Resilient P2P Streaming

Majed Alhaisoni, Mohammed Ghanbari
School of Computer Science and Electronic Engineering
University of Essex
Colchester, UK
malhai@essex.ac.uk
ghan@essex.ac.uk

Antonio Liotta
Electrical Engineering Department
& Mathematics and Computer Science Department
Technische Universiteit Eindhoven
The Netherlands
a.liotta@tue.nl

Abstract— **P2P streaming has shown an alternative way of broadcasting media to the end users. It is theoretically more scalable than its client-server based counterpart but suffers from other issues arising from the dynamic nature of the system. This is built on top of the internet by forming an overlay network. End-users (peers) are the main sources of the overlay network, sharing their bandwidth, storage and memory. Peers join and leave freely, which dramatically affects, both on QoS and QoE. Furthermore, the inter-connections among the peers are based on logical overlays, which are not harmonized with the physical underlay infrastructure. This article presents combinations of different techniques, namely stream redundancy, multi-source streaming and locality-awareness (network efficiency), in the context of live and video-on-demand broadcasting. A new technique is introduced to improve P2P performance and assess it via a comparative, simulation-based study. It is found that redundancy affects network utilization only marginally if traffic is kept at the edges via localization techniques; multi-source streaming improves throughput, delay, and minimizing the streaming time.**

Keywords- P2P; Multimedia; Redundancy; Multi-source; Locality-awareness

I. Introduction

Peer-to-Peer (P2P) streaming has evolved into one of the most popular Internet applications. P2P entails a highly attractive paradigm in a distributed fashion. It provides simple and efficient mechanisms to pool and share redeemable resources like CPU cycles, disk space or multimedia files. These advantages tolerate that any peer can join and leave without resulting in ill effect on the stream continuity and indexing, in contrast to the traditional client/server concept where a failure of the central server may affect the stream strongly. A P2P mode of operation, however, also has some downsides. P2P systems cause high traffic volumes, including extra traffic due to redundancy as well as signalling traffic to maintain the overlay topology. P2P network topologies reveal a high variability and P2P traffic patterns of P2P applications fluctuate strongly in time and space.

Overlay networks form the basis of the P2P networking applications where peers connect logically to form the virtual overlay. Peers can join and leave the overlay freely. In a P2P streaming application, multimedia contents are delivered to a large group of distributed users with low delay, high quality and high robustness [2]. P2P-based versions of IPTV, Video on Demand (VoD) and conferencing are, thus, becoming popular.

P2P streaming systems can sustain many hosts, possibly in excess of hundreds or even millions, with miscellaneous heterogeneity in bandwidth, capability, storage, network and mobility. An additional aim is to deliver the stream even under dynamic user churn, frequent host failures, unpredictable user behaviours, network traffic and congestion. To accomplish these goals, it is essential to address various challenges to achieve effective content delivery mechanisms, including routing and transport support.

This article is primarily directed towards finding effective ways to increase the resilience and scalability of the overlay network whilst at the same time minimizing the impact on the physical network (or underlay). It is found that P2P frameworks mostly fail to address the network locality which tends to cause severe network operational and management issues. In turn, this limits P2P scalability when traffic streams traverse, and thus congest, large portions of the network. Another issue is that existing P2P streaming systems are intrinsically best-effort. This fact, combined with their network unfriendly behaviour, often leads network operators to impair P2P traffic, with detrimental consequences for the resulting quality of service.

The key question we are addressing herein is whether and how it would be possible to increase the user quality of experience (QoE) in P2P streaming. Common techniques were developed, such as redundant streaming, i.e. to send multiple streams to the same user in order to reduce packet loss. The downside is that redundancy increases traffic, thus reducing network utilization and, hence, increasing congestion.

In order to retain the benefits of redundancy (QoE) and reduce its detrimental effects on the network (congestion and QoS degradation), in [1] we have studied the combination of two techniques, redundant-streaming and network locality. We found that by keeping traffic local among the peers and mainly at the edges of the network, the benefits of redundant-streaming outweigh their shortcomings.

Herein, we extend [1] considerably, including further evidence and carrying out an additional comparative analysis with another approach. We introduce multi-source streaming based on disjointness and without any type of redundancy. It is found that this approach further improves throughput, decrease delay and packet loss, and minimizes the overall streaming time. Both techniques are combined with network locality and computational load balancing, benchmarking against P2P TV system Joost.

Initial results indicate that both the multi-source and the redundant-streaming approaches lead to significant improvements, suggesting a promising direction in P2P streaming.

II. RELATED WORK

As we are dealing with network QoS, QoE, P2P locality awareness, stream redundancy and multi-source streaming we give first an overview of different studies that have looked at these topics individually. To the best of our knowledge, a comparative evaluation assessing combined strategies, as we do in this paper, has not been published before.

Ways to pursue efficiency between overlay and underlay have started to be investigated only recently. Authors in [3] 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 [4], where they measure the latency distance between the nodes and appropriate Internet servers called landmarks. A rough estimation of awareness among the nodes is obtained to cluster them altogether, as in [5] [6].

On the other hand, another study in [7] proposes different techniques where the video stream is divided into different flows that are transmitted separately to increase parallelism and, hence, reduce transmission latency. The authors use the PSQA technique that gives an estimate of the quality perceived by the user. This study was concerned on how to influence and improve on quality (as measured by PSQA). They introduce three cases: sending a single stream between nodes; sending two duplicate streams via different paths; and sending two disjoint sub-streams whose union recreates the original one. In our work we look at the case of multiple redundant streams and multi-source streaming, looking at the effects that redundant streams have on both the network load and the user QoE. Also we see how multi-source streaming alleviates some of the shortcomings of redundant streams. However, we emphasize on techniques to choose the inter-communicating peers based on their mutual proximity, to keep traffic local and minimize the impact on the network load.

Overlay locality is also studied by [8], 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 intercommunicating peers are forced to periodically handover, in order to distribute computational load as well as network efficiency (as explained in [13] and [14]).

Hefeeda et al [10] 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 [9], which employs a simpler version of our RTT approach based on continuous pinging of peers. Similarly, we adopt clustering to limit the signalling overheads associated with this process and prevent bottlenecks.

Other studies such as [11], 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 [12]. In particular, they show how to maximize the throughput in divisible load applications.

Moreover, Thomas *et al* [15] proposed a distributed hash table, which is suitable for highly dynamic environments. Their work was designed to maintain fast lookup in terms of low delay and number of routing hops. In their work, the hop-count was the main metric used to determine locality-awareness. According to their work, neighbouring nodes are grouped together to form a clique. Nodes share the same ID in a clique; moreover, the data will be replicated on all the nodes on the clique to avoid data loss.

A clique has an upper and lower bound in terms of the number of nodes, such that cliques are forced to merge or split. Another aspect of their work is to assume that all the nodes are distributed uniformly in a two-dimensional Euclidean space. However, this may not work in a large network such as the Internet. The link structure is updated periodically in order to establish a structured network. On the other hand, their proposal is based on pining nodes to join the closet clique which will drastically introduce extra signalling overhead.

Another study similar to [15] was conducted by Shah Asaduzzaman *et al* [16]. Their proposal was built on top of [22] with some modifications, introducing stable nodes (super-nodes) and replicating the data among the stable nodes only. However, their proposal elects one or more stable nodes of highest available bandwidth in each cluster and assigns a special relaying role to them.

Their work is based on a combination of tree and mesh architectures where the nodes on the clique form a mesh and the stable nodes are connected in a tree structure. For each channel, a tree is formed between the stable nodes including only one stable node in each clique. However, stable nodes are elected based on their live session. So, in this case a clique may have more than a stable node. The downside of this approach is that the relaying nodes (super nodes) are forming a tree, so reconstructing them in case of failures and peers churn will be costly and can introduce some latency.

By contrast to the abovementioned two works, our proposal aims not only to retain the benefits of redundancy (QoE) but also to reduce its detrimental side effects on the network. We study the combination of two techniques, redundant-streaming and network locality, while on [15] and [16] they are mainly concerned with network locality. We prioritize the choice of sources based on their mutual distance from the destinations. In essence we adopt a previously published hierarchical RTT monitoring approach [17] to maintain a list of sources $\{S_i\}$, ranking their order based on their distances from the recipient (R). Periodically, a new set of sources $\{S_{inew}\}$ is chosen from this pot and handover is forced from $\{S_i\}$ sources to $\{S_{inew}\}$ sources.

On the other hand, effective streaming mechanisms make use of the multi-path nature of P2P networks to satisfy the bandwidth requirements of media applications by using network resources. In [18], authors establish a generic framework for multi-path streaming. Different advantages gained by the exploitation of multiple transmission paths for media broadcasting consist of accumulative network bandwidth and delay reduction. Another experimental work on multi-path streaming was conducted in [19], which offer some insights concerning the selection of content sources and streaming paths, based on the jointness/disjointness of network segments.

However these findings cannot be applied directly in P2P scenarios, especially due to the lack of coordination among the peers. Hence, our paper introduces a multiple-source (stream) from different peers and clusters. In this regard, receiver makes the selection of those peers according to the algorithm published in [14]. GnuStream [25] uses multiple senders to stream a video to the receiver. GnuStream is, however, not robust to the churn of peers. This problem is improved in [26] by introduction of a central power peer responsible for sender selection and switching when such occasion happens. In real-life scenarios, the assumption of an always-available central power node cannot be justified. In [27], the authors proposed an algorithm where the receiver has control of rate allocation and packet partitioning among the senders.

In this paper we aim to establish to point up to which we can increase redundancy without triggering network congestion and, also, what is the optimum number of peers needed for transmission in case of redundancy and multi-transmission sources. Our work not only introduces locality-awareness, redundant-stream and multi-source but also it load balances computing and network resources. We introduce a new way of calculating the packet loss ratio, and end-to-end delay, with the stream redundancy.

Furthermore, QoS and QoE are tested by transmitting a video to examine the scalability and resilience of this proposition.

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

1) To study a new combination of existing techniques (cross-layer optimization, localization, forced handovers, redundant-stream, and multi-source-stream);
2) To take the perspective of the network operator, in trying to harmonize overlay and underlay networks;
3) To look for trade-offs between redundancy levels (to increase QoE) and network efficiency;
4) To make a comparison of two well know techniques redundant-stream, and multi-source-transmission accompanied by locality awareness and computational load distribution;
5) Introducing new techniques to gauge packet loss and end-to-end delay;
6) To compare against randomized approaches. (*mimicking the Joost application*).

III. PROPOSED APPROACHES

In this section two techniques are proposed and compared to each other. These techniques are: 1) redundant-streaming, whereby the stream redundancy is increasing from 1 to 5 sources (*more details will be given in the following section*); 2) and multi-source streaming, based on disjointness, whereby the stream is divided into flows and transmitted from 1 to 5 sources out of different clusters. Both techniques are combined with locality-awareness and computational load balancing. The emphasis of this paper is to examine the performance and compare each technique to a Joost-like system [20]. In essence, Joost chooses sources randomly and continuously handovers among sources to pursue computational load distribution.

A. Redundant- streaming approach

In this scenario the number of redundant streams is increased from 1 to 5 as shown in Figure 1. Sources {S1…S5} are chosen based on locality and are also continuously (periodically) forced to handover, choosing new sources from a pot of available sources. These are prioritized based on mutual inter-peer distances to ensure that traffic is kept as local as possible. On the other hand, forced handovers ensure that the important feature of computational load-balancing is maintained. This special aspect was discussed in previous publications [13] [14]. More details as to how network locality and computational load balancing are maintained from the receiver side can be found in [14].

Instead, herein we are mainly interested in understanding whether location-aware P2P techniques can actually reduce the detrimental effects of P2P redundant streaming. Under architectures other than P2P (unicast, multiple unicast and multicast), redundant streams increase QoE but have the side effect of increasing network congestion. An interesting proposition is that of finding the minimum redundancy level, which leads to the maximum QoE improvement. By contrast if we adopt a P2P approach that succeeds in keeping traffic away from the core network, we have a good chance that redundancy does not always result into network congestion. Our aim is to verify this hypothesis and better understand its implications, comparing it to localized multi-source based on disjointness but without introducing any type of redundancy.

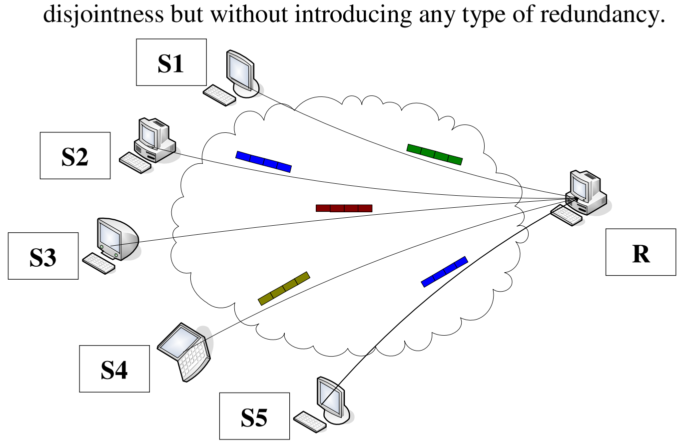

Figure 1 - Proposed architecture

Redundancy study

In order to study the effect of redundancy in relation to both QoS and QoE, we first measure relevant parameters (as detailed in section IV) for a Joost-like system [20] and use this as a benchmark. Redundancy is increased from 0 (1 source per destination) to 4 (5 sources per destination).

We then compare this with our proposed approach, in which sources are forced to handover continuously (as in Joost) - to ensure computational load balancing – but are not chosen randomly.

We prioritize the selection of sources based on their mutual distance from the destination. In essence we adopt a previously published hierarchical RTT (round trip time) monitoring approach [17] to maintain a list of sources $\{S_i\}$, ranked based on their distance from the recipient (R). Periodically, a new set of sources $\{S_i^{new}\}$ is chosen from this pot and handover is forced from $\{S_i\}$ sources to $\{S_i^{new}\}$ sources. Our hypothesis is that this forced handover strategy does not impact network congestion if traffic is kept away from the core network. We wish to establish, however, until which point we can increase redundancy without triggering network congestion. Details of how the peers are clustered and the inter-connecting and switching among the peers are maintained can be found in earlier publication [14].

B. Multi-source approach

Based on the derived findings of the redundant-stream results, it was found that locality-awareness and computational load balancing play a vital role in the scope of redundant-source-streaming. In the initial approach, it was shown how redundant-aware streaming has a positive impact over the quality of the received video. Now another scenario is proposed to improve the resilience of streaming without resorting to redundancy. This approach finds an effective way to improve the resilience and scalability of the overlay whilst at the same time not conflicting with the underlay network or overloading the network. Common techniques have been proposed to transmit the streams over multi-source or multiple paths. But a new combination of these two techniques (multi-source-path) complemented with locality-awareness and computational load balancing is examined here. This approach is compared to the former redundant-aware streaming.

Multi-source principle

Multi-sender methods are the best existing solutions for streaming video on P2P networks. However, in some cases multi-senders share a bottleneck and this impairs the throughput, increasing the delay and packet loss as well. In this method, multi-source streaming is combined with disjoint paths streaming, which ensure that peers don't share early bottleneck, which most likely happens over the access network. To illustrate and assess this method, the topology of figure 2 is used. Clique of peers are clustered together, giving the chance to alleviate the probability of having different peers over the same path (thus, creating a bottleneck or getting early congestion). In this method if a receiver R requests a certain video, a set of candidate senders (determined by the method in [14]) having the desired media are chosen from different clusters.

As in redundant-streaming, we prioritize the choice of sources based on their mutual distance from the destination. In essence we adopt a previously published hierarchical RTT monitoring approach [18] to maintain a list of sources $\{S_i\}$, ranked based on their distances from the recipient (R). Periodically, a new set of sources $\{S_i^{new}\}$ is chosen from this pot and handover is forced from $\{S_i\}$ sources to $\{S_i^{new}\}$ sources. Our hypothesis is that choosing senders from different clusters help in avoiding bottlenecks; furthermore, the forced handover strategy can be applied smoothly on the same cluster since peers in a clique share nearly the same RTT values [14]. This ensures that the proposed technique is applied efficiently in choosing the peers from different clusters and performing the enforced handover to maintain computational load balancing. The second part of the hypothesis is that transmitting the media file in this way will improve transmission time, increase throughput and decrease latency.

The video stream is divided into different flows which are sent from different sources, different clusters and over disjoint paths as shown in figure 1. This increases the parallelism and, hence, reduces transmission latency and also the transmitting time.

IV. Assessment Method

The effects of redundancy and multi-source streaming on QoS (Quality of Service) and QoE (Quality of Experience) are investigated, for each of the following two scenarios: 1) randomized scenarios (new sources are chosen randomly); and 2) locality-aware scenario (new sources are chosen based on minimal mutual distances from the recipient.

Sufficient Background traffic was added into the simulated network in order to simulate first lightly-congested networks and, then, heavily congested networks. Simulation scenarios, parameters and design are described below.

A. Simulation scenarios

Various simulation scenarios are considered. Each proposed scheme is benchmarked with a randomized behaviour. These scenarios are classified as follow:

- Localized-Redundant-Stream (L-R-S): in this scenario, redundant stream packets are streamed from 1 up to 5 sources, whereby every source is sending the same packets, to introduce redundancy. This technique is combined with locality-awareness and computational load balancing (*handover*).
- Randomized-Redundant-Stream (R-R-S): This scenario has the same characteristics of the L-R-S. The difference between the two is that L-R-S is locality-aware with load distribution mechanism, whereas R-R-S is purely based on random switchovers among peers (*this approach mimics a popular Video-On-Demand application known as Joost, used here as benchmarking for L-R-S*).
- Localized-Multi-Streaming (L-M-S): this approach is based on multi-source streaming but without creating any kind of redundancy. In this approach, packets are

scheduled and transmitted as flows from different sources starting from only 1 source and up to 5 sources. Packets are divided into flows and are scheduled to be transmitted from sources belonging to different clusters. This approach is combined with locality-awareness, computational load balancing and clustering. The most important factor besides multi-source streaming is clustering, which helps choosing the sources (peers) from different clusters to avoid the generation of hot spots.

- Randomized-Multi-Streaming (R-M-S): this approach is used as benchmarking for L-M-S. Connection and switching-over are built randomly though we transmit the stream from multiple-source (*This approach mimics a popular Video-On-Demand behaviour known as Joost, which is used here as benchmarking for L-M-S*).

B. Simulation setup

The proposed approaches were implemented and tested on the ns-2 network simulator (http://isi.edu/nsnam/ns/). A sample of the used topology is shown in figure 2. However, we have run our experiment with 200 nodes and found that the results are not changed owing to the nature of the proposed approach. In fact, the proposed approach always tries to keep the traffic at the edges of the network. Therefore, the traffic will not concentrate on particular part of the network although redundancy is introduced. The effect of redundancy is minimised as the traffic is kept at the edges and periodically connecting to new peers among the available peers with the aim of distributing the load. Furthermore, background traffic plays a vital role by running different sources and sinks over the network. Additionally, Although the used topology is fixed in the simulation, the proposed scenarios are treated in a different way. So, for the randomized scenarios, senders and receivers are chosen randomly for every run. On the localized scenarios, the senders are chosen based on locality and the receivers are selected randomly for every run. This gives the advantage of testing the L-R-S and L-M-R scenarios under different conditions over the used topology.

Moreover, various parameters were set on the used topology. First of all, each link has a bandwidth of 2 Mbps with equal latency (delay). However, the actual delay will be according to the nodes distance among each other; so, all the participants' peers have the same characteristics. IP is the network protocol and UDP is the transport protocol. For the redundant streaming simulation, video traffic of the "Paris" sequence of CIF resolution with 4:2:0 format was H.264/AVC coded and the same video packets were sent from one and, then, from multiple peers to the receiver.

On the other hand, for the multi-source video traffic simulation, the "Paris" video clips of CIF resolution with 4:2:0 format was H.264/AVC coded and the video flows were sent from one and multiple peers to the receiver. So, in this way, single video flows are transmitted from different peers according to a scheduling mechanism, whereby every flow has to go from different source to insure multiple sources and paths concurrently.

Secondly, in order to overload the network, it was imperative to set the CBR background traffic to vary the network load and enable us to study the various approaches under different loading conditions. The CBR traffic was setup from different sources to different destinations, with a 512 byte packet size. This background traffic operates during the whole duration of the simulations. This is added to the stream video on the running simulation.

For statistical significance, the proposed approaches were simulated independently and repeated 50 times. The presented results correspond to the average values of these simulations.

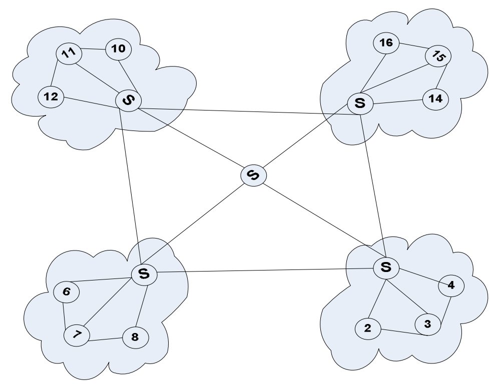

Figure 2 Simulation Topology

C. Evaluation metrics

Since we are dealing with two different approaches, redundant streaming and multiple-sources, it was important to devise suitable performance metrics. For the redundant stream, QoS factors such as packet loss and end-to-end delay is defined below in a new way. Next section shows how these metrics are defined in the scope of redundant streaming.

Throughput*:* is the average rate of successful delivery of the packets from the senders to the receiver. The throughput can be measured in different ways such as bit/s or packet/s.

Packet loss ratio*:* usually defined as the ratio of the dropped over the transmitted data packets. It gives an account of efficiency and of the ability of the network to discover routes. However, in P2P communication, a new way of calculating the packet loss ratio needs to be defined to study the particular issues relating to redundancy. In our case a packet is actually transmitted by several sources and is considered lost only if it is never received through any of the streams. This is formalized as follow:

Pi generic packet (i) sent by all source nodes

d_j sending or source nodes

$$X_{ij} = \begin{cases} 1 & \text{if } P_i \text{ sent by } d_j \text{ is lost} \\ 0 & P_i \text{ is received} \end{cases}$$

The decision as to whether a packet is lost or not will be according to the following Cartesian product:

$$PL_i = \prod_{j=1}^{d} x_{ij}$$

Therefore, if P is the total number of packets required to reconstruct a given stream, the packet loss ratio will be:

$$\mathbf{P}L = \frac{\sum_{i=1}^{p} PL_i}{P} \qquad \textbf{(1)}$$

Average end-to-end delay: is the average time span between transmission and arrival of data packets. In redundant P2P streams, the delay of each received packet is the minimum delay among all the received packets of the same type sent by all senders. This includes all possible delays introduced by the intermediate nodes for processing and querying of data. End-to-end delay has a detrimental effect on real-time IPTV.

Peak Signal to Noise Ratio: PSNR is an objective quality measure of the received video, taken as the user QoE. It is defined as the logarithm of the peak signal power over the mean squared difference between the original and the received captured video. This is formulized as in equation (2) and figure 3 shows how this is obtained through the simulation architecture.

$$\text{PSNR} = 10\log\frac{P^2}{E^2} \qquad \textbf{(2)}$$

Where p is the peak value for a given pixel resolution, e.g. for 8-bits $p = 255$

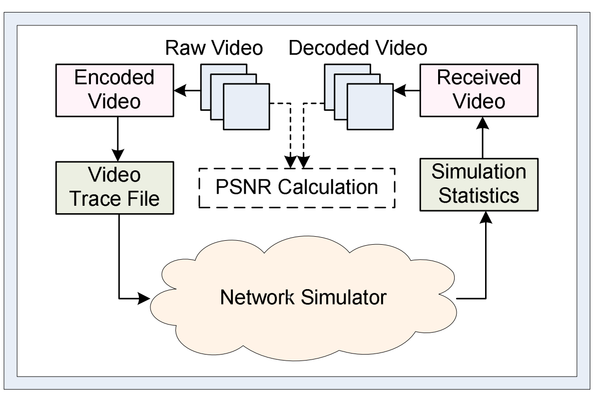

Figure 3 – Simulation Architecture

V. SIMULATION RESULTS

First of all, the redundant-streaming scenarios (L-R-S and R-R-S) will be compared to each other. Then, these results will be compared to the multi-source streaming scenarios (L-M-S and R-M-S).

A. Redundant-stream

Figures 4 and 5 show the packet loss ratio for the cases of lightly and heavily loaded networks, respectively. At light network load shown in Figure 4, both methods do not lead to any packet loss up to a redundancy level equal to 4. This means that for randomized redundant streaming (R-R-S), up to 4 sending peers, enough redundant packets can be received to compensate for any losses. However beyond 4, the added traffic creates congestion that leads to more losses, such that the backed up redundant packets may also be lost. With the localized-redundant-streaming (L-R-S), the figure shows that even going up to 5 redundant streams does not cause any packet loss. This is due to the fact that, no matter how much the network is congested, there is always enough number of redundant packets to be used by the receiver.

A network friendly behaviour of the L-R-S is even more apparent at higher network load, as shown in Figure 5. In this case, the network is brought close to congestion by the background traffic (not by the streams under scrutiny). The network is severely congested; then even one or two senders in action can lead to packet loss. By increasing the redundancy level (e.g., more senders), the packet loss is reduced in both methods. However, with the L-R-S, there is almost no packet loss after receiving from 3 senders. This is due to the fact that multiple copies of the same packets are now sent to the recipient.

By disparity, the randomized connection of Joost-like approach (R-R-S) cannot bring packet loss down to zero. In this case, even increasing the number of senders (more redundancy), the senders themselves create additional congestion such that, beyond 4 senders, congestion increases, as shown in figure 5.

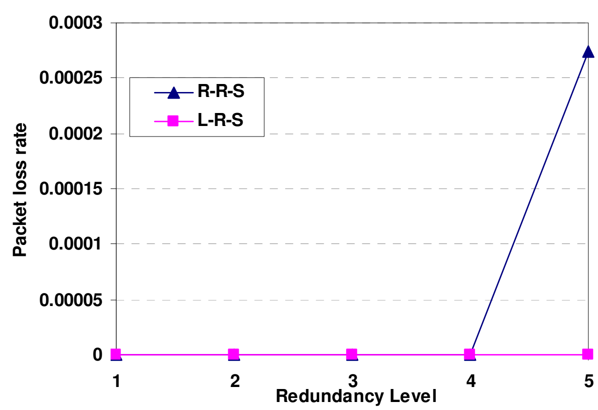

Figure 4 - Packets loss ratio – lightly loaded network

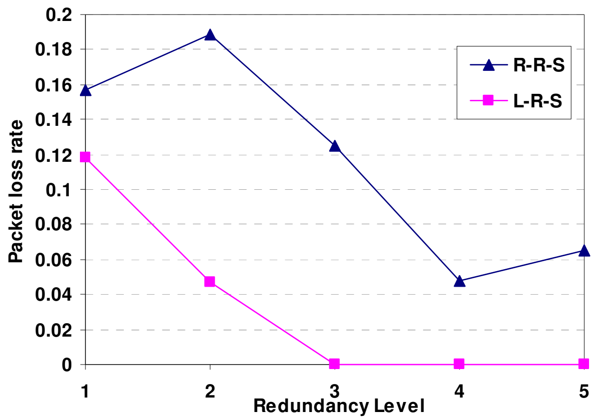

Figure 5 - Packets loss ratio – heavily loaded network

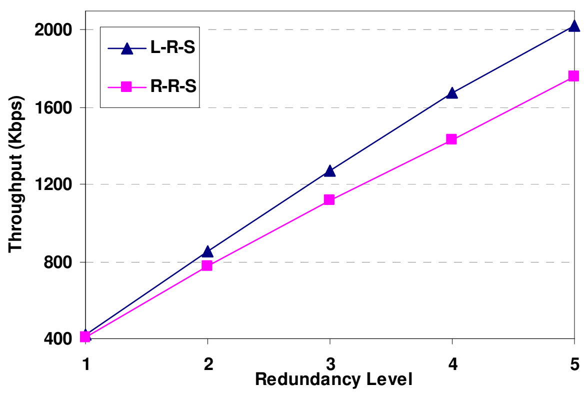

Figure 6 – Throughput

Figure 6 shows the effect of L-R-S and R-R-S the throughput. It can be noticed that the L-R-S, by managing the overlay, leads to considerable improvements. This approach reduces the average RTT among the inter-communicating nodes and, in turns, reduces the overall link utilization. In fact, the average throughput achieved with L-R-S is increasing with the number of sending peers.

Delay is another important quality of experience (QoE) parameter. Figures 7 and 8 show the average end-to-end network delay, for the lightly and heavily congested scenarios, respectively. It is important to note that at heavy network load, the end-to-end delay under localized connection has the least value at the redundancy level of 3 – 4 senders. Considering that packet loss rate is almost eradicated with just about 3 redundant senders (figure 5), it appears that the optimal redundancy level is comprised between 3 and 4.

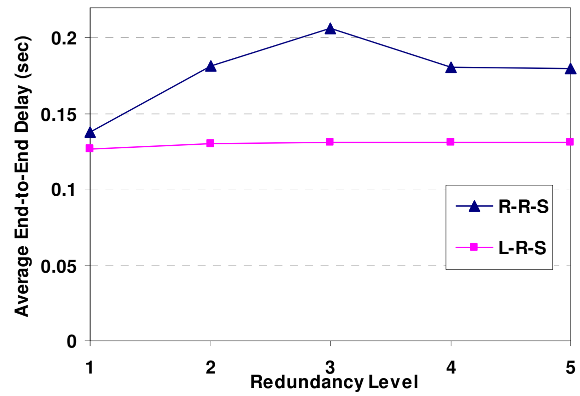

Figure 7 - Avg. E2E Delay – heavily loaded network

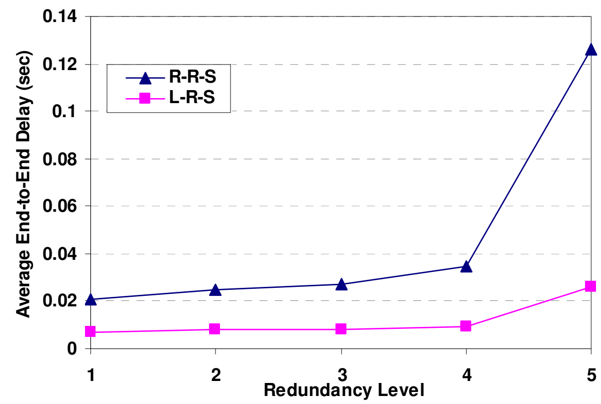

Figure 8 – Avg. E2E Delay – lightly loaded network

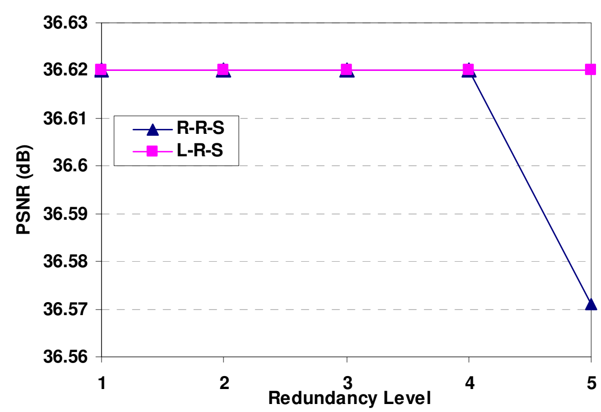

Figure 9 - PSNR – lightly loaded network

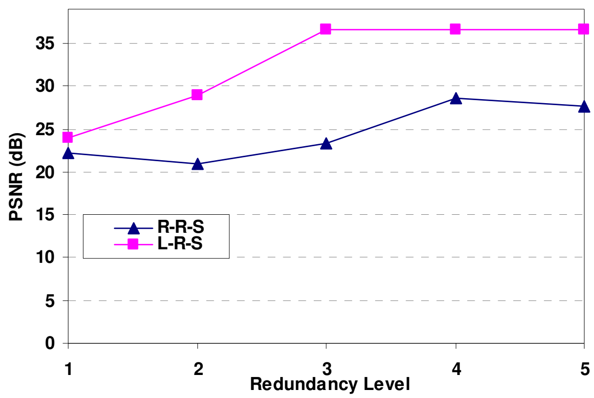

Figure 10 - PSNR – heavily loaded network

Finally, the subjective quality of the decoded video under both loading conditions is compared. The streams were coded and decoded with an H.264/AVC encoder/decoder of type JM15. Figures 9 and 10 show the objective video quality as measured by PSNR for lightly and heavily congested network, respectively. As expected they exhibit behaviour similar to that of Figures 4 and 5. When the network is lightly loaded, there is hardly any packet loss up to a redundancy of 4, at which point the randomized scenario generates congestions and, thus, PSNR drops.

The localized approach does not show any packet loss and maintains a constant level of QoE even with 5 injected redundant streams. Figure 10 is also consistent with this rationale. Noticeably, the localized approach improves PSNR steadily up to a redundancy level of 3, where the quality reaches its maximum theoretically achievable value (packet loss is zero at that point).

	Sending period (s)		*Throughput (Kbps)*	
Senders	**L-R-S**	**L-M-S**	**L-R-S**	**L-M-S**
1	35.59	35.59	424.16	409.98
2	35.59	17.88	850.78	829.69
3	35.59	12	1272.41	1243.39
4	35.59	9	1672.78	1645.92
5	35.59	7.32	2020.72	2041.21

Table 1 - Transmission Time with achieved Throughput

B. *Multi-source-streaming*

This section shows the results of multi-source-streaming compared to redundant-streaming (previous section). By examining the same parameters (*throughput, end-to-end delay, packet loss, and PSNR*), different pros and cons can be highlighted between the two approaches.

Table 1 shows the average data-rate over time when transmitting the Paris sequence over multiple sources and redundant-stream. Clearly, multi-streaming is able to achieve low sending time with high bandwidth. On the other hand, redundant-streaming is increasing the throughput but without reducing the sending period. Mean results were taken over 50 simulation runs to provide statistically significant results.

Figure 11 indicates the achieved throughput by every scenario; in the two proposed scenarios (L-M-S and L-R-S) the throughput is almost the same. However, L-M-S can be preferred over L-R-S since multi-streaming does not generate any redundancy load to the network and still achieving high throughput with the increase in number of sources. On the other hand, the randomized scenarios (R-M-S and R-R-S) are achieving less throughout for both cases and this is due to the pure randomness in selecting the sources, which leads to congesting different paths towards the receiving node.

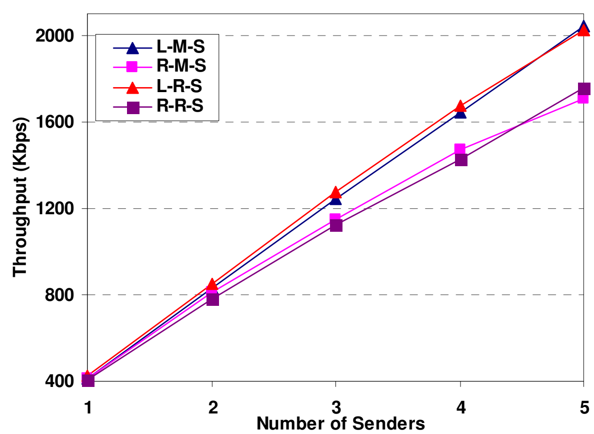

Figure 11 - Throughput

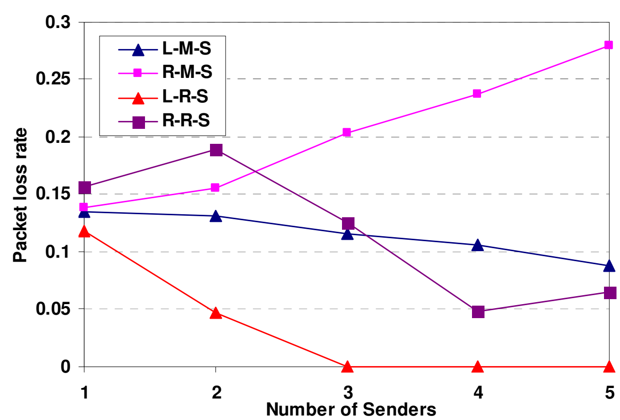

Figure 12 - Packet loss

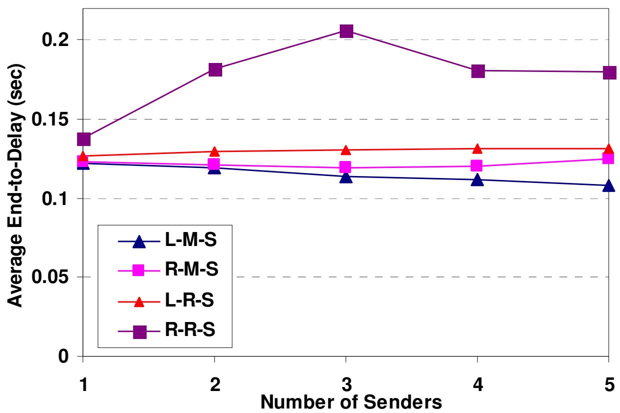

Figure 13 - End-to-End Delay

Looking now at packet loss, as shown in figure 12, it can be seen that L-R-S is performing the best due to the offered redundant-streaming by many sources. On one hand, this is positive but, on the other hand, this may at some point incur heavy load onto the network. L-M-S is not eradicating packet loss but with the increase in number of sources, packet loss is decreasing. However, the maximum occurred packet loss ratio is around 13%, which according to table 2, is acceptable as a threshold of QoE acceptability.

Another factor that is very important for streaming the video over P2P is end-to-end delay. This parameter plays a vital role with regards to the playback deadline mainly for real-time streaming. Any packet experiencing transmission latency will affect the video quality and, thus, the user QoE. In real time applications, packets missing their playback deadline are discarded and, hence, can be counted as lost packets. All these factors will diminish the continuity indexing for the video streaming.

Figure 13 depicts the end-to-end delay for the video packets. Looking at L-M-S and L-R-S, it can be seen that delay is slightly increasing with the increase in redundancy level. Whereas on the multi-source streaming the transmission latency is decreasing smoothly with the increase in number of transmission sources. In this regard, the value of dividing the streams into flows and transmitting that from different peers out of various clusters are obvious. On the other hand, for the randomized scenarios, in every case, it is clear that the most affected approach is the randomized scenarios with redundancy, which can be considered the worst.

Finally, the objective quality of the decoded video of all the scenarios is compared in Figure 14. The streams were coded and decoded with an H.264/AVC encoder/decoder of type JM15. Figure 14 shows the objective video quality as measured by PSNR for all the examined approaches. As expected, they exhibit behaviour similar to the packet loss findings of Figure 12. However, still the localized-redundant-streaming is performing the best. At the level of 3 redundant streams, it improves the packet loss ratio and, thus, the PSNR as received by the end-consumer.

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 2 - Quality of experience acceptability thresholds

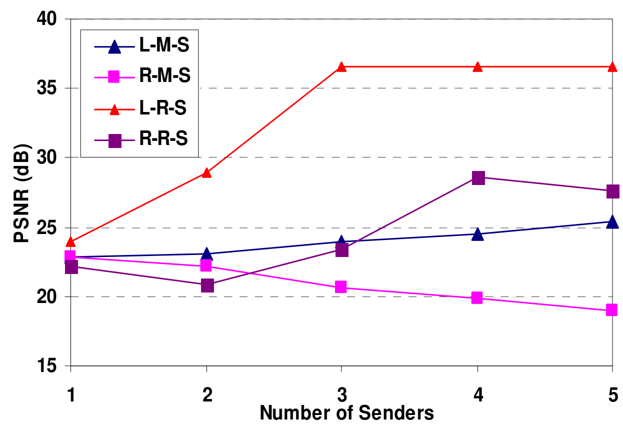

Figure 14 - PSNR

On the other hand, it is clear that the multi-source streaming is starting by an acceptable PSNR, at around 23 dB; but it keeps increasing with the increasing of dividing the stream into flows and transmitting it from more sources. This still reflects the value of dividing the video into flows and transmitting it over multiple sources and from different clusters

VI. DISCUSSION

In this paper, a localized redundant-source is proposed to offer a redundancy of the same content over the network with high quality and low end-to-end delay. This proposal has been tested and run under different scenarios and network conditions. In order to quantify its robustness, it has first been run under a point-to-point connection where there is only one sender and one receiver. Then, multi-connections were introduced where the receiver can connect from 2 and up to 5 senders (peers) simultaneously. Redundant streams are used in combination with locality-awareness to assess our initial hypothesis.

Varieties of connections have been run in two extreme congestion levels, lightly and heavily congestion, respectively. Different effective parameters on the network have been measured to show and validate how robust and practical is the proposed localized scheme with the offered redundancy by the chosen sending peers.

However, in order to adjudicate on the goodness of the localized-redundant-stream, a benchmark of a popular VoD application [20] is compared against this proposal, to show how localized-redundant-streaming behaves in contrast to the randomized scheme.

A vital network parameter is packets loss; to gauge this factor, a new way of measuring packet loss has been defined mainly to quantify the performance of the proposed localized redundant-stream, as shown in equation 1. The presented results show that the localized scheme is performing better across all QoS and QoE parameters. Consequently, by looking at packets loss ratio, it is apparent from figures 4 and 5 that the localized approach is better, particularly in case of the heavily congestion network. This was mainly due to the combination of location awareness with stream redundancy.

End-to-end delay is almost consistent on both congestion levels and particularly from 2 to 4 senders, as shown in figures 7 and 8. This was taken as the minimum delay among all the received packets. On the other hand, QoE is maintained appropriately on the localized-redundant-stream. Figures 9 and 10 provide evidence of the perceived video quality to the end-consumers. However, the most divergence can be seen between the two compared schemes on figure 10, where the network is heavily congested.

Another interesting point is the finding of the required optimum number of peers that should serve a client. According to the presented results by the localized-redundant -stream, it is so clear from all the figures that 3 to 4 peers are good enough to provide high quality to the end-users within the current configuration. In contrast to that, with the randomized approach, it is difficult to give a precise indication for that, as the inter-connections among peers are unpredictable.

We also considered localized-multi-streaming dividing the stream into flows and transmitting from different sources (1S-→5S). This was also complemented by locality-awareness and computational load distribution. An additional feature in this approach is de-clustering. In order to avoid hot spots, we made sure that higher weight was given to the peer selection process to those peers belonging to different clusters.

This was compared to the former localized-redundant-streaming over the most factors affecting the QoS and QoE. First and foremost, the achieved throughput in multi-source streaming is competing with redundant-streaming (Figure 11). Throughput is increasing since bandwidth is higher when there are multiple sources and each peer sends only part of the stream, as in L-R-S.

Looking now at packet loss, it was obvious that L-R-S is better than L-M-S. However, multi-source streaming is achieving an acceptable level of packet loss ratio starting from 13% and then decreasing till 8%, which is good according to the QoE acceptability threshold of table 2. On the other hand, from the network load perspective, multi-source streaming, with this percentage of packet loss ratio, may be preferable over redundant-streaming due to the traffic overheads incurred by redundant-streaming.

These two scenarios were also examined over the packets end-to-end delay. It was noticed that L-M-S reduces the transmission latency with the increase in number of sources. Packet loss may not affect as much as end-to-end delay since the video playback deadline is more stringent in this regard.

Another benefit of L-M-S is that streaming time is decreasing with the number of sources sending the stream. This is vital in case of short connectivity or coverage, such as mobile cellular networks.

Lastly, the perceived quality PSNR by the end-user is obtained by the decoder. However, packet loss plays a crucial role in this parameter since it is proportional to the deduction of packet loss. Therefore, redundant-streaming is showing higher PSNR than multi-source streaming.

The proposed two techniques have shown noticeable improvements over ordinary P2P streaming. From the users' perspective, minimizing network traffic is not as important as achieving a smooth QoE. In this case the localized-redundant-streaming approach (L-R-S) will be the favourable choice. On the other hand, from the network operators' perspective, multi-source streaming seems to be a better choice.

VII. Concluding Remarks

A large variety of popular applications, including VoD, live TV and video conferencing, make use of P2P streaming frameworks. These have emerged from the fundamental principles of insulation and abstraction between the network and the application layers. With this regard, several studies published recently (e.g. [21]), including also some by the authors of this article (e.g. [14] [22] [23]), have identified that when the P2P overlay is designed in isolation from the underlay physical network, the P2P stream has detrimental effects on the network itself. To aim for scalability and user QoE, P2P solutions adopt redundancy, multi-stream, caching, statistical handovers and other similar techniques, which generate substantial network management and control problems to the network operator.

This problem motivates our work aimed at studying ways to maintain the QoE and scalability of the overlay, whilst reducing its detrimental effects onto the underlay. This article represents our initial attempt to pursue network-friendly P2P streaming. Our initial hypothesis that the combination of network locality, redundant-stream, and multi-streaming can lead to significant improvements is reinforced by the findings presented herein.

The difficulty in realizing this approach in practical systems is that it entails breaking the concept of network insulation from the application. In our current work we plan to further study the potential of other cross-layer optimization techniques, which opens the way towards stimulating research. Once the overlay is made aware of the underlay, or vice versa, the potential of other techniques can be unleashed. For instance, we are studying the use of machine learning for the purpose of correlating network and application conditions and building a network-friendly overlay network.

REFERENCES

[1] M. Alhaisoni, A. Liotta, M. Ghanbari, "Multiple Streaming at the Network Edge", In Proc. Of the First International Conference on Advances in P2P Systems, and published by IEEE Computer Society, October 11-16, 2009 - Sliema, Malta.

[2] 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

[3] Y. Liu, X. Liu, L. Xiao, L. M. Ni, and X. Zhang, "Location-aware topology matching in P2P systems", in Proc. IEEE Infocom, pp. 2220-2230, 2004.

[4] Z. Xu, C. Tang, and Z. Zhang, "Building topology-aware overlays using globalsoft-state", in 23rd International Conference on Distributed Computing Systems, (ICDCS), RI, USA, 2003, pp. 500-508

[5] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable application layer multicast", Computer Communications Review, vol. 32, pp. 205-217, 2002.

[6] B. Zhao, A. Joseph, and J. Kubiatowicz, "Locality aware mechanisms for large-scale networks", in in Proc. Workshop Future Directions Distrib. Comput. (FuDiCo), Italy, 2002, pp. 80–83.

[7] Pablo RODRÍGUEZ-BOCCA (2008), "Quality-centric design of Peer-to-Peer systems for live-video broadcasting", PhD theses.

[8] 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.

[9] 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.

[10] Mohammed Hefeeda et al, "Promise: Peer-to-Peer Media Streaming Using Collectcast", ACM MM'3, Berkely, CA,November 2003, pp. 45-54

[11] Mohamed M. Hefeeda and Bharat K. Bhargava, "On-Demand Media Streaming Over the Internet", in The Ninth IEEE Workshop on Future Trends of Distributed Computing Systems, 2003, p. 279.

[12] Kovendhan Ponnavaikko et al, "Overlay Network Management for Scheduling Tasks on the Grid", In ICDCIT, Springer, 2007, pp. 166-171.

[13] 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, 21-24 April 2009 Valencia, Spain.

[14] 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 Streaming in P2P Environments.

[15] Thomas L., Stefan S., and Roger W. (2006), "eQuus: A Provably and Locality-Aware Peer-to-Peer System", In Proc. Of the Sixth IEEE International Conference on Peer-to-Peer Computing , On page(s): 3-11.

[16] Shah A., Ying Q., and Gregor B. (2008), "CliqueStream: An efficient and Fault-resilient Live Streaming Network on a Clustered Peer-to-Peer Overlay", In Proc. Of the Eighth IEEE International Conference on Peer-to-Peer Computing, On page(s): 269-278.

[17] 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

[18] L. Golubchik, J. Lui, T. Tung, A. Chow, and W. Lee, "Multi-path continuous media streaming: What are the benefits?" *ACM Journal of Performance Evaluation*, vol. 49, no. 1-4, pp. 429–449, Sept 2002.

[19] J. Apostolopoulos, T. Wong, W. Tan, and S. Wee, "On multiple descriptionstreaming with content delivery networks," in Proceedings of IEEEINFOCOM, vol. 3, 23-27 June 2002, pp. 1736–1745.

[20] www.Joost.com

[21] H.Kolbe, O.Kettig, E. Golic, "Monitoring the Impact of P2P Users on a Broadband Operator's Network", Proc of IM'09, NY 1-5 June 2009. IEEE Press.

[22] N.N. Qadri, A. Liotta, "Effective Video Streaming Using Mesh P2P with MDC over MANETS", Journal of Mobile Multimedia, Vol. 5 (3), Rinton Press, 2009.

[23] A. Liotta, L. Lin, "The Operator's Response to P2P service demand", IEEE Communications Magazine, special issue on the IP Multimedia Subsystem. Vol. 45 (7), pp.76-83, IEEE, July 2007.

[24] 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, pp.365-382.

[25] X. Jiang, Y. Dong, B. Bhargava, "GNUSTREAM: a P2P media streaming prototype", In Proceedings of ICME'03 2 (2003) 325–328.

[26] Y. Guo, K. Suh, J. Kurose, D. Towsley, "Peer-to-peer on demand streaming service and its performance evaluation", In Proceedings of ICME'03 2 (2003) 649–652.

[27] T. Nguyen and A. Zakhor, "Distributed video streaming over internet," in *Proc. Multimedia Computing and Networking*, San Jose, CA, Jan. 2002, SPIE, vol. 4673, pp. 186–195.