

Peer Assisted Web Content Delivery

Kushagr Gupta

Student

Atharva College of Engineering
Malad, Mumbai, India

Aanchal Lohia

Student

Atharva College of Engineering
Malad, Mumbai, India

Dhairav Mehta

Student

Atharva College of Engineering
Malad, Mumbai, India

Sanjay Chauhan

Student

Atharva College of Engineering
Malad, Mumbai, India

Nileema Pathak

Head of Department IT

Atharva College of Engineering
Malad, Mumbai, India

Abstract— In today's world, HTTP driven Internet traffic forms the backbone of information and content transfer around the globe. Content Delivery Networks(CDN) offer these systems the service of distributing this content wherever needed. However, these systems are often plagued with content delivery failures due to various constrains of CDN such as edge server failure/congestion. We therefore explore a browser based peer-assisted design to resolve content delivery failures. WebRTC is a novel way to deliver web content, but has the added complexity of handling dynamic resources and handling them efficiently. A fall back mechanism can be formulated to handle server and network failures by enforcing the protocols to support efficient and optimized content delivery. HTTP based services are currently used to deliver various rich media which has caused increase in the bandwidth load on the CDN and it is usually the cause of content delivery failures. The browser based peer-assisted scheme aims to reduce this load by leveraging the bandwidth of connected users to serve content to newer users. This method needs optimization to ensure that the most appropriate peer is selected. It suffers from problems that the conventional P2P method does not, such as peer disconnection as soon as page is loaded.

Keywords— Peer; rich-media; peer-assisted; web-torrent

I. INTRODUCTION

The exponential increase in internet adoption has led to a massive market of internet enabled services accessible over the web browser, which over time have incorporated rich media into their services. These services traditionally rely on CDNs to deliver rich, bandwidth consuming content such as video, audio and images to users. This method has become insufficient by itself due to performance issues of CDNs. In times of heavy user interaction, these CDNs have high latency and even sometimes fail to serve content to users. The edge servers of CDNs have only a limited amount of outward bandwidth capability which restricts the number of users they can serve.

We propose a peer-assisted web content delivery system, which utilizes technology supported by major web browsers to support existing content delivery systems. The system will use WebRTC protocol to create browser to browser connections and allow a mesh of users to exchange media data among themselves, which will ensure all users receive said data with reduced load on the CDN.

II. LITERATURE SURVEY

The digital revolution brings significant development and widespread use of rich media resources in Enterprise Media Network, Digital Marketing and others. Earlier work in the field of content delivery has been involved with utilizing data replication centers in combination with WebRTC protocol to protect against network failures [1].

With the increased emphasis on high quality media requirements to serve high resolution streaming devices, there is a need to provide this data uninterrupted. A reliable network of a peered architecture can be used to effectively utilize bandwidth along with high speed access using WebRTC.

Earlier efforts [8] involve dynamic load-balancing on a large-scale distributed computing environment. The scheduling processes are performed by independent distributed schedulers on individual computing resources. There exists a master that regulates the amount of jobs among workers. Workers build an overlay network among them with P2P communication, in order to quantify the impact of workers for scalability and fault tolerance.

The paper [1] proposes certain factors for choosing relay peers for a requesting user: (1) Diversity of peers' locations. (2) Diversity of peers' network ISP. (3) Diversity of peers workload and time-to-stay. These provide a structured system to interconnect all users in the minimum geographic radius in order to provide high numbers of peers to enrich the user experience. The effectiveness of their design can be compared with non-peer-assisted strategy and random peer selection strategy, their design significantly improves the successful relay ratio of web content items under network failures, e.g., their design improves the content download ratio up to 60% even when users located in a particular region (e.g., city) where none can connect to the regional CDN server.

Implementing advanced functions such as direct communications among users and file sharing, can be handled by Web browsers using JavaScript library such as PeerJS to implement P2P communication among Web browsers with WebRTC easily. In PeerJS, each client gets a PeerID to identify the client by the mediation server[2]. It mainly caters to volunteer computing which has yet to handle erroneous results from malicious workers

The paper [3] focuses on harvesting the use WebRTC protocol in regards to live video streaming. It realizes the diversity and capacity of web browsers and the ease of access of web based applications to maintain connections. It uses the concept of a WebRTC coordinator to maintain the network in which clients support the source publisher in content distribution. In their implementation, the WebRTC coordinator and source publisher are the same host. Since WebRTC is built into browsers as an API and JavaScript needs to be used to call functions of API. The most suitable function for their implementation was the RTCDataChannel interface, because it allows sending arbitrary data. To identify and register users it uses the JavaScript Library: PeerJS [4]. The limitations of PeerJS library is the range of browsers who support it. Thus there needs to be a universal mechanism to connect to peer clients across browser barriers. Against all restrictions they have successfully implemented a live streaming architecture based on the protocols of WebRTC.

There have also been some projects[5] which address the challenges of browser based peer to peer content sharing. It exchange messages asynchronously and facilitates new peer joins via existing peers in the network, thus reducing the dependency on bootstrap server.

Increased popularity and support of WebRTC protocols by major browsers motivates its usage for curbing the problem of rich content delivery.

III. METHODOLOGY

The central web server is the machine that serves the webpage to the user. The webpage is in the form of a HTML file, with the image and video tags intact. The webserver also acts as the primary location of the multimedia assets. It serves the content as a fallback in case the peer network is not feasible for serving the content. The client's browser acts as the primary actor in this methodology. On the loading of the DOM the multimedia content is converted into custom element and magnet link of the content is requested from the tracking server. A magnet link acts as both the identifier of the data being requested and the validity check of the data received as it contains the hash code of the data. This magnet link is then used to load the content using the web enabled bit-torrent protocol called web torrent using its browser based bit-torrent client. The tracking server supplies the individual webpages with the magnet links of the content. It also maintains a list of the known connected peers, refreshing them time to time. The individual requests are also supplied with a list of peers based on the user's logical location. The user's location is found out at the tracking server using IP Address. The location of the user is used to link peers from the same ISP to each other, as in most cases data within ISP limits is not

rate limited hence higher connection speeds, and data is zero-rated hence doesn't cost the individual peers money to send data to others in mesh.

IV. IMPLEMENTATION

The following steps list down the implementation plan of the proposed system, which can be seen in Fig. 1.

A. User – Web Server Interaction

The user will request the webpage from the webserver, which is the primary server of the service, run by the service provider and can be considered as the location of the content being served. The webserver, on receiving the request returns the code of the website to the user's web browser software. The websites code will contain special tags to signify the presence of rich media that will be served using the peering method. The code will also contain links to the peering and web torrent client JavaScript code which can be referenced without adding them to website code.

B. Browser-Tracking Server Interaction and DOM Manipulation

When the website code has been downloaded into the browser, a script removes all the links to rich media and replaces them with placeholders. This is done on Document Object Model (DOM) loading, before the rich media is requested from the location mentioned in the website code. The links of the rich media are then sent to the tracking server to receive the alternate 'magnet' links of the data. The tracking server stores the various website links and their associated magnet link. It sends back the magnet links requested by the browser. Magnet links have two uses; they are used to fetch and connect to a list of peers who have the same data, and used to check integrity of the data being received.

C. User Indexing and Peer Connection

The users of the peering service are indexed and information like their IP address, ASN number are used to link the correct peers to each other. The users from the same ISP are linked to each other first if possible, then priority is given to users in the same geographical region and then the connection is extended to all the peers.

This is done to increase the reliability of the content delivery. The connection speed and latency are two aspects that need to be considered for reliability of the delivery. The nearer the two peers are to each other, the higher the connection speed and lower the latency. This enables the mesh to stabilize faster on addition of new users, i.e new peer receiving the content.

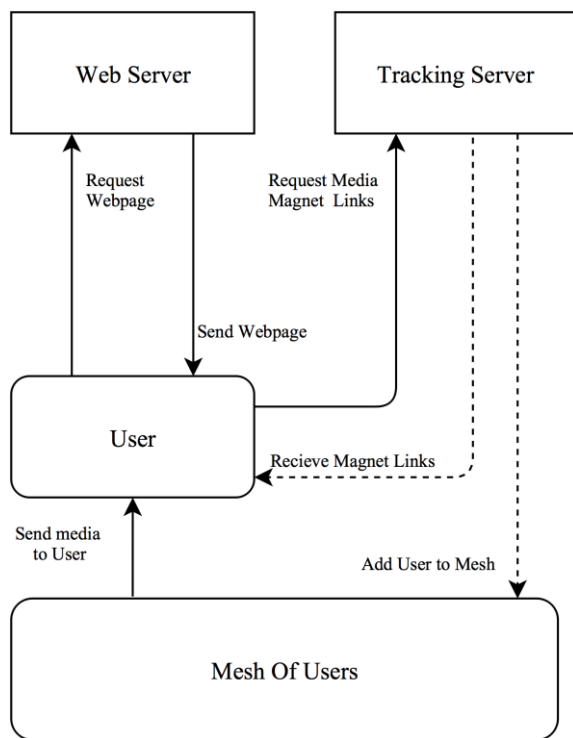


Figure 1: Proposed System Diagram

D. Content Delivery

The magnet link is added to the browsers web torrent client. The torrent client contacts the peers provided by the tracking server. The users in the mesh send the data to the user, and the client reconstructs the data using the blacks sent by the peers. The data can be sent in a streaming fashion, which allows videos to be sent to user for streaming.

V. CONCLUSION

The Internet media has consolidated itself as a very powerful globalized dimension, changing the face of business, and communication. This increases the load on traditional servers and content delivery systems. There is a need to accomplish a structure that can be easily accessible with optimal resources. Existing systems do provide a solution to tackle heavy data transfer but it is difficult to prepare for random surges in user interaction. Through this system we aim to devise a method that allows for the offloading of bandwidth to the mesh of users of a website, reducing the bandwidth costs as well as increasing the serviceable user counts of a single node. This also allows for higher bit rate videos being more accessible and providing an uninterrupted user experience.

REFERENCES

- [1] W. Hu, Z. Wang and L. Sun, "Path-aware peer-assisted web content delivery against network failures," 2015 IEEE 23rd International Symposium on Quality of Service (IWQoS), Portland, OR, 2015, pp. 79-80.
- [2] M. Kuhara, N. Amano, K. Watanabe, Y. Nogami and M. Fukushi, "A peer-to-peer communication function among Web browsers for Web-based Volunteer Computing," Communications and Information Technologies (ISCIT), 2014 14th International Symposium on, Incheon, 2014, pp. 383-387.
- [3] F. Rhinow, P. P. Veloso, C. Puyelo, S. Barrett and E. O. Nuallain, "P2P live video streaming in WebRTC," Computer Applications and Information Systems (WCCAIS), 2014 World Congress on, Hammamet, 2014, pp. 1-6.
- [4] PeerJS: <http://peerjs.com/> [date_of_visit: 27-12-16]
- [5] S. Vashishth, Y. Sinha and K. H. Babu, "Addressing Challenges in Browser Based P2P Content Sharing Framework Using WebRTC," 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA), Crans-Montana, 2016, pp. 850-857.
- [6] Node.js: <http://nodejs.org> [date_of_visit: 28-12-16]
- [7] WebRTC: <https://webrtc.org/> [date_of_visit: 27-12-16]
- [8] Y. Murata, T. Inaba, H. Takizawa and H. Kobayashi, "A distributed and cooperative load balancing mechanism for large-scale P2P systems," International Symposium on Applications and the Internet Workshops (SAINTW'06), Phoenix, AZ, 2006, pp. 4 pp.-129.
- [9] M. Zhang, Y. Xiong, Q. Zhang, L. Sun, S. Yang, "Optimizing the throughput of data-driven peer-to-peer streaming", IEEE TPDS, vol. 20, 2009, pp: 97 – 110
- [10] Webtorrent : <https://webtorrent.io/> [date_of_visit: 27-12-16]