PeerLive – Multimedia Distribution using Overlay Networks

Massimo Battestini and Dr. Miguel Rio

University College London

Abstract: PeerLive aims to create an overlay network of peers used to broadcast television channels or other multimedia content. PeerLive differentiate from similar projects for some key-features like swarming, QoS overlay, incentives and virtual currency and the ability in uploading early fragments of the content even before being viewed. The attention is focused on aspects such as buffer size, query methodology, delays and incentives and evaluation criteria.

1. Introduction.

Distribution of medium to high bandwidth audio-visual content is typically achieved by analogue or digital broadcast means over terrestrial or satellite distribution systems. However, the air interface is a limited resource and very large quantities of content streams cannot be made available through traditional mechanisms. For example, large events (such as the Olympic Games) present problems to broadcasters who are unable to devote channels to every game.

PeerLive aims to propose a content delivery architecture suitable for broadcaster to provide large numbers of services, including minority-interest programmes. By applying p2p concepts the content providers are freed of the need for deploying expensive high-capacity servers and access link to provide the service. The application layer only approach of PeerLive is another important factor because allows rapid deployment without requiring modifications to the underlying network infrastructure.

2. Key features of PeerLive.

Swarming. PeerLive is acquiring the content through swarming. Swarming consists in getting fragments of the channel from several senders (peers). This can be seen an extension of Application Level Multicasting (ALM) and provides the advantage that the system becomes more efficient since every host can participate independently of its upload capacity. Thus, heterogeneous device support becomes easier.

QoS Overlay. Internet only provides a best-effort service. The overlay p2p network will therefore be used to implement added value QoS-based services. Our approach in providing QoS has two aspects: the first focuses on the identification of suitable peers, based on fragment availability; the latter on network performance metrics such as bandwidth and delay.

Incentives and Virtual Currency. Since p2p swarming requires end systems to participate in the data replication process, its efficiency may be compromised by the fact that end systems may behave selfishly by limiting their participation in uploading stream fragments. We propose a virtual currency based mechanism to create incentives for cooperation. A peer wishing to receive data, register with the original sender and periodically receives virtual credits. These credits are then passed in exchange of the data feed from any receiver. The receiver of credits may then use them to pay for its own feed or to buy a better QoS (negotiating a better and more expensive place in the distribution tree with less delay). All the above stimulates peers in having children to gain more credits, and, therefore, more services.

3. PeerLive simulator.

PeerLive simulator has been used as a test bed where to try and test the efficiency of our solutions with a network formed by an arbitrary number of peers. In several occasions, we have considered a population of 100 peers, which represents a medium to small network, and is perfect to emulate a real contest.

3.1. Network traffic.

When a peer is willing to view a channel, he firstly has to ask for the channel content, spreading a query over the network. Two different kind of content queries are on the study: expanding rings and limited flooding. Expanding ring query consists in associating a *TTL* (Time To Live) to each query message, which decreases passing from one peer to another and that makes the query expire when TTL=0. Therefore, the query keeps local and doesn't weight the entire network. Limited flooding consists in flooding the query from one peer to its neighbours and, just in the case that the neighbour cannot provide feeding, it will flood the query to its neighbours too and so on; this justifies the name 'limited' and permits crossing the network in the search of rare content.

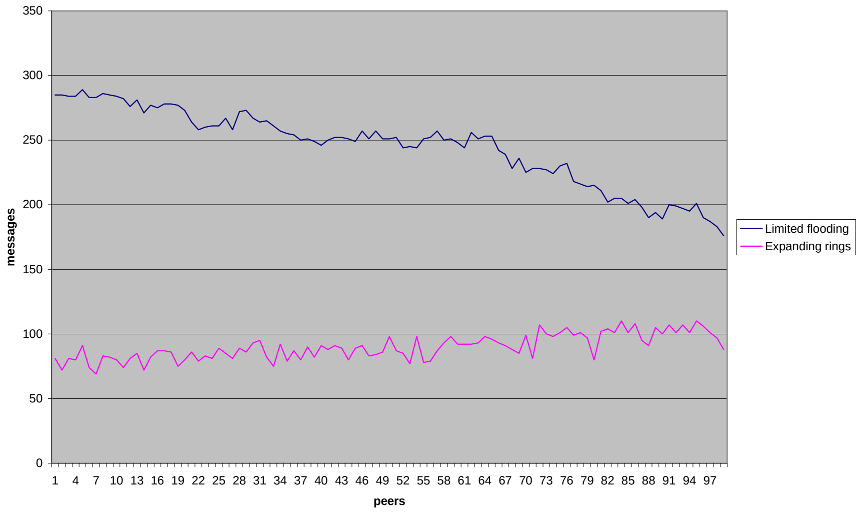

Figure 1 - messages transmitted considering expanding rings and limited flooding queries

Figure 1 is considering a 100 peers network population, where the number of peers that have joined a channel group is raising from 1 to 100. The figure shows the number of messages transmitted during a query at different channel group dimension stages. As we can see, with the expanding ring methodology (TTL=2) the number of exchanged messages is slightly increasing when the number of peers that have joined a channel is rising. In fact, even if the query is still flooded locally, more peers have got the content and can provide feeding, thus, they are now available to reply. Completely the opposite trend is followed by the limited flooding query, where the number of messages is decreasing. In fact, an increased number of peers, that have joined the channel group, means more probabilities for an immediate reply instead of a query flooding.

3.2. Offers criteria.

The answer to a query is always a content availability offer. Thus, in the end the peer who did the query has to choose the best offer among all those received. A critical aspect of the simulator consists in defining the criteria on how to prepare an offer and on how to choose the best one.

Regarding the preparation of an offer, assuming that each peer wants to cover its feeding expenses, the credit wanted for each fragment would be:

$$credits = \frac{expenses}{max\ children} \quad (1)$$

Where *max children* indicates the maximum number of peers that the peer is able to feed.

When a peer has to choose the best offer between all those received, it has to consider two parameters: credits and delay. Some users would be happy to pay more to receive a stream with less delay, others would prefer to save credits, and thus even a stream with large delay would be satisfactory. Therefore, the choosing criteria are strictly linked to user's preferences. The solution we have proposed gives a mark to each offer according to the value of the parameters credits and delay. Thus, if we consider all the offers in a Cartesian graph, offers with the same mark would be represented by an ellipse with the eccentricity varying between 0 and 1, depending on user preferences.

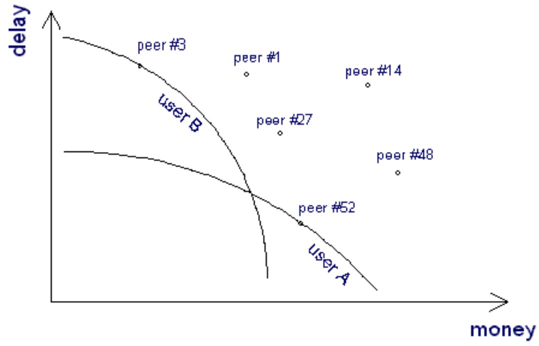

Figure 2 - example of offers choose

Figure 2 is showing the offers' domain. User A and user B have different preferences: user A is willing to pay more for a little delay; totally the opposite is user B, which prefers paying little even if with considerable delay. Thus, equimark lines are represented by two ellipses with different eccentricity. In this example, user A would choose the offer of peer #52, whilst user B the offer of peer #3. According to the ellipse solution, we are using the following formula to assess the mark:

$$money^2 \cdot \left(1 - e^2\right) + \left(k \cdot delay\right)^2 = mark^2 \quad (2)$$

Where *e* is the eccentricity varying between 0 and 1, and based on user's preferences and *k* is a scaling factor of the delay.

3.3. Delays and buffer size.

Another important aspect to evaluate is the delay. The channel stream accumulates delay passing from one peer to another; this is due to both transmission delays between peers and computation delays in each peer. The delay problem is mainly addressed in the selection process of the suitable peers to get the content from. The algorithm described before is responsible in choosing the right channel feeders reducing expenses and maintaining a near real-time stream at the same time.

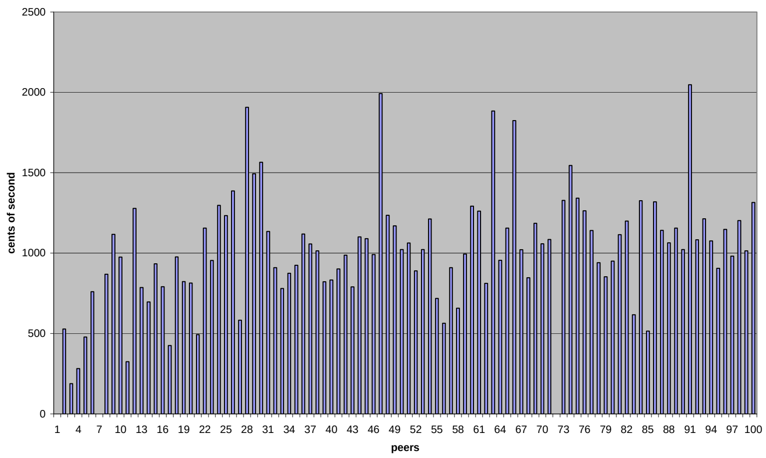

Figure 3 - delays in channel viewing for a population of 100 peers

Figure 3 is showing a delay snapshot in channel viewing in a network formed by 100 peers and considering a maximum transmission and computation delay of 10 seconds. Results are quite good, being possible to view the channel with an average of 8 seconds delay and a maximum delay of 21 seconds.

Each channel is formed by several fragments following different path to reach a peer, therefore carrying different delays. The positive aspect of early fragments is that they can be forwarded to other peers, even before being viewed, making a substantial contribution in lowering delays; but the negative aspect is that they need to be buffered because the channel view can start just after all the fragments are present.

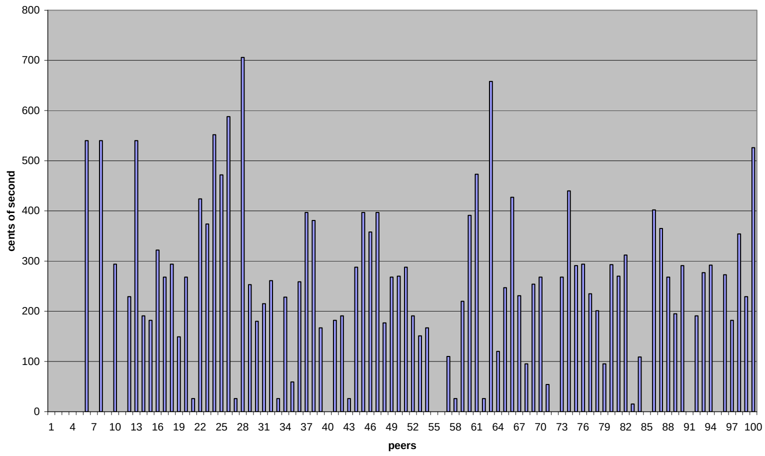

Figure 4 - Buffer utilisation for a population of 100 peers

Figure 4 is showing the buffer utilisation expressed in cents of second, considering a channel stream formed by four fragments. From the simulation, the minimum size of the buffer to guarantee a proper view of the stream is 7.06 seconds. The corresponding size expressed in bytes depends on the stream quality. For example considering a stream of *256kb/s*, it is obtained:

$$256kb/s \cdot 7.06s = 1.76Mb \quad (3)$$

Unfortunately, the minimum buffer size doesn't prevent viewing interruptions in the interval of time a peer loose its connection and restore it (reconnection time); a supplementary buffer and recovering techniques have to be used.

4. Conclusions.

During the research project many practical aspects of the multimedia distribution have been considered like: query spreading methodology over the network, delays and incentives, evaluation criteria, buffer size and network stability.

Particularly, two original solutions are differentiating our research from similar ones: the possibility of uploading early fragments, as soon as they are received before being buffered or viewed; the use of incentives and virtual currency. The former reduces noticeably the stream delay along the distribution tree, but adds complexity and buffer size and recovering techniques after a receiving interruption are still an open issue to be solved in the near future. The latter promotes a fair cooperation between peers and rewards the more altruistic ones.

References.

[1] Yu-Wei Sung, Michael Bishop and Sanjay Rao, Enabling Contribution Awareness in an Overlay Broadcasting System, Sigcomm'06, September 2006

[2] S.Banerjee, B.Bhattacharjee, C.Kommareddy, Scalable Application Layer Multicast, ACM Sigcomm 2002

[3] Alberto Blanc, Yi-Kai Liu and Amin Vahdat, Designing Incentives for Peer-to-Peer Routing, INFOCOM 2005

[4] Y.Chu, J.Chuang, H.Zhang, A Case for Taxation in Peer-to-Peer Streaming Broadcast, PINS 2004

[5] I.Jannotti, J.Gifford, D.K.Johnson, K.Kaashoe and W.James, Overcast: Reliable multicasting with an overlay network, Tech.Rep. TR-101, RSA Labs, 1995