

Video Over Intranet (VDOIT)

Deepak Tatayji Ahire ‡

‡ Walchand College of Engineering, Sangli, India

Corresponding author: Deepak Tatayji Ahire (ahiredeepak20@gmail.com)

Reviewed v1

Received: 09 Sep 2020 | Published: 10 Sep 2020

Citation: Ahire DT (2020) Video Over Intranet (VDOIT). Research Ideas and Outcomes 6: e58504.

<https://doi.org/10.3897/rio.6.e58504>

Abstract

Video calling or voice calling through the use of the internet is a very common thing in today's world. Nevertheless, users have to pay charges directly or indirectly, tolerating all the delays even if you use the facility within the same network or sub-network. Over 30 million US households used Voice over IP (VoIP) until the end of 2010. Specialist found that 31.4% of the household used VoIP as the main and only home phone line by 2012. Users up till now have multiplied by a huge factor. There are several drawbacks of using video calling over the internet within a small network such as delay of packets (travel to the main server outside network and reenter the same network again), direct or indirect costs, users need to be always connected online and efficiency and speed of the user's system vary according to connection quality and bandwidth. In this project, we implemented a WebRTC enabled web application which provides free peer to peer video calling and conferencing tool for any private organization on top of their existing private network.

Keywords

webrtc, video-streaming, javascript, node.js, websocket, materialized-css, stun-server, turn-server, huffman-encoding, data-compression, encryption-techniques, networking, video-over-intranet, networking

Problem statement

To implement an on-demand, on-premise, and secure video calling service over the private cloud of private organizations that can be used without the use of any external hardware and free of cost.

Motivation

A lot of students in our college could not afford mobile phones and computing machines. On a big campus of over 90 acres, communication was impossible when internet connections were interrupted or social media sites were down. Communication over the intranet was the only solution that was feasible, affordable, and scalable over the bandwidth of the college's network. To satisfy the above needs, we adapted the efforts of WebRTC project (WebRTC 2019).

Technical areas explored

The technical areas explored are as follows:

1. Google WebRTC - STUN and TURN servers (Dutton 2013).
2. Networking using WebSocket API and Node.js (Fujimoto et al. 2020).
3. Data compression and encryption techniques (Huffman Encoding and Decoding Technique) (Kidner 1995).
4. VoIP analogous protocol (for Application layer) (Arora 1999).
5. P2P Communication using browsers (Shepherd et al. 2019).
6. JavaScript, jQuery & Materialized CSS (for frontend) (Wang 2014).

Application domain

The project was deployed on the private cloud of the Computer Science and Engineering department of Walchand College of Engineering (WCE), Sangli, for the users of the same college.

Application users

The registered users (students and faculty) having a verified account on the service deployed on the cloud and who has access to any computing machine which is connected to the college's LAN and qualifies the system requirements.

System configuration required

The system configurations required are as follows:

1. No external hardware requirements.
2. Systems/devices should be the part of Local Area Network (LAN).
3. A network with a bandwidth of 40-200kbit/s for real-time audio and at least 500kbit/s for real-time video streaming Ber 2015.
4. A standalone signalling server deployed on the private cloud connected to LAN.
5. Supported browsers (Deveria and Schoors 2020):
 1. Chrome 18.0.1008+.
 2. Safari 11.2+ on iOS; 11.1+ on Mac.
 3. Firefox 17+.
 4. Opera 18+.

Methodology

The running application flow is as follows:

1. Clients notify the signaling server as they become online(log in).
2. Signaling server notifies every other connected peer about the client and thus all peers will exchange data with SDP via signaling server.
3. Caller sends a call request to the server.
4. If the receiver is available, then the server signals availability of the receiver to the caller.
5. The caller tries to connect the receiver directly (P2P).
6. If the receiver accepts the call, a peer connection is established.
7. When either client terminates the call, the connection will be closed.
8. Meanwhile, all the availability and network changes will be updated at the server database.

UML Digrams

Fig. 1 depicts the use case diagram for VDOIT.

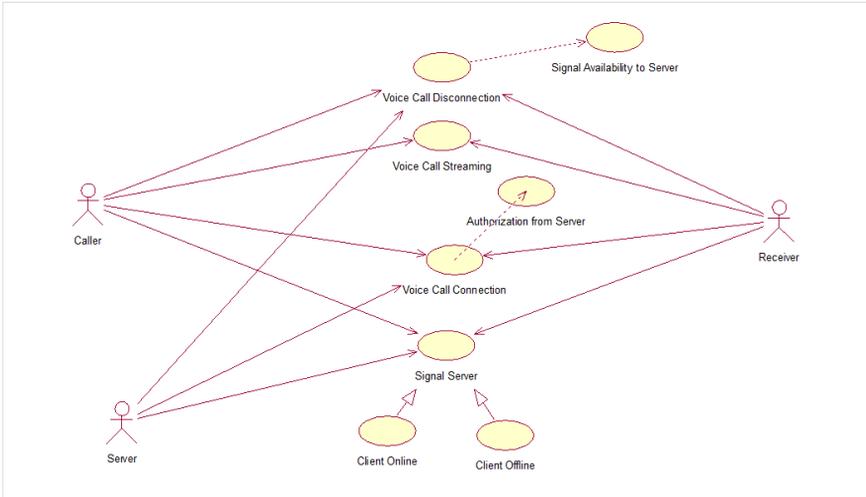


Figure 1. doi
VDOIT use case diagram.

Fig. 2 depicts the sequence diagram for VDOIT.

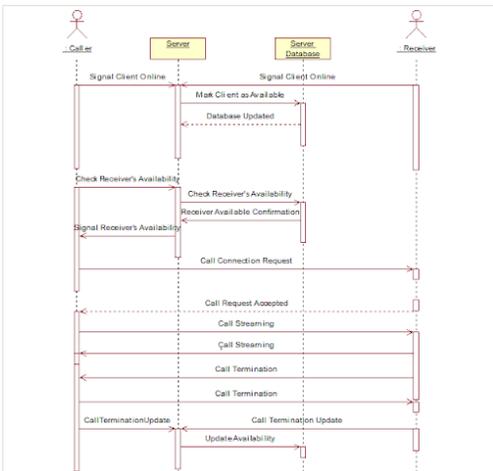


Figure 2. doi
VDOIT sequence diagram.

Fig. 5 depicts the deployment diagram for VDOIT.

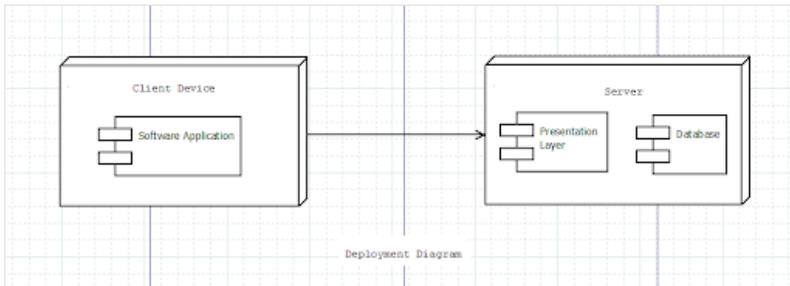


Figure 5. doi

VDOIT deployment diagram.

Fig. 6 depicts the deployment diagram for VDOIT from user's perspective.

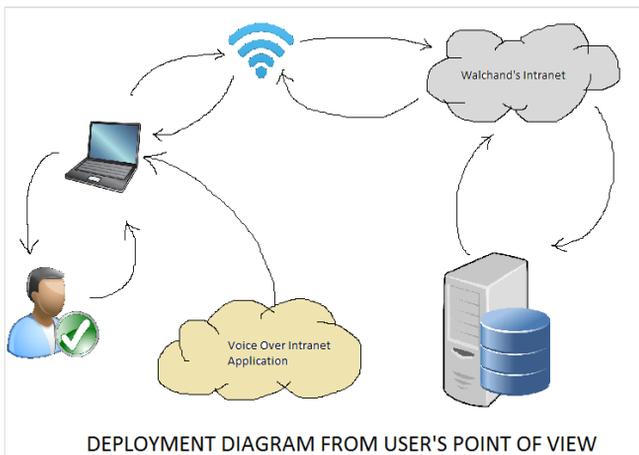


Figure 6. doi

VDOIT deployment diagram from user's perspective.

Project outcomes and current issues solved

The project outcomes and the issues solved with the help of this project are as follows:

1. The registered users were able to make voice and video calls free of cost, without the requirement of internet connectivity, and any external hardware.
2. If someone attempts to make a voice call over the WCE's intranet, then they will be benefited with much less delay than pre-existing systems such as WhatsApp calls, Hangouts call, Facebook video call, etc..
3. No direct or indirect cost is be charged by this service.

Future Scope

This project can be extended to provide a personalized experience for the faculties to carry out online lectures and internal assessments and for the on-campus medical staff to carry out medical examinations.

Conclusion

The project was completed and received positive feedback from the users of WCE. The project received the second runner up prize at the Innovata Project Fair 2017, held on 15th April 2017, organized by the Department of Computer Science and Engineering Innovation, Incubation and Entrepreneurship (IEE) cell.

The code repository was successfully hosted at Github and to contribute to this project, please send pull requests to <https://github.com/adeepak7/Video-Over-Intranet>.

Live demonstration of the working project which is tested on the WCE's intranet is hosted on Youtube at <https://www.youtube.com/watch?v=nAwfaog15N8>.

Acknowledgements

This project was completed as part of the course "Mini project-II" (3CS342), 3rd year, B.Tech Computer Science and Engineering Programme at WCE. I would like to acknowledge the efforts of The Association of Past Students, WCE for supporting the project fair, the network, and server administrator of WCE.

References

- Arora R (1999) Voice over IP : Protocols and Standards. URL: https://www.cse.wustl.edu/~jain/cis788-99/ftp/voip_protocols/
- Ber A (2015) [WebRTC bandwidth requirements](https://stackoverflow.com/questions/29854622/webrtc-bandwidth-requirements). Stack Overflow. URL: <https://stackoverflow.com/questions/29854622/webrtc-bandwidth-requirements>
- Deveria A, Schoors L (2020) Can I Use ? <https://caniuse.com/#search=webrtc>. Accessed on: 2020-8-13.
- Dutton S (2013) WebRTC in the real world: STUN, TURN and signaling. <https://www.html5rocks.com/en/tutorials/webrtc/infrastructure/>. Accessed on: 2020-8-13.
- Fujimoto M, Mills C, Scholz F, Alam MM (2020) The WebSocket API (WebSockets). https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API. Accessed on: 2020-8-13.
- Kidner RE (1995) Real-time data compression using a fixed Huffman encoding scheme. United States.
- Shepherd E, Scholz F, Teoli, Zartner S, Patonnier J, Jouanneau L (2019) https://developer.mozilla.org/en-US/docs/Web/Guide/API/WebRTC/Peer-to-peer_communications_with_WebRTC. <https://developer.mozilla.org/en-US/docs/Web/>

[Guide/API/WebRTC/Peer-to-peer communications with WebRTC](#). Accessed on: 2020-8-13.

- Wang A (2014) A modern responsive front-end framework based on Material Design. URL: [A modern responsive front-end framework based on Material Design](#)
- WebRTC (2019) Real-time communication for the web. <https://webrtc.org/>. Accessed on: 2020-8-13.