Simulation of P2P TV system using OMNeT++

Arkadiusz Biernacki
Chair of Communication Networks
Chemnitz University of Technology
Chemnitz, Germany
Email: arkadiusz.biernacki@gmail.com

***Keywords*-Network Architecture and Design; Distributed applications; Wide-area networks; Emerging technologies;**

I. Introduction

The use of peer-to-peer overlay networks (P2P) to deliver live television on the Internet (P2P TV) is gaining increasing attention. P2P TV systems are growing nowadays at increasing speed as a result of the opening of video content producers to new business models. This process is supported and accelerated by larger bandwidth availability of Internet and better algorithms for compression, reproducing, and receiving video streams.

Traditional Internet TV services based on simple unicast approach are restricted to small group of clients. The overwhelming resource requirement makes this solution impossible when the number of user grows to thousands or millions. By multiplying the servers and creating a content distribution network (CDN) the solution will scale only to a larger audience with regards to the number of deployed servers, which may be limited by the infrastructure costs. Other solution is based on multicast protocols, unfortunately their deployment and availability is currently limited.

Therefore the use of the P2P overlay networks to deliver live television on the Internet is achieving popularity and has been considered a promising alternative to IP unicast and multicast models. Nodes in the network, called peers, exchange TV content between themselves serving simultaneously as clients and servers. In a results, with increasing number of network peers, the number of servers in the network also increases leading to a smoother exchange process. Consequently, such an approach has the potential to scale with group size, as greater demand also generates more resources. The raising popularity of this solution is confirmed by the amount of new P2P TV applications that have became available: PPLive, SOPCast, TVants and TVUPlayer, Joost, Babelgum, Zattoo, and by the fact that the traffic generated by such applications has recently increased significantly.

Although from the users as well as from the broadcasters' perspective P2P applications may be very useful, network operators may not share this opinion due to their concerns about the capability of asymmetric underlying network to support large number of users acting as servers uploading their content. Therefore it seems important to gain an understanding of the potential influence that P2P TV applications may impose on the underlying network. Since the most widely deployed commercial systems cited above have closed architecture and are proprietary, only an experimental behavioural (blackbox) characterisation of traffic injected by such systems is in general possible. Reverse engineering of such systems may be costly and not give answers to all nurturing questions regarding their performance.

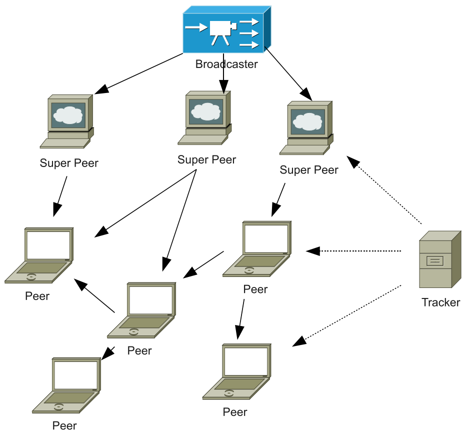
To avoid those disadvantages we propose a simulation analysis of a P2P TV system which allows us fast prototyping, provides the possibility to perform large scale experiments, and offers a common reference platform for experimentation.

II. Simulator

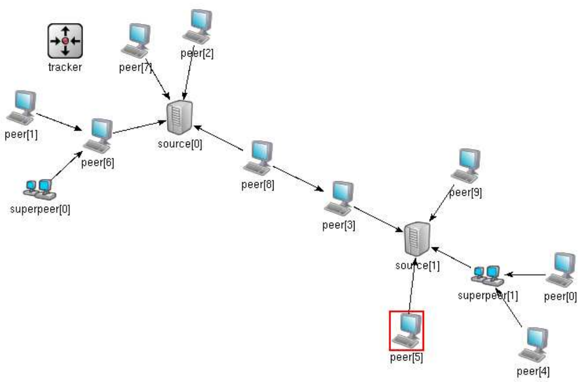
Simulating the operation of P2P TV is a difficult task due to the inherent protocol complexity and large number of possible strategies for peer selection and choking decisions. Thus, in order to isolate important protocol aspects, we ignored the influence of the underlying protocols and focused on the overlay network and application logic being conscious of introduced certain degree of inaccuracy in our simulation. Underlying transport protocol characterisation, dynamics, propagation delays, the potential queueing of packets in routers and geographical location of the system nodes are important factors that can affect various aspect of the simulation e.g. peer selection decisions.

The system our simulator implements is based on real P2P TV delivery platform GoalBit [1]. GoalBit is capable of distributing high-bandwidth live-content using a BT-like approach where the stream is decomposed into several flows sent by different peers to each client. Goalbit has been actively developed and its references are present in academic papers.

There are four different types of components in the GoalBit network: broadcasters, super-peers, peers, trackers, see Figure 1(a). The broadcaster is responsible for the content to be distributed i.e. for getting it from some source and to put it into the platform. This critical component plays a main role in the streaming process. If it leaves the network, the streaming obviously ends. The super-peers are highly available peers with large capacity. Their main role is to



(a) GoalBit



(b) Simulation

Figure 1. System components

help in the initial distribution of the content – they get the data streams from the broadcaster and distribute it to the peers which are the final users of the system. The peers, representing the large majority of nodes in the system, are the final users, who connect themselves to the streams for playout. The tracker is in charge of the management of the peers in the system. For each channel the tracker stores a reference to the peers connected to it. A client periodically learns about other peers connecting the tracker and parsing the peer list returned. The client joins the swarm by establishing connections with some of those peers.

All the above components were implemented in our simulator as descendants of *cSimpleModule* class, however with functionality reduced to establishing and tearing down connections, monitoring the bandwidth status and basic control traffic exchange, see Figure 1(b). Control traffic messages, based on *cMessage* class, are simplified to several fields including the sender, recipient, type of message and its value. The audio-video traffic is represented as a certain status of pairs of nodes thus the diffusion of data chunks is represented as a continuous flow – we do not distinguish between individual chunks. Every network node is described by several attributes, amongst them: performance, bandwidth limitation, coefficient of audio-video stream repeatability, and maximum number of incoming and outgoing connections. Other globally controllable parameters involve number of transmitted TV channels, number of network nodes: peers, super-peers and sources. Currently our simulation support only single tracker and does not implement peers churn.

In our approach we used the event driven approach, where a scheduler maintains a list of simulation events. This is opposite to discrete event simulation approach used in some others P2P simulators.

In our simulator the system topology is created dynamically from scratch. Peers and super-peers periodically query the tracker and obtain lists of their neighbours. Each peer creates ranking of neighbouring peers and tries to connect to the peer with the highest upload goodput. If the number of connections is equal to maximum allowed number of connection, peer disconnects from the neighbour with worst upload goodput and tries to reconnect to a better one, thus making the network topology constantly evolve.

We performed an analysis of the influence of maximum number of allowed connection on upload and download goodput of the network peers. The examination results were presented on three different levels: transient behaviour of a single peers, where we monitor a behaviour of a single peer in the function of time; an analysis of average values, where we summarise a behaviour of single peers during the simulation execution; and an analysis of global values, where we summarise and aggregate a multiple peers behaviour in the function of consecutive simulation runs.

REFERENCES

[1] Mara Elisa Bertinat, Daniel De Vera, Daro Padula, Franco Robledo, Pablo Rodriguez-Bocca, Pablo Romero, and Gerardo Rubino. GoalBit: the first free and open source Peer-to-Peer streaming network. In *LANC ’09: Proceedings of the 5th international IFIP/ACM Latin American conference on Networking*, New York, USA, September 2009. ACM.