IPTV-P2P clients at home

Alexandro Sentinelli[(1)], Gustavo Marfia[(2)], Giovanni Pau[(2)], Luca Celetto[(1)]

[(1)]Advanced System Technology – STMicroelectronics
Agrate Brianza (Mi), Italy
[alexandro.sentinelli | luca.celetto] @st.com

[(2)] University of California of Los Angeles (UCLA)
Los Angeles, California
[gmarfia | gpau] @cs.ucla.edu

***Abstract*— The investigation around the use of peer-to-peer in IPTV environments goes on. The debate about the best compromise between the cost reduction offered by the peer-to-peer (P2P) approach and the quality requirements by the mass market is still open. Though some successful protocols are already worldwide used, there are still issues and optimization problems to solve. The main contribution of this work has been the deployment on PlanetLab (Plab) of a large testbed in order to investigate the performance of [P2P client – streaming player] systems in static and dynamic overlays: the goal was to recreate a real IPTV-like heterogeneous scenario with home users enjoying the same stream and cooperating via P2P. IPTV needs bandwidth and computational resources that single peers often struggle to provide. We chose the SopCast client for our experiments, mainly because of its popularity and its flexibility. Although SopCast analysis was not our main research purpose, we are able to point out some of its characteristics by observing its behavior in large and dynamic overlay, with different churn patterns (i.e. zapping channel behaviors).**

Keywords-component; P2P, home environment, streaming, video delivery, PlanetLab.

I. INTRODUCTION *(HEADING 1)*

The whole Internet video market grows as a synergy among broadband adoption, wireless subscribers, and pervasive video contents in web-sites and services. The IP convergence plays a key role in the wired/wireless integration as much as wireless providers became conscious of the urgency in following the evolution of the wired broadband: content producers have to be able to deliver their products to every user, as well as every user wants to access the same content independently from the type of network connection. *Video is the market drive* of such massive broadband adoption. Global IPTV Service Revenue Forecast is $7.2 billion in 2008 growing to $31.6 billion in 2012 [1], with a compound annual growth rate of 34.5%. Distributed systems, decoders, service providers, hardware manufacturers and content broadcasters are following the migration, or we may also think of an extension, of the TV to the Internet Protocol network. Set Top Boxes (STB) match well the IPTV market especially in the case of broadband householders. According to [2], 70% of European homes will own at least one HD-capable TV by 2012. Aside HD-TV, P2P recently started to find a proper place in the business scenarios. Small broadcasters, pro-sumers and (controlled) file-sharing communities, look for P2P engines to lower the cost of network infrastructure. High churn rates can heavily affect the stability of a P2P network distribution system. Commercial clients, on the other side, dedicate servers to ensure a certain QoS, especially to its end users with high definition requirements. However, for low resolution video streams, many commercial content providers are beginning to rely on P2P. In this work we explore the implementation of a full IPTV Home-like system in a heterogeneous environment. To do this, we deploy a large set of clients on PlanetLab and measure the performance figures of the videos that is received at these clients. In order to make this more realistic, we enforce churn patterns and observe how the P2P network recovers from each perturbation. In this work we explore the implementation of a full IPTV Home-like system in a heterogeneous environment. To do this, we deploy a large set of clients on PlanetLab and measure the performance figures of the videos that is received at these clients. In order to make this more realistic, we enforce churn patterns and observe how the P2P network recovers from each perturbation. This paper is organized as follows. In the next section we describe the testbed, giving an idea of what type of software has been used on each node. We then show the results we gathered from the overlay, for a static situation. Finally, we move to describe the network's behavior under dynamic conditions. The conclusion section closes suggesting possible new directions of work and describing our next objectives in this research scenario.

II. TESTBED DESCRIPTION

Worldwide successful, usable and available clients for Linux, SopCast and Videolan (VLC) have been our choices. The main metric to compare video performances for streaming application at the end user side is the frame loss, that VLC makes available only through the visual interface. We modified the VLC code to print frame statistics into a log file. In our setting we installed VLC on each node to test the whole IPTV-like (SopCast+VLC) system's behavior simulating either steady or dynamic environment. The trace's collection lasted 4 months, simulating either steady or dynamic scenarios where users are watching the same content while other zapping through channels. Aside we made traffic analysis based on tcpdump traces to get a deeper comprehension of the network behavior under different solicitations. A stream that we wish to broadcast is originated through VLC server and accepted by SopCast (Sop) server as a live content. SopCast, at the server side, tags the stream with a channel ID and informs the SopCast main tracker that this content is available. The SopCast clients that first select this content for viewing will download it from the server, later clients will connect to peer clients, generating a mesh network. The typical buffer time that we observe is of ~50 sec, this amount of data is shared via P2P. In Fig.1 we find the testbed's logical scheme. We observe that

The work has been partially funded by the European Commission under the projects ICT-214063/SEA (www.ist-sea.eu).

VLC's buffer is negligible compared to SopCast's buffer (1.2 sec vs ~ 50 sec): we therefore can say that VLC is not interfering with the natural behavior of the p2p infrastructure.

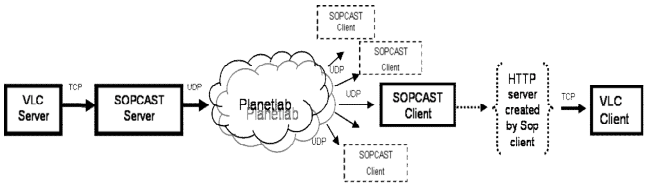

Figure 1. Testbed - implementation scheme.

We place the source of content, the server, in the Network lab at UCLA. This acts as a small broadcasting station by delivering a live stream using a server-less infrastructure (pure p2p). One important aspect we should highlight in our experiments is PlanetLab heterogeneity and resources availability. The research community is as spread just as PlanetLab is around the world, overloads are unpredictable in space and time, thus the performance of the peer-to-peer network, even keeping the same settings across all experiments, is subject to changes. This is true especially for those experiments that require a high bandwidth and which are CPU-intensive. On the other side, the heterogeneous nature of this environment represents much better a real Internet scenario, where the network congestions, computational resources and the bandwidth links dynamically change the state of the system and of the overlay. PlanetLab policies, in order to guarantee a fair use of its resources to the whole Plab community, enforce constraints that limit the CPU consumption and the daily traffic per user account (or slice). Internet Service Providers (ISP) apply similar constraints (especially for the bandwidth) to reduce the P2P traffic, since P2P traffic is capable of saturating the bandwidth of an ISP's infrastructure. Actually, many P2P clients are just designed following a best-effort approach that may lead to saturate local resources. To give an idea of this problem, in PlanetLab, 50 SopCast peers downloading at a 1.5 Mbps bitrate exhaust in a few nodes the traffic share per day per slice after 20 minutes! We collected traffic statistics to investigate such issue. The main video reference metric in our investigation is the Continuity Index (CI), which is the number of video frames read in the right order over the number of total frame in the stream (thus CI is in [0,1]). If chunks do not reach the destination on time VLC skips the frames contained in those chunks and freezes the playback, causing a video playback quality deterioration.

III. Experiments with Steady Overlay

The first scenario sets up a steady overlay at 3 different bitrates: 102 (cell phone), 380 (handset), 1511 (IPTV) kbps (these are all bitrate at VLC server/client side before/after the SopCast protocol). Since there are no changes in the overlay we averaged several sessions with specific settings measuring the dwnl/upl bitrate, the CI and the duration of VLC before it crashed. VLC's duration does not represent a "scientific" index of performance, but an index that may give us some information about the workload of each node. Performances are often affected by the overload of the CPU and the high dwnl/upl bitrate (function of the content resolution or the overlay network size), just as end users use their platform for various purposes. These factors can push a VLC client to crash. Even with a static overlay, we observed a good variability in the performance of clients placed on the PlanetLab nodes. Averaged over 10-15 trials, here we have the experiments for three different overlay size (10, 50, ~110).

A. *Traffic statistics*

From tcpdump traces there are no particular remarks about the download statistics: SopCast delivers the same amount of data to every node for the three bitstreams, though the upload contribution is much different for each peer. In Fig. 2 we see the average upload per peer.

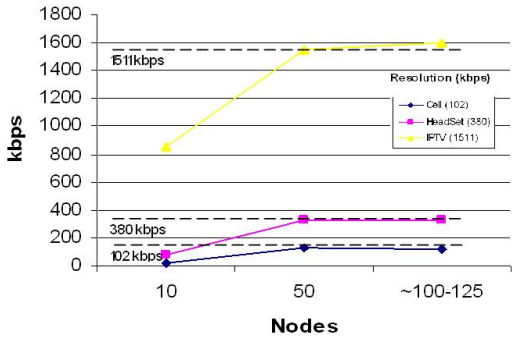

Figure 2. Avg Upload in a steady environment for different video resolution.

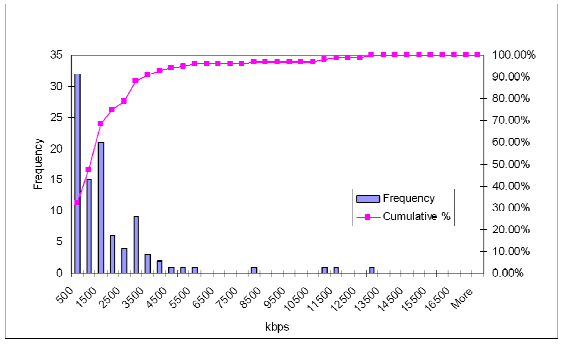

Figure 3. Bin histogram Upload Bitrate. Source 1511 kbps.

The average contribution from each peer to the overlay is given by (N-K)B/N, where N is the overlay size, K the number of contemporary streams provided by the source and B the stream bitrate. The asymptote is reached after the source saturates its resources (first peers ask the source). Then, any new peer that wishes to download the stream must give on average an equal upload contribution to preserve the overlay flow balance. Since resources in PlanetLab are not homogenously distributed it is interesting to see how the protocol assigns tasks in a large overlay. For IPTV quality (Fig. 3) some peer contributes for 13Mbps vs the majority that uploads less than what it downloads. For lower bitrates, some peers behave like mirror suppliers for the rest of the overlay.

B. *Video Performance*

The VLC process duration and the average CI characterize video performances at the end user side. As we said before, we

witnessed a huge variability in the video performance in our network.

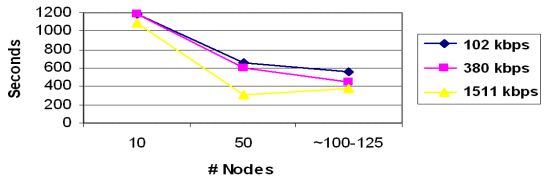

Figure 4. Average VLC duration in a steady environment.

We observed how long VLC lasts before crashing. As we already observed, this is an empirical indicator of the amount of resources required at each endpoint. To do this we proceeded in the following way: we first choose a group of 10 nodes (first points in figure 5) that show a more robust behavior ("alive" nodes), we then add the rest of the peers.

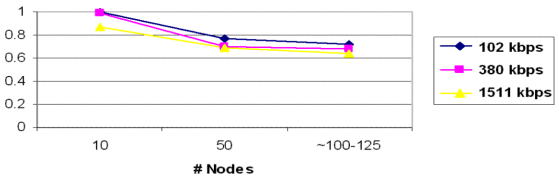

Figure 5. Average Continuity Index in a Steady Environment.

With 20 minutes sessions, as expected, as the overlay size increases, the average lifetime value considerably decreases. A surprising results is that the "good" nodes suffer the computational load testing the IPTV scenario at 1,5 Mbps. Keeping the same approach we observed the CI accounting for the read/loss frames while VLC is running. The "good" nodes perform well while streaming the first two bitrates, but significantly deteriorate in performance at 1,5 Mbps.

IV. DYNAMIC OVERLAY EXPERIMENTS

The nodes we defined as "good" nodes in the previous experiments are now referred as probe nodes (PN). These node's clients are up and running for the whole experiment timeline, we monitor the probe nodes for the whole duration of the experiment while the rest of the overlay dynamically changes. This experiment tries to emulate the behavior that would be seen when a set of end users watch their favorite channel, though sharing the content for a long time session while other end users hop from a channel to another. In this scenario the nodes change their status at specific session instants while the PN, around 10-15, share the content for the whole duration of the experiment. After 5 minutes, the time necessary to reach a "steady" environment, the overlay enlarges at steps of ~25 nodes every 30 sec until it reaches a population of 100~120 peers. Then, with the same rate, the overlay drops back to the original size. Moreover, it is important to point out that we limit the source upload bandwidth to 1 Mbps to stress the overlay's performance and verify how well the mesh network is able to organize connections to provision all clients. Nodes randomly join and leave the network during both the ascending and descending phase. We observe that stepping from 10 to 30 nodes has severe consequences than stepping from 120 to 100, the shorter the interval, the harder the hit on video performances.

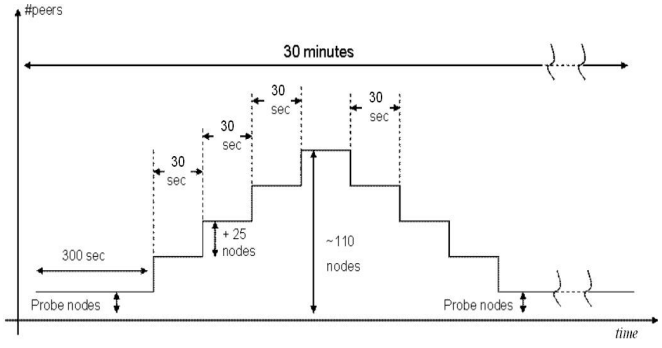

Figure 6. Logic scheme of the Dynamic experiment.

1) *Continuity Index analysis*

As we mentioned before, in this session we applied a bandwidth limiter on the source of 1Mbps, thus, with a 380 kbps bitstream, the source cannot feed more than 3 nodes. During the first 300 sec the overlay becomes stable (only the probe nodes are up).

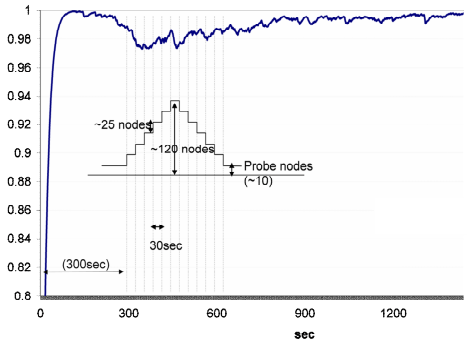

Figure 7. Avg PN CI(*t*). Intsec=30 sec. 1Mbps Upl filter on the source.

Fig. 7 represents the CI for a dynamic experiment where the peer population increases every 30 seconds. The curve has been drawn averaging over time across all nodes and experiments. Network reconfiguration may affect video performance on some nodes more than others. The curve says that the most dramatic change happens at the beginning of the experiment, when the overlay size doubles.

2) *Traffic Statistics analysis*

Parsing tcpdump traces we observe the instantaneous bitrate. PN download curves (Fig. 8) have a pretty similar shape: a burst of chunks is downloaded at the beginning, while the amount of chunks downloaded after this phase varies as the overlay is increases in size. The first peak is a reasonable protocol design's choice to fill the buffer as soon as possible. Instead, a rise occurs when the overlay starts to enlarge. Since the bitstream is 380 kbps, the download traffic is higher than needed for the pure video rendering. This inefficiency is caused by the sudden arrive of multiple peers all at once, causing the PN to download multiple copies of the same chunk.

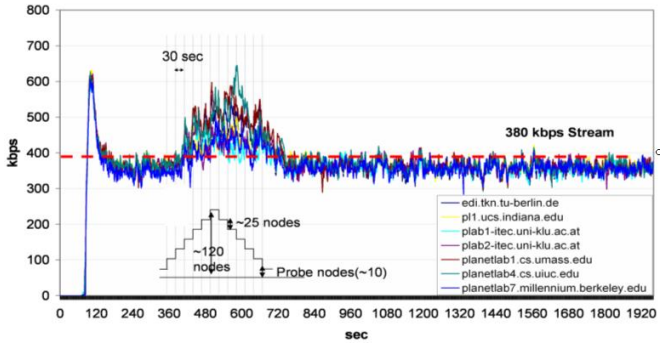

Figure 8. Dwnl(t) of PN. Intsec=30. 1Mpbs Upl Filter.

In general, good performances depend on having good parents providing all chunks in the right order. We found the parents supplying the 80% of the content for each peer and the contribution coming just from the source, which we discovered being the first supplier for most of the probe nodes. A slight dependence by the geographical distance from the source is visible in Fig. 9. European nodes rely on the source less than US nodes, although UCLA is still the best parent in terms of reliability.

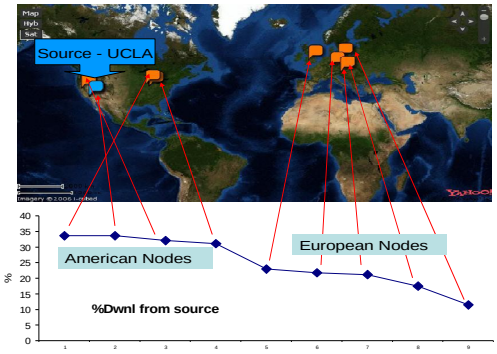

Figure 9. Source contribution to the PN and geographical correlation.

Aside bandwidth and computational power, Skype supernodes [3] become so when they are online most of the time. SopCast doesn't choose parents according to their reliability; these appear to be randomly chosen. In Fig. 10 we show a single client's session length (in seconds) and its contribution in terms of upload bandwidth to the overlay network. The limited bandwidth on the source means that the overlay must cooperate to share chunks. The peak upload bandwidth again corresponds to the UC Riverside, which has an average upload bitrate of 2 Mbps, far higher than the other probe nodes that are online for the entire session. In case of full server bandwidth the contribution from probe nodes reduces in absolute terms, but increases relatively to the overall contribution (source is not included). In this particular experiment we have at the beginning of the dynamic session two different aspects that determine two opposite effects: an event changing the overlay configuration and a broader number of possible "parents" after each step.

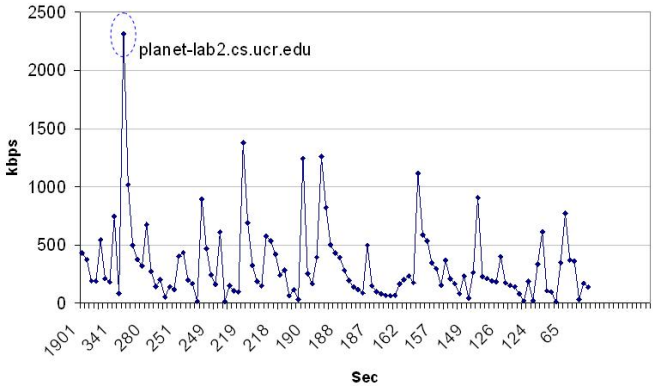

Figure 10. Upload vs Sopcast Duration for one single experiment.

Every event determines a request of chunks from new peers (either increasing or decreasing the overlay size because some peers may loose contact with the old parents and ask around for new ones), while a bigger overlay means a bigger set of possible parents. Stable peers are the ones who have joined the overlay for enough time, estimable at 1.5 or 2 times the buffer size (in seconds). Designers must keep in mind the kind of service that clients need to deliver in order to set the right buffer dimension. An important football match, though critical and subject to very low user-tolerance, might be less risky than a song festival because users follow the event until the end without changing the overlay configuration.

CONCLUSIONS

Results in this paper are far from being able to give a definite answer on the P2P approach for IPTV solutions, nevertheless, they open a number of questions that need to be answered to go in such direction. Processing power, as network connectivity, might be an issue on devices such as STBs. Network strategies should be tailored to the application. Broadcasting a live event is different than broadcasting a regular TV show. In one case we might be more worried about the delay at the receivers, while in the second case there might be a higher chance of hopping from a channel to another. A better understanding of mass behaviors can lead to more efficient solutions. Clearly, "best-effort" random peer download schemes cannot fit this scenario, while they are excellent for low quality WebTv applications.

ACKNOLEDGEMENT

This publication is based on work performed in the framework of the Project SEA IST-214063, which is partially funded by the European Community.

REFERENCES

[1] Electronics.ca: "IPTV Global Forecast - 2008 to 2012" 2008

[2] Strategy Analytics. "Europe's High Definition Homes: High Definition TV and Video Devices Forecast." 2007

[3] S. Baset and H. Schulzrinne. "An analysis of the skype peer-to-peer internet telephony protocol". Technical Report CUCS-039-04, Computer Science Department, Columbia University, New York, NY, Sep.