P2PWeb: a Client/Server and P2P Hybrid Architecture for Content Delivery over Internet

Majd Ghareeb, Soufiane Rouibia, Benoît Parrein, Mohamad Raad, Cedric Thareau

P2PWeb: a Client/Server and P2P Hybrid Architecture for Content Delivery over Internet

Majd GHAREEB[1], Soufiane ROUIBIA[2], Benoît PARREIN[3], Mohamad RAAD[1], Cedric THAREAU[2]

[1] Computer and Communication Engineering
Lebanese International University
146404 Mazraa, Beirut, Lebanon
firstName.lastName@univ-nantes.fr

[2] TMG
44230 St Sébastien sur Loire
France
lastName@tmg.eu

[3] LUNAM University, University of Nantes, IRCCyN UMR CNRS 6597, Polytech Nantes, France
benoit.parrein@univ-nantes.fr

***Abstract*— In the domain of content delivery over Internet, each of the Client/Server and P2P communication modes has its pros and cons. In this scope, hybrid network architectures have been recently proposed as a relevant solution. In this paper we propose a new hybrid architecture that is called P2PWeb, between the centralized client/server and the non-centralized P2P architectures for content delivery. The main objective of this proposal is to reduce the load over the server in order to provide a better Quality of Service (QoS) for the end-users. A new P2PWeb communication protocol has been implemented and deployed to reach the objective. The experimentation results and the performance evaluations that we have made show the efficiency of the proposed system in terms of QoS evaluations.**

Keywors-- Content delivery; Client/Server; P2P; BitTorrent; Hybrid architectures; P2PWeb.

I. INTRODUCTION

The rapid development of networking technologies has tremendously facilitated the content delivery over Internet. In this trend, the traditional client/server (C/S) communication mode has suffered from several drawbacks that necessitated the deployment of alternative solutions such as the Peer to Peer (P2P) communication mode. However, in order to provide an efficient reliable and scalable content delivery service, neither a C/S-based, nor a P2P-based architecture can reach the target alone. The centralized server of C/S architecture may not support the huge loads over the Internet, which undermines the system scalability and leads to a bad content delivery service. It requires high accessibility features that may not be easily and cheaply provided [1]. On the other hand, P2P architecture needs a sufficient number of 'seeders' (content sources) to launch the content delivery service. Besides, it is not easy to fairly determine the contribution of each peer, which may also affect the reliability of the system. To cope with these issues, different researches have been already proposed, to provide hybrid compromise network architectures between C/S and P2P architectures [1,2,3,4]. In some of these works, authors propose to release the server after a given instant and to switch to a pure P2P communication mode [1,2]. Other approaches propose to start the content delivery with a P2P communication mode, and then to redirect the clients' requests to the content server in case of need [3]. In this article we propose new hybrid architecture (P2PWeb) that helps to integrate the P2P technologies in a web environment in order to provide an efficient content delivery service over Internet. In our proposal, the server continues to response to clients' requests as long as it is not overloaded. Then, when it arrives to a saturation threshold at which it cannot deliver the content to any new client, it stays in the system and it starts to redirect the new clients (peers) to retrieve the content from the other clients/peers that are already present in the system. P2PWeb first goal is to reduce the load over the server. Moreover, the new proposed P2PWeb communication protocol helps to improve the (Quality of Service) and QoE (Quality of Experience) of the system. P2PWeb communication protocol develops the role of BitTorrent typical tracker by leading it to deliver a chunk of the content to the new arriving peers (leechers), besides the list of the already existing client/peers (seeders). This enhancement (i) helps to reduce the start-up delay that is normally generated because of the lake of seeders at the beginning of the content delivery session; moreover, (ii) it augments the contribution of the new arriving peers (leechers) in the content delivery compared to the already existing client/peers (seeders) contribution.

Section II of this paper presents the hybrid P2PWeb architecture and its communication protocol. Section III shows the advantages of using this architecture by demonstrating some of the experimentation results and the performance evaluations that we have obtained. Furthermore, Section IV concludes the article and lists some of the future work steps.

II. THE HYBRID P2PWEB ARCHITECTURE AND DESIGN

We start this section by introducing the hybrid architecture P2PWeb. Then, we explain its network protocol and the P2PWeb compatible browser-plug-in that we have developed.

A. System Architecture

In P2PWeb architecture, the server plays the role of a "*content provider*" and of an "*index server*" at the same time. For a given content, the delivery start in a C/S mode. Then, at a given clients' number threshold at which the server becomes saturated, the system changes the delivery to a P2P mode among the already connected P2PWeb clients and that potentially accept to share their downloaded chunks. Figure 1 represents the system architecture blocks. The first server's task is to provide the different types of contents (web pages, text, images, audio, video, etc) that will be distributed in the P2PWeb system. The content passes through a Content Preparation phase, in which it will be divided into variant number of chunks to be injected in the system. Chunks

978-1-4673-5307-6/13/$31.00 ©2013 IEEE

generating process depends on the type of content. For example, for the real-time media streaming, each chunk will represent the data to be delivered for a given portion of time. Hence, some important factors such as time stamp and chunk order should be taken into consideration.

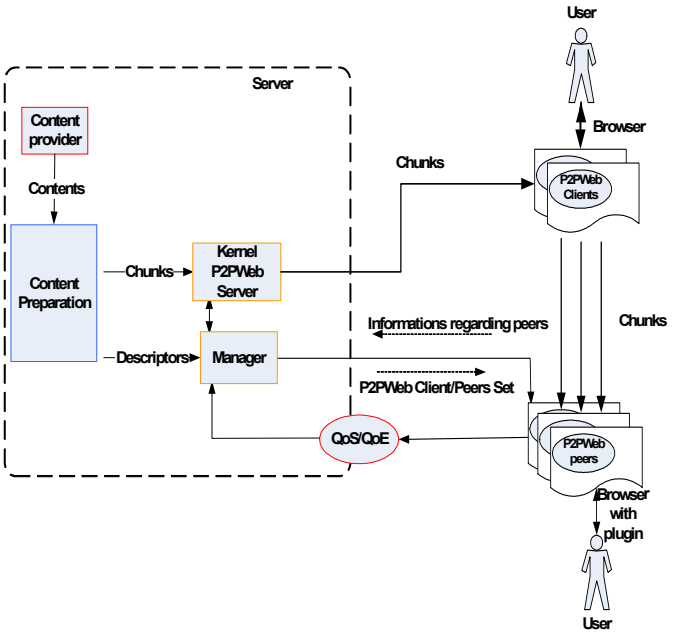

Fig. 1. P2PWeb hybrid architecture.

On the other hand, non-real-time content downloading has typically less restrictions in term of delay. Each content file will have its own related list of chunks that is signed by the content preparation service and that will be transmitted to the Manager by the kernel P2PWeb Server. This list contains the associated content ID, the chunk IDs, the chunk time stamp and the chunk hashes. To ensure the safety of the contents, these hashes are managed in a centralized manner via the Manager. P2PWeb Manager is the general coordinator that has a global knowledge about the system state. It knows (i) what content is available, where it is located and in which quality; (ii) which P2PWeb clients/peers are on-line and the resources/content they have; (iii) which and how many P2PWeb clients/peers request a given content.

As long as the server is not saturated, the kernel P2PWeb server responses to the new arriving request in a C/S content delivery mode, by sending the chunks that are formatted to be easily used in the P2P exchange between peers. Moreover, it declares the new client to the Manager as a new content owner. Although P2PWeb clients are receiving the content directly from the server, but we have to be aware that they are not classical clients, since they are receiving the content in terms of chunks. Besides, these clients become sources for this content at a given instant. When the server arrives to a saturation threshold, with which it will not be able to deliver the content to any new client, the Manager takes the responsibility of responding to the new requests. This will be done by sending meta-data information about the content with a list of P2PWeb clients and peers that are already downloading this content to the requested user. A special P2PWeb plug-in that is installed on the user browser will analyze the received information and will consequently start to download the content. Here, we can notice that as long as the server is at its saturation threshold, all the new arriving nodes will not be treated as clients, but as peers. Hence, instead of downloading the content chunks directly from the server, they will retrieve the requested chunks from the P2PWeb clients/peers that are already connected to the system and that are downloading the content in question.

P2PWeb hybrid architecture can be supported with an adaptive QoS/QoE management technique, like the one presented in [5]. With such a technique, P2PWeb clients and peers will feedback the Manager with activity reports about the perceived quality and the user satisfaction, so it will be able to improve the delivery service accordingly and in real-time.

B. P2PWeb Communication Protocol

During the implementation of the P2PWeb communication protocol, we had to consider the different operational modes of our proposed architecture. P2PWeb starts up with a classical C/S communication mode before reaching a clients' number threshold. Then, it passes to a first hybrid operational mode, in which the content server selects a group of clients to keep working with them in C/S mode, while new arrivals will be treated as peers. In the second hybrid operational mode, the content server keeps running C/S mode with its already chosen set of clients, while with the others (peers), it uses a P2P policy for choosing and sending content chunks. In this last mode, the server sends a list of the clients/peers that are already existed on the system to the new arrived peer. Besides, it sends it a chunk of content, in order to accelerate its contribution into the swarm as much as possible, and consequently improve the QoS parameters. Different mechanisms could be used to determine the chunk of content that will be sent to each peer (random chunk, rarest chunk, sequential-based chosen chunk, etc). However, as we will see in the experimentation results section, even by sending a randomly chosen chunk, the quality of the provided service would be enhanced substantially.

In order to achieve the different operational modes of P2PWeb and instead of starting from crash, we adopted the BitTorrent Open Source code, developed in Python Language that has already proven a good and reliable performance. The modifications that we have applied consist of adapting the queries for a Web exchanges. The first chunks to be downloaded are organized in order to have sequential and not random chunk downloading. We also added some parameters to measure the QoS and manage the overlay construction using these measurements, as we will see in details in the performance evaluation section.

C. P2PWeb Use-case Design

Figure 2 depicts the sequence diagram of a use case for a content delivery process in P2PWeb system. In the first case, when the server did not reach its clients' number threshold yet, a simple content delivery process between the server and the user will be done in a C/S mode. On the other hand, the second case represents the steps of content delivery in P2P mode when the server threshold is reached. In this case, P2PWeb plug-in will obtain the meta-data information and a set of P2PWeb

clients/peers that are already downloading the content from the Manager. Then it will start to retrieve the content from this.

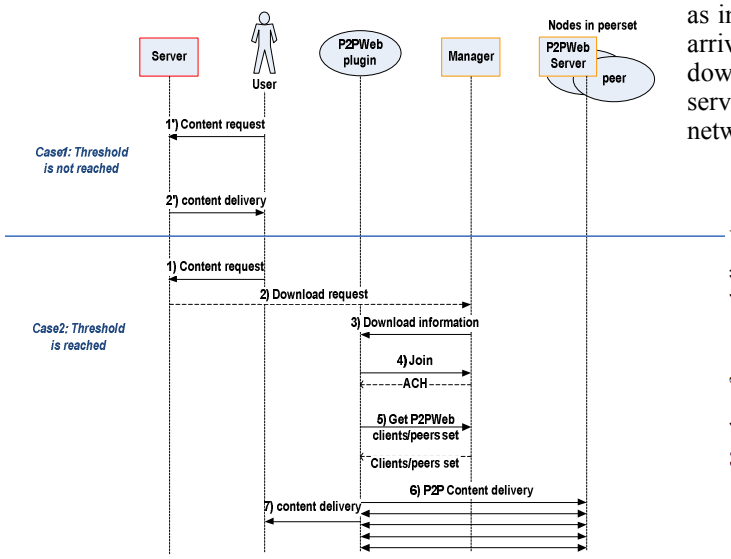

Fig. 2. A use case for content delivery process in P2PWeb system

III. EXPERIMENTATION RESULTS AND PERFORMANCE EVALUATIONS

In this section we first present comparison results to show the advantage of using P2PWeb hybrid architecture against a traditional C/S transmission mode. Then, we study the performance of our hybrid proposal in different scenarios. Our experimentations have been done in a virtualisation environment that contains a server and two clusters of 23 and 50 virtual machines respectively, with public IP-addresses for the different machines. Several content files with different sizes have been tested to validate our experimentations. We present the results obtained by using a JPG content file of 39 MB size. At this step of experimentation, the other file sizes did not present a significant difference.

A. P2PWeb vs pure Client/Server architecture

The objective of the first set of experiments was to show the advantage of the proposed P2PWeb hybrid architecture against a traditional C/S communication architecture in terms of QoS parameters represented by the delay. For the two scenarios (C/S and P2PWeb), we ran a server with the same downloading rate of 2MB/s using the first cluster of virtual machines. In the former scenario, the 23 nodes will act as pure clients and will download the content from the server directly. Furthermore, in P2PWeb scenario the server will start to upload the content to the first 8 arrived nodes as pure clients. Starting from the 9th arrived node to the network, the server responses to user requests by sending a list of P2PWeb clients/peers that are already connected to the network and that accept to share the content with a maximum uploading rate of 300KB/s for each client/peer. We can notice in Figure 3 how in C/S scenario, the more the number of connected clients is, the more the demanded time for downloading the content will be. On the hand, by using P2PWeb architecture, the first 8 nodes (clients) will take the same time to download the content as in C/S scenario. While, starting from the 9th node, the new arriving nodes (peers) will take considerably less time to download the content, since they will not be limited by the server, and they will use the other clients and peers on the network as content providers.

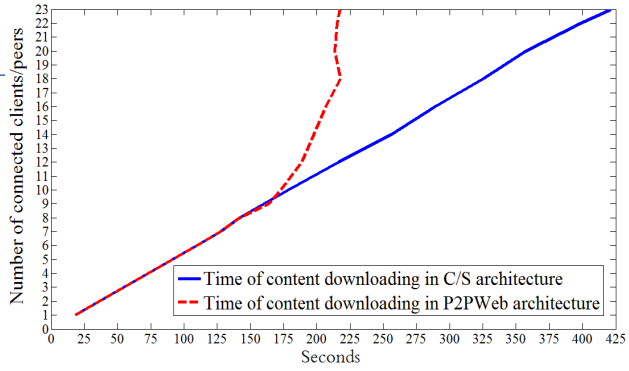

Fig. 3. P2PWeb vs pure Client/Server architecutre

B. QoS P2PWeb performance

After illustrating the benefits of P2PWeb hybrid architecture, our goal in the second set of experiments is to study the performance of the proposed architecture in a way that gives better results in terms of QoS and potentially QoE. Two main contributions are addressed for this objective. The former is to compare a passive P2PWeb tracker that plays a role of an index server, with an active tracker that passes a chunk of data to the new peers. However, the selection strategy of the data chunk to be sent could significantly affect the quality of the provided service. Hence, our second contribution in the performance evaluations is to compare two selecting strategies, a randomly chosen chunk and the rarest chunk. We aim at demonstrating the ability of applying our hybrid architecture in large scales. Hence, we extended the platform of 23 nodes that we have used in the first set of experimentations with another platform of 50 nodes. Furthermore, in order to give more realistic results, we made our measurements on a machine that is put behind a firewall like the most part of the connected terminals over the Internet [6]. Moreover, we limited the measurement time to the first 60 seconds. This minute corresponds in many P2P applications to the TTL (time to live) of chunk delivery [7]. Then, we tried to find the most suitable P2PWeb topology (clients vs. peers) for this precise period. We studied the impact of three important parameters variation on the hybrid P2PWeb architecture, which are:

- the number of P2PWeb clients that retrieve the content from the server directly;
- the number of peers that retrieve the content from P2PWeb clients/peers which are already present in the system and receiving that content;
- the uploading rate for the peers in the network.

- ***P2PWeb: Active tracker vs. Passive tracker:***

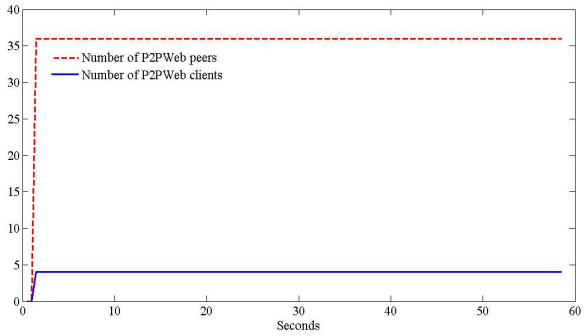

Fig. 4. Number of connected P2PWeb Clients and Peers per second

In this scenario, we use a data rate of 60KB/s. Besides, we assume that the server can serve 5 clients directly in a C/S mode. The 68 arriving nodes to the network -besides the measurement peer- will be then served as peers. In this case, the Manager will pass to each arriving node a set of 50 P2PWeb clients/peers that are connected to the network and that are concerned about the content in question. Figure 4 represents the number of P2PWeb clients and peers that are connecting to the measurement peer during the first 60 seconds of the content delivery process.

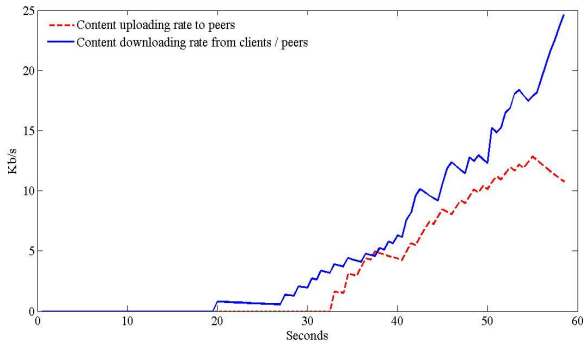

Fig. 5. Passive tracker: Uploading /Downloading rates

On the other hand, Figure 5 shows the interaction between the measurement peer and P2PWeb clients/peers. By mapping Figure 5 to Figure 4, we can notice that although the measurement peer is connected to 5 P2PWeb clients and to more than 30 peers as soon as it arrives to the network, the interaction with these clients/peers will not start directly, and a remarkable "startup" delay takes place before starting the content downloading. The reason is that the new arrived peer has no content to share yet. Hence, at its arrival, nobody on the network will be interested to exchange chunks with it. For these reasons, active tracker (that passes a data chunk with the list of clients/peers) has been proposed as a solution. In the scenario of Figure 6, we keep the same previously configuration. On the other hand, we use our enhanced active tracker that passes a randomly chosen chunk of content to the new arrived peers.

We can clearly remark the improvement in the start-up delay that could be achieved due to the rapid contribution of the peers in the content delivery process, as soon as they are present at the system.

However, the configuration that we have used in the previous two experimentations has been chosen to maximize the contribution of the peers in the system compared to the contribution of the P2PWeb clients. This configuration could clearly show the advantage of using an active tracker, but on the other hand, 60KB/s data rate will limit the measurement peer to download less than 20% of the entire content in 60 second for the two experimentations, which will necessitate the re-tuning of the experimentation parameters in order to obtain better results.

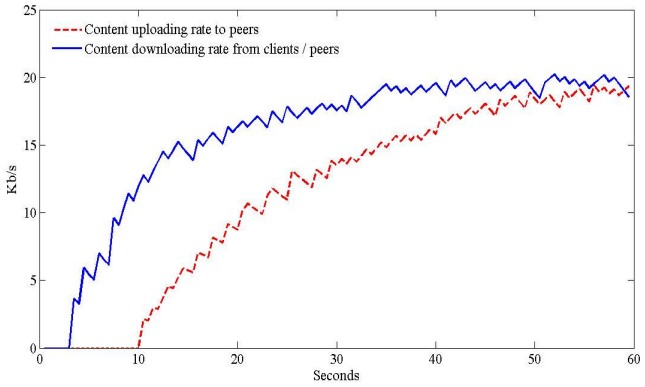

Fig. 6. Active traker (random chunk): Uploading/Downloading rates

- ***P2PWeb: Random chosen chunk towards rarest chunk***

In our second scenario, we chose to use an uploading data rate of 300KB/s for each client/peer that corresponds to a reasonable and realistic data rate over the internet nodes. Besides, we increased the number of P2PWeb clients into 10 instead of 5, while the number of peers became 63.

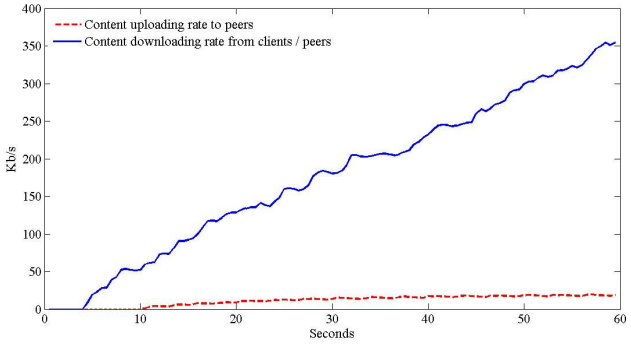

Fig. 7. Active traker (random chunk): Uploading/Downloading rates

Figure 7 shows the interaction between the measurement peer and P2PWeb clients/peers with an active tracker that sends a randomly chosen chunk to each new connected peer. We can notice how the largest part of the downloaded content has been obtained by the clients, while a modest contribution has been provided by peers. This result was obtained when we used the passive tracker, since all the peers are connecting to the network at almost the same time and thus will not have

enough chunks to exchange with the new arriving peer as the P2PWeb clients do. However, we can see that the active tracker (random chunk) in Figure 7 has solved the problem of the start-up delay, but could not solve the problem of the unfair contribution of the new arrived peers. Beside, as Figure 8 shows, the measurement peer could only download 45% of the entire content in 60 second, which means the ineffectiveness of the random chunk selection strategy.

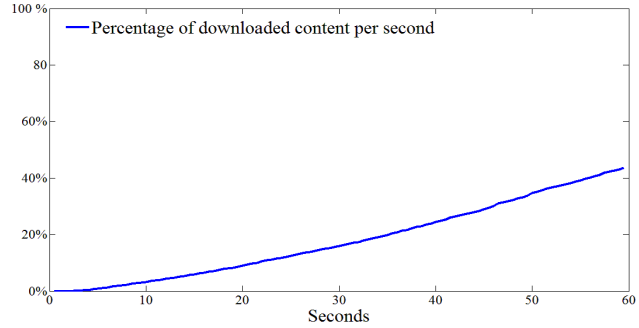

Fig. 8. Percentage of downloaded data over the meauserment peer

These limitations motivated us to adopte another chunk selection strategy, which is the rarest chunk. As shown in Figure 9, the tracker will pass the rarest chunk to the new arriving peer. Consequently, the contribution of the peer will be remarkably accelerated in the content delivery process as soon as it enters to the system. The new arriving peers will continue to fastly share their chunks in the system, until a certain limit (second 40 in Figure 9) in which the number of content providers (clients/peers) will be high enough to undermine any new contibution.

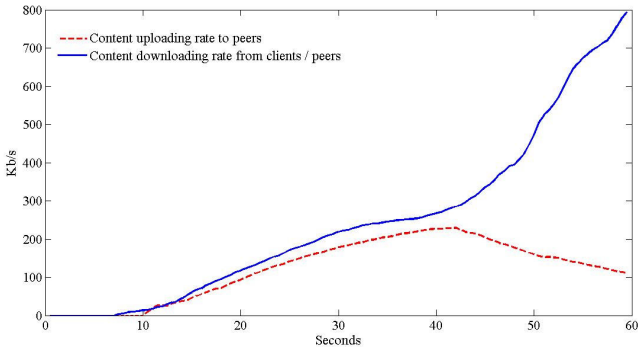

Fig. 9. Active traker (random chunk): Uploading/Downloading rates

Moreover, as we can notice in Figure 11, with 10 P2PWeb clients, 63 peers, an uploading rate of 300 KB/s for each client/peer and the rarest chunk strategy, the measurement peer could download up to 8% of the content in the first 60 second of the content delivery process.

IV. CONCLUSION

Our main objective in this paper was to provide an efficient reliable and scalable content delivery service. We proposed a new hybrid architecture that is called P2PWeb, between the centralized Client/Server and the non-centralized P2P architectures for content delivery over the Internet. This architecture helps to reduce the load over the server in order to provide a better service for the end-users. To reach the goal, we

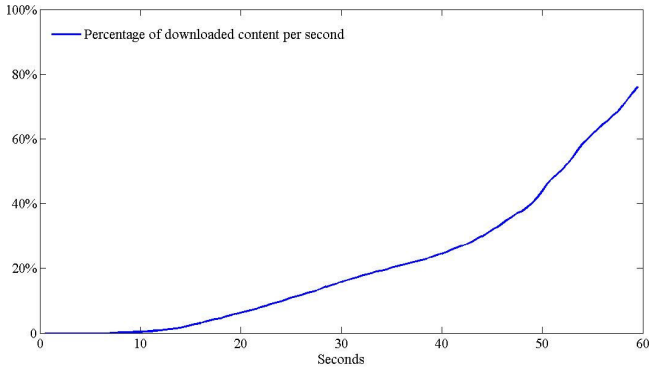

Fig. 10. Percentage of downloaded data over the measurement peer

based on an open source BitTorrent protocol to implement a P2PWeb network protocol. By our experimentation results, we prove the advantage of our proposed architecture against the traditional C/S communication mode. Besides, we studied the different network configurations in term of client's number threshold and maximum uploading rate in order to select the best one for our contribution. Moreover, we enhanced our P2PWeb tracker functionality by allowing it to send a randomly chosen or the rarest chunk of data to the new arrived peers in order to facilitate their contribution at the content delivery service. However, finding the rarest chunk of content is a critical point that should be deeply investigated in future works. Currently, we are studying the feasibility of the proposed architecture for different applications, such as the real-time streaming of scalable video content by using the Scalable Video Coding (SVC).

ACKNOWLEDGMENT

The work presented in this paper has been supported by P2PWeb project with support from OSEO and the Bretagne and Pays de la Loire regions. The P2PWeb project is labeled by the "Images et Réseaux" cluster.

REFERENCES

[1] D. Xu, S. Kulkarni, C. Rosenberg, and H. Chai, "A CDN-P2P hybrid architecture for cost-effective streaming media distribution," Computer Networks, vol. 44, pp. 383-399, 2004.

[2] Y-C Tu, J. Sun, M. Hefeeda and P. Sunil, "An analytical study of peer-to-peer media streaming systems", ACM Trans. Multimedia Comput. Commun. Appl, vol. 1, no. 4, pp. 354-376, 2005.

[3] A. Bakker, R. Petrocco, J. Gerber, V.r Grishchenko, D. Rabaioli and J. Pouwelse, "Online Video Using BitTorrent and HTML5 Applied to Wikipedia", Int. Conf. on Peer-to-Peer Computing, pp.1-2, 2010.

[4] R. Mattson, "Enhancing HTTP to improve page and object retrieval time with congested networks", Ph.D. thesis, La Trobe University, Melbourne, Australia, 2008.

[5] M. Ghareeb, A. Ksentini and C. Viho, "A multipath Video Streaming Approach for SNR Scalable Video Coding (SVC) in Overlay Networks," IEEE CCNC: Consumer Communications and Networking Conference, pp. 605-610, Las Vegas, NV, USA, 2011.

[6] L. D'Acunto, J. Pouwelse, and H. Sips, "A measurement of NAT & firewall characteristics in peer to peer systems," in Proceedings of 15th ASCI Conference, pp. 1-5, June 2009.

[7] B. Cheng, H. Jin, and X. Liao. "RINDY: A Ring Based Overlay Network for Peer-to-Peer On-demand Streaming". IEEE conference on Ubiquitous Intelligence and Computing ,pp. 1048-1058, Wuhan, China, September 2006.