Network Friendly P2P-TV: The Napa-Wine Approach

L. Abeni¹, Á. Bakay⁴, M. Biazzini¹, R. Birke², E. Leonardi², R. Lo Cigno¹, C. Kiraly¹, M. Mellia², S. Niccolini³, J. Seedorf³, T. Szemethy⁴, G. Tropea⁵

¹University of Trento, Italy; ²Politecnico di Torino, Italy; ³NEC Laboratories Europe, Germany; ⁴Netvisor Ltd, Hungary; ⁵LightComm s.r.l, Italy

I. BACKGROUND & SCENARIO

P2P-TV systems have become part of the Internet landscape¹. The architecture of these (normally proprietary) applications is generally receiver-driven, in that receivers actively search for suitable peers to download from, trying to maximize their performance. This results in very aggressive applications that generate huge and non optimized traffic loads.

Fig. 1. Logical architecture of a P2P live streaming system interacting with an ALTO server

The Napa-Wine EU project² proposes an architecture where P2P-TV clients exploit network measures to reduce their network footprint and, if available, exploit ALTO-like services to optimize topology and performance, and to minimize the overall network load. ALTO³ is a proposal for supporting the interaction between network operators and P2P applications, whose goal is providing applications with enough information on the network structure to enable the optimization of the overlay topology and the content dissemination, while minimizing network costs. Fig. 1 pictorially represents the logical structure of P2P Live Streaming applications using ALTO for optimized peer selection. Each peer, when discovers new peers to update its neighborhood, sends the list of these candidates peers to an ALTO server, which returns either a sorted list of the candidates, or the same list with some metrics associated.

II. THE NAPA-WINE CLIENT

Fig. 2 presents the functional architecture of a Napa-Wine P2P-TV client, which is based on chunkization of the media and swarming on a generic mesh overlay. The client is based on generic libraries we developed, and is organized in different modules, which are briefly outlined in the following.

User Layer. It is the module in charge of ingesting and rendering the media. It does support different encoding, but most of all it takes care of smart chunkization of the stream, so that priority diffusion can be implemented by schedulers.

Topology Management & ALTO Interaction. Building the overlay for a real-time P2P application is one of the most critical tasks. This module takes care of optimizing the overlay taking into account both application-level performance and network-level costs, also interacting with an ALTO server to obtain information that cannot be accessed at the application level (e.g., link costs).

Chunk Trading. The exchange of chunks on the overlay requires both signaling capabilities (e.g., to exchange buffer maps) and scheduling capabilities to optimize the diffusion of chunks. On the one hand, this means careful selection of chunks so that they diffuse evenly and, on the other hand, it means selection of peers based on network-level metrics to meet performance goals while minimizing the network footprint.

Monitoring Layer. Metrics like the RTT, the number of hops from other peers, the available bandwidth, etc. requires network-level measures, which are implemented in this module. Some metrics can be retrieved from an external repository instead of measured locally.

Messaging Layer. An abstraction of the networking interface is provided by the ML, so that other modules do not have to take care of detailed networking functions like NAT traversal, shaping (if required), peer naming, chunk and message segmentation and reassembly, etc.

The Topology Management and Chunk Trading are implemented on top of a set of generic libraries (GRAPES – Generic Resource Aware P2P Environment for Streaming) which provide all the basic functionalities needed for peer sampling, for chunk scheduling, for exchanging buffermaps and chunk ID sets, for sending and receiving chunks, etc. The
GRAPES libraries only depend on some basic functionalities for sending and receiving messages, and the Messaging Layer described above has been used for this purpose. Programs based on GRAPES just need to implement some basic logic gluing the libraries’ functionalities together (this is generally implemented as a simple control loop receiving messages from the network and passing them to the libraries).

III. DEMONSTRATIONS & MEASUREMENTS

The goal of this demo is to show the impact of various P2P streaming options and the efficiency of the Napa-Wine approach (compared to more traditional approaches) by running real streaming clients in realistic conditions. To make this comparison possible, the software developed in Napa-Wine is highly modular and configurable, allowing the user to test different topology management and chunk trading techniques developed within the Napa-Wine project, as well as to configure it to mimic other chunk/peer selection strategies known from literature. The Napa-Wine P2P-TV clients (with different dialects and versions, testing the impact of different policies and architectures) are already running over the Internet in about 100 physical nodes hosted by the project partners, and our demonstration reproduces the most significant results achieved through these tests. To this end, some live streams are distributed over the Internet. TV sources are DVB channels transcoded directly from standard terrestrial and satellite distribution systems, and the video quality can be selected to run experiments with different bitrates and frame sizes (one of such possibilities is to distribute the DVB stream without transcoding).

Since large scale, distributed experiments over the Internet are complex, and measures collection is a challenge in itself, we developed an ad-hoc repository for the collection of data and the visualization of the system properties and measures.

4Each physical node can host more than one logical peer, so that large scale experiments can be run easily.