# On Traffic Locality and QoE in Hybrid CDN-P2P Networks

**Moisés Rodrigues[1], Márcio Neves[1], Josilene Moreira[1], Djamel Sadok[1], Arthur Callado[2], Per Karlsson[3], Victor Souza[3]**

| Centro de Informática<br>Univers. Federal de Pernambuco<br>Recife – Brazil<br>{moises,mmn2,josilene,jamel}@gprt.ufpe.br | Campus de Quixadá<br>Universidade Federal do Ceará<br>Quixadá - Brazil<br>arthur@ufc.br | CDN and Cloud Computing Lab<br>Ericsson Research<br>Stockholm – Sweden<br>{per.p.karlsson, victor.souza}@ericsson.com |
|---|---|---|

Keywords: Discrete event simulation, hybrid, CDN, P2P, traffic locality

**Abstract**

Hybrid CDN-P2P applications, such as P2P applications, tend to sometimes ignore traffic costs at ISPs and generate large amounts of undesirable cross-ISP traffic. As blocking the traffic does not seem to solve the problem, some simple cooperation between the neighboring peers will be beneficial. In this case, some biased neighbor selection helps peers connecting to others within the same Autonomous System, hence keeping communication local. We developed a detailed simulation model to evaluate the problem and discovered that using a biased neighborhood selection in a hybrid CDN-P2P content network can even enhance user's quality of experience (QoE) and fundamentally reduce the cross-ISP traffic even with the presence of a transit ISP making de-localization decisions.

## 1. INTRODUCTION

Content distribution is today an important service provided by the Internet but in a classical client-server application and even in a single Content Distribution Network (CDN) the costs could be very high. Therefore, some content providers are starting to opportunistically enlist the help of peer-to-peer (P2P) hosts in distributing content additionally. Here, P2P hosts collaborate with others in distributing content hence resulting in an additional upload capacity for the network. The additional upload capacity of a P2P overlay helps distribute content increasing the content availability and could be essential for a user's quality of experience during a flash crowd event. This happens, for example, when the CDN capacity, either overall or local, does not cover all the demand from its clients.

To the best of our knowledge, hybrid CDNs are still at a preliminary development stage and have received limited study. Although [7] and [8] adopt a hybrid architecture, their P2P overlay only works for surrogate cooperation while leaving the clients as regular or non-cooperating entities. Several other studies take further steps into the use of P2P to deliver content and more specifically multimedia ones [8][9]. Although the efficiency of using a P2P approach to deliver content has been proven by many well known systems [10][11][12], the frequent transfer of large multimedia content across ISPs poses additional unwelcome costs.

The impact of P2P traffic in recent years has been so significant that some ISPs are deploying packet filters in an attempt to curve such cross traffic [13]. It has become a cat-and-mouse game where protocol developers found new ways to bypass these filters, for example, by obfuscation(altering traffic patterns to avoid detection). As this conflict does not seem to end, solutions based on collaborative work between P2P and ISPs seem to be promising. Consequently, several traffic localization strategies have been proposed and analyzed for several kinds of networks. Those designs give preference to connecting peers with others in its vicinity, keeping traffic local. This approach raises some questions with regard to some widely used Internet P2P systems. For example, when using BitTorrent-like [12]systems, the high cross traffic is due to the random selection of neighboring peers! This brings a very interesting question: is such random or location-blind selection necessary? As we can see in [15], for a simple torrent P2P system the answer is no, but would this answer be the same for a CDN-P2P hybrid Video On Demand (VoD) network? This is the question that our work aims to shed some light on and eventually give some insights into answering.

In this paper we, describe the implementation and analysis of two traffic locality strategies, per-hop count and per-AS, in a VoD hybrid CDN-P2P network as we aim to answer the question previously stated. Further we evaluate the impact of a transit ISP making de-localization rather than localization strategy to artificially increase cross traffic as proposed in [14]. In the remainder of this paper, we present related traffic localization work in Section2. Section 3describes the simulation tool used and the overlay implemented to evaluate the strategies. Section 4explains all metrics used in our simulations and section5 describes the locality strategies used. Section 6 describes the scenarios and the network topology followed by the results in Section 7 and Summary and future work in Section 8.

## 2. RELATED WORK

Several studies suggest the use of a Hybrid CDN-P2P to address these issues [26] [30]. Although [24] and [25] adopt hybrid architecture, their P2P overlay only works for surrogate cooperation while leaving the clients as regular non-cooperative entities. Several other studies take further steps into the use of P2P to deliver multimedia content [26] [9]. In [27] collaboration between clients is proposed, but clients are unable to receive data from different sources, such as from the peering community and CDN entities simultaneously. In [28] and [29] a hybrid architecture is proposed but without any traffic locality concern.

CDNSim [33] seems to be, at the time of this writing, the only CDN specific simulator mostly acknowledged by the community and have been based on the OMNnet++ open software. Similarly, the INET open source simulation framework is also based on OMNET++ and includes some CDN support. It nonetheless lacks any hybrid simulator support. Others such as [31][32] where used in the analysis of different CDN strategies while frequently leaving out important aspects of a CDN infrastructure, such as cache capacities and request routing [31][32].

Concerning traffic locality the work in [1] introduces the idea of an oracle service for each ISP. This oracle helps peers choose neighbors from a candidates list. This seems to be a good way to solve the trust issues between ISPs because each ISP has its oracle and none of the internal topology details needs to leave the ISP. In [2] a new aspect for the selection of peers was introduced as peers are ordered by last-hop bandwidth along with inter AS distance and AS-hop distance. A similar concept to the oracle is the P4P concept of an iTracker described in [16], the main difference is that information from the ISP topology is sent to others and this could be seen as a bad thing in terms of privacy disclosure and network security by ISPs.

In [3] authors propose an ISP independent way of making a biased neighbor selection using some well known content distribution networks. Each peer has a score based on the caches that the peer was redirected to and this score is used to group peers. Although very interesting, this strategy has a low accuracy, approximately only 33% of the time a peer selects peers along paths that are within the same AS. In [4] the locality strategy is based on limiting the number of external ISP connections. This is done by the tracker monitoring the number of active peers with external connections and if a new peer comes and asks for neighbors the tracker can give him only intra-AS peers or some intra-AS peers and one inter-AS peer at most. The paper [15] proposes a biased neighbor selection algorithm to improve the traffic locality using the concept of limiting external connections. External connections are identified by mapping IP addresses to AS numbers to identify peers and then select peers according to peers' AS number. However, this algorithm cannot handle peering agreements between ISPs therefore could not use ISP's services efficiently in such circumstance.

The idea in [15] is extended [5] by proposing a locality strategy based on a controller which will work as part of the tracker to make the peer selection. This selection will be done using some info from the ISP topology sent by the ISP which turns this strategy a little bit intrusive and one that can result in implementation difficulties. In [14] several comments are made about some effects and impacts of locality strategies leading to important results such as locality strategies being often responsible for increasing application performance, without an increase of up-link bandwidth usage while also decreasing cross traffic.

In [21] authors thoroughly examine the impacts of biased localization considering torrent stratification effect due to the unchoke/choke algorithm. This work aims on BitTorrent network directly thus some of the analyses can't be directly used in our VoD CDN-P2P BitTorrent based network.

In [22] a PPLive analysis is made. Clients were distributed around the world and data was collected from several channels of PPLive network [11]. Results show that PPLive traffic is highly concentrated, up to 85% of data transmissions, at the ISP level due to a decentralized, latency based and neighbor referral peer selection policy without infrastructure support.

Although [23] mentions the possibility to use biased locality strategies in all BitTorrent's levels there were no experiments where locality was considered in all levels. One of the main results is that locality, in all levels analyzed, can clearly reduce download time.

## 3. SIMULATION TOOL AND OVERLAY NETWORK

We developed a tool which aims to realistically simulate CDN networks, hybrid or not. This section aims to describe this simulator and how it was constructed.

Overall, the simulator has strategically been divided in three parts, as seen in Figure 1.The first of which is at the heart of any simulation system and is called Simulator Core. It deals with creating the simulation clock, creating and scheduling events, managing sessions from start to end of a simulation and collecting data. The second part, the Network Infrastructure, is concerned with the network aspects of the simulator. It is therefore responsible for managing and collecting network level information such as packet delay, link congestion events, throughput, etc. The third part of the simulator, the Hybrid CDN-P2P Overlay, reflects the design of the overlay system under study. Application, P2P and CDN related design issues are present at this level.

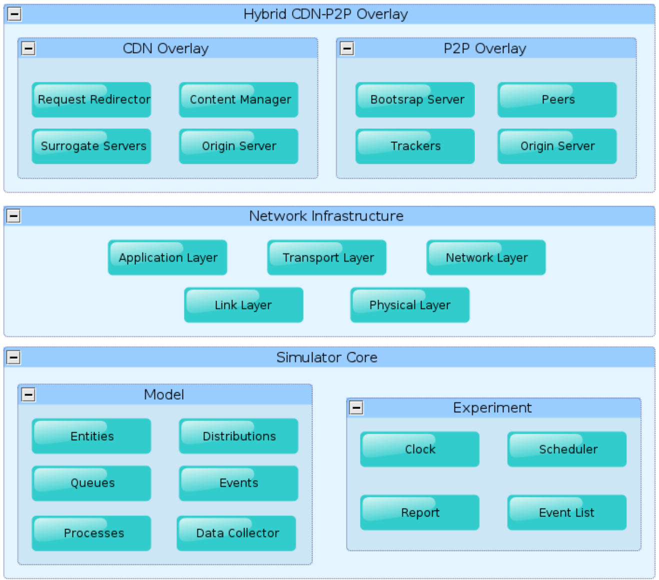


**Figure 1**. Simulator architecture

The simulator's entities responsible for the execution of given tasks have been implemented in the form of processes similar to those from an operating system. These are mapped into events in the simulator produced in a discrete manner. Consequently, in order to create a new entity, the designer must extend a given abstract class implemented within the simulator and specialize it with new capabilities to reflect the new entity. This is very similar to the way it is done in object-oriented languages such as Java.

At the network level, our simulator is different from other P2P and CDN simulation environments in that it allows for an absolute control of the network topology and the characteristics of its links and routers. These include the choice of a packet queuing strategy, delay values, buffer size for devices and packet loss for each link. These aspects are thoroughly simulated and not just approximated and there is an actual packet level exchange simulated within the overlay network.

Looking back at the programmer's task, our aim was to make it as easier as possible with scenarios ranging from simple overlay simulations to those that define new entities and/or strategies. For example, a Web server is in fact seen as a process executing over a network node. It inherits functionalities from an abstract class according to our previous explanation. This process is then mapped by the simulator into a number of discrete events. It also offers an API for sending and receiving datagrams, similar to Java Socket. The simulator then processes these events according to the network level conditions.

The simulator was initially developed to evaluate strategies in hybrid CDN P2P networks. It is based on the Desmo-J [17] framework which handles basic event driven simulation aspects like the clock and event scheduling.

The overlay used in this work is a CDN with a torrent-based P2P network between the clients. This overlay network is composed of three basic entities: Origin server, Bootstrap server and Tracker/Cache server. The Origin is the server with all the content and has only one instance in the network. A client should never be redirected to the Origin server. The Bootstrap server is the initial contact server for a client, which is responsible for redirecting the client to its Tracker/Cache server according to the strategy used. The Tracker/Cache server acts both as the Tracker server to the P2P network and as the Cache for the CDN network. A client asks for neighbor clients watching the same video, the Tracker could answer only with local information, meaning no cooperationbetween Trackers, or can request information about clients watching that video to other trackers (this will only happen if the tracker cooperation is turned on). To turn Trackers cooperation on or off you just have to change a variable value in a simulator's properties file. If the cache does not have the content then it requests the content to the Origin server and the client waits until the first chunk is downloaded by the Cache to be delivered next to the client requesting it.

In contrast with a normal P2P file sharing system, in a CDN-P2P hybrid system other entities are deployed in the overlay and have great influence on the overall system behavior. For instance, on a normal P2P file sharing system the P2P seeds are the only source for the full content and if all seeds from a specific content leave the network that content is temporally unavailable. In a CDN-P2P hybrid network there are caches with parts of popular content that never leave the network and an Origin server with all parts of all content being requested by clients.

Our video objects are fragmented based on [9]. Our overlay claims to have information about the ISPs topologies, ASes division and nodes proximity. This information could be easily obtained using an Oracle [1] system without being intrusive to the ISP, meaning no ISP needs to publish information about its internal topology. Using such entity the simulator can indicate addresses from the same AS and sort nearby clients both needed for our proposed locality strategies.A limited features web version of our simulator is available at http://cdn1.gprt.ufpe.br.

## 4. METRICS

Our simulator implements a set of metrics that aims to understand and evaluate the network overall state and QoE of the clients. In this section we give a short description of all the metrics used in this report.

### 4.1. Startup Delay

The Startup Delay represents the period between the time that the peer starts to download the video and the time the user starts watching. The user will only start to watch the video when a minimum playable part of the video is

This work is supported by Ericsson do Brasil.

downloaded. This minimum playable part is considered as a number of seconds of the video (we used 5 seconds for our simulations), which can be set in the simulator properties file. Then, this delay is found by calculating the time to transfer these first bytes needed to start watching that video.

### 4.2. Video Download Time

The Video Download Time metric aims to analyze the total video download time for each peer (the set of chunks that composes a video).

### 4.3. Playback Continuity

This is a metric that evaluates one aspect of the Quality of Experience (QoE). It measures whether the client was able to see the whole video without glitches. At every simulation second the peer calculates the total amount of data needed to watch the video continuously, using total time elapsed and bitrate of the video, and compares to the total amount of data downloaded. If the total amount of data is lower than the needed, the playback loss is incremented.We consider a buffering time equals to the minimum playable unit used in calculating the startup delay (we used 5 seconds for our experiments) to restart the checking of the playback continuity. At the end of the simulation, each peer has a playback continuity value that is an integer between 1 and 100 meaning the percentage of times that the playback was ok.

### 4.4. Total Bytes Transferred

The Total Bytes Transferred metric is collected to measure total bytes transferred in the network. Each and every packet that passes every link that connects every node is counted and entered in this metric. If a packet goes from a node A to a node D passing throughnodes B and C,it will increase the total bytes transferred for every link it goes (from A to B, B to C and then C to D).

### 4.5. Routers Cross Traffic

We use these metric to evaluate the traffic between ASes by accumulating the size of every packet that passes through a link between two ASes.

## 5. LOCALITY STRATEGIES

Our study focuses on two possible strategies for dealing with locality, as seen in Table 1. Each strategy affects how one entity communicates with another in three levels: Bootstrap behavior, Tracker behavior and Peer behavior.

The first possible strategy is to actually not use any peer biased selection at all. In this manner, called none, each tracker will handle a set of objects and peers will be redirected to the tracker that handles requests for the object that the client is requesting. All other selections are random. The second possible strategy is to consider network proximity as the metric for selection. In this approach, the client contacts the bootstrap, which responds with the nearest tracker to the client (in number of hops). The client contacts this tracker and it responds with the nearest peers that have the video (also based on number of hops). Then, the client connects to these peers to download the video.

**Table 1.** Comparison of the two methods used for each simulation run

| Strategy | Bootstrap | Trackers | Peers |
|---|---|---|---|
| ***None*** | The server that handles all requests for the object requested | Send random peers | Connects to random peers |
| ***By Hop*** | The nearest tracker in number of hops | Send the nearest peers based in number of hops | Connects to nearest peers in number of hops |
| ***By AS*** | Send tracker from the same AS or the nearest one otherwise | Send peers from the same AS or the nearest ones otherwise | Connect to peers from the same AS or the nearest ones otherwise |

The other possibility is to choose trackers and neighbor peers prioritizing the AS where the request is originated. In this strategy, when a client contacts the bootstrap for a video, it responds with a tracker located in the same AS of the client or (if this is not possible), with the nearest tracker. Likewise, when contacted for a peer list, this tracker will respond with peers located in the same AS of the client or (if there are too few peers or none in the same AS) the nearest peers to the client. The only modification from the previously described strategies behavior is the use (insertion) of a de-localization strategy by a transit ISP. Since our locality strategies are based on the use of an Oracle service in each ISP which would sort nodes by distance and indicate whether a node belongs to this ISP's AS or not [1], some transit ISP could intentionally answer the requests with de-localization (peers with worse location) information. For instance, it could make a client request for content be responded with neighbors belonging to any other AS when there are plenty of neighbors from the same AS (the transit ISP then profits from the increase in cross-traffic). With this issue in mind our simulator covers the possibility to have one AS or more where the locality strategies works as the exact opposite of what was described. For instance, if AS1 is marked to use de-locality and AS2 and AS3 are not, then when the overlay uses AS locality strategy all nodes belonging to the AS1 when choosing between possible nodes will always give priority to nodes from other ASs and nodes belonging to AS2. Similarly, AS3 will always give priority to nodes belonging to the same AS. Using hop count locality nodes will always give priority to nodes with the highest hop count.Therefore our locality strategy could be called a class 4 locality strategy according to [18].

## 6. TOPOLOGY AND SCENARIO DESCRIPTION

We use a 500 nodes topology with 10 ASs and 50 nodes on each AS. Nodes are divided in router nodes and border ones. The topology was generated using the tool BRITE with topology type Top down topology [6]. The edge connection model was the Smallest k-Degree with k=10 and we used default AS configuration parameters as well as routers configuration except for n which was 50. The model chosen was the Waxman with default parameters. Figure 2 shows an overview of the 500 nodes topology used in our simulations.

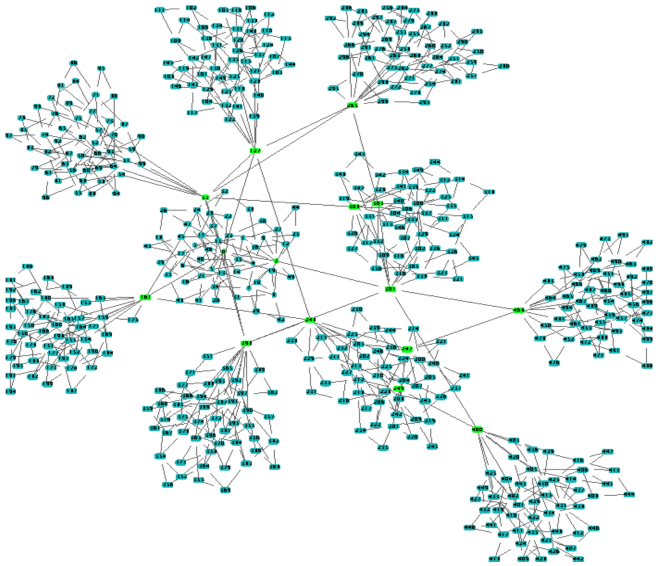

**Figure 2**. Overview of the 500 nodes topology generated with BRITE used in our simulations

To generate the evaluation scenario, the ProwGen [20] tool was used with a client arrival equals to one client per second, 6 different videos with mean size of 400 MB and a total of 741 clients with a mean permanence of 3202 seconds. All clients are placed in three different AS, out of ten available on the topology, and with 2Mbps symmetrical links to the router with which the client is connected. Routers have 10Gbps links between them and other CDN entities, for instance origin servers and replica servers. Since our main focus was to analyze traffic locality strategies, all replica servers have enough space to store all requested content. In other words, all content requested by clients to replica servers will be requested to origin servers and downloaded to replica server's cache.

Considering that the simulator is deterministic, and that repetition of a given scenario yields exactly equivalent results, there is no need for replications in this experiment and confidence intervals are not used in the evaluations.

## 7. RESULTS

We ran a series of experiments aiming to evaluate our locality strategies. One experiment for each locality strategy was run and one using no locality strategy just for reference. As described above, the AS locality strategy aims to keep as much as possible traffic inside the AS. This is done by redirecting peers to a cache that is inside the AS, choosing the nearest one if there isn't any, and making the peer chose between his neighbors the ones that are in the same AS or the nearest one (when there isn't one in the same AS). In per-hop locality the AS is totally ignored and the nearest one (in terms of number of hops) is always selected (nearest tracker and nearest peers).

In these experiments, we analyzed the impact of the algorithms involved in the selection of nodes (selection of tracker, selection of neighbors) in the network performance. In Figure 3 we see that the video startup delay, an important metric for CDNs, using the "Per-AS" locality strategy (Per-AS Locality represents the location of the autonomous system) is slightly smaller than the "per hop" locality strategy. The result is similar when looking at the total time to download video (Figure 4) and playback continuity (Figure 8): we can see that the only strategy where the playback continuity is stable and good (100%) is Per-AS.

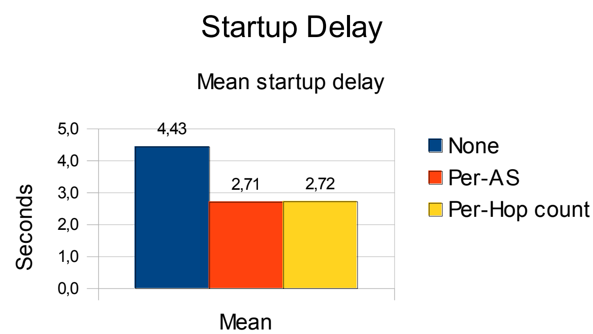


**Figure 3**. Start up delay comparing Per-AS, Per-Hop and none locality strategies

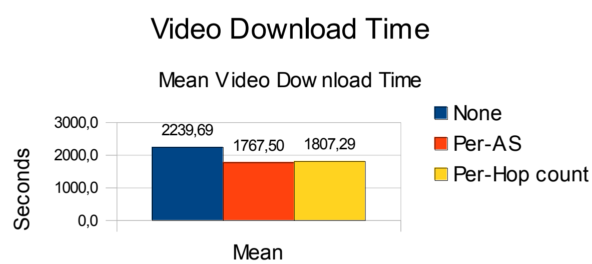


**Figure 4.** Time to download the video comparing Per-AS, Per-Hop and none locality strategies

This is due to the random selection of using no biased selection and in the Per-Hop count strategy the possibility that a peer connects to a cache from another AS therefore using links between ASes which are frequently more congested than inner ASes links. This difference exists due to the overload on the links between ASes as the worst locality strategy uses more those links. On both figures, we can see that using the locality strategy "None" is the worst choice possible. Observing Figure 5, one notes that using Per-AS locality the total bytes transferred is automatically increased. This increase, in principle undesirable, is caused by the impossibility of using one node in another network to download the video, even though it is closer in number of hops. This impossibility results in more traffic as with more hops the total traffic increases (the traffic is counted once per link it passes through). However, analyzing Figure 6 it is possible to observe that one can reach around 60% decrease in cross traffic using the locality strategies. This decrease agrees with what was found in other P2P studies [16] [1] that show an expected cross traffic decrease varying from 60% to 80%.

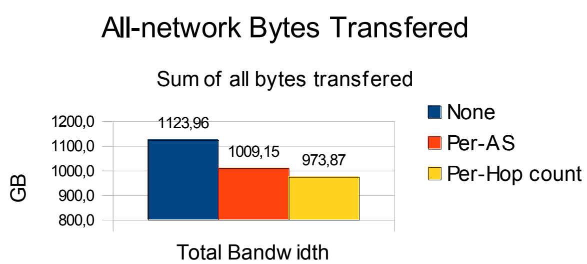


**Figure 5.** Total bytes transferred during the simulation comparing Per-AS, Per-Hop and none locality strategies

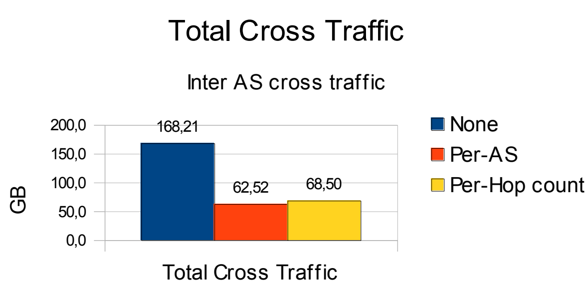


**Figure 6.** Inter-AS total bytes transferred comparing Per-AS, Per-Hop and none locality strategies

Figure 7 shows the behavior of cross traffic over time showing that using biased locality strategy the cross traffic stabilizes earlier than using no biased selection. This is due to the caches and the stabilization of the P2P network as using biased selection the P2P localized swarms along with the caches will result in insignificant or even zero cross traffic.

Between Per-AS and Per-hop locality strategies there is 9% less cross traffic using Per-AS locality strategy. Thus, even though there is an increase in traffic when using the Per-AS locality strategy, there will still be less traffic between different networks, which implies lower cost for network providers. From our experience we see that this cross traffic decrease is due to our topology and caches placement which results in the closest cache sometimes being a cache from a different AS than the peer's AS.

Another study made with the same scenario was to measure the impact of transit ISP de-locality. For this, we analyzed the de-locality strategy to modify the biased peers selection in the transit AS of the same scenario used earlier, which has a total of three ASs involved. It is important notice that only the transit AS was doing de-locality, all entities from the other two ASes where using Per-AS locality strategy normally.

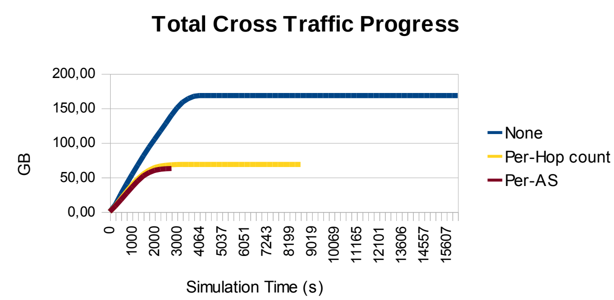


**Figure 7.**Inter-AS total bytes transferred progress over simulation time comparing Per-AS, Per-Hop and none locality strategies

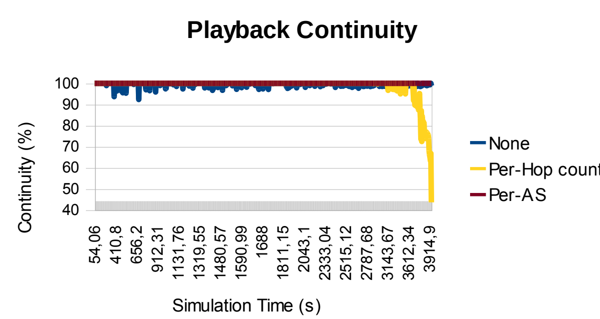


**Figure 8.** Peers reports of Playback continuity over simulation time comparing Per-AS, Per-Hop and none locality strategies

As one can see in Figure 9, there is an increase of about 25% in cross traffic resulting in an almost 30% gain compared to not using any traffic locality at all. The cross traffic using AS locality here was higher than previous graphs presented in this section because, although the same scenario was used, the tracker cooperation is used on our de-locality strategy implementation, which means that some of the traffic will be redirected to other caches.For a fair comparison, the Per-AS strategy simulation was run again with trackers cooperation turned on. Figure 10 is similar to Figure 7and presents the behavior of cross traffic over time showing that using biased locality strategy the cross traffic stabilizes earlier than using no biased selection. Another interesting aspect is that the QoE decreases as the mean startup delay increases almost 17% and the mean video download time increases 4% as we can see in Figure 11 and Figure 12, respectively.

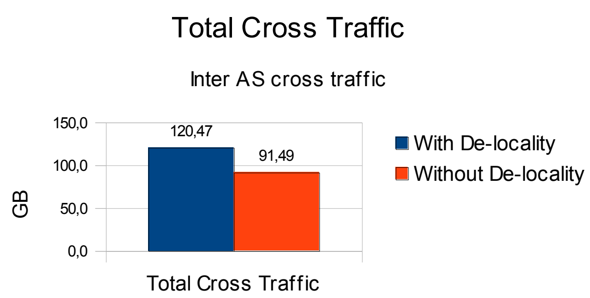


**Figure 9.**Transit ISP de-locality analyses comparing with pure AS locality strategy.

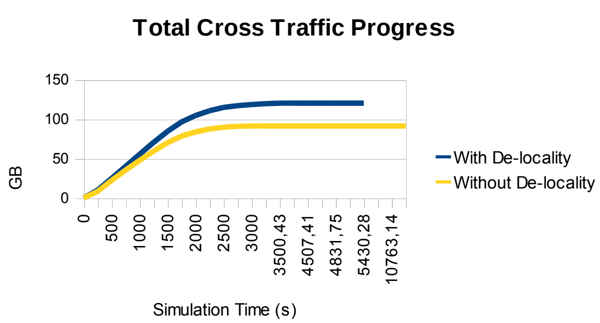


**Figure 10.**Transit ISP cross traffic over time comparing with pure AS locality strategy with and without de-locality

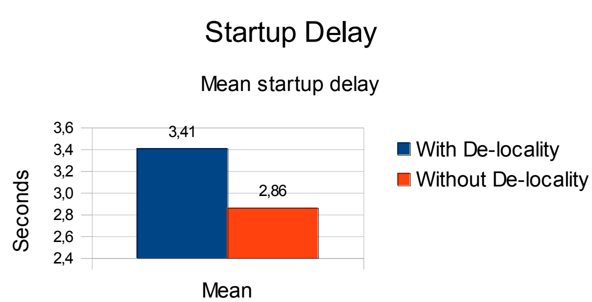


**Figure 11.**Startup delay mean comparing transit ISP de-locality impact and pure AS locality strategy

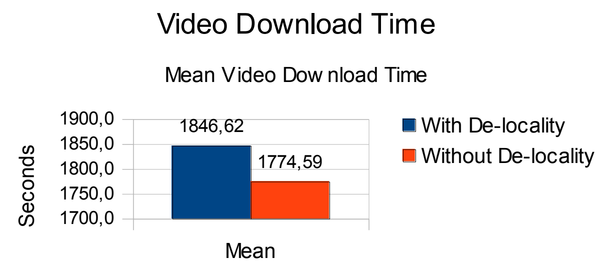


**Figure 12.**Video complete download time mean comparing transit ISP de-locality impact and pure AS locality strategy

## 8. SUMMARY AND FUTURE WORK

This paper has shown through the simulations undertaken that biased peer selection strategies slightly enhance client's QoE but also significantly reduces the cross-traffic between ASes. Our experiments show that along with a reduction of 60% in cross traffic the studied locality strategies decreased startup delay, video download time and increased playback continuity reaching 100% continuity using Per-AS in our scenarios. This answers our main question, showing that as reported for P2P-only systems in [15], one can reach less cross traffic with limited or even no loss to user's Quality of Experience (QoE) in a hybrid CDN-P2P VoDsystem. It is important to notice that a CDN-P2P environment is closely related to the topology and to the placement of caches.

We also notice that if the transit AS in our experiments decide to promote de-locality the gain on using a locality strategy decreases 25% along with worst mean startup delay and mean video download time.

Our next research steps would consider analyzing the impact of clients with asymmetric links such as those using ADSL and ADSL2+ and traffic locality strategies in the context of hybrid CDN-P2P systems. Another extension should consider bigger scenarios for the de-locality analyses and cache replacement strategies.

## 9. REFERENCES

[1] Aggarwal V, Feldmann A, Scheideler C. Can ISPS and P2P users cooperate for improved performance? *ACM SIGCOMM Computer Communication Review*. 2007;37(3):29. Available at: http://portal.acm.org/citation.cfm?doid=1273445.1273449.

[2] Aggarwal V, Akonjang O, Feldmann A. Improving user and ISP experience through ISP-aided P2P locality. *IEEE INFOCOM 2008 - IEEE Conference on Computer Communications Workshops*. 2008:1-6. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4544640.

[3] Choffnes DR, Bustamante FE. Taming the torrent. *ACM SIGCOMM Computer Communication Review*. 2008;38(4):363. Available at: http://portal.acm.org/citation.cfm?doid=1402946.1403000.

[4] Blond SL, Legout A, Dabbous W. Pushing BitTorrent Locality to the Limit. Technical Report, Inria, Dec. 2008. [Online]. Available at: http://hal.inria.fr/inria-00343822n v1/

[5] Ruan B, Xiong W, Chen H, Ye D. Improving Locality of BitTorrent with ISP Cooperation. 2009:2-6.

[6] Medina, A. Bottom-up Hierarchical Topologies. Available at:

http://www.cs.bu.edu/brite/user_manual/node19.html, last visited March, 2011.
[7] Mulerikkal J.P., Khalil, I. An Architecture for Distributed Content Delivery Network. IEEE International Conference on Networks, pp. 359-364, Nov. 2007.
[8] Kang S., & Yin H. (2010). A Hybrid CDN-P2P System for Video-on-Demand. *2010 Second International Conference on Future Networks*, 309-313. Ieee. doi: 10.1109/ICFN.2010.83.
[9] Huang Y., Fu T. Z., Chiu D., Lui J. C., & Huang, C. Challenges, design and analysis of a large-scale p2p-vod system. *Proceedings of the ACM SIGCOMM 2008 conference on Data communication - SIGCOMM '08*, 375. New York, New York, USA: ACM Press. Doi: 10.1145/1402958.1403001.
[10] “Emule”, http://www.emule.com/.
[11] “PPLive”, http://www.pplive.com/.
[12] "BitTorrent", http://www.bittorrent.org/.
[13] Forum DSLReports. Comcast is Using Sandvine to Manage P2P Connections. http://www.dslreports.com/forum/r18323368-Comcast-is-using-Sandvine-to-manage-P2P-Connections, last accessed in February 2011.
[14] Marocco E., Fusco A., Rimac I., Gurbani V. Improving Peer Selection in Peer-to-peer Applications: Myths vs. Reality. March 08 2010.
[15] Bindal R., Chan W., Medved J., Suwala G., Bates T., Zhang a., et al. (n.d.). Improving Traffic Locality in BitTorrent via Biased Neighbor Selection. *26th IEEE International Conference on Distributed Computing Systems (ICDCS'06)*, 66-66. Ieee. doi: 10.1109/ICDCS.2006.48.
[16] Xie H., Yang Y., Krishnamurthy A., Liu Y., and A. Silberschatz, "P4P: Explicit Communications for Cooperative Control Between P2P and Network Providers", in ACM SIGCOMM Computer Communication Review, vol. 38, no. 4.
[17] “Desmo-J”, http://desmoj.sourceforge.net/home.html.
[18] D'an G., Hossfeld T., Oechsner S., Chołda P., Stankiewicz R., Papafili I., Stamoulis G. Interaction patterns between P2P content distribution systems and ISPs. *IEEE Commun. Mag.*, Revised Aug. 2009.
[19] “BRITE”, http://www.cs.bu.edu/brite/.
[20] Busari M., Williamson C.. ProWGen: a synthetic workload generation tool for simulation evaluation of web proxy caches. Journal of Computer Networks, 38(6):779–794, April 2002.
[21] Cuevas R., Laoutaris N., Yang X., Siganos G., Rodriguez P. Deep diving into bittorrent locality. Technical Report, available from: http://arxiv.org/abs/0907.3874, 2009.
[22] Liu Y., Guo L., Li F., Chen S. A case study of traffic locality inInternet P2P live streaming systems. in Proc. IEEE ICDCS, Jun. 2009.
[23] Liu B., Cui Y., Lu Y., Xue Y. Locality-awareness in BitTorrent-like P2P applications. *IEEE Trans. OnMultimedia*, vol. 11, no. 3, pp. 316–371, 2009.
[24] Mulerikkal J.P., Khalil I. An Architecture for Distributed Content Delivery Network. IEEE International Conference on Networks, pp. 359-364, Nov. 2007.
[25] Zhang G., Xing C., Chen M.. "A distributed multimedia CDN model with P2P architecture,"International Symposium on Communications and Information Technologies ( ISCIT '06), pp. 152-156, Oct. 2006.
[26] Kang, S.; Yin, H. A Hybrid CDN-P2P System for Video-on-Demand. 2010 Second International Conference on Future Networks, 309-313. Ieee. doi: 10.1109/ICFN.2010.83.
[27] Xu D., Kulkarni S. S. Analysis of a CDN-P2P hybrid architecture for cost-effective streaming media distribution. Multimedia Systems, pp. 383- 399, Mar. 2006.
[28] Jiang H., Li J., Li Z., Liu J. Efficient Hierarchical content distribution using P2P technology. IEEE conference on networks, 2008.
[29] Liu X., Yin H., Lin C. A Novel and High-Quality Measurement Study of Commercial CDN-P2P Live Streaming. International Conference on Communications and Mobile Computing, 2009.
[30] Huang, C.; Wang, A. Understanding Hybrid CDN-P2P:Why Limelight Needs its Own Red Swoosh. In Proceedings of NOSSDAV '08 Braunschweig, Germany.
[31] Bent L., Michael R., Geoffrey V. M., Zhen X. Characterization of a large Web site population with implications for content delivery. In Proceedings of the 13th international conference on World Wide Web. 522 – 533.
[32] Wang L., Pai V., Peterson L. The effectiveness of request redirection on CDN robustness. In Proceedings of the 5th Symposium on Operating System Design and Implementation. 345 – 360.
[33] Stamos K., Pallis G., Vakali A., Katsaro, D., Sidiropoulos A. Manolopoulos Y. CDNsim: A Simulation Tool for Content Distribution Networks. ACM Transactions on Modeling and Computer Simulation, Volume 5, 2009.