Improving P2P Streaming Methods for IPTV

Majed Alhaisoni
University of Essex
Colchester, UK
malhai@essex.ac.uk

Antonio Liotta
Eindhoven University of Technology
The Netherlands
A.Liotta@tue.nl

Mohammed Ghanbari
University of Essex
Colchester, UK
ghan@essex.ac.uk

Abstract

Peer-to-Peer (P2P) IPTV applications have increasingly been considered as a potential approach to online broadcasting. These overcome fundamental client-server issues and introduce new, self-management features that help improving performance. Recently, many applications such as PPlive, PPStream, Sopcast, and Joost have been deployed to deliver live and Video-on-Demand streaming via P2P. However, the P2P approach has also shown some points of failure and limitations. In this paper we analyze, assess and compare two popular Live and Video-on-Demand P2P streaming applications, Sopcast and Joost. Fundamental shortcomings of existing applications are then countered by our approach where we employ a cross-layered method aimed at improving network efficiency and quality of experience. We show how simple modifications to existing protocols have the potential to lead to significant benefits in terms of latency, jitter, throughput, packet loss, and PSNR.

Keywords: P2P streaming; Ne; Network efficiency; Load balancing; QoS; QoE.

1. Introduction

Peer-to-Peer (P2P) video streaming is becoming a viable alternative to conventional IPTV systems for distributing video content over the Internet. The underlying mechanism is based on the distribution of the stream through a self-managed, application-level overlay including the user terminals in the role of peers i.e., content distribution relays. This is in contrast to other IPTV approaches which are based on content distribution networks. These require a dedicated multicasting infrastructure whose cost increases dramatically with the scale and dynamics of the system. On the other hand, in the P2P approach any capable (user) terminal becomes a distribution hub for any incoming stream, reducing in this way the possibility of failure points and bottlenecks that are traditionally associated with servers. Moreover, as the number of connected users increases, the number of distribution points grows too. As a consequence, the system scales much better than any client-server counterpart.

The P2P streaming concept has nowadays been deployed into several trial P2P streaming systems, such as Sopcast [2], Joost [3], and Zattoo [4]. The online broadcasting arena is then evolving, mainly due to the clear commercial interest for these new technologies. Many hosts can be supported by a P2P multimedia system, possibly in excess of hundreds or even millions, with miscellaneous heterogeneity in bandwidth, capability, storage, network and mobility. However, although most of the P2P applications are designed to automatically load-balance computing resources, they fail to pursue network efficiency.

Accordingly, this article begins by describing experiments to assess the network efficiency in two of the most popular P2P applications that is Sopcast for the real-time broadcasting and Joost for the Video-on-Demand (VoD) services. By analyzing traffic traces we calculate and compare the network efficiency of these systems, in terms of network locality and percentage of P2P traffic (as a fraction of CS (Client-Server) traffic). The percentage of P2P traffic, gives another form of computational efficiency since the aim of P2P systems is to minimize server intervention. We find, however, that due to the particular nature of P2P streaming, it is not always possible to do without server support and that various systems address this issue in radically different ways. P2P traffic percentage gives an indication of both computational and network load balancing. We also look at the latter property from the view point of network locality, which is the ability to keep traffic local (in addition to being spread out). Our analysis reveals that network efficiency is being exploited poorly in the conventional P2P streaming systems, which indicate new opportunities for designing future applications aiming at network computational and cost efficiency.

Following our analysis of two representative P2P platforms, we then propose ways in which simple modifications to existing protocols have the potential to lead to significant benefits in terms of latency, jitter, throughput, packet loss, and PSNR (Peak signal to noise ratio). The important aspect of locality shows up in our results as it confirms that user satisfaction and network locality cannot be treated independently. For instance in order to satisfy more users, network resources should be treated efficiently which will in return help in supporting more users. Our method aims at improving network utilization and locality whilst at the same time not degrading computational load balancing. The proposed scheme is run under the ns-2 simulator [27].

2. Related Work

Many studies have been published about P2P streaming, but very few actually focus on the analysis of the inter-relationship between computational and network efficiency deriving from self-managed, P2P networks. Existing work aims at understanding the underlying algorithms of current applications which are mostly proprietary. Single-systems analysis has been carried out, like for example in the case of Joost. Consideration on network utilization and locality are given in [5] and [6].

Other studies are based on the comparison of two or more P2P applications for video streaming [7]. However the focus is on examining similarities and differences from the structural point of view, looking at protocol utilization, percentage of download and upload traffic, and signaling overheads. This is the case, for example, of the work published by Silverston and Fourmaux about Sopcast, TVAnts, and PPlive [7], and by Ali et al. about Sopcast and PPlive [8] [9]. Other studies have considered different applications such as [10] and [11]. Moreover, Gustavo Marfia et al [12] have conducted an experimental study on Video-On-Demand system and they were concerned about the performance of the system on different environments such as campus and residential. Thus, they gave some results about protocol utilization and the start-up delay. Additionally, they gave brief descriptions for the existing architectures such as Tree and Mesh.

On the other hand, ways to pursue efficiency between overlay and underlay networks have started to be investigated only recently. Authors in [13] propose a technique, where the peers on the overlay are chosen based on their mutual physical proximity, in order to keep traffic as localized as possible. A similar approach is described in [14], where they measure the latency distance between the nodes and appropriate Internet servers called landmarks. A rough estimation of awareness among nodes is obtained to cluster them altogether, as in [15] [16].

Overlay locality is studied also by [17], where the authors make use of network-layer information (e.g. low latency, low number of hops and high bandwidth). We use though a different distance metric, based on RTT (round trip time) estimations, to prioritize overlay transmissions. Additionally, we use a cluster management algorithm whereby communicating peers are forced to periodically handover, in order to pursue computational as well as network efficiency (as explained in [1] and [23]).

Hefeeda et al [18] have proposed a mechanism for P2P media streaming using Collectcast. Their work was based on downloading from different peers. They compare topology-aware and end-to-end selection based approaches.

The latter approach is also the subject of [19], which employs a simpler version of our RTT approach based on continuous pinging of peers. Similarly, we adopt clustering to limit the signaling overheads associated with this process and prevent bottlenecks.

Other studies such as [20], propose relevant methods to serve multiple clients based on utility functions or clustering. A dynamic overlay capable of operating over large physical networks is presented in [21]. In particular, they show how to maximize the throughput in divisible load applications.

Looking at previous studies, we can say that our main contributions are:

1. To complement other on-going studies with a view to better understand the impact of P2P streaming onto the network. Our main attention is on network efficiency and locality, with the aim to identify intrinsic deficiencies of existing platforms.
2. To propose a new approach to improve network locality.
3. To study a new combination of existing techniques (cross-layer optimization, localization, switching over, forced handovers).
4. To take the perspective of the network operator, in trying to harmonize overlay and underlay networks

5. To quantify the impact of the network-aware approach on QoS (Quality of Service) and QoE (Quality of Experience) factors.

3. Experimental Setup

Our experiments were conducted in the United Kingdom. Measurements and data collection have been carried out during Euro 2008, a major event when thousands of users adopted P2P Sopcast application to watch football matches live. We collected a unique data set, measuring most of the Euro games. We have collected 1.10 GB and that was observing the whole match which is around 1 hour and 30 minutes.

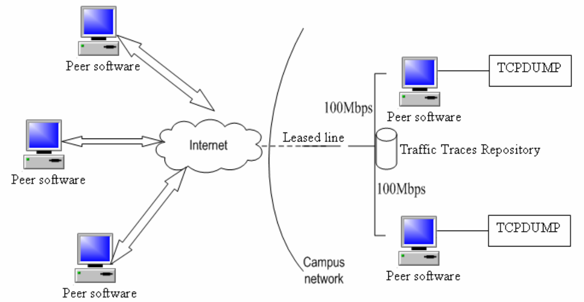

Figure 1 - Experimental Environment

On the other hand, VoD data was collected from P2P application Joost version 1.1.4. The overall size of our trace files was 277 Mbytes. Moreover, it is worth mentioning that the collected data for both live and on demand includes the video of the game which was sent to our client in UK. However, Figure 1, outlines the set up of the two machines used for the collection of traces, which were then used for packet analysis. The machines were connected to a 100Mbps Ethernet, connected to the campus Internet leased line. This ensured that both inbound and outbound bandwidth were considerably higher than the minimum required for the correct functioning of Joost and Sopcast which is 300-350 kbps for Sopcast and 500 kbps for Joost [5].

4. Performance Metrics

Every single application can make use of different strategies in order to provide a video streaming service. However, there are some features that a P2P streaming system should have in order to fully benefit from resource distribution and load balancing. First of all, for efficient network utilization, locality is one of the main goals. Network locality is the ability to maintain the P2P overlay in such a way as to create logical connections among peers who are physically close to each other. The ideal condition occurs when the most intensive data exchanges happen among nearby peers.

4.1. Network Efficiency

In order to evaluate the network efficiency of the two scrutinized applications, we first had to come up with a way to measure it, taking into consideration factors that have an influence. Our approach has been to weight positively those characteristics that improve efficiency and *vice versa*.

We considered 15 observation windows, each of one minute duration. Within each window we considered the relative distance among all pairs of peers in terms of average Round Trip Time (RTT) and its relation with the amount of exchanged traffic volume (expressed in Mbytes). The result is weighted by the minimum RTT obtained during the whole observation period (15 minutes) divided by the total of traffic expressed in Mbytes. Moreover, it is commonly agreed that the lower is the "RTT", the more would be the offered data rate (bandwidth). Thus, peers offering lower RTTs are considered as having a higher available bandwidth and that may be shared fairly among the services.

$$Ne = \sum_{i=1}^{n}\left(\frac{t_i}{\sum_{j=1}^{n} t_j} * \frac{\min(d_i)}{d_i}\right) \qquad \text{[Eq1]}$$

We have come up with Eq 1 which gives an account of network efficiency.

Where Ne is the network efficiency and i = 1 …. n identifies the set of connected hosts; t_i is the traffic coming into the node under scrutiny (or client node) from each peer *i*, (expressed in Mbytes); $\sum t_j$ represents the total traffic coming from all the connected hosts (considering all the observed peers in a window); d_i represents the distance between the intending client and each peer *i*, expressed as Round Trip Time (RTT), min (d_i) indicates the minimum distance between the client and all other peers across the observed window. Hence, we can define the efficiency factor for each of peer in a minute window.

4.2. Peers vs. Servers

In order to ensure uninterrupted streaming experience, P2P applications make use of servers that kick in the streaming process when P2P traffic cannot meet the necessary delivery deadlines. Hence, another measure of efficiency is the ratio between client-server (CS) and P2P traffic incurred in the network. Clearly, an efficient system is the one in which most of the traffic comes from peers and, particularly, when peers are physically sitting close to each other.

Unfortunately, we could obtain an exact measurement of this CS to P2P ratio only in the case of Joost, as Sopcast seems to rely on foreign companies (or websites) that supply CS content from inaccessible locations. Nevertheless Sopcast makes also use of a large number of back-end servers in order to increase quality and availability.

5. Results

In this section, we show our results categorized by type of streaming models, which is Real Time and Video-on-Demand. We look at two difference scenarios: a peak-demand scenario exemplified by the Euro 2008 event, and normal operational model.

5.1. Real Time

We have based our observations on Sopcast, which is currently a popular real-time P2P streaming application.

5.1.1. First Scenario (peak demand)

The Euro 2008 event gave us a unique opportunity to assess the system behavior, particularly network locality, when a large number of users connected globally.

Given that our test-bed was in the United Kingdom, we wanted to make sure that a large user population was based in Europe to be able to collect a representative data set. Hence we recorded traffic traces during the match between Czech Republic and Switzerland. It's noticeable from figure 2 that the network locality (as measured with Eq. 1) starts at low values but gradually increases as it manages to prioritize connections based on the mutual distance of peers.

This positive trend is, however, reversed when the number of peers starts increasing more rapidly, that is after about the first 7 minutes of the match. Figure 3 shows a significant increment of peers (*Note that the total number of peers of figures 3, 5 and 7 relate to the peers that connect to our UK test-bed. This number is much smaller than the total number of peers connecting to the P2P system but it gives an indication of the trend of peer number and churn*).
after 7 minutes which is followed by a sudden decrease in network efficiency (Figure 2).

When we looked a bit more closely at what happened during this sharp fall, we realized that not only peers were increasing but there was also a significant churn of peers *(peers leave and join the session regularly)*, that is many peers where disconnecting while even a greater number was actually joining the video stream. Between minutes 8 and 12 the churn of peers was still quite considerable and Sopcast did not manage to improve on network efficiency. A fall in connected users between minutes 12 and 13 was followed but a sharp improvement in efficiency but this was reversed by a sharp increase of users at min 14.

These results tell us that Sopcast manages to pursue network efficiency even in the large scale, but only if churn is limited.

5.1.2. Second Scenario (normal operation)

The second scenario that we studied was during an ordinary show. The most apparent result from Figure 4 is that, apart from the initial start-up period, the system manages to maintain a more stable level of network efficiency.

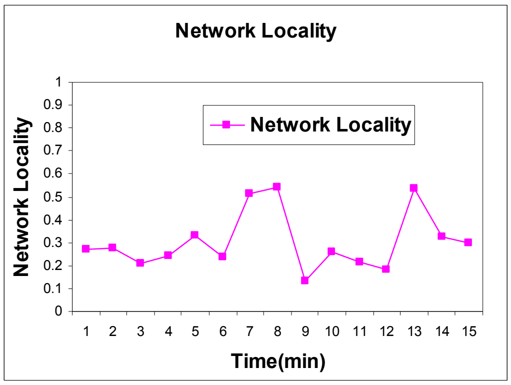

Figure 2 - Network Locality (Sopcast) (Euro 2008)

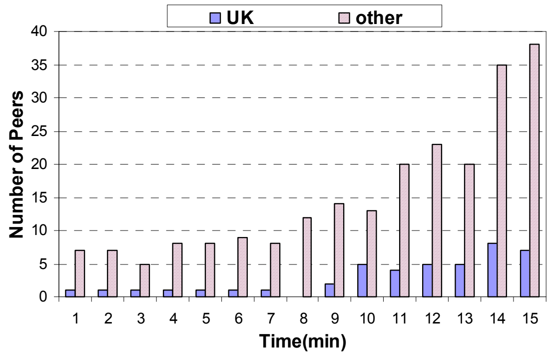

Figure 3 - Number of Peers (Euro2008)

This is because, at steady state (i.e. once people have decided that they will watch the show) the level of churn is limited and Sopcast can manage the overlay more efficiently. Looking more closely at what happens during the first 4 minutes; we can notice that this is the time users are making a decision as to whether they will watch the show. Figure 5 indicates a sharp decrease in watchers. Our in-depth observations also showed that there was also significant churn, hence the poor network performance.

We also notice that, although most of the traffic is coming from outside the UK, there is a good level of locality. This was due to the fact that the network was relatively uncongested with relatively low values of RTTs.

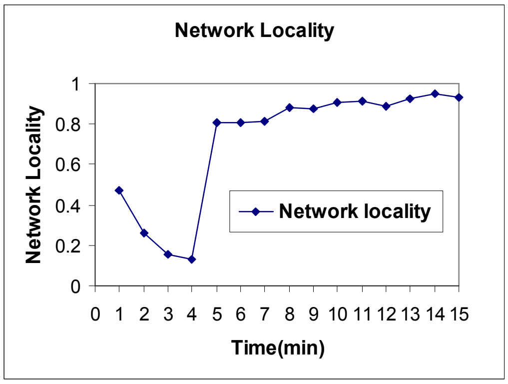

Figure 4 - Network Locality (Sopcast) (Ordinary Day)

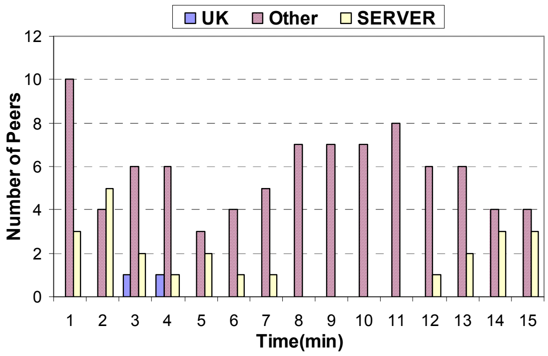

Figure 5 - Number of peers (Ordinary)

5.2. Video-On-Demand

Video-On-Demand traces were taken from Joost which has emerged more recently but is already gaining significant attention. Joost operates in a rather different way than Sopcast [5] [6]. It employs a statistical load-balancing algorithm which seems more efficient than other systems from the computational viewpoint [5]. Our study shows, however, that this benefit is achieved at the expenses of network efficiency. Our experiments were conducted running Joost in our UK-based test-bed. Traces relating to the most popular Joost channels were collected. Equation 1 was used to benchmark network efficiency as for the case of Sopcast.

By looking at the combined results from Figures 6 and 7 we can draw some lessons about the approach followed by Joost. The most apparent difference with the results obtained from Sopcast is that in Joost there is a continuous fluctuation in network efficiency. This results from the fact that Joost continuously forces handover among the intercommunicating peers, with the aim to maximize computational load-balancing.

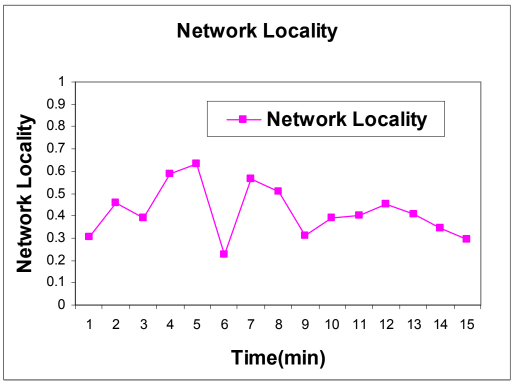

Figure 6 - Network Locality (Joost)

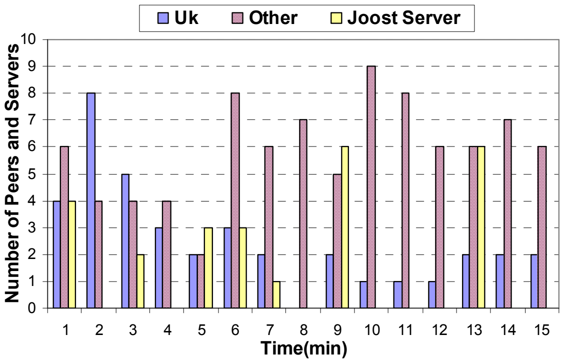

Figure 7 - Number of peers (Joost)

Because inter-communicating peers are forced to handover, network efficiency is dominated by varying link conditions (i.e. RTT values).

To counter the best-effort nature of those links, Joost deploys streaming servers in strategic locations. Currently there appear to be 2 servers in the US and 2 in Europe (one of which in the UK). Each server has many clients to upload the content to the receivers and this is clear from figures 6 and 7 where we sometime notice number of servers up to 5.

When those servers kick in, network efficiency drops, as is visible for instance at minutes 6 and 9. Figure 7 actually shows that the server-originating content is considerable in Joost. Also the amount of P2P traffic coming from outside the UK is significantly higher than local traffic. Again, this indicates the network-unfriendly behaviour of Joost. Hence, even though the user-perceived quality is good, this is achieved through an inefficient means, at least from the network operator's point of view.

6. Proposed Approach

The analysis of most of the existing popular P2P streaming applications, indicate that network efficiency (locality) is not given much attention. Our experimental results showed that Joost's locality is poor and this due to the random connect and disconnect policy, which incurs a negative effect on network locality. On the other hand, Sopcast achieves good levels of network efficiency but only in relatively stable conditions. High levels of churn lead to a considerable degradation in network efficiency. By contrast, Joost is not network efficient due poor locality. Also, a substantial traffic has to originate from servers, to compensate for the best-effort nature of the network and also for poor network locality (the RTT between intercommunicating links is sub-optimal).

Therefore, it is needed to take advantage of the network locality and the load balancing in P2P streaming applications. In this section, a new method for peer-to-peer streaming is proposed and its improved network locality is verified. In addition, QoS and QoE factors were examined and showed additional improvements in different metrics.

6.1 Proposed method

According to our experimental data, we found that a client connects to different peers on the overlay network across the globe. To make clients locally aware, different distance metrics such as RTT, number of hops, and the geographic location may be imposed on connections to maintain the relations between the participant peers. In the proposed method, RTT has been used as a tool to reflect the locality of other peers to any client. A cross-layer approach has then been implemented between the overlay and the underlay network to obtain the RTT values of the participant peers. In our work, though, we have not really focused on new RTT monitoring techniques since this is actually ongoing research topic that is being studied by many authors such as [24] [21]. One way of estimating RTT values is the monitoring method for the intending nodes. Details of various monitoring methods have been published in [24] [25] by one of the authors. Moreover, this is managed by clustering the peers into groups and every cluster is lead by a Super-Node likewise KaZaA. Thus all the queries will go through these Super-nodes instead of tracking all the peers across the network. This strategy helps in avoiding of extra signaling overhead. The aim of this method is to improve the network locality (efficiency), which will be demonstrated in next sections.

6.1.1 Network locality

As mentioned earlier, network efficiency (locality) is the ability to keep the traffic as local as possible, which can be achieved by considering the peers which are nearby with varying the sources among the participants. Therefore, in the proposed method, a decision is made among the participant peers based on the offered RTT values by the monitoring system. Peers are prioritized based on the lower RTT values, and the connections are setup based on the RTT values. Consequently, this will not only maintain the network locality among the inter-communicating nodes but it will also improve the QoS and, hence, the user's quality of experience (QoE).

However, offering network locality only without varying the sources among peers, will be drastically affecting load balancing or in other words, the load distribution between the network and computing sources. Therefore, different techniques are embedded to the proposed method. The main aim of these techniques is to distribute the load among the participants and at the same time having the network locality not affected. This can be shown in the next section.

6.1.2 Computing and network resources load

In order to maintain the load balancing among the contributor peers, different handover techniques have been embedded into the proposed approach. Two conditions trigger handover among interconnected peers:

Switching over: Since the network may experience various constraints such as congestion, bottleneck and link failures, the RTT values will be severely affected and may not be reliable. Additionally, these stochastic conditions will drastically affect the network locality and degrade quality of service (QoS) parameters such as throughput, packet loss, and end-to-end delay. There is also another important requirement arising directly from the adoption of P2P: peers are not reliable entities and cannot be assumed to be always connected. Nodes may leave and join at unpredictable times. So, we must adopt a mechanism which allows receiving peers (in client mode) to maintain a smooth video, although the streaming peers (in server mode) are not constantly available.

One solution to this problem is that any intending client should regularly update the peers' list and re-order them based on the lower RTT values. In our implementation, we keep a ranking list based on RTT distances. Each peer streams to and from 3 other ones. Hence, when this list changes, we apply a switch-over to the new top 3 peers (those with minima RTT to the peer under consideration). This approach has been chosen according to previous findings published in [4] where we found that the average of the active peers that usually a node is downloading from is 3 to 4. Therefore, in this model, the maximum number of sender nodes has been set to be three.

Enforced handover: Another favorable property in the proposed method is the computational efficiency. This can be achieved when the load is periodically distributed among the peers. However, under normal network conditions, peers with lower RTT are selected, but if the condition changes, switch over is applied to the new peers with lower RTT values.

However, some peers may not experience any constraints such as congestion, bottleneck, and link failures. The RTT values will not be affected and may not be changed, so those peers may become the best in every periodical check. Therefore, selecting them regularly would impair the load balancing between the computing and network resources among the peers and the network locality, so an enforced handover is applied.

Furthermore, to avoid pure randomness on the enforced handover process, network locality is applied into clusters of peers, named super-peers, similar to the one adopted in KAZA [27]. Thus, peers are grouped and they are managed by a special peer, or super node. Our experiments have confirmed that the peers on the same cluster share nearly the same RTT values.

7. Performance Evaluation

7.1 Simulation setup

The proposed method has been tested using the ns-2 simulator [27]. The network topology considered for simulations is shown in Figure 8. Moreover, different parameters have been set on for the target topology. First of all, links bandwidth is distributed to all the links evenly as 2Mbps for every link with the same delay; so, all the participants' peers have the same properties. IP as the network protocol and UDP as the transport protocol have been chosen. For simulation of video traffic, the "Paris" sequence of CIF resolution with 4:2:0 formats, was H.264/AVC coded and its packets (chunks) were sent from different peers to the receiver to be assembled on-the-fly by the decoder. These techniques mimic the mesh-based approach on P2PTV streaming. Additionally, to simulate the proposed mechanisms in a more realistic environment, CBR background traffic with UDP transport protocol of packet size of 512-bytes for an average of 1.5Mbps was injected on to the network.

7.2 Simulation scenarios

To verify the proposed method, two scenarios have been implemented, run, evaluated, and compared to each other. The two scenarios have been considered in this paper. The first one considers the proposed method; the second one mimics a typical P2PTV system where peers connect and switchover to and from different peers randomly (*in results section called Randomized approach*), striving for computational load-balancing, as described in [1] [23]. Both scenarios are applied to the same network and loading conditions.

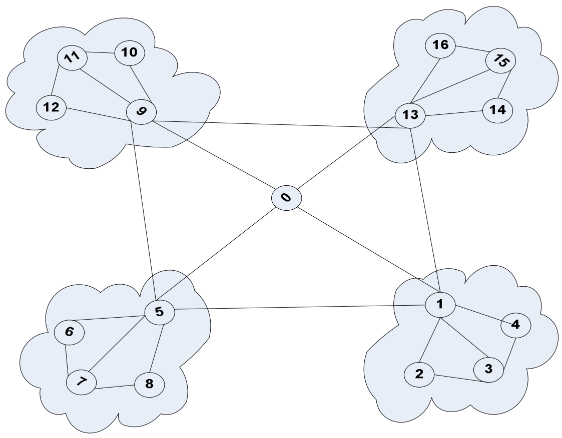

Figure 8 - Simulation Topology

7.3 Evaluation metrics

In order to assess the proposed scheme, different network parameters should be identified, tested and evaluated. The following parameters have been considered:

- Network Locality
- Quality of Service (QoS)
- Quality of Experience (QoE)

8. Results

8.1 Network locality

Network efficiency is the first demand of the proposed method where the inter-connections among the peers should not be done randomly. Thus, it can be seen from figure 9 how the approach is supporting the receiving node to keep the traffic as close as possible, by considering the local peers (*the lower RTT*) at the initiating phase of the connection and in each handover point. So, looking at figure 9, it is clear that the receiving node started with a set of peers which were nearby on the overlay network and achieved almost 0.7 of the network efficiency followed by smooth diminish in few percentages.

This small reduction can be interpreted due to the contribution variation among the participant peers where each peer experiences its own network conditions such as congestion, bottleneck, and sometime link failures. However, the proposed method shows that based on the prioritization among the participants' peers with the lower RTT values and the switching over, network efficiency has increased further (at time 15, 25, and 30) as shown in figure 9.

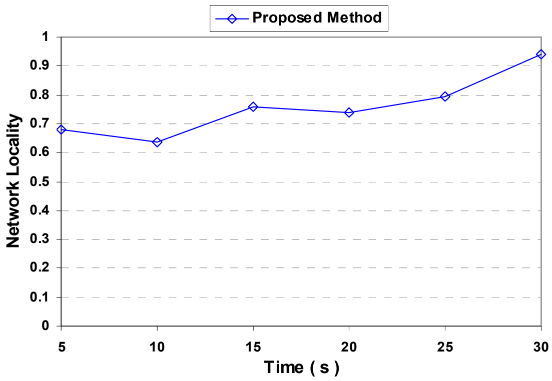

Figure 9 - Network Locality

8.2 Quality of Service

The most important factors that affect the QoS have been considered to determine whether the proposed approach is also actually affecting the quality at the application (or user) level.

In P2P networking, Quality of Service is linked to different metrics about the network. These metrics are intrinsic to each other. For instance, when peers keep downloading from a specific node which offers high bandwidth, but without presenting any sense of balance, the quality of service will be degraded drastically. This also will have other side effects on the computational efficiency (*load distribution*).

For this reason, to quantify and test the proposed method, different effective QoS parameters have been presented and used here as follow:

- Throughput: is the average rate of successful delivery of the packets over the network. The throughput can be measured in bit/s or packet/s.
- Packet loss ratio: This is the ratio between dropped and transmitted data packets. This gives an account of efficiency and the ability of the network to discover routes.
- Average end-to-end delay: The average time span between transmission and arrival of data packets. This includes all possible delays introduced by intermediate nodes for processing and querying of data. End-to-end delay has a detrimental effect on real-time IPTV. This can be countered only to some extent by increasing buffering.
- Delay variation or jitter is another parameter, particularly important for video streaming, as the packets at the receiver need to be decoded at a constant rate. It is desired, if the delay variation to be as minimal as possible.

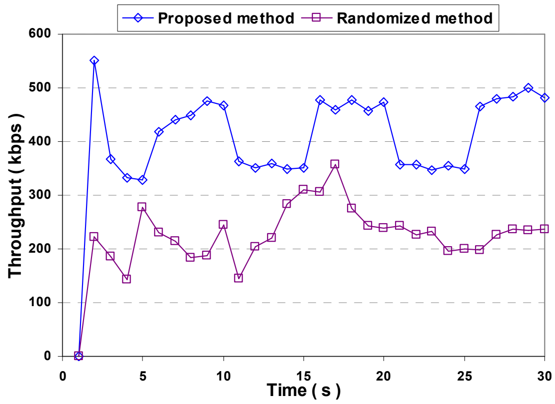

Figure10 - Throughput

Figure 10 shows the effect of the proposed method on the throughput of the two run scenarios. It can be noticed that our local-aware approach in managing the overlay leads to considerable improvements. Our approach reduces the average RTT among inter-communicating nodes and, turns, reduces the overall link utilization. In fact, the average throughput achieved with our approach is 404 kbps, which is almost twice the average throughput obtained by the randomized approach used as a benchmark (210 kbps). If we consider that, according to the literature, P2PTV applications incur a traffic comprised between 300 and 500 kbps [1] [23], we can conclude that our approach brings considerable advantages.

Another parameter which is considered most effective on the QoS is the packet loss rate. This metric is influenced by the congestions on the network, where due to limited buffer capacity overflow packets or packets delayed more than the human perception limit, can be discarded. Figure 11 gives an indication of the advantage of the proposed method since it is trying to reduce the packet loss by switching over among the peers in case of congestion. On the other hand, the randomized approach is showing packet loss and which is due to the randomness in selecting and switching over among the peers.

Packet loss can be considered the most crucial factor in video decoding and, consequently, a determining factor for video Quality of Experience (QoE). It is therefore essential to verify that the P2P overlay is optimized with this regard. Table 1 summarizes findings published in [28], correlating packet loss ratio with video streaming Quality-of-Experience (QoE), for the case of single-layer coding. It is noticeable that any value above around 14% leads to poor quality. According to this table, the proposed method shows a smooth video to the end-users.

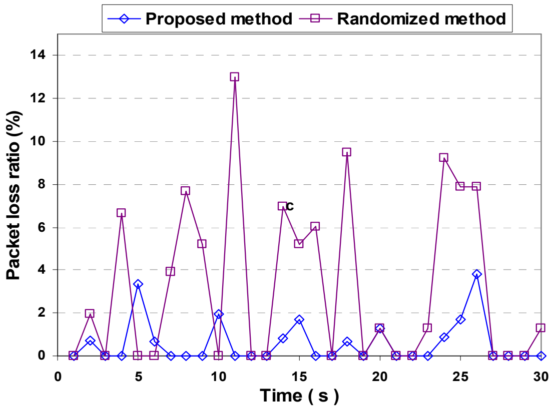

Figure11 - Packets loss ratio

Packet loss ratio [%]	QoE acceptability [%]	Video quality playback
0	84	Smooth
14	61	Brief interruptions
18	41	Frequent interruptions
25	31	Intolerable interruptions
31	0	Stream breaks

Table 1 Quality of experience acceptability thresholds

Another important performance metric in video streaming is end-to-end latency (or delay), which is critical in the process of meeting strict playback deadlines. In turn, this has a direct impact on quality of service and quality of experience.

Figure 12 shows the average end-to-end delays of both scenarios. Noticeably, our approach gives a lower average latency (order of 0.02 seconds); also, we can see that the delay is consistent, due to the fact that connections are not chosen randomly. On the other hand, delay in the randomized approach is fluctuating. This variation in delay may in turn increase packet loss with detrimental effect on QoE.

Figure 13 illustrates this attribute in terms of jitter. A nearly constant jitter derives from the fact that peer handover is governed by a prioritization process (based on RTT values), which ensures that nearby nodes are chosen first. The small variation in jitter in our approach is due to the fact that we still need to force handover even in the case of optimal connections, to maintain a good computational load balance.

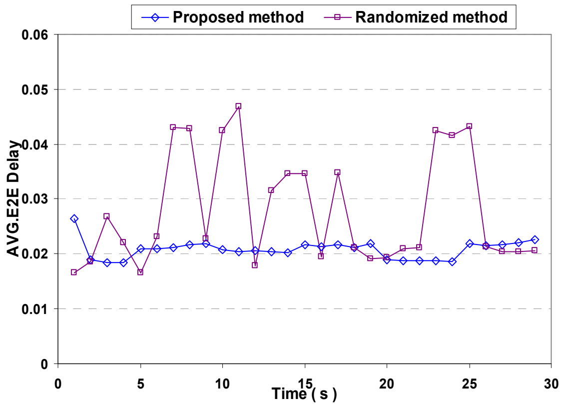

Figure12 - End-to-End Delay

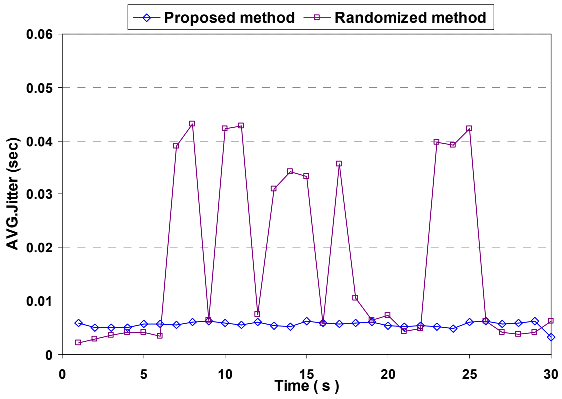

Figure13 - Jitter

8.3 Quality of Experience

Finally, the objective and subjective quality of decoded video under the proposed and randomized methods are compared. The received packets are decoded using error concealment of H.264/AVC encoder JM15. The resulting PSNR is shown in fig. 14 comparing both the algorithms. It is clear that the PSNR improvement is more than 2 dB in most cases, showing the efficiency of our proposed algorithm. The key reason behind the improved PSNR is the smaller degree of packet loss of the proposed method, as shown in fig 11.

Figures 15 and 16 compare the subjective quality of the proposed method against the randomized method respectively, quality of picture on the table, where on the randomized method, the cup, pen, and papers are missing, but in the proposed method, picture quality is considerably better.

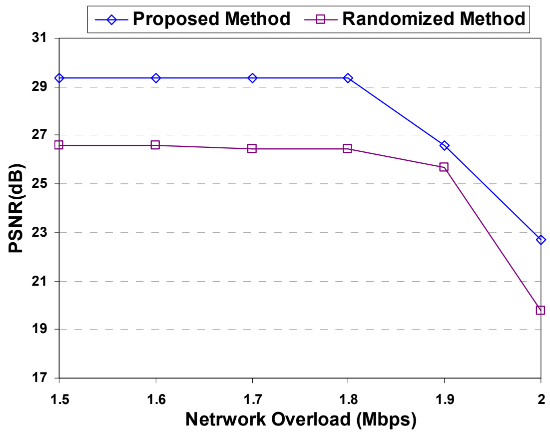

Figure14 – PSNR

6. Concluding Remarks

In this paper, we first evaluate the state of the art in P2P streaming by carrying out a methodical study base on two popular frameworks. Once we identify key shortcomings, we move into an attempt to find simple yet practical and effective ways to address them.

We have looked at two different approaches (Sopcast and Joost), two different streaming models (real-time and VoD), and two different scenarios (peak and ordinary demand levels). In each of the above cases, the aim of the designer was to build a self-managed network overlay that could handle large-scale demand, regardless of fluctuating network conditions and user location.

The P2P paradigm addresses the issue of scalability by limiting the amount of server intervention. As of today, however, P2P application designers do not seem to have placed sufficient emphasis on the need to come up with network-friendly solutions. In this article we have introduced a formula to measure network efficiency in a way that captures network locality (distance among interconnecting peers), link conditions (RTT of those links), and degree of server intervention (ratio between CS and P2P traffic).

A key outcome of our study is that existing approaches do not achieve a good trade-off between computational and network efficiency. Sopcast achieves good levels of network efficiency but only in relatively stable conditions. High levels of churn lead to a considerable degradation in network efficiency. By contrast, Joost is not network efficient due poor locality. Also, a substantial traffic has to originate from servers, to compensate for the best-effort nature of the network and also for poor network locality (the RTT between intercommunicating links is sub-optimal).

Figure 15 - Video Quality (Proposed)

Figure 16 - Video Quality (Randomized)

Existing P2P streaming applications (such as Sopcast and Joost) are appealing from the point of view of the application (or P2P framework) provider as well as the content provider due to the reduced digital distribution costs. However, from the network provider's perspective, such applications do not represent an ideal solution. Problems include:

- Network inefficiency: as demonstrated by our study, network resources are used sub-optimally.
- Traffic aggregation: it is very hard, if not impossible to perform traffic aggregation since traffic sources and destination are not chosen deterministically.
- Manageability: it is very hard to forecast and monitor traffic since user behaviour is unpredictable and traffic sources and destination are not deterministic. Network dimensioning, planning, and control are also difficult.
- Economic models: it is hard to think of a way for the network operator to profit from P2P streaming when most charging models are based on flat rates.

Our proposal is to look for algorithms that employ a certain degree of cross-layer optimization but that retain the simplicity of existing approaches to the best possible extent. In the second part of this article we give an example of a possible way to improve network efficiency and, ultimately, quality of experience.

Our study suggests that the prospects of P2P streaming seem very good if the scalability of this approach is assessed from the application (or even the user) view point. However, if we look from the angle of the network operator the scenario is rather different. Tackling the complex issue of P2P streaming in isolation from network and network management and dimensioning problems is not promising. Simple yet effective approaches based on cross-layer techniques are significantly more powerful.

7. References

[1] M. Alhaisoni, A. Liotta, M. Ghanbari, “An assessment of Self-Managed P2P Streaming”, In Proc. Of ICAS2009 (the 6th International Conference on Autonomic and Autonomous Systems and published by IEEE Computer Society, pp.34-39, 21-24 April 2009 Valencia, Spain.

[2] www.Sopcast.com. Last Access was on 15th of October 2008

[3] www.Joost.com. Last Access was on 27th of October 2008

[4] www.zattoo.com. Last Access was on 20th of October 2008

[5] M. Alhaisoni,, A. Liotta. (2009). Characterization of signaling and traffic in Joost. Peer-to-Peer Networking and Applications, 2(1), 75-83.

[6]

[7] Jun Lei, ,Lei Shi and Xiaoming Fu, (2007)An Experimental Analysis of Joost Peer to Peer VOD Services, Technical Report, Computer Networks Group, University of Gottingen ,Germany.

[8] T. Silverston and O. Fourmaux, “P2P IPTV Measurement: A Comparaison Study”, Universit´e Pierre et Marie Curie, Paris, France.

[9] Ali, S., Mathur, A., and Zhang, H. 2006. "Measurement of commercial peer-to-peer live video streaming". In Proceedings First International Workshop on Recent Advances in Peer-to-Peer Streaming (WRAIPS), August 10, 2006, Waterloo, Ontario, Canada,(pp. 1-12).

[10] Hei,X., Liang,C. ,Liu,Y. and Ross,K.(2007) ‘A measurement stud y of a large scale P2P IPTV system’, IEEE Transactions on Multimedia, Volume: 9, Issue: 8, page(s): 1672-1687

[11] Yong Liu, Yang Guo, Chao Liang, "A survey on peer-to-peer video streaming systems". The Journal of P2P Networking and Applications, Springer, Volume 1, Number 1 / March, 2008, pp. 18-28

[12] Tobias Hoßfeld et al , "A Qualitative Measurement Survey of Popular Internet-based IPTV Systems" 4-6 June 2008, page(s): 156-161

[13] Susu Xie, Bo Li Keung, G.Y. Xinyan Zhang .(2007) ‘Coolstreaming: design, theory ,and practice’ ,IEEE Transactions on Multimedia,Vol.9,No.8,December, page(s): 1661-1671

[14] Y. Liu, X. Liu, L. Xiao, L. M. Ni, and X. Zhang, "BLocation-aware topology matching in P2P systems", in Proc. INFOCOM, Date: 7-11 March 2004 Volume: 4, On page(s): 2220- 2230 vol.4

[15] Z. Xu, C. Tang, and Z. Zhang, “Building topology-aware overlays using globalsoft-state”, in Proc. ICDCS, 19-22 May 2003 On page(s): 500- 508

[16] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, “Scalable application layer multicast”, Dept. of Comput. Sci., Univ. of Maryland, 2002, Tech. Rep. UMIACS.

[17] B. Zhao, A. Joseph, and J. Kubiatowicz, “Locality aware mechanisms for large-scale networks”, in Proc. Workshop Future Directions Distrib. Comput. (FuDiCo), 2002.

[18] Mushtaq, M., Ahmed, T. “P2P Object-based adaptivE Multimedia Streaming”, In J Netw Syst Manage (2007), Volume 15 , Issue 3, Pages: 289 - 310

[19] T. P. Nguyen and A. Zakhor. “Distributed video streaming over Internet”, In Proc. SPIE, Multimedia Computing and Networking., volume 4673, pages 186–195, December 2001.

[20] T. Nguyen and A. Zakhor. “Distributed video streaming with forward error correction”, In Proc. Packet Video Workshop, Pittsburgh, USA, April 2002.

[21] M.Hefeeda, A.Habib, B.Botev, D.Xu, and B.Bhargava (2003), "PROMISE:peer-to-peer media streaming using CollectCast".

In Proc.of ACM Multimedia 2003,pages 45–54, Berkeley ,CA ,USA, November.

[22] Mushtaq, M., Ahmed, T. "Adaptive Packet Video Streaming over P2P Networking Using Active measurements", In ISCC 2006, Proceeding of the 11th IEEE Symposium on Computers and Communications, pp. 423-428, IEEE Computer Society, Los Alamitos.

[23] Hefeeda ,M.M and Bhargava ,B.K.(2003) 'On-demand media streaming over the internet', In Proceedings of the The Ninth IEEE Workshop on Future Trends of Distributed Computing Systems, Page: 279

[24] M. Alhaisoni, A. Liotta, M. Ghanbari, "Resource awareness and trade off optimization in P2P Video Streaming", accepted in International Journal of Advance Media Communication, special issue on High-Quality Multimedia Streamingin P2P Environments.

[25] Ragusa C, Liotta A, Pavlou G. "An adaptive clustering approach for the management of dynamic systems", IEEE Journal on Selected Areas in Communications, 2005; 23(12) 2223–2235

[26] Liotta A, Pavlou G, Knight G. "Exploiting agent mobility for large-scale network monitoring", IEEE 2002; 16(3), 7–15

[27] KaZaA: http://www.kazaa.com. Last Access was on 27th of Sep. 2009

[28] http://isi.edu/nsnam/ns/ Last Access was on 27th of Sep. 2009

[29] Agboma, F., Smy, M. and Liotta, A. (2008) QoE analysis of a peer to-peer television system. In Proceedings of IADISInt. Conf. on Telecommunications, Networks and Systems, Pages 111-116