Understanding the effect of streaming overlay construction on AS level traffic

Reza Motamedi and Reza Rejaie
Information and Computer Science Department
University of Oregon
e-mail: {reza.motamedi, reza}@cs.uoregon.edu

Rubén Cuevas
Departamento de Ingeniería Telemática
Universidad Carlos III de Madrid
e-mail: rcuevas@it.uc3m.es

Abstract—In this paper, we examine the effect of overlay structure in live peer-to-peer (P2P) streaming applications, on the inter-AS traffic. We identify a collection of edge ASes whose clients are likely to participate in P2P streaming applications. We derive the AS-level path between all pairs of selected edge ASes by simulating BGP routing over a snapshot of AS-level topology. We consider three different overlay construction strategies that try to minimize the following aspect of inter-AS traffic: (i) the total number of external connections for edge ASes that are mapped to customer-provider links to their providers, (ii) the number of connections that reach tier 1 ASes, (iii) the AS-level hop length of individual overlay connections. Our analysis reveals how constructing an overlay based on one of these metrics affect properties of the overlay with respect to other metrics as well as delivered stream quality in realistic settings.

Index Terms—Overlay, Underlay, AS-level characterization, P2P streaming

I. INTRODUCTION

P2P-streaming applications implement a random bootstrapping process by which peers select randomly other peers to exchange traffic with. This is easy to implement and provides a reasonably good performance. However, this process constructs a random overlay that leads to an unoptimized utilization of the underlay physical network. Due to the large amount of traffic that p2p-streaming applications generate this unoptimized utilization may harm Internet Service Providers (ISPs) in several ways. For instance, P2P streaming clients located in an AS may establish connections to remote peers located in other ASes despite the availability of peers within the local AS. These remote connections can be established through transit paid links and thus have a serious impact on the AS operational costs. In another example, an AS not hosting P2P streaming clients can experience that traffic associated to P2P streaming applications is being routed through it, for instance from a provider AS, thus incurring again on undesired costs for the former AS.

The research community have recently started to work in defining underlay-aware overlay construction algorithms. Most of the proposed solutions focus on localizing the generated overlay by forcing the peers to prefer establishing connections with other peers within the same AS [1]–[4]. This mitigates the first problem described above. However, even in the case of these localized overlays, few inter-AS links are needed in order to guarantee that the video stream reaches all the ASes forming the overlay. To the best of the authors knowledge none previous works have studied how these inter-AS overlay links impact the underlay topology. For instance, current solutions makes this selection at random what may still produce the second problem presented above.

This paper aims to understand the impact of different overlay construction algorithms at the inter-AS level in the underlay physical network. Towards this end we utilize a systematic methodology that guarantee the practical applicability of the obtained results. First, we identify a collection of relevant edge ASes that are likely to host a large number of p2p-streaming clients. These relevant ASes will be use to create the p2p-streaming overlay. Second, in order to evaluate the impact of the p2p-streaming overlay on the underlay network, we use state-of-the-art data and tools to construct a realistic underlay that connects the selected edge ASes. Based on the topological properties of this underlay we can identify three problems that a random inter-AS overlay (as that produced by current p2p streaming applications as well as proposed localization solutions) may cause in the underlay network: (i) the unnecessary use of longer AS-Paths that involves more ASes than needed in the distribution of the stream; (ii) the generated paths typically cross tier 1 ASes even if it is not necessary. This may lead to create an unnecessary traffic overload in the core of the Internet; (iii) selected relevant ASes may tend to use paid transit links to establish overlay links to other ASes despite the availability of free (e.g. peering links). This may lead to increase the cost associated to p2p streaming traffic for these ASes.

After identifying these problems, we propose three different underlay-aware overlay construction strategies to address each one of them. By extensive simulations we analyze the performance of these overlay construction algorithms and compare them to a random overlay. Our results suggest that on the one hand implementing any of the underlay-aware proposed algorithms will lead to mitigate the three presented problems compared to a random overlay. However, this reduction of the impact on the underlay network produces an slight degradation in the p2p streaming application performance.

The rest of the paper is organized as follows: Section II introduces the most important facts that motivates our study.
Section III details the methodology utilized and Section IV discusses the conducted simulation study and the obtained results. Finally, Section V concludes the paper.

II. MOTIVATION

P2P-streaming applications typically define a random bootstrapping process in which each node select (or is assigned) a random set of other peers to exchange traffic with. This defines a peer-level overlay topology in which peers exchange traffic with long-distance peers despite having other closer peers. This leads to form a random overlay topology also at the AS level, that may have a negative impact in the underlay topology (e.g. using unnecessary longer AS-Paths). Figure 1 shows graphically this phenomenon. In this simple example we observe that peer 1 (marked as “1” in the Figure) located at AS1 is connected to peers in AS2 and AS3 despite it has available peers in its own AS. Furthermore, P2P-streaming applications generate an important amount of traffic, then the described random bootstrapping process brings traffic to costly inter-AS links (e.g. transit links) for the AS. This fact, has motivated the research community to develop Locality techniques [1]–[4]. These techniques make the peers to select as neighbors other peers within the same AS. Just few links with external peers are established. These external links guarantee that the distributed stream reaches the AS. Typically, an AS establishes n incoming and n outgoing links, being n the number of substreams associated to the distributed stream. However, all these locality solutions worry exclusively about the local AS, and the ASes to which an external connection is established are still randomly selected. Figure 2 illustrates this scenario. In this example, peers within AS1 are now connected to each other as result of the locality algorithms. However, AS1 establishes two inter-AS connection to AS3 and just one to AS2, despite AS2 is one hol closer to AS1 than AS3.

In this paper we zoom out our focus from the peer level and study the impact of the AS overlay formed by the P2P-streaming application in the underlay physical topology. To this end, we perform an extensive simulation study considering both a random overlay construction strategy and different underlay-aware overlay construction algorithms. The obtained results allow us to evaluate the performance of the different proposed strategies. Note that our simulations are run on top of a realistic set-up that considers both relevant ASes to P2P-streaming applications and an accurate underlay topology obtained from BGP information.

III. METHODOLOGY

Our main objective is to understand the impact of relevant underlay-aware overlay construction schemes in the underlay topology as well as in the application performance. Towards this end we define a three steps methodology: (i) we identify those representative ASes for our study hosting a large number of P2P-streaming clients; (ii) we compute a detailed underlay topology connecting the previously selected ASes. This underlay topology includes information about the ASes relationships (peering vs client-provider) as well as the tier associated to each AS. For this purpose we rely on real BGP data; (iii) we define different overlay construction strategies including both random and underlay-aware strategies. Furthermore, we evaluate the performance of the different strategies throughout extensive simulations. Figure 3 provide a summary of the different parts of our methodology.

Next, we will provide a detailed description of each of these steps.

A. Identifying relevant ASes

The first step of our methodology consists on identifying relevant ASes for our study. Since world-wide large-scale P2P streaming applications do not exist at the time of this study¹ we focus on the ASes which host a large fraction of P2P file sharing nodes in the Internet. The rationale for

¹The most important P2P streaming applications such as PPLive [5] or SopCast [6] are mainly popular in China. Therefore the representative ASes in this case are reduced to few chinese ones.
adoption of this choice was that since these ASes are hosting a large number of P2P file sharing nodes, they will likely host large number of P2P streaming nodes when this service becomes pervasive. In particular, we utilize the population of Gnutella clients per AS obtained in a previous work [7] in order to find the most representative ASes for our study.

First of all we map the IP address of each Gnutella client to its correspondent AS. For this purpose, we use RouteViews’ [6] BGP snapshots to perform a prefix matching between the IP addresses and the announced prefixes of the ASes. We find that around 14K ASes host P2P clients, but most of them provide Internet connection to only a handful of them. Hence we filter out these ASes and consider only ASes that host more that 50 peers. This value has been selected since it lead to an important reduction in the number of ASes to be considered but the decrease in the number of peers is minimal. Specifically, the number of ASes decreases to 4083 whereas the number of peers shrinks from 72M to only 70M. This step is marked as (1) in Figure 3.

In order to further filter the ASes to find those most relevant to our study we use the information regarding the tier of each individual AS. For this purpose we use TierClassify [8]. The algorithm used in this tool relies mainly on the assumption that all tier-1 ASes should be interconnected with one another. Therefore it tries to find a clique among the ASes with highest degrees. Once the tier-1 clique is identified, the algorithm simply follows provider-customer relationships and classifies other ASes such that each tier-\(n\) AS can reach the tier-1 clique in \(n - 1\) hops. Based on the tier information we realize that tier-2 and tier-3 ASes host more that 90% of peers. This result reveals that tier-2 and tier-3 ASes are the most representative to our study so that we filter out the ASes at other tiers.

Finally, we apply a third filter based on the extent of the geographical area covered by an AS. Specifically, we use results from a previous paper [7] that demonstrate that 90% of P2P nodes’ population is hosted in both country and state level ASes. Therefore, we consider these type of ASes for our study. Note, that the last two filtering techniques are marked as (2) in Figure 3.

Overall, after the applied filtering techniques our final dataset includes 2902 ASes (20% of the original set) and 65M of P2P clients (90% of the original set). Therefore, the applied filtering techniques allow to identify the set of most representatives ASes to our study.

B. Inferring ASes underlay topology

Once we have identified the relevant ASes for our study, we need to define the underlay topology connecting them, so that latter on we can evaluate the performance of different overlay construction algorithms. In particular, we determine the AS-path between every pair of the identified representative ASes. To this end, we use C-BGP [9] which is a simulation of BGP to produce pairwise routes between ASes. C-BGP abstracts the AS-level topology as a collection of interconnected routers, where each router represents an AS. It receives as input the BGP routing policies for each relation between connected ASes. We generate this input based on snapshots of the AS-level Internet topology provided by CAIDA [10] where each link between two ASes is represented with the relationship between them. Using C-
BGP we perform a detailed simulation of BGP routing over these annotated snapshots of the AS-level underlay to infer the corresponding AS-path for each pair of ASes in our dataset.

There are some interesting aspects to highlight from the underlay topology generated following the described methodology. First, Figure 4 shows the CDF of the AS-Path length associated to each pair of ASes within our dataset. We observe that the median number of AS hops between any pair of ASes is 4. Since current P2P-streaming overlays are formed by inter-ASes links generated at random, then we expect that the median value of the AS-Path for the P2P-streaming communications is 4. More interestingly, since the overlay inter-AS links are generated at random, it is likely that most of the ASes are using longer inter-AS links than needed, thus making the P2P-streaming traffic to cross more ASes than needed.

Second, for each pair of ASes in our dataset we have computed the highest tier crossed by the AS-Path. Our result reveal that 75% of the AS-Paths include at least one tier-1 provider, whereas the other 25% includes at least one tier-2 provider. Again since the current P2P-streaming overlays are randomly generated we would expect that 75% of the inter-AS links established pass through a tier-1 provider. This suggests that in the future P2P-streaming traffic may have a serious impact in the core of the Internet.

Furthermore, ASes are mainly worried about the cost associated to P2P-streaming traffic. Intuitively, the more P2P-streaming traffic goes to transit paid links the higher the associated cost is. A transit paid link is a link established between a customer and a provider ASes. The customer pays for both the upstream and downstream traffic crossing that link. Furthermore, two ASes can establish a peering link. The traffic crossing this link is free for both ASes. Therefore, in order to minimize the cost associated to the P2P-streaming traffic an AS would prefer to use links to its customers or peers. For each pair of ASes in our dataset we have computed the type of link used on each end to establish the overlay connection. Therefore, we have 4 possible combinations: (i) both links use transit links then both have to pay for the traffic, (ii) the first AS uses a free link (a peering link or a link to a customer) and the second a transit link, (iii) the second AS uses a free link and the first a transit one and, (iv) both ASes use a free link. Table I shows the percentage of overlay links associated to each one of these cases within our dataset. The construction of a random overlay would lead to a distribution of costs similar to that one shown in the Table. Therefore, it seems that there is room to reduce the ASes costs.

In summary, the presented results suggest that construction of random overlays may have an important impact in the underlay topology namely, using longer AS-Paths than needed, imposing an important overhead in the Internet core or increasing the cost of those ASes hosting a large number of P2P-streaming clients. Therefore, it would be interesting to understand whether the definition of new underlay-aware overlay construction strategies may help to reduce the impact of P2P-streaming applications in the underlay topology and how such strategies may affect the performance of P2P-streaming applications. Towards this end, in the next subsection we define three underlay-aware overlay construction policies that are motivated by the obtained results. Furthermore, in Section IV we conduct extensive simulations to assess the performance of each one of the defined strategies.

### C. Overlay construction strategies

In the previous subsections we have defined the set of relevant ASes to our study and used state-of-the-art tools to infer the underlay topology connecting these ASes. Furthermore, we have discussed what is the impact of using a randomly generated overlay (as the current P2P streaming systems do) in the underlay topology.

In this subsection we define three different underlay-aware overlay construction strategies in addition to the random construction policy. Each of these strategies addresses one of the issues unrevealed in the previous subsection, i.e. AS-Path length, impact on the Internet core and ASes cost associated to P2P-streaming traffic. Note, that our objective is evaluating the performance of these strategies, then how to implement them is out of the scope of this paper and we leave this discussion for future work. Next we describe each one of the four strategies:

- **Random**: The inter-AS overlay connection are built totally at random. The resultant AS overlay topology
is the one generated by any existing P2P-streaming application in which peers select at random other peers to communicate with. Furthermore, existing locality techniques aim to localize traffic within an AS limiting the number of external connections to other ASes. In this case the inter-AS overlay links are again generated at random. Then this strategy also captures the overlay topology associated to current locality-biased P2P-streaming solutions.

- **Shortest AS Path**: In this case every AS creates an inter-AS overlay link to the n other ASes with the lowest associated AS-Path. Note that we are assuming that an ideal locality mechanism is implemented within each AS, thus n represents the minimum number of inter-AS links needed to guarantee the correct distribution of the stream within an AS. Previous locality studies [ADD REF] demonstrate that n is equal to the number of sub-streams forming the stream. This strategy minimizes the number of ASes that the P2P-streaming traffic has to cross, thus reducing the impact of P2P-streaming traffic in the underlay topology. In order to implement this strategy, we use a greedy algorithm in which for each AS in our dataset we calculate the AS path to every other AS within the dataset. Afterwards, we rank the other ASes from the one having the shortest AS-Path to the one having the longest and select the n ASes with the shortest associated AS-Paths. Note, that the filtering process described in Subsection III-A helps to improve the scalability of this greedy algorithm.

- **Minimization of impact in the Internet core**: This strategy tries to minimize the usage of ASes located in the higher tiers. This is, AS-paths passing exclusively through ASes in tier-3 are preferred than AS-Paths including one or more tier-2 ASes, that are in turn preferred to paths including one or more tier-1 ASes. We use a similar greedy algorithm as the one described above. However, in this case the destination ASes are ranked based on the criterion we have just defined. Therefore, this strategy minimizes the P2P-streaming traffic penetration in the core of the network, then reducing the impact of P2P-streaming traffic in the Internet core.

- **Cost minimization**: This strategy aims to minimize the cost for both ends of the overlay inter-AS link. Therefore, overlay links in which both ends use a peering link to its immediate AS are preferred to those overlay links in which one of the end ASes needs to use a transit link to an immediate provider AS. These in turn, are preferred to those overlay links in which both end ASes need to use a transit link for the communication. As in the previous cases, we implement a greedy algorithm in which each AS in our dataset establishes an overlay link to the other top n ranked ASes based on this cost metric.

### IV. Evaluation

In this section we present our evaluation methodology and describe the obtained results. Furthermore, we compare the performance of the different underlay-aware overlay construction strategies defined before.

#### A. Simulation Set-up

Our simulation experiment emulates a typical P2P-streaming application in which we have a single source that distributes the video by dividing it in n substreams. For our simulation we configure n = 10. Furthermore, we consider that an optimal traffic localization algorithm is used to localize traffic within each AS. Based on this each AS establishes a maximum number of 10 incoming and outgoing overlay links to other ASes. In order to guarantee the scalability of our simulation the simulation engine randomly selects 1000 ASes from the set of 2902 relevant ASes found in Section III-A. Furthermore, these 1000 ASes build four different P2P-streaming overlays following the four construction algorithms described in Section III-C. These overlays are formed on top of the underlay topology constructed following the methodology described in Section III-B. Finally, in order to provide statistically meaning results we perform 1000 simulation runs. In each run we select a different set of 1000 ASes and construct the four different overlays based on the selected set of ASes.

#### B. Results

Based on the previously described simulation experiments we compute the following 4 metrics for each of the four overlay construction strategies:

- **AS-Path length**: We calculate the AS-Path length associated to each established overlay link. In this case, for each simulation run we obtain a a vector including one value per each overlay link established among the 1000 ASes.

- **Overall Percentage of overlay links reaching Tier 1, 2 and 3 ASes**: For each simulation run we compute three values that correspond to the three previous percentages.

- **Overall Overlay cost**: For each simulation sample we measure the total number of transit links used by the overlay ASes. In this case, the output data is a single value for each simulation run.

- **Average Playback quality**: For each simulation run we measure the total number of substreams (out of the 10 transmitted by the source) received by each AS in the overlay. Afterwards we calculate the average value across all the ASes. Therefore, the output data is a single value per simulation run.

Table II summarizes the obtained results. It presents in the columns the different overlay construction strategies and in the rows the described metrics. For each metric we present the average and the standard deviation values across
the 1000 simulation runs. First of all we observe that the standard deviation value is in any case (at least) two order of magnitude smaller than the correspondent average value. This suggests that any considered scenario will produce similar results to those presented in the table. Given that the configured set-up is based on realistic data, we believe that the obtained results captures quite well the expected behaviour on a real environment.

Furthermore, on the one hand, we observe that any of the underlay-aware construction strategies offer better results than random when considering the underlay-relevant metrics. Hence from the underlay network perspective it would be recommendable to use one of these polices. However it is not clear which one is the most suitable since the optimization of a metric have different impact in the other ones leading to a trade-off situation. On the other hand, the results suggest that the utilization of any of the proposed underlay-aware overlay construction strategies leads to a slight degradation on the playback quality between a 3% and a 7% compare to a random overlay. Although this performance degradation is small, it is not clear what are the incentives that a P2P-streaming provider has to modify its application to implement an underlay-aware algorithm if this leads to a performance degradation.

V. CONCLUSION

In this paper we have studied the performance of underlay-aware overlay construction strategies at the inter-AS level. Toward this end we have defined a comprehensive methodology that allows to reproduce a realistic environment composed by a P2P-streaming overlay topology including relevant ASes in top of an accurate and detailed underlay network. Furthermore, we have used this set-up in order to evaluate and compare the performance of different overlay construction algorithms including three underlay-aware and a random construction strategy. The result reveal that any of the underlay-aware construction algorithms offer better results for the underlay-related metrics but a random construction strategy offers a better playback rate. This leads to a trade-off scenario in which interests of ASes and P2P-streaming providers seems to be confronted.

<table>
<thead>
<tr>
<th>Results Summary</th>
<th>Random</th>
<th>Shortest AS-Path</th>
<th>Internet core impact minimization</th>
<th>Cost Minimization</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS-Path length</td>
<td>3.94, 0.0008</td>
<td>2.43, 0.0006</td>
<td>3.58, 0.0009</td>
<td>3.84, 0.0009</td>
</tr>
<tr>
<td>% Overlay Links reaching Tier 1</td>
<td>74.16, 1</td>
<td>50.17, 0.2</td>
<td>24.74, 0.1</td>
<td>69.03 , 0.2</td>
</tr>
<tr>
<td>% Overlay Links reaching Tier 2</td>
<td>24.55, 0.1</td>
<td>46.69, 0.2</td>
<td>69.28, 0.2</td>
<td>28.24, 0.1</td>
</tr>
<tr>
<td>% Overlay Links reaching Tier 3</td>
<td>1.00, 0.01</td>
<td>2.37, 0.05</td>
<td>4.67, 0.1</td>
<td>1.47 , 0.03</td>
</tr>
<tr>
<td>Overall Overlay Cost</td>
<td>18770, 15</td>
<td>18.570, 17</td>
<td>18.330, 17</td>
<td>17.940, 21</td>
</tr>
<tr>
<td>Play-back Quality</td>
<td>9.92, 0.002</td>
<td>9.25, 0.007</td>
<td>9.7, 0.004</td>
<td>9.74, 0.004</td>
</tr>
</tbody>
</table>

TABLE II

PERFORMANCE EVALUATION FOR THE DIFFERENT OVERLAY CONSTRUCTION STRATEGIES. EACH CELL REPRESENT THE AVERAGE AND THE STANDARD DEVIATION (AVG, STD) FOR THE CORRESPONDENT METRIC.

REFERENCES