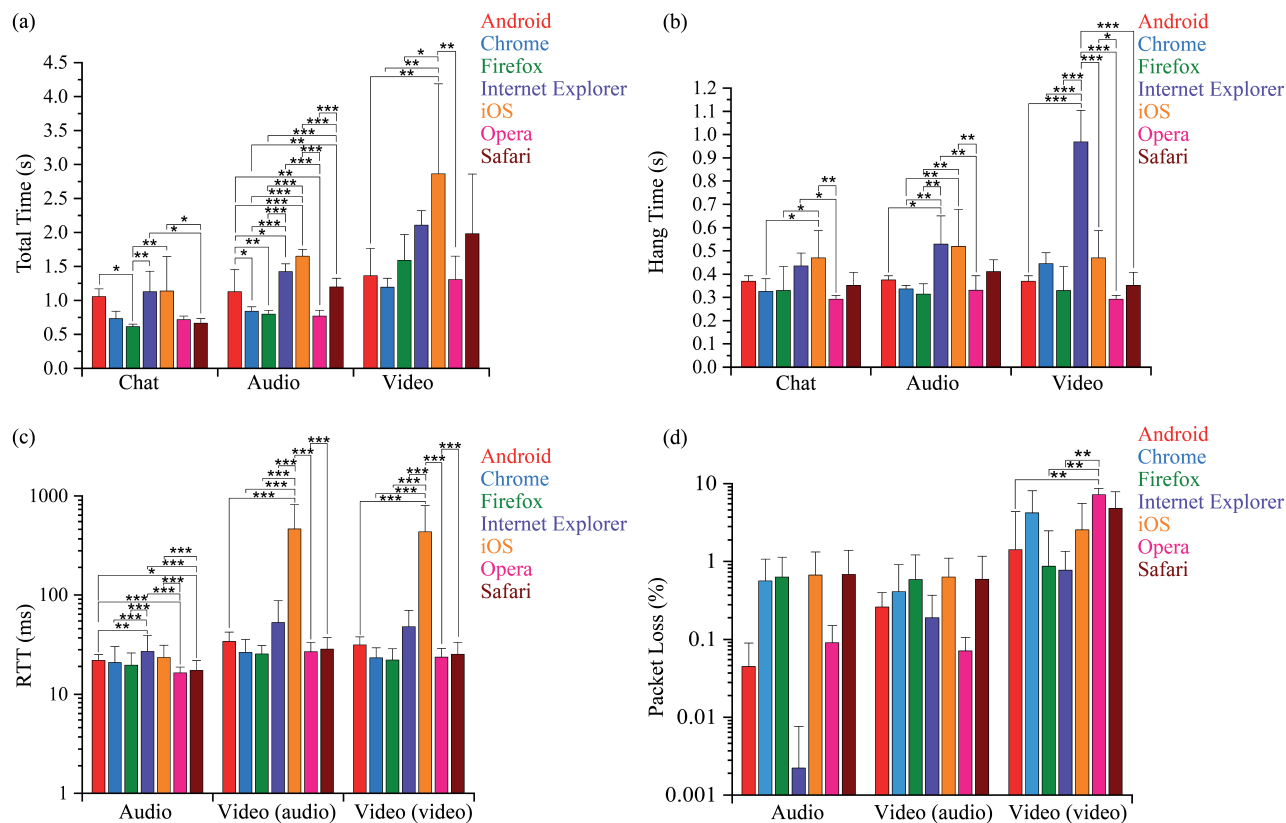


Figure 6. Results for different browsers and mobile operating systems: (a) total connection time, (b) hang time, (c) RTT, and (d) packet losses shown for different call types in Android, Chrome, Firefox, Internet Explorer, iOS, Opera, and Safari browsers and mobile operating systems. *, **, and *** are used to indicate $P < 0.05$, $P < 0.01$, and $P < 0.001$, respectively.

There was no significant difference in packet loss for audio packets of video calls (Figure 5d). However, packet losses were higher in the DSL connection compared to the other connection types and it was significantly different from the 3G network in audio calls and also significantly different from all other networks for video packets of video calls. Packet loss reached its maximum value of $4.21 \pm 3.88\%$ for video packets of video calls in the DSL connection. On the other hand, the 4G network’s packet loss remained under 0.01% for audio and video packets of video calls. For different connection types, 3G, which has the lowest Internet connection speed, showed the worst performance in total time, hang time, and RTT. On the contrary, packet loss for video packets was increased 31-fold in DSL, which has the second lowest Internet connection speed, compared

**Microsoft Teams (2018). Media Quality and Network Connectivity Performance in Skype for Business Online [online]. Website <https://docs.microsoft.com/en-us/skypeforbusiness/optimizing-your-network/media-quality-and-network-connectivity-performance> [accessed 22.07.2019].

to the 3G connection. Packet loss in other connection types remains below 1% , which is good for streaming audio or video. Packet losses in 3G, 4G, and local networks were statistically indifferent. Moreover, average frame rates of video communication were in the interval of 29.4–30 fps for different networks and no statistical difference between networks was found (Figure S1). Based on these results, the presented application showed good communication quality (RTT <100 ms and packet loss <1%) in different network scenarios. However, video packet losses were statistically high in DSL (>1%). This value indicates a problem in communication quality for video calls in the DSL network.

Performance analysis of the application was also performed on different browsers and mobile operating systems (Figure 6). The iOS mobile operating system had the longest total connection times among the others with 1.13 ± 0.5 s, 1.64 ± 0.1 s, and 2.86 ± 1.32 s for chat, audio, and video calls, respectively (Figure 6a). For chat calls, total time was below 1.5 s for all systems, whereas total times were below 1 s only for Chrome, Firefox, Opera, and Safari. Similar trends in total time values were observed for audio calls, except in iOS and Safari. Total time values for iOS and Safari reached 1.64 s and 1.2 s, respectively. Total times were below 2.11 s for all systems except iOS and it reached 1.19 ± 0.13 s for the Chrome browser, which was its minimum value for video calls. As shown in Figure 6b, the longest hang time was obtained for Internet Explorer during video calls and it was 0.97 ± 0.13 s, whereas the hang time for the other systems stayed between 0.29 and 0.47 s. The RTT value of Internet Explorer was higher than the other systems for audio calls (Figure 6c). This RTT value was 26.97 ± 12.17 ms, which was not statistically different than the RTT value of iOS. RTT values of iOS were higher than those of other systems for video calls, which were 466.03 ± 353.13 ms and 436.06 ± 363.02 ms for audio packets and video packets, respectively (Figure 6c). Therefore, we encountered several video freezing problems during iOS tests that might have been caused by the Cordova iOSRTC plugin. On the other hand, RTT values remained under 53 ms for other systems, assuring good streaming quality. Packet losses remained under 1% for all audio calls and audio packets of all video calls. There was no significant difference between these calls. Video packet loss was under 1% only for Firefox and Internet Explorer (Figure 6d). However, packet losses reached $7.20 \pm 1.46\%$ for Opera and it was significantly different from Chrome, Firefox, and Internet Explorer systems for video packets of video calls. The remaining systems had less than 5% packet loss. Furthermore, average frame rates of browsers and mobile operating systems in video communications were in the interval of 29.4–30 fps. Safari had the lowest frame rate, which was also statistically different from other systems except iOS (Figure S2). With these results, it was revealed that audio streaming showed good quality (RTT <100 ms and packet loss <1%) in all tested systems in the DSL network. iOS with high RTT (>100 ms) and packet loss (>1%) values and Android, Chrome, Opera, and Safari with high packet loss (>100 ms) in DSL networks can affect video streaming quality [37, 38]. However, packet losses could be minimized by using different networks, such as 4G (Figure 5d), in order to improve communication quality in browsers and mobile operating systems.

Tests revealed that on average 52.03 MB of data were received during 5-min video calls. During the calls, 640×480 sized video at 29.89 fps average was streamed, which consumed 828-fold and 36-fold more data than chat and audio calls, respectively. Although this huge consumption of data could be costly for mobile users, the increased popularity of unlimited data plans may dissipate this problem.

Communication quality can be analyzed through RTT and packet losses [38]. We defined high, medium, and low communication quality as follows: (i) RTT is lower than 50 ms and packet loss is lower than 0.5%; (ii) either RTT is higher than 50 ms and packet loss is lower than 0.5% or RTT is lower than 50 ms and packet

loss is higher than 0.5%; (iii) RTT is higher than 50 ms and packet loss is higher than 0.5%. In this regard, 29% and 63.5% of total calls are of high and medium quality, respectively, for video communication, whereas in audio communication, 83.5% and 15% of total calls are of high and medium quality, respectively. Total time and hang time values were examined with respect to video communication quality. When the total time was less than 1 s, high quality communication was reached for all calls (Table 1). The majority of calls (98%) had at least medium communication quality when the total time was between 1 s and 2 s. In this 1–2 s total time interval, 26%, 72%, and 2% of the calls were of high, medium, and low quality, respectively. Beyond this time interval, the percentage of low quality calls (33.5%) was increased significantly and only 16.5% of calls were of high quality. For hang time values lower than 0.3 s, 25% and 75% of calls were of high and medium quality, respectively (Table 2).

Table 1. Quality of video calls with respect to percentage of calls at different total time intervals.

Total time	0–1 s	1–2 s	>2 s
High quality *	100%	26%	16.5%
Medium quality **	0	72%	50%
Low quality ***	0	2%	33.5%

* Packet loss <0.5% and RTT <50 ms

** (Packet loss <0.5% and RTT >50 ms) or (Packet loss >0.5% and RTT <50 ms)

*** Packet loss >0.5% and RTT >50 ms

Table 2. Quality of video calls with respect to percentage of calls at different hang time intervals.

Hang time	0–0.3 s	0.3–0.6 s	>0.6 s
High quality *	25%	28%	37.5%
Medium quality **	75%	66%	37.5%
Low quality ***	0	6%	25%

* Packet loss <0.5% and RTT <50 ms

** (Packet loss <0.5% and RTT >50 ms) or (Packet loss >0.5% and RTT <50 ms)

*** Packet loss >0.5% and RTT >50 ms

The high and medium quality ratio was slightly reduced to 94% (i.e. 28% and 66% of total calls are of high and medium quality) when the hang time was between 0.3 s and 0.6 s. After this time interval, the ratio of low communication quality was increased to 25%. For hang time, no clear time interval was found to distinguish high and medium communication quality. Based on these results (Table 1 and Table 2), the total time could give a better probability of predicting communication quality for certain time intervals than the hang time. For instance, low total time (<1 s) indicated high quality communication in all tests and moderate total time (1–2 s) indicated high or moderate communication quality in most of the tests (98%). Thus, communication quality is correlated with the total time and it can be simply analyzed using the total time parameter without continuous monitoring of WebRTC performance parameters obtained from GetStats API.

The application was tested with 84 different nonautomated and unsupervised video calls. In these calls,

the mean values of total connection time and hang time were found as 6.03 s and 0.55 s, respectively (Figure 7). In these tests, 6% of the calls had a total time of less than 1 s, indicating high communication quality (Table 1), while 47.6% of the calls had a total time of less than 2 s. Hence, nearly half of the video calls were expected to have a high probability of reaching at least medium communication quality based on the results in Table 1, which indicates that our application has sufficient quality for online consultation.

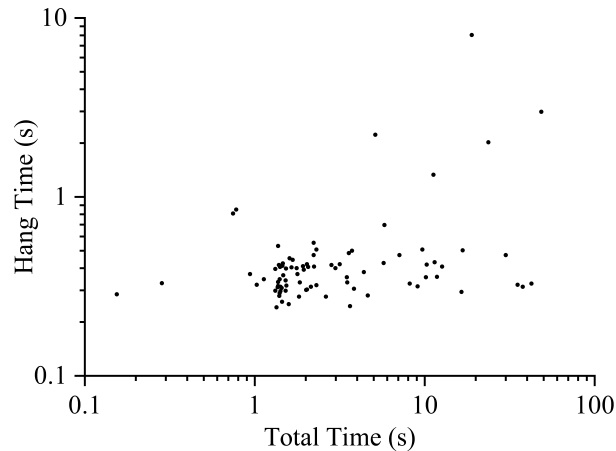


Figure 7. Corresponding total connection time and hang time of 84 different video calls made using the application.

4. Conclusion

A WebRTC-based online consultation application was presented, and it was tested in different networks, browsers, and mobile operating systems for the first time to reveal its performance in different usage scenarios. Audio calls showed good communication quality in all tests. On the other hand, video calls with Android, Chrome, iOS, Opera, and Safari in the DSL network had a worsening of communication parameters, such as RTT and packet loss. This indicates communication problems, which degrade communication quality. For instance, video-freezing problems can be observed in iOS having high RTT (>100 ms) and packet loss ($>1\%$) values. Moreover, total time values showed a good relation with communication quality, which can eliminate continuous monitoring of RTT and packet loss parameters. For telemedicine services, good streaming conditions are necessary for uninterrupted services. Test results revealed good communication conditions for interconnection types and interbrowsers. Hence, this presented application can be utilized in various usage scenarios in different networks and browsers. 5G network technology shows significant performance gains in terms of connection reliability, spectral efficiency, system capacity, and transmission range [39]. By the deployment of the 5G network, the application can be examined in the 5G network to reveal the effect of 5G on communication quality. As a future work, the proposed application can be tested with patients and healthcare providers to reveal its usability for online medical consultation. Moreover, ultrahigh resolution images are needed for dermatological diagnosis. By improving video resolution and network quality, WebRTC applications may stream high quality videos that can lead to improved remote diagnosis. It is evident that WebRTC can open new paths in the field of telemedicine, like developing disease-specific medical applications with secure teleconsultation capability.

Acknowledgments

This work was funded by the Turkish Industry and Technology Ministry (0971.TGSD.2015). The first author acknowledges the support of the Turkish Council of Higher Education for a 100/2000 CoHE doctoral scholarship. The authors would like to thank Engin Özcivici, PhD, from the Department of Bioengineering, İzmir Institute of Technology, for helpful discussions and suggestions; Sena Yaman and Yekta Gülmez from the Department of Bioengineering, İzmir Institute of Technology, for their help with Figures 3 and 4; and Başak Tekin from Saniber Ltd. for critically reviewing the manuscript.

References

- [1] De La Torre-Díez I, López-Coronado M, Vaca C, Aguado JS, de Castro C. Cost-utility and cost-effectiveness studies of telemedicine, electronic, and mobile health systems in the literature: a systematic review. *Telemedicine Journal and E-health* 2015; 21 (2): 81-85.
- [2] Lindberg B, Nilsson C, Zotterman D, Söderberg S, Skär L. Using information and communication technology in home care for communication between patients, family members, and healthcare professionals: a systematic review. *International Journal of Telemedicine and Applications* 2013; 2013 (1): 461829.
- [3] Krishna S, Boren SA, Balas EA. Healthcare via cell phones: a systematic review. *Telemedicine Journal and E-health* 2009; 15 (3): 231-240.
- [4] Sheikh A, Sood HS, Bates DW. Leveraging health information technology to achieve the “triple aim” of healthcare reform. *Journal of the American Medical Informatics Association* 2015; 22 (4): 849-856.
- [5] Cueto-Manzano AM, Gallardo-Rincón H, Martínez-Ramírez HR, Cortés-Sanabria L, Rojas-Campos E et al. A pilot study of a mobile phone application to improve lifestyle and adherence of patients with kidney disease. *Journal of Telemedicine and Telecare* 2015; 21 (2): 119-120.
- [6] Caffery LJ, Smith AC. Investigating the quality of video consultations performed using fourth generation (4G) mobile telecommunications. *Journal of Telemedicine and Telecare* 2015; 21 (6): 348-354.
- [7] Azevedo J, Pereira RL, Chainho P. An API proposal for integrating sensor data into web apps and WebRTC. In: *1st Workshop on All-Web Real-Time Systems*; Seoul, South Korea; 2015. pp. 1-5.
- [8] Ekeland AG, Bowes A, Flottorp S. Effectiveness of telemedicine: a systematic review of reviews. *International Journal of Medical Informatics* 2010; 79 (11): 736-771.
- [9] Sood S, Mbarika V, Jugoo S, Dookhy R, Doarn CR et al. What is telemedicine? A collection of 104 peer-reviewed perspectives and theoretical underpinnings. *Telemedicine Journal and E-health* 2007; 13 (5): 573-590.
- [10] Puel A, Von Wangenheim A, Meurer MI, De Macedo DD. BUCOMAX: Collaborative multimedia platform for real time manipulation and visualization of bucomaxillofacial diagnostic images. In: *IEEE 27th International Symposium on Computer-Based Medical Systems*; New York, NY, USA; 2014. pp. 392-395.
- [11] Kumar S. Introduction to telepathology. In: Kumar S, Dunn BE (editors). *Telepathology*. 1st ed. Heidelberg, Germany: Springer, 2009. pp. 1-4.
- [12] Mueller KJ, Potter AJ, MacKinney AC, Ward MM. Lessons from tele-emergency: improving care quality and health outcomes by expanding support for rural care systems. *Health Affairs (Millwood)* 2014; 33 (2): 228-234.
- [13] Latifi R, Weinstein R, Porter J, Ziemba M, Judkins D et al. Telemedicine and telepresence for trauma and emergency care management. *Scandinavian Journal of Surgery* 2007; 96 (4): 281-289.
- [14] Istepanian RS, Petrosian AA. Optimal zonal wavelet-based ECG data compression for a mobile telecardiology system. *IEEE Transactions on Information Technology in Biomedicine* 2000; 4 (3): 200-211.
- [15] Whited JD. Teledermatology research review. *International Journal of Dermatology* 2006; 45 (3): 220-229.

- [16] Massin P, Chabouis A, Erginay A, Viens-Bitker C, Lecleire-Collet A et al. OPHDIAT©: A telemedical network screening system for diabetic retinopathy in the Île-de-France. *Diabetes and Metabolism* 2008; 34 (3): 227-234.
- [17] Rees CS, Haythornthwaite S. Telepsychology and videoconferencing: issues, opportunities and guidelines for psychologists. *Australian Psychologist* 2004; 39 (3): 212-219.
- [18] Marescaux J, Leroy J, Rubino F, Smith M, Vix M et al. Transcontinental robot-assisted remote telesurgery: feasibility and potential applications. *Annals of Surgery* 2002; 235 (4): 487-492.
- [19] Eron L. Telemedicine: the future of outpatient therapy? *Clinical Infectious Diseases* 2010; 51 (2): 224-230.
- [20] Farnan JM, Sulmasy LS, Worster BK, Chaudhry HJ, Rhyne JA et al. Online medical professionalism: patient and public relationships: policy statement from the American College of Physicians and the Federation of State Medical Boards. *Annals of Internal Medicine* 2013; 158 (8): 620-627.
- [21] Finkelstein SM, Speedie SM, Potthoff S. Home telehealth improves clinical outcomes at lower cost for home healthcare. *Telemedicine Journal and E-Health* 2006; 12 (2): 128-136.
- [22] The IWG ASIA Task Force on Telemedicine. Roadmap for Telemedicine, Key Considerations and Recommendations. Innovation Working Group, 2013.
- [23] Liu WL, Zhang K, Locatis C, Ackerman M. Internet-based videoconferencing coder/decoders and tools for telemedicine. *Telemedicine Journal and E-Health* 2011; 17 (5): 358-362.
- [24] Johnston AB, Burnett DC. WebRTC: APIs and RTCWEB protocols of the HTML5 real-time web. St. Louis, MO, USA: Digital Codex LLC, 2012.
- [25] Jesup R, Loreto S, Tuexen M. WebRTC Data Channels. IETF, 2015.
- [26] Antón D, Kurillo G, Goñi A, Illarramendi A, Bajcsy R. Real-time communication for Kinect-based telerehabilitation. *Future Generation Computer Systems* 2017; 75: 72-81.
- [27] Bharath R, Vaish P, Rajalakshmi P. Implementation of diagnostically driven compression algorithms via WebRTC for IoT enabled tele-sonography. In: *IEEE EMBS Conference on Biomedical Engineering and Sciences*; Kuala Lumpur, Malaysia; 2016. pp. 204-209.
- [28] El Jaouhari S, Bouabdallah A, Bonnin JM, Lemlouma T. Toward a smart health-care architecture using WebRTC and WoT. In: *World Conference on Information Systems and Technologies*; Porto Santo Island, Portugal; 2017. pp. 531-540.
- [29] Gillis J, Calyam P, Bartels A, Popescu M, Barnes S et al. Panacea's glass: Mobile cloud framework for communication in mass casualty disaster triage. In: *3rd IEEE International Conference on Mobile Cloud Computing, Services and Engineering*; San Francisco, CA, USA; 2015. pp. 128-134.
- [30] Jang-Jaccard J, Nepal S, Celler B, Yan B. WebRTC-based video conferencing service for telehealth. *Computing* 2016; 98 (1-2): 169-193.
- [31] Hong J, Kong HJ, Yoon HJ. Web-based telepresence exercise program for community-dwelling elderly women with a high risk of falling: randomized controlled trial. *JMIR mHealth and uHealth* 2018; 6(5): e132.
- [32] Anton D, Kurillo G, Yang AY, Bajcsy R. Augmented telemedicine platform for real-time remote medical consultation. In: *International Conference on Multimedia Modeling*; Reykjavik, Iceland; 2017. pp. 77-89.
- [33] Loreto S, Romano SP. Real-Time Communication with WebRTC: Peer-to-Peer in the Browser. Sebastopol, CA, USA: O'Reilly Media Inc., 2014.
- [34] Schulzrinne H, Casner S, Frederick R, Jacobson V. RTP: A transport protocol for real-time applications. Internet Engineering Task Force, RFC, 2003.
- [35] Holmer S, Shemer M, Paniconi M. Handling packet loss in WebRTC. In: *IEEE International Conference on Image Processing*; Melbourne, Australia; 2013. pp. 1860-1864.
- [36] Kumaravel K. Comparative study of 3G and 4G in mobile technology. *International Journal of Computer Science Issues* 2011; 8 (5): 256-263.

- [37] Mok RK, Chan EW, Chang RK. Measuring the quality of experience of HTTP video streaming. In: 12th IFIP/IEEE International Symposium on Integrated Network Management and Workshops; Dublin, Ireland; 2011. pp. 485-492.
- [38] Nafaa A, Taleb T, Murphy L. Forward error correction strategies for media streaming over wireless networks. IEEE Communications Magazine 2008; 46 (1): 72-79.
- [39] Tehrani M, Murat U, Halim Y. Device-to-device communication in 5G cellular networks: challenges, solutions, and future directions. IEEE Communications Magazine 2014; 52 (5): 86-92.