Conversas Hiperligadas: Novo Paradigma de Comunicação e Colaboração, potenciado pela Tecnologia WebRTC

Henrique Lopes Rocha
email1: henrique.rocha@ist.utl.pt

Instituto Superior Técnico

Abstract. In here put your abstract. Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem.

Keywords: keyworkd1, keyworkd2, keyworkd3

1 Introduction

The need to build a global comunications network in an era when almost nobody had access to it, caused that some protocols weren't suitable for a huge increase on the amount of publicly known users. Internet Protocol Version 4 (IPv4) limits the number of public addresses in such a way that today is scarse [5]. One way to overcome this problem was the development of a mechanism that groups multiple address into a single one, the machine that is assigned that address is then responsible to redirect messages to members of its group through their private addresses, each member of the private network is identified publicly by the same Internet Protocol (IP) address but different port, this technique is also known as Network Address Translation (NAT).

Initially NAT offered an alternative for address exhaustion and a minimal sensation of security. There are four types of NAT, "Full Cone NAT", "Restricted Cone NAT", "Port Restricted Cone NAT", "Symmetric NAT".

Full Cone NAT maps public each IP address and port to a private IP address and port, any external host can communicate with private hosts throught their mapped public address and port. This represents the least restrictive type of NAT and as we will later, the unique type of NAT that enables real time communications from point to point.

Restricted Cone NAT requires that a private client must first send a message to an external host before it can receive messages from the same host. With this type of NAT, the private client can be contacted from any port of the same external host.

Port Restricted Cone NAT works in the same way as Restricted Cone NAT but only allows communications from the same external host's IP address and port, ignoring all messages from other applications within the same external host.

Symmetric NAT maps different ports for each connection, as we will see later, this represents a problem on real time communications.

Asymmetric NAT became a vulgar configuration on the web. As a direct result, problems started to appear, the amount of ports that IP disponibilizes is also small compared to our current needs, worse than that, NAT also difficult end-to-end communication, forcing most of applications that follows this model to be implemented ineffectively.

Applications based on multimedia and file sharing were one of the most strained by NAT. Those kind applications require real time communication in order to achieve the best performance. Session Traversal Utilities for NAT (STUN) and Traversal Using Relays around NAT (TURN) [6] servers are a possible solution to overpass NAT, although, none of those can establish direct connections on multiple level NATs.

STUN servers are quite simple, they receive requests from NATed clients, the source address of a request is the public address that NAT mapped to the client, STUN servers will reply the mapped public address to the client, so it knows its public IP address and port. Because Symmetric NAT changes IP port for each different connection, STUN servers will reply the IP address and port of their connection, which will be useless to clients connections, that's why Symmetric NAT represents a problem for real time communications.

On the other hand, TURN uses public servers to redirect traffic between private endpoints, it may use a P2P network relay to find the best peer but, after that, the behavior is much like client-server. Direct communication is only achieved by STUN when NAT is a type *full cone*. Interactive Connectivity Establishment (ICE) uses STUN when it's possible and TURN otherwise.

Most of client-server applications aren't affected by NAT when the servers are public, but they're inadequate for real time communication between two private endpoints. Clearly this type of communication requires a more expensive infrastructure and, in most cases, more network usage, leading to a worse quality of service. The requirements of video communication makes this kind of model unsuitable.

When connection is established, either in a direct or indirect way (via TURN servers), Web Real-Time Communication (WebRTC) cames to simplify how audio and video are transmited.

Skype is a stand-alone application which is compatible with all major operating systems (Windows, Linux, MacOS) and mobile operating systems (Android, iOS, Windows). This enables voice, IM and video communication. It's main advantage is that it has a huge installation base (approx 299 million users).

Compared to Skype, WebRTC is an open source technology that allows web browser real time communications without installing any aditional application or plugin.

2 Signaling

Signaling is the most important proccess for applications to exchange connection information about peers and servers, their capabilities and metadata.

WebRTC doesn't implement signaling, different applications may need different protocols, there is no single answer that fits all problems. Amongst multiple options, signaling can be done by using Session Initiation Protocol (SIP), Extensible Messaging and Presence Protocol (XMPP), WebSockets, Socket.io or by implementing a custom protocol.

WebRTC uses Session Description Protocol (SDP) to define peer connection properties, types of supported media, codecs, protocols used and network information.

One of signaling requisites is bi-directional communication over Hypertext Transfer Protocol (HTTP). HTTP works on a request followed by a server response, by other words, follows a unidirectional communication. Sometimes it's required that some informations are obtained in real time, as we saw, some NAT's don't suport callbacks from servers, one technique to overcome this problem is polling.

Polling consists on sending periodic messages that the server responds imediately with empty content or fresh information. Because real time communications are unpredictable, if the time between periodic requests is short, most of time the server will return empty results wasting network bandwith and energy. On the other hand, if the time between periodic requests is large, newer messages may arrive later.

A technique called long polling consists on making the server hold the request until there is fresh information or expiring after some time, after the receival, the client makes another request. This results on a better network usage and a faster server response, but both simple polling and long polling requests are sent with HTTP headers, which adds data overhead, specialy for small messages.

The WebSocket protocol (rfc6455[1]) provides bidirectional communications trough a full-duplex socket channel. WebSocket handshake phase specifies an HTTP header in order to upgrade to websocket type of communication, the remainder messages are done without HTTP headers, which leads to much smaller messages and better network usage. WebSockets may not be avalailable on every web browser, frameworks like *socket.io*[2] and *SockJS*[3] uses HTTP when there is no support for WebSockets.

[1] `http://tools.ietf.org/html/rfc6455`
[2] `http://socket.io/`
[3] `http://github.com/sockjs`

Bidirectional-streams Over Synchronous HTTP (BOSH) is a technique based on long polling, that uses two socket connections and allow sending client messages to server while a previous request is holded.

SIP (rfc3261[4]) is protocol used for negotiation, creation, modification and finalization of communication sessions between users. SIP follows a client/server architecture with HTTP like messages and it can be used as signaling protocol. The advantage of SIP is the ability to make video and voice calls applications over IP networks.

The work group SIP for Instant Messaging and Presence Leveraging Extensions (SIMPLE) proposed the creation of SIP extensions, namely presence information (rfc5263[5]) and instant messaging (rfc3428[6]).

SIP is used in Voice Over IP (VoIP) applications due to its compatibility with Public Switched Telephone Network (PSTN). Service providers are disponibilizing access to their SIP infrastrucures through WebSockets. Frameworks like *jsSIP*[7], *QoffeeSIP*[8] and *sipML5*[9] are used on client side to parse and encode SIP messages, making SIP accessible to web based applications.

SIP with WebSockets can be used as a signaling method for WebRTC applications, it allows web browsers to have audio, video and Short Message Service (SMS) capabilities like mobile phones. For instance, it's possible to interoperate web communications with SIP networks, mobile and fixed phones.

XMPP was initially developed for instant messaging (Jabber[10]). It is nowadays an open technology for real-time communications.

XMPP messages are eXtensible Markup Language (XML) based, which are attractive for applications that need structured messages and rich hypermedia. XMPP advantage is the addition of extensions, for example XEP-0096[11], which adds file transfer capabilities between two entities and XEP-0045[12] which enables multi-user chat.

XMPP's bi-directional communication is achieved through BOSH (XEP-0206[13]), which basically consists on long polling. This kind of communication is also possible through WebSockets (rfc7395[14]).

[4] `http://tools.ietf.org/html/rfc3261`
[5] `http://tools.ietf.org/html/rfc5263`
[6] `http://tools.ietf.org/html/rfc3428`
[7] `http://jssip.net/`
[8] `http://qoffeesip.quobis.com/`
[9] `http://sipml5.org/`
[10] `http://jabber.org/`
[11] `http://xmpp.org/extensions/xep-0096.html`
[12] `http://xmpp.org/extensions/xep-0045.html`
[13] `http://xmpp.org/extensions/xep-0206.html`
[14] `http://tools.ietf.org/html/rfc7395`

Amongst multiple XMPP servers softwares are: ejabberd[15], Metronome[16], Openfire[17] and Prosody[18]. Ejabberd is the one that implements more Request For Comments (RFC) specifications and XMPP Extensions (XEP)s[19].

3 Hypermedia

Since the early days of video technology, one of the problems that raised with it consisted of how to add more information onto it without generating multiple versions. Some implementations like [4] added hypermedia information to empty space present on Moving Picture Experts Group (MPEG) frames in order to provide interactive television, the MPEG coder and decoder were changed in order to handle hypermedia content.

The need to translate movies, raised the problem whether it is apropriate to change the original video or audio. For example subtitles should be an entity independent from the video, in order to be personalized or replaced easily.

Amongst multiple formats for subtitles, Synchronized Accessible Media Interchange (SAMI), and SubRip Text (SRT) are used by video players that support them. Although those formats have styling available, they are quite limited to text.

Hypervideo is a kind of video that contains links to any kind of hypermedia, including links to skip part of it. An example of hypermedia application could be a search engine over hypermedia content, like subtitles, in order to jump to a specific point in time. HyperCafe [7] was an experimental project to expose hypervideo concepts that consisted on an interactive film by switching between conversations inside a cafe.

Detail-on-demand is a subset o hypervideo that allow us to obtain aditional information about something that apears along the video, like obtaining information about a painting that appears in a particular segment. Hyper-Hitchcock[8] is an editor and player of detail-on-demand video.

In order to navigate through a dynamic video, one must be aware of time synchronization and the multiple time flows, it's important that all time, causality and behavior rules are well defined.

HyVAL[10] is an XML based language that was proposed for modeling composition, synchronization and interaction of video, the structure of HyVAL is derived from traditional video, which divides video into segments, scenes, shots and frames hierarchically. This approach is quite limitative if we want to apply hypervideo concepts to videos that don't follow this structure.

Synchronized Multimedia Integration Language (SMIL)[1] was introduced to describe temporal behavior of multimedia, for instance, it could be used to overlay subtitles on films. With SMIL it's possible to synchronize multiple sections of

[15] http://jabberd.im/
[16] http://lightwitch.org/metronome
[17] http://igniterealtime.org/projects/openfire/
[18] http://prosody.im/
[19] http://en.wikipedia.org/wiki/Comparison_of_XMPP_server_software

video, either in parallel or in sequence, reproduce a different audio track, overlay user interface elements with hyperlinks, amongst multiple other functionalities.

In order to create a multimedia rich hypercall, SMIL fits our goals, but it lacks on browser compatibility. Ambulant [2] was one of the SMIL players that were developed for browsers, although this player implements most of SMIL 3.0 [9] specifications, it needs to be installed on browsers as a plugin.

SmillingWeb [3] attempts to implement SMIL 3.0 with javascript and jQuery, which doesn't need to be installed and shouldn't have incompatibility issues. But SmillingWeb isn't fully implemented yet and their scheduler engine loads the SMIL file only once, which could raise problems when leading with SMIL changes in real time.

Scalable Vector Graphics (SVG) is a format that incorporates the animation feature of SMIL. Currently SVG allow us to add movement and animate attributes of elements. When embedded on HyperText Markup Language (HTML), it allows dynamic changes to inner content in real time, besides that, it also allows to call javascript functions on events such as animation end, mouse over and mouse click.

Video functionalities are already embedded in HTML5, like SVG it is also possible to bind javascript functions for different kinds of events over videos.

By using techonologies that relies only on web standards, like Cascading Style Sheets (CSS), HTML5, Javascript and SVG, it's possible to raise communications to a new level. For example, with Application Programming Interface (API)s like WebGL[20], it is now possible to manipulate a three dimensional environment in the context of a hypercall. Another example would be a collaborative spreadsheet using WebRTC. With this, hypercalls are not limited to only audio, image, text and video, but also interaction with complex graphical user interfaces that changes over time.

In this project our goal is to enrich hypercalls with no limits, every user should be free to choose how it wants to be contacted and it wants to share its contents.

In order to give users a personalized communication channel, each user must have a personal web page where its available plugins could be downloaded from other peers, after that they can talk in the same language whatever it is.

4 Applications

Real time collaboration applications have become a huge help on team tasks, providing a great boost on business, research and investigation velocity. Technologies like this are appearing along this days, but they couln't be possible years ago because technology was limited or unavailable. Although todays technology is limited on some aspects, we are doing progress in order to improve the web ecosystem, by creating standards and migrating to newer technologies.

Our first concern on real time collaboration applications, besides the communication itself, is the data storage and representation. Because most browsers

[20] `http://khronos.org/webgl/`

are recommended to limit local storage to at least five megabytes per origin, storing multimedia content is not a viable solution.

If, for instance, one whants to rewind a real time video, recordings will be needed from who is streaming the video.

[Video Recording]

[Video Playback - Past and Present]

4.1 Context

4.2 Problem Statement / Solution Statement

4.3 Thesis Contributions

4.4 Article Structure

4.5 Methodology

4.6 Planned Schedule

5 Conclusions

5.1 Summary

6 Conclusions

References

1. Bulterman, D.C.: Smil 2.0 - part 1: Overview, concepts, and structure (2001)
2. Bulterman, D.C., Jansen, A., Cesar, P.: Video on the web: Experiences from smil and from the ambulant annotator (2007)
3. Gaggi, O., Danese, L.: A smil player for any web browser (2011)
4. Gradvohl, A.L.S., Iano, Y.: Combinando tv interativa e hipervideo (2007)
5. IEEE-USA: Next generation internet: Ipv4 address exhaustion, mitigation strategies and implications for the u.s. (2009)
6. Lin, Y.D., Tseng, C.C., Ho, C.Y., Wu, Y.H.: How nat-compatible are voip applications? (2010)
7. Sawhney, N., Balcom, D., Smith, I.: Hypercafe: Narrative and aesthetic properties of hypervideo (1997)
8. Shipman, F., Girgensohn, A., Wilcox, L.: Creating navigable multi-level video summaries (2003)
9. W3C: Smil 3.0 (2008), `http://www.w3.org/TR/SMIL3/`
10. Zhou, T.T., Jin, J.S.: A structured document model for authoring video-based hypermedia (2005)