IPTV using P2PSP and HTML5+WebRTC

Cristóbal Medina-López,
Juan Pablo García Ortiz and
J. A. M. Naranjo
Luxunda S.L.

L. G. Casado and
Vicente González-Ruiz
Department of Informatics,
University of Almeria*

February 3, 2014

Abstract

This work presents a novel implementation of the P2PSP (Peer-to-Peer Straightforward Protocol) over the WebRTC/HTML5 framework, resulting a Web client that can be used to retrieve in real-time multimedia content from a streaming server. This solution, compared to non P2P-based ones, minimizes the bandwidth comsumption at the server side, provides a good quality of service and avoids the need to download plugins or third party software.

Keywords: P2PSP, IPTV, WebRTC, HTML5

1 Introduction

Massive distribution of real-time video content is one of the big challenges in the Internet. Nowadays, there are several proposals that approximate to this goal, but none of them provide a QoS (Quality of Service) comparable to the DVB (Digital Video Broadcasting). This is a direct consequence of the design of the Internet and an unefficient use of the capacity of the network. The Internet was created to provide the so called best-effort service that basically means that you can transmitt data through the network but the transmission time is unknown a priori. That time depends on several factors such as network failures, the network load and, obviously, the amount of sent data. These two last factors are directly related to the capacity of the network, a term also refered as (network) bandwidth.

In order to provide an acceptable QoS in real-time streaming scenarios one of the most important requirements to fulfill is to have a lower bound of the network capacity. However, most of the current solutions fail to take advantage of this resource. One of the reasons is that IP multicast has not achieved the expected popularity. This forces content sources to replicate the same data for different receivers, resulting in a linear growth of the transmmited data when the number of receivers increases. In this situation, P2P (Peer-to-Peer) overlays can improve the performance of the real-time streaming services: given that all peers manage (and thus can share) the same content, the trasmission requirements at the source are drammatically reduced.

There are a number of proposals for P2P video streaming. Of these, the PPSP [4] stands for two reasons: (1) it is a standard (though currently only available as a draft) and (2) it can be used to implement both IPTV and VoD (Video on Demand) systems. However, we propose to use an alternative called P2PSP. As it will explained in the next section, the P2PSP is quite functional and very easy to implement [7].

Consequently, this document introduces P2PSP (Peer-to-Peer Straightforward Protocol) and discusses how it takes advantage of HTML 5 and WebRTC. P2PSP is a set of transmission and machine behavior rules that helps increasing the QoS of real-time streaming systems. Basically, the P2PSP mimics the IP multicast solution by taking advantage of the transmission capacity of the peers, which is typically wasted in pure C/S (Client/Server) services. We would like to stress that, although the P2PSP can be an standalone solution for small scenarios, its performance can be increased in massive scenarios by taking advantage of both paradigms: the C/S model and the P2P model.

2 HTML5 and WebRTC

Today, the Web browser is an indispensable tool to access the information available on the Internet and HTML is the language used to described the Web objects that the browser interprets and displays to the users. One of the most interesting additions in the version 5 of the HTML is the `<video>` label which enables the playback of a video embedded in the Web page. This possibility has opened a range of multimedia possibilities and avoids the need of third party software to handle multimedia.

*This work is partially supported by grant TIN2012-37483 from the Spanish Ministry.

On the other hand, the W3C and the IETF are currently engaged in a process of standardization of WebRTC API [9]. Through this API two HTML5 [10] applications (running on two different Web browsers) can communicate with each other without the need for any intermediate server. Thanks to this, if a user A is playing a video in his browser and other user B wants to also play the same video in his browser (as happens for example in the TV), then user A can serve video to user B, thus reducing server load at the video source. This is of great interest to improve the services offered by P2PSP.

3 The Peer-To-Peer Straightforward Protocol

P2PSP [5, 8] is a application layer protocol for real-time streaming of multimedia content over the Internet, i.e., users playing the same stream in a synchronized way (all peers follow the same playback point). It can be used to build a variety of live-streaming services that ranges from small hangouts to large IPTV (Internet Protocol TV) systems.

3.1 Main P2PSP characteristics

These are some of the P2PSP features:

- P2PSP is not aware of the broadcasted content, the bit-rate, the format, etc. Any type of stream can be transmitted without having to modify the protocol at all.
- At least one working implementation of P2PSP can be found in Launchpad [7]. It can be used/modified/expanded for free as long as the GNU GENERAL PUBLIC LICENSE [2] guidelines are followed.
- P2PSP has a modular architecture. The number of modules used depends on the final requirements.
- The most basic module is simple enough to run the peer process in systems with very low computing resources. The rest of modules add functionality to the protocol, such as connectivity across NATs, parallel streaming, data integrity and information privacy.
- If native IP multicast is available (even locally, as it happens in most of the local area networks), the P2PSP can use it.
- The P2PSP facilitates the use of error concealment techniques in the received stream because lost packets are spreaded along the time.
- Peers can be hosted in private networks, even if they are placed behind symmetric NATs.
- The protocol is fully compatible with multiresolution and bandwidth-adaptive streaming services. Simulcast [1], scalable video coding [6] and multiple description video coding based solutions [3] are possible.
- P2PSP has been conceived for P2P real-time streaming services but it can be used to deploy hybrid C/S-P2PSP systems.

3.2 A modular design

The protocol is organized in six modules, each of them defined by a set of rules that provide a different functionality. Only one of the sets (DBS) is compulsory to implement in order to create a P2PSP team. The rest are optional and, in general, there is no a dependency between them. This means that almost any combination of sets of rules is possible.

1. **Data Broadcasting Set (DBS) of rules:** This set of rules implements the most basic behaviour of the protocol and it has been designed to be efficient in transmitting a data-stream from a source node to peers in the network.
2. **Adaptive Chunk-rate Set (ACS) of rules:** The DBS forces all peers in a team to provide the same uploading capacity. This module relaxes this requirement by transferring upload bandwidth requirements from weak peers to better-performing peers when needed. This rule can be useful in those PPV (Pay-Per-View) where the stream must be guaranteed to those users that have paid for receiving the stream. However, notice that in this last case, the C/S model should be also considered.
3. **End-point Masquerading Set (EMS) of rules:** This set of rules handles those situations where two or more peers are behind a NAT device that performs IP masquerading (a situation commonly found when peers run in private networks).
4. **NAT Traversal Set (NTS) of rules:** Connection-filtering-NATs are becoming increasingly frequent, a situation that dificults the connectivity between peers. This set of rules introduces the extra P2PSP functionality to handle peers that are behind restricted-cone NATs and symmetric NATs.

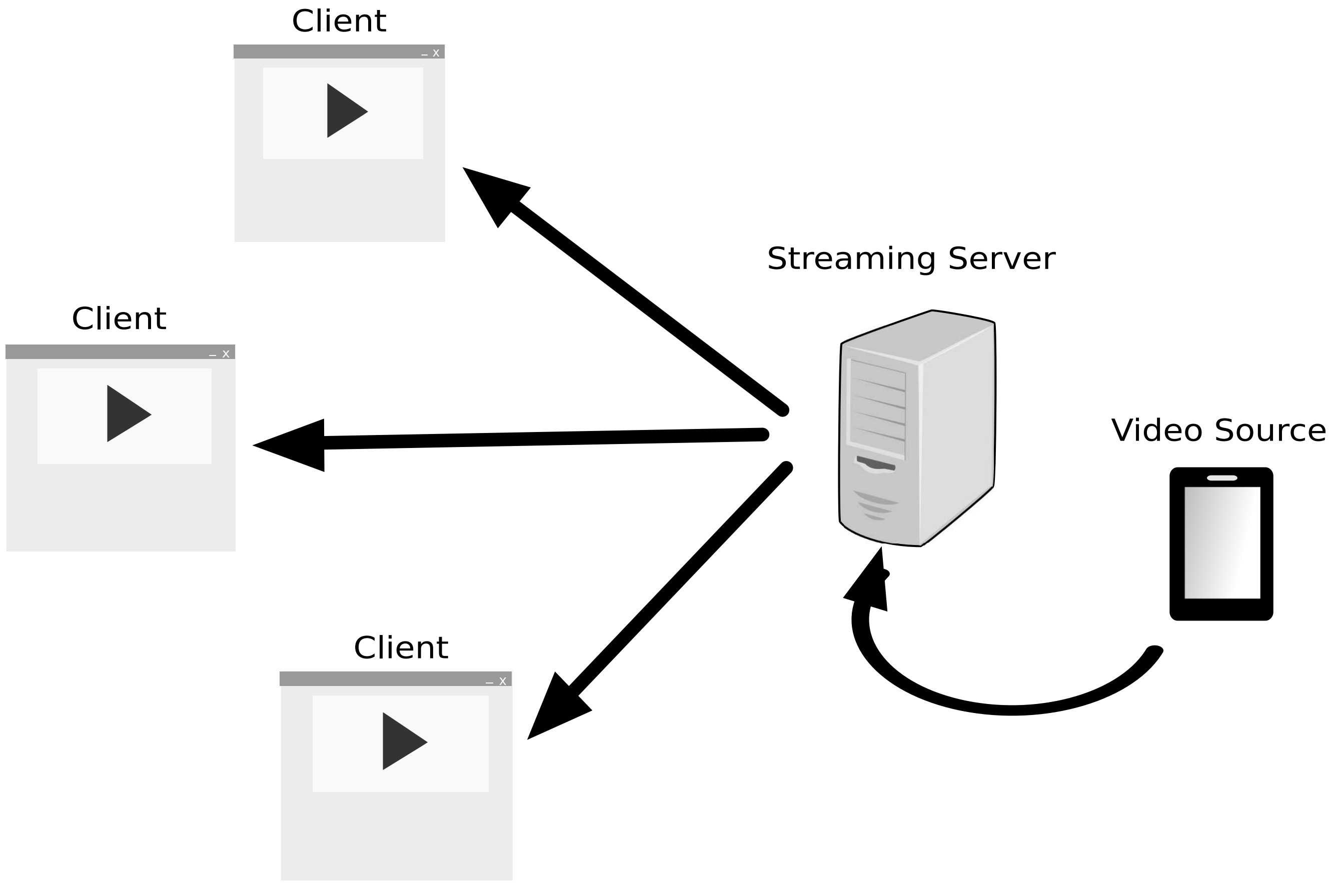

Figure 1: A typical real-time streaming scenario using the C/S model. The Video Source (in this case a mobile device) captures the video and sends it to a Streaming Server. Each Client retrieves a copy of the stream from the Streaming Server, which sends the same content several times.

5. **Multi-Channel Set (MCS) of rules:** In contexts where there is sufficient available bandwidth, peers can decide to subscribe to more than one stream (channel) in a given period of time. A peer that implements this set of rules communicates with several teams concurrently.
6. **Data Integrity Set (DIS) of rules:** In some contexts, the network needs to protect itself from hostile peers which could produce the poisoning of the stream, a denial of service, etc. Those peers will be identified and rejected from the P2PSP overlay by this set of rules.
7. **Data Privacy Set (DPS) of rules:** There is a collection of rules ensuring the transmitted stream is played only by allowed peers. This set can be useful to implement, for example, PPV streaming services.

4 Proposal

Figure 1 presents an example of a pure C/S system used to transmit a capture from a mobile camera to three clients. Three main elements can be recognized:

1. **The video source:** A device that generates the video and sends it to the streaming server.
2. **The streaming server:** A process running in a host reachable from the public Internet which is in charge of relying the video in real-time. Notice that the number of concurrent clients depends on the video encoding bit-rate and the upload capacity of the server.
3. **The client:** A Web browser with HTML5 support. The Web page interpreted by the browser can be downloaded from any avaiable Web server, which redirects to the Streaming Server.

Both technologies, the WebRTC API and the P2PSP can be used together to develop more efficient Web-based IPTV systems than the presented in Fig. 1. More specifically, the idea is to run P2PSP peers in Web browsers and to use the DataChannel facility provided by the WebRTC as the transport protocol.

Figure 2 shows a similar configuration as Fig. 1, but in this case the streaming server only needs to send a copy of the stream to the P2PSP splitter. The content is scattered over the peers (the Web browsers) by the splitter and those

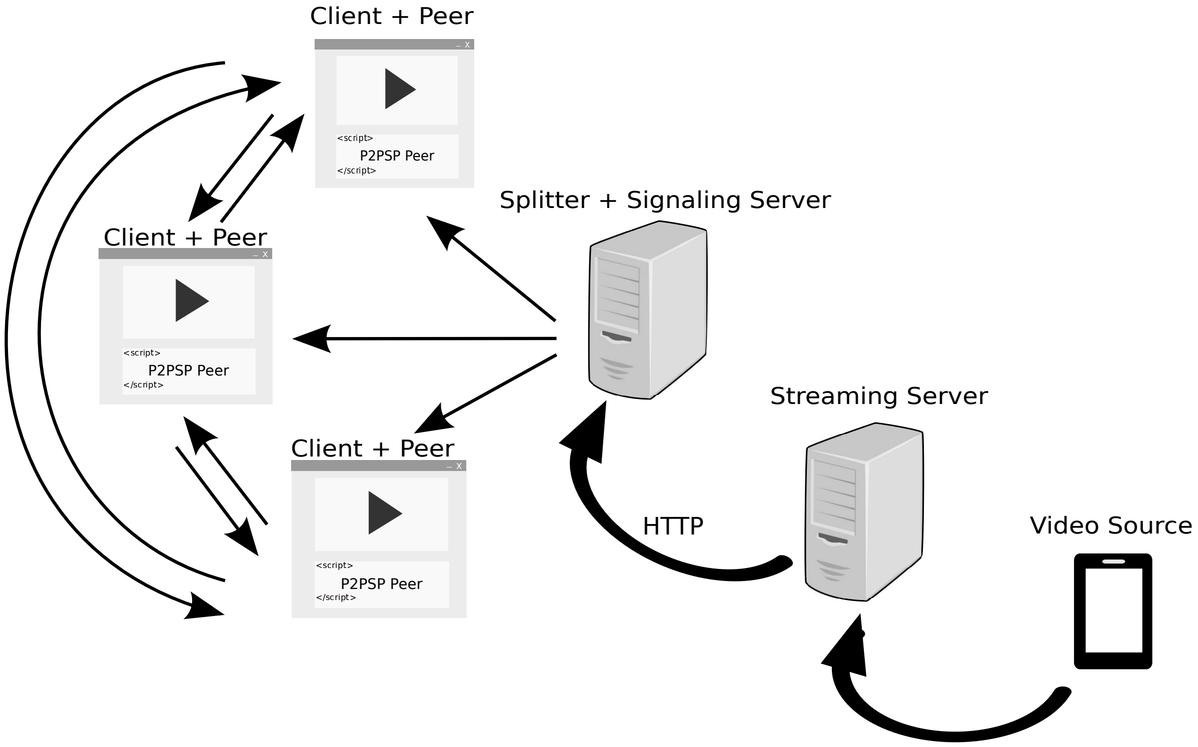

Figure 2: An equivalent scenario to the exhibit in Figure 1 but using the P2PSP. Notice that in this case the Streaming Server sends only one copy of the stream. As the Streaming Server, the Splitter only sends a copy of the stream and the peers are in charge of sharing the content between them.

retransmit the chunks of data that the rest of peers need in order to reconstruct the video completely. More specifically, we need to include:

1. **Splitter+Signaling Server:** Retrieves the video from the Streaming Server (using the HTTP) and broadcasts it to the P2PSP team.
2. **Client+Peer:** Receives, shares and plays the stream. The main steps taken in this process are: (1) downloading the list of peers from the Splitter+Signaling Server and (2) performing an P2PSP communication with the rest of Client+Peers.

Basically, the proposed system provides the following advantages:

1. As in the C/S model, users do not need to download nothing more than a Web page in order to start downloading the video. No extra plugins or third party programs are required.
2. The clients supply their upload capacity in order to make the system much more scalable.
3. The extra requirements of the P2PSP/WebRTC are quite reduced. Only an extra port is required at each client and the computing resources are minimal (the logic of the P2PSP is very lightweight).
4. The C/S model and the P2P model are not incompatible. For example, it it possible that a premium user (which is not forced to share the stream) retrieves the stream directly from the streaming server using a C/S model.

5 Summary

IPTV systems require an optimal use of the available network capacity in order to provide a good quality of service. In this context, a Web-based IPTV real-time streaming system has been proposed. It uses three recent technologies, P2PSP, HTML5 and WebRTC, which allow to deploy efficient P2P overlays without needing to install extra software. Therefore, HTML5 and WebRTC standards could be considered in other P2P video streaming protocols.

References

[1] C. Bouras, G. Kioumourtzis, and A. Gkamas. Simulcast transmission for video applications: performance evaluation with an integrated simulation environment. In *Proc. 12th inter. conf. on Perf. Eval. of Computer & Telecommunication Systems*, pages 339–346, Piscataway, NJ, USA, 2009.

[2] Free Software Foundation. `http://www.gnu.org/licenses/gpl.html`.

[3] ViVek K Goyal. Multiple description coding: Compression meets the network. *IEEE Signal Processing Magazine*, pages 74 – 93, September 2001.

[4] IETF. Peer to Peer Streaming Protocol (PPSP). `http://datatracker.ietf.org/wg/ppsp/charter/`.

[5] Cristobal Medina-López, J.A.M. Naranjo, Juan Pablo García-Ortiz, L. G. Casado, and Vicente González-Ruiz. Execution of the P2PSP protocol in parallel environments. In Guillermo Botella y Alberto A. Del Barrio Garcia, editor, *Actas XXIV Jornadas de Paralelismo (http://www.congresocedi.es/images/site/actas/ActasParalelismo.pdf)*, pages 216–221, Madrid, Septiembre 2013.

[6] Huifang Sun, Anthony Vetro, and Jun Xin. An overview of scalable video streaming: Research articles. *Wirel. Commun. Mob. Comput.*, 7(2):159–172, February 2007.

[7] P2PSP Team. Implementation of the P2PSP (Peer to Peer Straightforward Protocol) in Lauchpad. `https://launchpad.net/p2psp`.

[8] P2PSP Team. Peer to Peer Straightforward Protocol. `http://p2psp.org/index.php/en/p2psp-protocol`.

[9] WebRTC Team. Web Real-Time Communication. `http://www.webrtc.org/`.

[10] W3C. HyperText Markup Language, versión 5. `http://www.w3.org/html/wg/drafts/html/master/`.