Efficient File Sharing Scheme Based on WebRTC

Quanfeng Duan[1, a *], Zhenghe Liang[2,b] and Limin Wang[3,c]

[1]School of Computer and Information,Hohai University,Nanjing 211100,China

[2]School of Computer and Information,Hohai University,Nanjing 211100,China

[3]School of Computer and Information,Hohai University,Nanjing 211100,China

[a]duanquanfeng@163.com, [b]zhliang@hhu.edu.cn, [c]15751872027@163.coml

Keywords: WebRTC; HTML5; P2P; File Sharing; File Fragment

Abstract. WebRTC is a relatively hot technique for web in the recent years, simple Javascript API can achieve the communications of common data types even audio and video data types between browsers supporting HTML5. Moreover, it possesses the features of no plug-ins, cross platform, low development and usage cost etc. At present, vast of related applications have been proposed, and file sharing is one of the most important research issues among them.The present way to realize the web file transfer is uploading the file to the server, then browsers can download the files which results in that the server undertakes all burdens, so we need more requirements for the computational capacity and bandwidth resources; furthermore, the cooperating operations should be done after the user loads all the resources in the cooperating system which causes the long waiting time for users then affects the user experience. Aiming at the problems above, we raise a file sharing scheme with web P2P based on WebRTC and HTML5, it can reduce the burden of the server, it makes good use of upstream resource of web with transferring the file fragment which improves the speed of the transfer speed, then decrease the waiting time of users.

Introduction

File-sharing is an important function of web applications.The phenomenon that the network cloud is widely used in the era of the cloud shows us the importance of file-sharing.The traditional method by uploading and downloading files increases the burden of server side and frequently fails due to too large file size or network issues.With the development of HTML 5, a new solution depending on it has been proposed[1].If web applications can transfer file on pure browsers without any plugins by P2P just like the C/S applications,then the burden of servers can be reduced.We can not realize the goal until the new web technology of WebRTC which enables two browsers to communicate without any servers and transmitted data between two peers is encrypted by DTLS protocol to ensure the security of data.The interest towards the web P2P technology is increasing[2].Since this skill provides a new way to share files,more and more new applications have been implemented such as PeerCDN and SwarmCDN. [3] introduces a means that how to share files between two browser peers.

ShareFest is a web BitTorrent application which has been realized to enable users share files on web side.When user choose target shared file,the application will return a link which can be shared to other users.When one receiver open the link,this peer will establish a connection with the file sender.The receiver begins to receive the file data and the received file will be automatically downloaded to the local after get the whole file.Besides,if the current page keep opening,the peer can be anew file source for other peers downloading.While a new peer participates in the communication,there will be more sources endpoints to send file data so the speed of file-sharing can be improved.But in one case that vast users require the file from source endpoint at the same time such as cooperating system and no one get the whole file, the process of file-sharing is restricted by upstream bandwidth of sender.If we can use the free upstream of receivers to transmit file data,the speed of file-sharing will be increased.Based on this point,this paper introduce a design which improve web P2P file-sharing.It uses HTML 5 File API read and slice file to pieces on source peer,then distributes the slices to every peer by a certain strategy and the receiver will broadcast the pieces of file data to other peers once receive the slices.At last all peers can get the whole file data and can be merged into one file for downloading.

WebRTC and File API

The arrival of HTML5 sparks web technologies revolution and it brings us more web technologies such as web worker,local storage,canvas for WebGL,WebSocket and so on.Now various kinds of HTML5 applications are coming into our life without any plugins and provide perfect user experience.Traditionally browsers have no access to deal with local files,but now File API allow us to read files on web side with the permission of users.A File object which inheritance in Blob object is file data selected from local.A Blob object is a large binary object data and can be sliced to pieces.FileReader object can convert File data to other types such as ArrayBuffer,Base64 DataURL and Text.The API enable web developers have more privileges to deal with local files on browsers side.

Google opened the source of WebRTC in 2011,aiming at establishing P2P and real-time connection by simple JavaScript API,no need to install any plug-ins between browsers will be able to establish a connection for media and non-multimedia data transmission.Currently,IETF(Internet Engineering Task Force) and W3C(World Wide Web Consortium)jointly develop its standardization.IETF develops the Internet-based protocol WebRTC standard,which is also known as RTCWeb(Real-Time Communication in web-browsers).W3C is responsible for the development of client-side JavaScript standard API interfaces.It has been announced three API:MediaStream,RTCPeerConnection and RTCDataChannel[4],using the first two API enables audio and video real-time communication between the two browsers,the current video communication can be established through the use of three or more full connection,after two API enables transmission of text and binary data,transmission is encrypted and not through the server.

Establishing WebRTC peer connection requires a signaling mechanism,the role of signaling mechanism is to pass information both SDP and ICE candidates which peer requires.SDP is a protocol used to describe sessions and manage sessions when initiating sessions.However,relying solely on SDP for P2P transmission is not able to achieve NAT traversal[5].WebRTC using ICE framework which integrates STUN and TURN to realize this function.Collected candidates (consist of IP and port)sorted by priority will be send to peer browser using the signaling mechanism in order to establish P2P connection. WebRTC does not specify which must be chosen to send such information so SIP,XMPP,WebSocket,Socket.io and XHR can be used.Currently the most popular way is to use the WebSocket[6] that provides practical simple API[6],achieving the function that the server can push message to browsers and it has smaller header for transmission of data than HTTP protocol.

Using RTCPeerConnection API can achieve efficient and stable data stream,its API for web developers shielding complexities of the underlying protocols and coding method,which controls the whole process of establishing JavaScript Session Establishment Protocol(JSEP)[7] by using Offer/Answer mechanism.Once the negotiation that choose a candidate pair is completed the data channel can be created.In case how to set up a data channel between A browser and B browser as follows:

①Each of peers creates a RTCPeerConnection object,then we will get peerA and peerB.

②PeerA call its createOffer method to get its SDP,then using setLocalDescription to set local SDP.At the same time peerA sends SDP to peerB by signaling server.

③When peerB receive SDP from peerA, it will set it as a remote SDP by using setRemoteDescription method.PeerBget its SDP by calling createAnswer which will be set as local SDP of peerB and will be sent to peerA by the same signaling server.

④Once peerA get the SDP from peerB,peerA will set it as remote description.

⑤When both sides onicecandidate called and finishing collecting candidates information ,they will get candidates and add candidates to local peer by addIceCadidate method.

⑥PeerA uses createDataChannel method to create a RTCDataChannel object,the data channel connection will be established after the completion of peerB triggers onDataChannel event.

A data channel can be created between the two browsers through the above process,the process of SDP information exchange is shown in Fig.1.

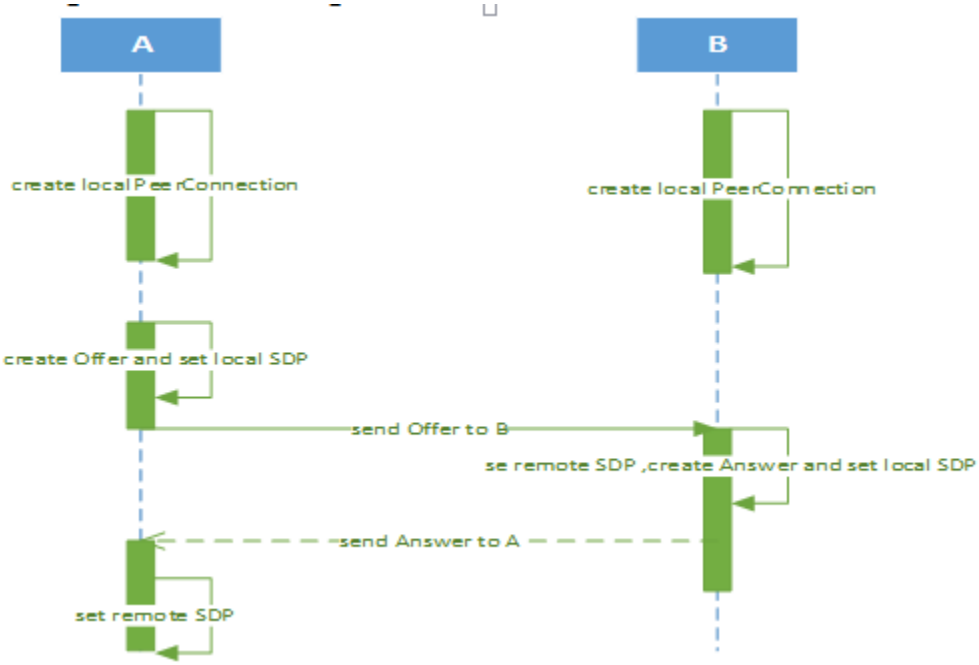

Fig.1 A、B exchange SDP

Implementation

The running environment of the application designed in this article is browser that supports HTML5 and WebRTC,the underlying implementation is using WebRTC to establish the data channel ,HTML 5 File API and the browser’s build-in functions.The whole structure can be divided into two parts:the first part is the web page display,using <input type=” file” > element to allow user to select local file wanted to be shared and other elements for web page display;Secondly,on one hand using JavaScript slice the selected file to fragmentation and convert file data format,on the other hand establishing a full data channel connection among all peers.Specific procedures are as follows:

①Reading the selected file and slicing the file to pieces.Get File object by JavaScript API,using FileReader object converting type of file.If the file is small enough the can be sent directly by File or ArrayBuffer data type.Otherwise,chosen file need to be sliced to pieces by File API and sent to peers.Besides,limited to the use of browsers that have different max size limit to transferred data (such as Chrome allows a maximum of 64K,Firefox allows maximum 16K),recommended that each fragment size should not exceed 16K.Typically the transferred data will be convert to more convenient Base64 code,besides,the data converted to Base64 code will become 4/3 times the original,and therefore a corresponding reduction in the size of the file chunk.Then construct block number chunk data,block number and other file meta data into a JSON format and sent to peer node by data channel.

② To establish a full connection based on the second chapter describes the process of establishing a data channel.Full connection means that each pair of nodes need to establish connection,so if there are n nodes ,there will be n*(n-1)/2 connections to be established.Typically to establish full connection uses such a strategy:Suppose connection between A and B has been completed by the above manner,the third node C apply to join.C sends a connection request to A and finish the connection,A send the notice all of its peers to connect C,so connection between B and C can be achieved and then a full connection of three peers has been implemented.Fig.2 shows how node D join in the connection,through repeat the above procedure can establish communication path with more connected nodes.

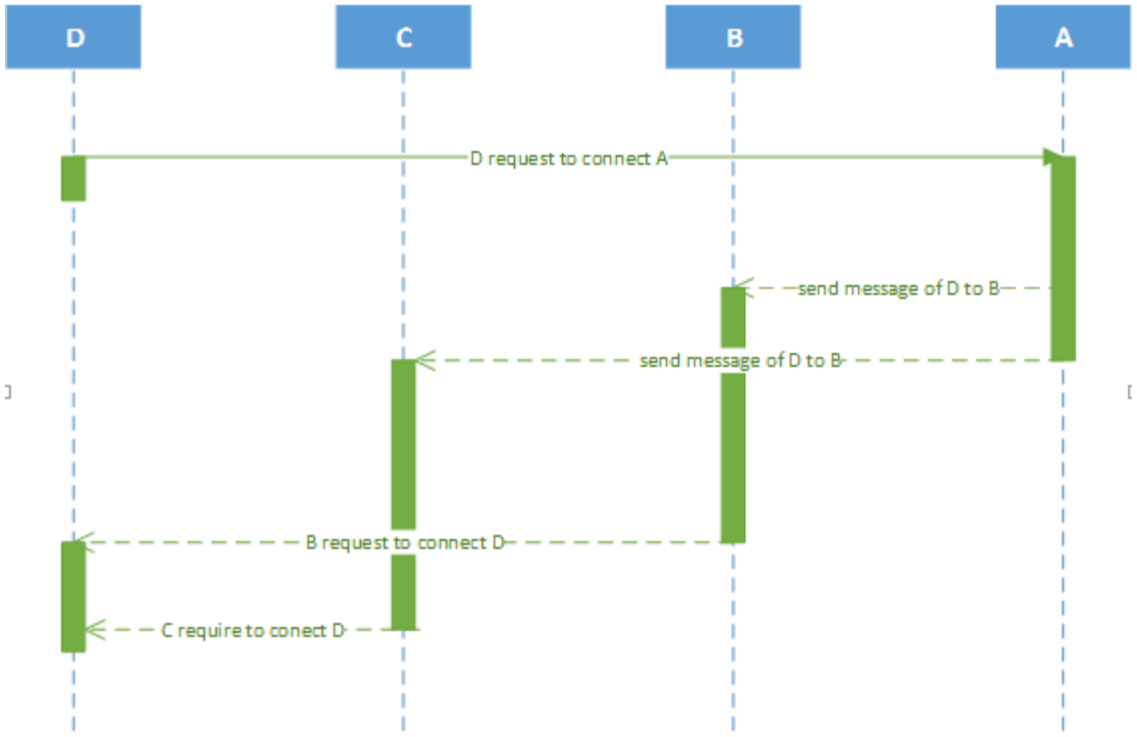

Fig.2 D joins the connection

③Slice file and send file fragmentation.On the sender side,configure every file chunk as a JSON object described in step①.The file source node sequentially send all data objects to each peer through the data channel,the other receives the data stored in the local cache and broadcasts the data to all other nodes except the source nodes via various channels.File source node will notice all nodes when finish the last sending chunk.Other nodes will also notice its peers the end sending of itself.So one completes receiving can be determined where it has finished all end message form its peers.Suppose there are five nodes,A,B,C,D and E,E shares its file which is divided into 8 chunks.E sends 1-4 chunks to the four nodes in accordance with the order and then send the 5-8 chunks in the same way.Other four nodes a data fragment and will share to the other three nodes.So A shares 1st and 5th file chunks,B shares 2nd and 6th file chunks and so on,each node wiil eventually receive all the data fragments.

④Check and merge file data.If lack of file fragments is found,the node will require data from its peers even source file peer.Then begin to merge all the file chunks,after that a local URL that is created based on the file data should be bind to the page element for downloading.Note that ,Base64 code URL is limited to the size,it should be converted to Blob type of URL.

Conclusions

While WebRTC standard has not been complete nor stable,browser vendors do not provide a unified manner to achieve, but it appears that some of the current experimental applications demonstrate that it has a powerful data transmission,its browser-based operating environment achieve a cross platform and cross device,the future of the technology will have broad application space on web development.

This article describes the file-sharing scheme,as apposed to the traditional distribution that reduces the burden of server side;with respect to the introduction of multi-file source of sharing,the application designed in this paper slices file to pieces on file source side and then send to different peers,each peer sharing received fragments,in this way breaking the single-peer upstream speed limit and full using the upstream bandwidth of each peer.This can improve overall throughput and the speed of file-sharing.Each peer is almost complete file receiving at the same time,this case that multi-user request to download the same file can reduce overall waiting time.

Acknowledgements

I would like to express my sincere thanks to a lot of people.Without their assistance,the accomplishment of this thesis would have been impossible.First of all,I would like to take this opportunity to show my sincere gratitude to my supervisor,Mr Liang for his instructive advice and useful suggestions on my thesis.Secondly,I'd like to express my gratitude to my classmate.She proposes many suggestions and tells me many professional terms in English.Last but not least,I am indebted to those leaders, teachers and working staff for their review to my thesis.

References

[1] Xiaojin C. Research on file upload based on HTML5[C]//Service Systems and Service Management (ICSSSM), 2014 11th International Conference on. IEEE, 2014: 1-3.

[2] Xianghui Zhang,Jiaqing Huang,Kangheng Wu,et al.The review of real-time audio and video communication based on WebRTC in Chinese[J].Computer Science,2015, 42(2): 1-6.

[3] LanaL.WebRTCFileShareApplication.https://www.software.intel.com/en-us/android/articles/

webrtcfilesharing-application

[4] Jennings C, Narayanan A, Bergkvist A, et al. WebRTC 1.0: Real-time Communication Between Browsers[J]. W3C, W3C Working Draft, Feb, 2015.

[5] Ford, B., Srisuresh, P., & Kegel, D. (2004). Peer-to-peer (P2P) communication across middleboxes. Oct. 2003. Internet draft draft-ford-midcom-p2p-01. txt .

[6] Wang V, Salim F, Moskovits P. The WebSocket API[M]//The Definitive Guide to HTML5 WebSocket. Apress, 2013: 13-32.

[7] Jennings C, Uberti J, Rescorla E. Javascript Session Establishment Protocol[J]. 2014.

[8] Rosenberg J, Schulzrinne H. An offer/answer model with session description protocol (SDP)[J]. 2002.