Technical Report CIS-TR-10-01, Department of Computer and Information Science, University of Oregon, January 2010

Characterizing the Global Impact of the P2P Overlay on the AS-level Underlay

Amir H. Rasti, Reza Rejaie
University of Oregon
{amir,reza}@cs.uoregon.edu

Walter Willinger
AT&T Labs-Research
walter@research.att.com

***Abstract*—This paper examines the problem of characterizing and assessing the global impact of the load imposed by a Peer-to-Peer (P2P) overlay on the AS-level underlay. Toward this end, we make three contributions: (i) We describe the major challenges in addressing this problem (e.g., capturing overlay snapshots, determining the load generated by of individual connections, and identifying the AS-level paths taken by individual overlay connections) and present existing techniques for addressing these issues along with their limitations. (ii) We present a methodology that combines a collection of best practices for tackling the above challenges. (iii) We apply the proposed methodology to characterize several aspects of the load imposed by a real-world P2P overlay (e.g., Gnutella) on the AS-level underlay. This study represents an initial attempt at deepening our understanding of how overlays get mapped into the AS underlay.**

I. Introduction

Peer-to-Peer (P2P) applications, such as P2P file sharing or P2P video streaming, have become increasingly popular over the Internet during the past few years. In these applications, participating peers form an overlay and contribute their outgoing bandwidth by forwarding some pieces of the content to their neighboring peers in the overlay. These P2P applications contribute a significant fraction of Internet traffic. The likely increase in the access link bandwidth of average Internet users can be expected to contribute an even higher volume of P2P traffic on the Internet.

The large volume of traffic associated with P2P applications has led to a growing concern among ISPs, especially edge ISPs or Autonomous Systems (ASes), that need to carry the P2P traffic relayed by their costumers. Recently, researchers and practitioners have focused on the idea of reducing the volume of external P2P traffic for edge ISP by localizing the connectivity of the P2P overlay [2]–[4]. This approach only deals with the local effect of an overlay on individual edge ASes. Despite the large and growing volume of P2P traffic on the Internet, assessing the *global* impact of a P2P overlay on the network underlay, namely individual ASes, remains a challenging problem and is not well understood. This is in part due to the fact that investigating this problem requires a solid understanding of an array of issues in two areas: *(i)* design and characterization of overlay-based applications, and *(ii)* characterization of AS-level topology and BGP routing in the underlay. Another significant challenge is dealing with inaccurate, missing, or ambiguous information about the AS-level underlay topology, AS relationships and tier properties, and BGP routing policies.

This paper investigates the problem of assessing the load imposed by a given overlay on individual ASes in the network that we call AS-level underlay. It describes a number of significant steps towards exploring this issue and makes the following main contributions. First, we show that assessing the impact of an overlay on the underlay requires tackling three challenging problems including *(i)* capturing accurate snapshots of the desired overlay, *(ii)* estimating the load associated with individual overlay connections, *(iii)* determining the corresponding AS-level path in the underlay for individual overlay connections. We describe existing approaches to address these problems as well as their assumptions and limitations.

Second, we present a methodology for assessing the impact of an overlay on the AS-level underlay. Our methodology incorporates a collection of the best known practices for capturing accurate snapshots of an overlay and, more importantly, for determining the AS-level path corresponding to each overlay connection. We rely on snapshots of the AS-level Internet provided by CAIDA for the same dates that our overlay snapshots are captured. In these snapshots, each inter-AS edge is annotated by the estimated type of relationship between connected ASes. To improve the accuracy and consistency of the provided AS relationships, we test them against another commonly used inference algorithm [5]. We also infer the tier information for individual ASes using *TierClassify* [6] tool applied to BGP routing table snapshots retrieved from RouteViews [7] archive. Relying on the *C-BGP* [8] tool, we perform a detailed simulation of BGP routing over these annotated snapshots of the AS-level underlay and we infer the corresponding AS-level path for each overlay connection to determine the load experienced by individual ASes.

Third, we demonstrate our methodology by assessing the impact of four overlay snapshots of the widely deployed P2P

This technical report is an extended version of our paper in the *Passive and Active Measurement Conference 2010* [1].

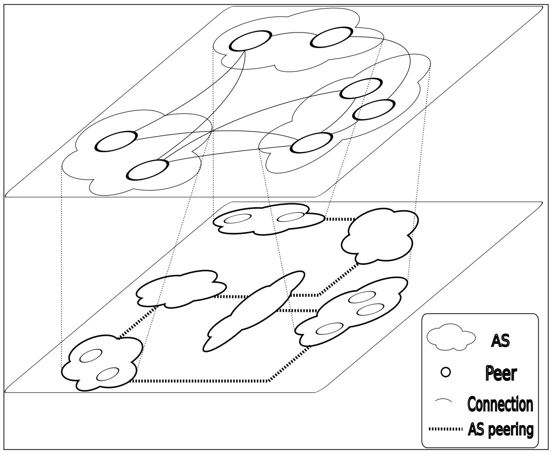


Fig. 1. Mapping the connections of a p2p overlay to the AS-level underlay.

application Gnutella on the AS-level underlay. These overlay snapshots are collected over a four year period and illustrate the growth of an application-level overlay.

Our methodology produces an estimate of the aggregate load that an overlay imposes on individual ASes and connections between neighboring ASes in the underlay. Once this rather detailed information is collected, an important issue is to represent the load in some aggregate but useful fashion. Toward this end, we map peers in each overlay snapshot to *edge ASes* in the corresponding underlay snapshots and characterize the resulting ASes in terms of size and tier membership. More importantly, we present a few temporal and spatial characteristics of overlay load on the underlay.

The rest of this paper is organized as follows: In Section II, we further elaborate on the problem of mapping an overlay on the AS-level underlay, present an array of challenging issues that arise in addressing this problem, and present existing techniques to deal with these issues and their limitations. Section III presents our methodology for assess load of an overlay to AS-level underlay. We use our proposed methodology to characterize the mapping of Gnutella overlay topologies on the corresponding AS-level underlay in Section IV. Section V provides an overview of related work. Finally, we conclude the paper and sketch our future plans in Section VI.

II. The Problem and Challenges

The main problem that we are addressing in this paper is mapping the traffic associated with a P2P overlay to the AS-level underlay. In this process, the input is a P2P overlay consisting of the IP addresses and port numbers of the participating peers together with their neighbor lists. The output of the process is a global AS-level load matrix in which the amount of traffic between each pair of connected ASes is provided in both directions. In Figure II, the overlay is shown on top and ...

In this section, we describe an array of challenges that arise in quantifying the load imposed by an application-level overlay on the network-level underlay consisting of individual ASes and the connections among them. This is the most obvious granularity for assessing load on the underlay—individual ASes are managed as independent networks and are primarily concerned with how traffic is exchanged with their neighboring ASes because of of its immediate financial and operational implications. The load on the AS-level underlay can be expressed in terms of an *AS-level load matrix* where for two connected ASes AS_i and AS_j, the element l_{ij} represents the traffic load destined to AS_j from AS_i.

Deriving the AS-level load matrix for a given overlay requires the following steps:

- Capturing the topology of application-level overlay,
- Estimating the load on individual connections of the overlay,
- For each overlay connection, identifying the AS-level path in the underlay,

Adding the load of each overlay connection to all the inter-AS links for ASes along the corresponding AS-level path provides the aggregate load of all overlay connections on each inter-AS link in the underlay, *i.e.*, the load on link between AS_i and AS_j (l_{ij}) in the AS-level load matrix.

While the basic idea is straight forward, some of these steps, especially steps 2 and 3, pose serious practical challenges. Next, we describe common practices for achieving each one of these steps and their related challenges.

A. Capturing Overlay Topology

Capturing the overlay topology for a given application is feasible if each peer provides its list of neighbors in response to a query (*e.g.*, Gnutella). A crawler can start with contact information for a few peers and progressively query known peers to discover other peers and connectivity among them. While capturing a snapshot of an overlay topology is feasible, the captured snapshots is likely to be *distorted* due to the dynamics of peer participation (*i.e.*, churn) and the resulting evolution of the overlay [9]. More specifically, a non-negligible fraction of discovered edges of an overlay topology may be reported by only one node implying that the status of this connection has changed during the crawling process. We call these *uncertain edges*. Both inclusion or exclusion of these uncertain edges introduces distortions of the captured snapshots. Increasing the speed of the crawler can reduce the number of uncertain edges and improve the accuracy of the resulting overlay snapshot.

B. Estimating the Load of Individual Overlay Connections

The observed load of individual connections is collectively determined by several factors including: *(i)* the number of peers that generate traffic (*i.e.*, sources) and the pattern of traffic generation by these peers, *(ii)* the location of source peers in the overlay, *(iii)* the topology of the overlay, *(iv)* the deployed strategy by each peer to relay (*i.e.*, forward) all or part of the received traffic. For example, the data exchange rate among different P2P connections in a BitTorrent swarm could be significantly different based on their pairwise available bandwidth and their available content. In contrast, different connections of a tree-based P2P streaming are likely to observe the same load since each peer should relay all of its received packets.

The observed load on individual connections of a flat signaling overlay (*e.g.,* top-level overlay in Gnutella) depends, among other things, on the structure of the overlay topology. If we assume that the probability of issuing a query for all nodes is roughly the same, and each query is flooded within a certain scope of its source (*e.g.,* n hops), the observed load of each connection depends on its *betweenness*, and may significantly vary depending on the structure of the overlay topology. In summary, the load on individual connections of the overlay depends on many variables and we are not aware of an existing model for estimating per connection load in a general case.

C. Identifying the AS-path for each Overlay Connection

Identifying the AS-level path for a given overlay connection is the most important and most challenging step in mapping (or assigning) the load of each connection onto the AS-level underlay. The two approaches to tackle this problem are: *1) Measuring AS-level Paths,* and *2) Inferring AS-level Paths*.

1) Measuring AS-level Paths: In this method, using a large number of geographically distributed vantage points across the Internet, one can conduct end-to-end measurements to estimate the AS-level path between all possible pairs of end-points. This approach has two significant limitations: First, the router-level paths between end points are usually measured using traceroute which is known to have a variety of problems, particularly, in the presence of routers implementing load balancing [10]. Even if an accurate router-level path can be captured, to our knowledge, there is no reliable approach to accurately translate this into an AS-level route due to IP address aliasing (*i.e.,* using multiple addresses on the same router) [11]. Second, to capture a complete set of AS-paths, we need at least one vantage point at each edge AS where peers are likely to exist. However, the number of vantage points in unique ASes is often significantly smaller than the total number of edge ASes. This in turn limits the coverage of discovered end-to-end paths by this approach.

2) Inferring AS-level Path: In this approach, one can emulate BGP routing over a snapshot representing the AS-level underlay topology to determine the AS-paths between any pair of ASes. To infer the AS-level paths for a set of end-to-end connections through simulation, one needs at least the following two pieces of information: *(i)* an accurate and complete snapshot of the Internet AS-level topology, and *(ii)* realistic relationships between connected ASes in terms of the associated BGP policies that each AS uses for prioritizing incoming updates and filtering outgoing updates. However, as discussed below, obtaining this information is a challenging task in its own right.

D. Capturing AS-level Underlay Topology

There are two popular approaches to capture the Internet AS-level topology, namely *traceroute-based topology discovery* and *BGP-based topology discovery*.

1) Traceroute-based Topology Discovery: As discussed earlier, previous studies used traceroute to discover router-level paths between any pair of end points. Using a large number of geographically scattered vantage points, one can discover a large number of such end-to-end paths [11]. However, the captured routes may only reveal a portion of the AS-level topology. Furthermore, known limitations of traceroute [10] can result in significant errors to the captured paths (*e.g.,* incorrect edges in the topology). Finally, accurately mapping a router-level path to AS-level path is a non-trivial task and prone to error as mentioned earlier.

2) BGP-based Topology Discovery: Each received BGP update includes the AS-paths that the packets will take to get to the origin AS of that BGP updates. Therefore, each BGP router keeps the AS-paths to all reachable ASes over the Internet. This can be used to infer a partial but reliable view of the Internet AS-level topology from the viewpoint of the receiving router. Public BGP archiving and monitoring centers (*e.g.,* RouteViews [7]) often establish BGP peerings with a number of volunteer ISPs across the Internet to get access to BGP tables from multiple viewpoints and effectively have a more complete picture of the AS topology. They usually provide current and archived BGP tables that include AS-paths from each participating AS to all other reachable ASes. One can extract all the AS links from these AS-paths and form an inferred AS topology. While the produced AS-level topology is known to miss a significant portion of AS links [12], [13] (esp. peering links between lower tiered ASes), this approach is very popular by networking researchers and engineers for generating estimated connectivity structures of the real-world AS-level topology.

E. AS Relationships & Policies

Individual ASes are often unwilling to reveal their relationship with their neighboring ASes and the associated policies due to security and business concerns. Furthermore, the number and nature of these relationships and associated policies may change with time. Internet registries (*e.g.,* RIPE and ARIN) provide such information but they are likely to be incomplete, inaccurate or obsolete. Given the large number of ASes (*i.e.,* 30,000+), soliciting this information from individual ASes (even if they are cooperative) is prohibitively expensive.

In the absence of explicit information on AS relationships, prior studies have relied on heuristic methods to categorize these relationships into the following three types: *(i)* Customer-Provider, *(ii)* Peer-Peer, and *(iii)* Sibling-Sibling. Gao [5] uses the relative degree of neighboring ASes to infer the type of each relationship. In [14] the authors extend Gao's method and use new heuristics to increase the inference accuracy. CAIDA uses the method proposed in [14] and provides AS-level topologies that are annotated by the AS-relationship type.

These techniques often assign "tier" information to individual ASes as well. The inferred AS relationships by these algorithm may result in error when the underlying heuristics are not true or the relationship does not exactly fall in one of these three categories. Despite the potential error, these heuristic techniques are considered as best known practices for estimating AS relationships.

BGP routing is mostly based on the policies defined by each AS. The usefulness of any simulations involving policies will depend strongly on the accuracy of the policies implemented. However, the most common policy in effect is commonly referred to as *valley-free routing* according to which no customer will provide transit service between two of its providers. In order to implement this, ISPs should filter out any routes that they have received from their providers in the outgoing announcements to other providers. t

F. Quantifying the Impact on the Underlay

So far we have discussed the challenges in accurately estimating the impact of an overlay on the AS-level underlay in the form of AS-level load matrix. We note that the load matrix is a large (N*N) matrix where N is the number of affected ASes in the Internet. While the load matrix is likely to be sparse, (*i.e.*, has many elements that are zero or very small), it provides a rather detailed representation of load on the underlay which is difficult to comprehend. Therefore, it is essential that we measure the impact with a small number of aggregate properties that represent overall characteristics of impact and provide useful insight. This raises two basic questions: *(i)* how does one identify useful aggregate properties of load?, and *(ii)* how does one ensure that the proposed metrics are not too coarse; that is, unable to reveal informative fine-grained features due to aggregation/averaging?

III. Our Methodology

This section describes our methodology for addressing some of the challenges that we discussed in Section II. Our methodology composes a collection of best practices in addressing each one of the stated challenges. While our choices in addressing each problem are specific, our methodology is generic in the sense that if new and improved techniques for dealing with any the challenges are developed, they can be plugged into our methodology to increase its accuracy.

An Overview: At a high level, our approach can be summarized as follows. We capture the overlay topology of widely deployed P2P applications using a fast crawler. To assess the load on individual connections, we make the simplifying assumption that all connections of the overlay experience roughly the same average load. To identify the AS-level path for individual connections of the overlay, we leverage the best practices for capturing AS-level topology, and identifying the AS relationships and tier information. Finally, using detailed simulation of BGP routing over the annotated AS-level topology, we infer AS-level paths between all edge ASes in order to assess load on the AS-level overlay. In the rest of this section, we elaborate on each one of the above steps in our methodology.

A. Capturing Overlay Topology

In our study, we use multiple snapshots of the top-level overlay of Gnutella. Gnutella is a widely-deployed P2P file sharing application. Gnutella has a two tier overlay topology with a subset of peers, called ultrapeers, forming an unstructured top-level overlay, and the remaining peers, called leaf peers, connecting to the overlay through a few ultrapeers. We only focus on the top-level overlay among ultrapeers as examples of a large-scale P2P overlay. We use a high performance P2P crawler called Cruiser (see [15]) to capture snapshots of the Gnutella overlay in less than 10 minutes. To our knowledge, Cruiser is the fastest P2P crawler available and thus captures the most accurate and complete snapshots of the Gnutella overlay.

The Gnutella overlay is primarily used for exchanging control messages (*e.g.*, query and responses). Therefore, the load on individual connection of the overlay is likely to be less than the load in an overlay used for content delivery (*e.g.*, BitTorrent or CoolStreaming). However, we are not aware of any technique to *reliably* capture overlay topology in content delivery P2P systems. Clearly, our methodology can be applied to any other overlay topology.

B. Estimating the Load of Individual Overlay Connections

Without loss of generality, we assume in our analysis that all connections of the overlay experience the same average load in both directions. This simplifying assumption allows us to focus on the mapping of overlay topology on the underlay AS-level topology. If a reliable model for load of individual connections is available, it can be easily plugged into our methodology by assigning two weights (one in each direction) to each connection of the overlay. In this paper, we simply assume that the weight is one for all connections in both directions.

C. Mapping Overlay Connections to the Underlay

Identifying the AS-path for each overlay connection is the most challenging piece in our methodology. Toward this end, we take the following steps:

1) Building AS-Grouped Overlay: We group all peers within each AS and identify those ASes as edge ASes for a given overlay. This provides an “AS-grouped” view of an overlay where each node represents an edge AS, and each edge represents overlay connection(s) between two ASes. We also associate a weight to each AS to indicate the number of P2P connections that are mapped to an edge in the AS-grouped overlay. When peers at both ends of a P2P connection are located in the same AS, we simply drop that connection since it does not impose any inter-AS load on the underlay.

We use BGP routing table snapshots taken on the same date as the overlay snapshots and provided by the RouteViews project [7], to map the IP address of a given peer to its corresponding AS. In the process, we identify the AS that has the longest matching prefix with the peer’s IP address by leveraging CAIDA’s Coral Reef package.

2) Determining AS-level Underlay Topology: We use snapshots of the AS-level Internet topology provided by CAIDA [16]. These snapshots of the AS-level topology are constructed using BGP routing tables from the RouteViews project for the corresponding dates. Furthermore, each inter-AS link in the topology is annotated with the relationship between connected ASes. These relationships are estimated by an algorithm applying the "valley-free routing" constraint on the topology. Briefly, the algorithm works by *(i)* assuming that the AS with the highest degree along each path has the highest position in the hierarchy (top of the hill) , and *(ii)* assigning chains of costumer-provider relationships to edges that are hanging at each side of the top AS all the way to the ends of the path. Furthermore, other heuristics (*e.g.*, degree difference) and information from routing registries (e.g., RIPE, ARIN) were used to infer peer-peer and sibling-sibling relationships [14].

Second, we leverage the "LogRelInfer" tool developed at the University of Massachusetts - Amherst (implementing the algorithm proposed by Gao in [5]) on CAIDA's annotated snapshots of AS-level topology to explore any inconsistencies between this algorithm and CAIDA's algorithm. Our comparison revealed that more than 95% of derived relationship are consistent. Furthermore, CAIDA's derived topology contains a larger number of ASes and relationships. Therefore, we simply use these snapshots after the mentioned validation. We also used another tool "TierClassify" (implementing the algorithm presented in [6]) to classify ASes into tiers. This tool mainly relies on the fact that all tier-1 ISPs should be interconnected with one another and therefore tries to find a clique among the ASes with highest degrees. As we used this tool with the AS relationships obtained from CAIDA, we faced a large number (~ 30) ASes classified as tier-1.While the commonly accepted tier-1 networks are about 10-15, we slightly modified the tool's parameters to reduce the number of tier-1 ASes to 15.

D. Inferring AS-level Paths of Each Overlay Connection

We use detailed simulations of the BGP protocol over our annotated snapshots of the AS-level underlay topology to derive the AS-level path for each connection of the overlay. Toward this end, we use the *C-BGP* simulator[1]. *C-BGP* is a multi-purpose BGP simulator [8] that can compute the outcome of the BGP decision process on a desired set of interconnected routers with arbitrary BGP policies. *C-BGP* can also take an annotated AS-level topology as input and convert it into a router topology where each AS is represented by a router. The annotated relationship for each link is then translated into a predefined set of BGP policies that are applied to each side of the respective BGP peering. The basic idea behind the policies is to implement *valley-free* routing. Table I shows the common routing policies that are used by *C-BGP*. This table shows the following important points *(i) communities* are used to mark incoming and outgoing routes, *(ii) local-preferences* give first, second and third priorities to routes received from customers, peers and providers, respectively, *(iii) route filtering* is used to prevent redistributing provider and peer routes to other providers and peers in order to ensure valley-free paths. In summary, in the absence of reliable information about deployed (potentially heterogeneous) routing

[1]developed at UCL, Belgium; http://cbgp.info.ucl.ac.be/

TABLE I
BGP POLICIES USED TO IMPLEMENT AS RELATIONSHIPS IN *C-BGP*

```
->customers:
    community remove COMM_PROV
    community remove COMM_PEER
<-customers:
    local-pref 100
->peers:
    deny if (community COMM_PROV) || (community COMM_PEER),
    community remove COMM_PROV,
    community remove COMM_PEER
<-peers:
    community append COMM_PEER,
    local-pref 80
->providers:
    deny if (community COMM_PROV) || (community COMM_PEER),
    community remove COMM_PROV,
    community remove COMM_PEER
<-providers:
    community append COMM_PROV,
    local-pref 60
->siblings: (no filter, no community change)
<-siblings: (no filter, no community change)
```

policies by individual ASes, we adopt a commonly accepted and intuitive set of policies in our simulations.

We note that representing each AS as a single router in our simulation results in producing a single AS-level path between each given pair of edge ASes. This implies that potential multiple AS-level paths that may exist between two ASes in practice (as presented in [17]) are not incorporated in our simulations. While this assumption somewhat simplifies the problem, we are not aware of any existing technique to capture these subtle behavior of BGP routing. While the basic idea of using simulation is rather straight forward, we have faced a few problems that are worth mentioning.

Relationship Cycles in the AS-level Topology: In our initial simulations over annotated underlay AS-level topology from CAIDA, the BGP protocol did not converge. Further examinations revealed that the annotated topologies contained several relationship cycles. For instance, a simple 3-hop cycle is formed when ISP_1 is a customer of ISP_2, ISP_2 is a customer of ISP_3, and ISP_3 is a customer of ISP_1. In addition to being intuitively incorrect, these cycles led to infinite oscillations of the BGP protocol. To fix this problem, we identified all such loops in each topology and then broke the loop by changing the relationship on the edge that was most counter-intuitive. In particular, we changed the relationship on the link where customer degree was larger than the provider degree. In cases where more than one such link was identified, we selected the link with the largest difference in the degree of the customer and provider ASes.

Policies for Sibling Relationship: Unknown policies for Sibling relationship introduce another set of unknown variables to our BGP simulations. Sibling ASes are simply defined as those owned by the same company, however, the deployed policies between sibling ASes may significantly vary across different companies based on the size of ASes, their locations and purposes.

Considering the shared ownership status between sibling ASes, we initially used the policy of sharing all routes and

TABLE II
SNAPSHOTS OF GNUTELLA OVERLAY TOPOLOGY

Snapshot	Date	#Peers	#Connections
G-04	2004-11-20	177k	1.46M
G-05	2005-08-30	681k	5.83M
G-06	2006-08-25	1.0M	8.64M
G-07	2007-03-15	1.2M	9.80M

TABLE IV
MAPPING PEERS TO THE EDGE ASES

Snapshot	#Peers	#Edge ASes	# Edge ASes (w 100+ peers)
G-04	177k	1,872	154
G-05	681k	2,670	290
G-06	1.0M	3,462	414
G-07	1.2M	3,684	460

their preferences between sibling ASes as if they were one AS. However, this strategy also resulted in formation of cycles effectively preventing BGP from converging. Further investigations revealed that sharing all routes and preferences between sibling ASes may not always be appropriate [2] Since assigning proper routing policies to each pair of sibling ASes is not feasible, we modified all sibling relationships to Peer-to-Peer relationships over which no transit service is provided.

IV. INITIAL CHARACTERIZATION OF THE EFFECT OF OVERLAYS ON THE UNDERLAY

This section presents the effects of a Gnutella top-level overlay on the AS-level underlay from several angles. First, we present our datasets and then we examine the mapping of peers to their corresponding edge ASes. Finally, we map overlay connections to the AS-level underlay and assess the load on core ASes.

A. Datasets

1) Snapshots of Gnutella Top-level Overlay: We use four snapshots of the top-level overlay (*i.e.*, connectivity among ultrapeers) in the Gnutella file sharing application. These snapshots are captured using a high-speed P2P crawler, called *Cruiser* These snapshots are collected in 4 consecutive years starting at 2004. Table II summarizes the main information associated with the collected overlay snapshots including date of capturing, number of peers, and number of connections. We use the labels G-xx to refer to these snapshots throughout this section.

Table II reveals that the number of coexisting peers in the top level overlay of Gnutella has become more than six times larger over the 4 year period. The number of edges has also proportionally grown.

[2]For example, consider a small overseas branch of a large ISP as its sibling AS. In this case, the BGP policies should be strictly asymmetric to prevent any transit traffic to go through the small AS. Alternatively, a national level ISP that uses two sibling ASes on each coast of the U.S., should adopt a symmetric set of policies between them with minimal filtering to maximize the benefit of the peering relationship.

TABLE III
SNAPSHOTS OF BGP ROUTING TABLE

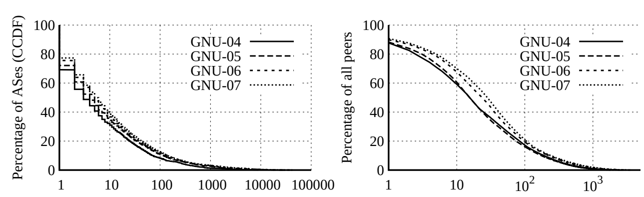
Date	# Prefixes	# Originating ASes	# Total ASes
2004-11-20	165,406	18,499	18,733
2005-08-30	185,256	20,424	20,628
2006-08-25	210,098	23,038	23,262
2007-03-15	229,237	24,669	24,923

2) Snapshots of BGP Routing Table: We used daily snapshots of BGP routing tables from the RouteViews archive for the same days that our overlay snapshots were collected as shown in Table III. This ensures that each overlay is mapped to a corresponding AS-level underlay. The table summarizes important statistics about each BGP snapshot including the number of reported prefixes in each snapshot, the number of unique ASes that announced at least one network (*i.e.*, originating ASes), and the total number of unique ASes observed in the snapshot. The significant growth in the number of prefixes and ASes (both originating and total) during the four year period demonstrates the growth of the Internet. The difference between originating and total ASes (around 200-300 ASes) can be associated with transit ASes that do not announce any prefixes.

B. Mapping Peers to the Underlay

Identify Edge ASes: We start by mapping peers in our overlay snapshots to the AS-level underlay to identify edge ASes. Table IV summarizes the number of edge ASes (*i.e.*, ASes where peers are located) and the number of major edge ASes that host more than 100 peers. Given the number of active ASes in the Internet for each year from Table III, Table IV indicates that *(i)* the edge ASes cover 10%-15% of all ASes in the Internet, *(ii)* only 1% to 3% of the edge ASes have more than 100 peers. These results demonstrate that our overlay topology has a rather wide footprint across the Internet.

Distribution of Peers per Edge AS: Figure 2 depicts the distribution of peers among edge ASes for all four overlay snapshots. In Figure 2(a) the distribution of AS peer population is shown in a semi-log scale CCDF plot for all four overlay snapshots. This Figure shows two interesting points: *(i)* the distribution of peers across ASes is very skewed. Depending on the overlay snapshot, 20-30% of edge ASes host only a single peer, and 60-70% of ASes host less than 10 peers, while less than 10% of ASes host between 100 to 10,000 peers. *(ii)*



(a) AS peer population(CCDF) (b) Peer dist. among ASes(CCDF)

Fig. 2. Distribution of peers over ASes

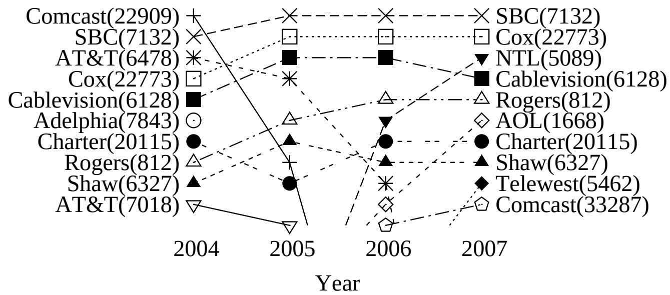


Fig. 3. Identity & evolution of top-10 edge ASes hosting highest number of peers

while the collected snapshots are far apart in time, they exhibit a very similar trend. Figure 2(b) depicts the distribution of all peers among the ASes sorted by peer population in semi-log scale CCDF format. As shown in this figure, the top 10 ASes represent between 30-40% of all peers and the top 100 ASes represent about 80% of all peers.

Identity and Evolution of Edge ASes: We now turn our attention to the identity of major edge ASes and examine their evolution over time. Figure 3 shows the top-10 edge ASes (ordered from top to bottom on the y axis) based on the number of peers they host, and the evolution of top-10 edge ASes across the four overlay snapshots. While the population of peers and thus the ranking of top-10 ASes change over time, 60% of the edge ASes consistently remain in the top-10 during the 4 year period. We note that the change in the population of peers in an AS could be due to a variety of reasons. For example, in some cases an ISP could change its allocated IP addresses, or its policies for using AS numbers. *e.g.*, AS numbers 22909 and 33287 both belong to Comcast. Interestingly AT&T has two top-10 ASes in 2004 but none in 2007.

Distribution of Peers & Edge ASes across Tiers: Another interesting issue is to examine the position of peers and edge ASes in the AS-level hierarchy (*i.e.*, AS tiers). Table V shows how the peers are distributed in ASes at different tiers. This table shows that a majority (65% - 82%) of peers are located in tier-2 ASes. This is reasonable because most of the large residential providers are tier-2 ASes. A small portion (2% - 13%) of peers are in tier-1 ASes. These are large telecom companies with both residential services and backbone infrastructure. Another interesting trend in Table V is that peers seem to be moving from tier-3 and tier-1 ASes to tier-2 ASes during this 4 years. The large fraction of peers in tier 1 in the 2005 overlay snapshot is primarily caused by AS7132 (SBC). In the 2005 snapshot, SBC hosts the highest number of peers and is classified as a tier-1 AS whereas in other years SBC is classified as tier 2. This suggest that the large fraction of peers in 2005 is a result of a mis-classification of SBC in 2005.

TABLE V
DISTRIBUTION OF PEERS ACROSS EDGE ASES WITH DIFFERENT TIERS.(% OF PEERS IN EACH SNAPSHOT)

Snapshot	Tier 1	Tier 2	Tier 3	Tier 4
G-04	6.5	70.6	22.5	0.4
G-05	12.7	64.9	21.9	0.4
G-06	2.6	80.4	15.8	1.1
G-07	1.9	81.1	16.3	0.7

TABLE VI
DISTRIBUTION OF EDGE ASES ACROSS DIFFERENT TIERS (% OF ASES)

Snapshot	#Ases	Tier 1	Tier 2	Tier 3	Tier 4
G-04	1,872	0.8	50.2	43.8	4.9
G-05	2,670	0.6	47.7	44.9	6.3
G-06	3,462	0.4	42.4	49.0	7.8
G-07	3,684	0.4	42.5	45.9	10.5

Table VI shows the distribution of edge ASes across different tiers for all four snapshots. Clearly, a majority of edge ASes are evenly divided between tier-2 and tier-3 while a small but growing fraction of these ASes are tier-4. Comparing this table with Table V reveals that there is a large number of tier-4 and tier-3 ASes with small peer populations whereas the average peer population in a tier-2 AS is much larger.

C. *Mapping Edges to the Underlay*

We now characterize the load imposed by overlay connections on the AS-level underlay by focusing on "transit ASes". These are the ASes along AS-paths for individual overlay connections excluding the first and last ASes (edge ASes) where peers at the end of each connections reside. Table VII shows the number of connections, the number of corresponding AS-paths and the number of AS-paths with 100+ connections for all four overlay snapshots. Both the number of overlay connections and unique AS-paths are growing over time. But the number of AS-paths is roughly 2 orders of magnitude smaller than then number connections.

To examine the mapping of overlay connections to AS-paths more closely, Figure 4(a) depicts the CCDF distribution of the number of overlay connections per AS-path between any pair of connected edge ASes in log-log scale for all overlay snapshots. This distribution is very skewed showing that about 10% of paths have less than 10 connection while 1% of paths observe more than 200 connections.

Observed Load by Individual Transit ASes: Since we assumed that all connections have the same load, we simply quantify the load on each transit AS by the number of

TABLE VII
MAPPING OVERLAY CONNECTIONS TO THE AS-LEVEL UNDERLAY

Snapshot	#P2P Conn.	#AS Paths	%AS Path (w 100+ conn.)
G-04	1.46M	192k	2.0
G-05	5.83M	384k	2.9
G-06	8.64M	605k	2.8
G-07	9.80M	684k	2.7

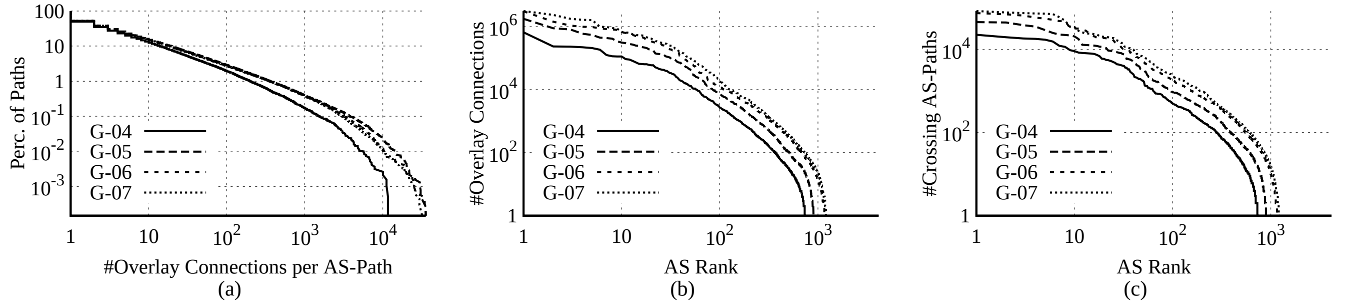


Fig. 4. (a) Dist. of number of overlay connections per each AS-paths, (b) Number of overlay connections passing through each transit AS, (c) Number of AS-paths crossing through each transit AS.

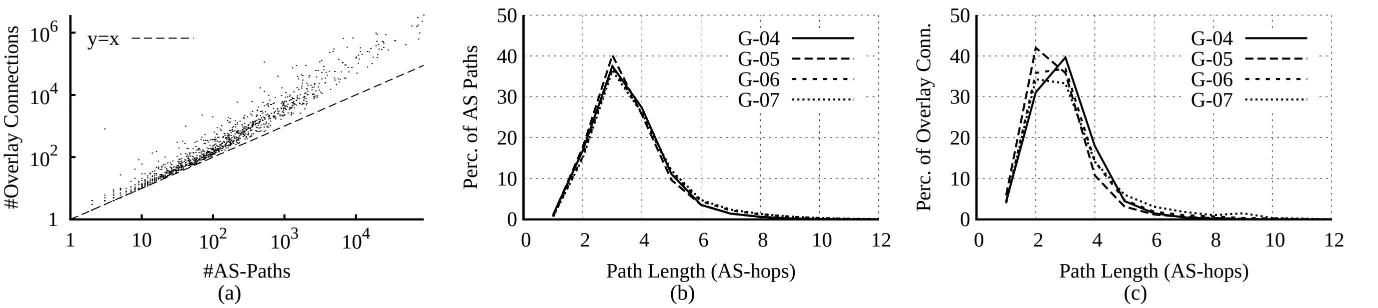


Fig. 5. (a) Scatterplot of number of crossing AS-paths vs. number of overlay connections passing through each AS, (b) Dist. of AS-path length between connected edge ASes, (c) Dist. of AS-path length for all overlay connection.

overlay connections crossing that AS, *i.e.*, number of overlay connections for which this AS appears on the corresponding AS-level path. Figure 4(b) shows the number of overlay connections that cross each transit AS in log-log scale where ASes are ranked from high to low based on their overall observed load. This figure demonstrates that the load on transit ASes is very skewed. A small number of ASes carry a large volume of traffic while the load on most transit ASes is rather small. Interestingly, the shape of this histogram for all four snapshots is very similar except for the outward shift in more recent snapshots due to their larger population and number of connections. This similarity in the load histogram could be due to two observations: *(i)* stability of most top-10 edge ASes over the four year period, and *(ii)* the constraint of valley free routing over the hierarchical structure of AS-level underlay.

To further investigate the underlying causes for the observed skewed histogram of load among transit ASes, we examine the number of unique AS-paths (associated with overlay connections) that pass through each transit AS in Figure 4(c). This figure shows that the histogram of AS-paths crossing per transit ASes has a very similar shape. This suggests that the number of crossing connections for individual ASes is primarily determined by BGP routing. Figure 5(a) validates this observation by showing the number of crossing AS-paths and crossing connections through each transit AS (as x and y values) in a scatter-plot. This figure essentially connects the previous two distributions, and confirms that the observed load by individual ASes primarily depends on their appearance across associated AS-paths. Interestingly, once the number of cross AS-paths exceeds a certain threshold (a few hundred), their observed load increases at a much faster pace .

Distribution of AS-Path Length: One way to quantify the impact of an overlay on the AS-level underlay is to characterize the length of AS-paths for individual overlay connections. Figure 5(b) shows the histogram of AS-path length across all AS-paths used by the overlay for each of the 4 snapshots and can be used as a reference. This figure shows that around 40% of the used paths are three AS hops long while 80% of the paths in each overlay are at most 4 AS hops.

Figure 5(c) depicts the histogram of AS-path length across all overlay connections for each overlay snapshots. In essence, this can be viewed as a "weighted" version of Figure 5(b) where each path is weighted by the number of corresponding connections. The figure shows similar patterns across all overlay snapshots despite the change in the number of peers and their connections. The two histograms are very similar but the path length for all connections of the overlay is slightly shorter indicating that a higher fraction of connections are associated with shorter paths.

Identity and Evolution of Transit ASes: To investigate the effect of transit ASes more closely, we examine the identity of the top-10 transit ASes that observe the highest number of crossing overlay connections, and their evolution over time in Figure 6. The top transit ASes are ordered vertically from top to bottom for each overlay snapshot. This figure shows that only 4 transit ASes (including top 3 ASes) remain stable across all four snapshots. However, the changes in the other transit ASes is more chaotic. This is due to the fact that ranking of

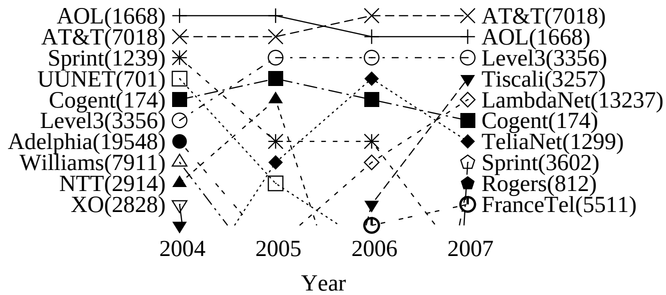


Fig. 6. Identity and evolution of top-10 transit ASes observing the largest volume of traffic

TABLE VIII
PERCENTAGE OF PATHS/CONNECTIONS REACHING EACH TIER OF AS HIERARCHY.

	Tier-1		Tier-2		Tier-3	
Snapshot	Path	Conn	Path	Conn	Path	Conn
G-04	51	84	46	16	2.4	0.0
G-05	59	73	38	27	3.0	0.0
G-06	52	64	38	36	10	0.0
G-07	55	63	41	37	3.6	0.1

transit ASes is affected by a combination of factors including changes in the topology of AS-level underlay, changes in routing policies, and the location of peers.

We have randomly rewired the connections of each overlay to generate a comparable random graph and then calculated the associated AS-paths over the same AS level topology. The histogram of AS-path length in the randomized overlay (not shown) is nearly identical to Figure 5(c) for all overlay snapshots.

Diffusion of Traffic in the AS Hierarchy: The most interesting way to quantify the impact of an overlay topology on the AS-level underlay is to determine the fraction of load that is propagated upward in the AS-level hierarchy towards top-tiered ASes. Table VIII shows the percentage of paths and percentage of overlay connections whose top AS is tier-1, tier-2 and tier-3 in each overlay snapshot. The columns marked "unweighted" show the percentage of the used AS-paths reaching each tier while the columns marked "weighted" represent the percentage of paths weighted by the number of overlay connections using that path, effectively showing the percentage of overlay connections reaching each tier. This table shows that more than half of the paths reach tier-1 ASes and roughly 40% of the paths peak in a tier-2 AS.

The percentage of connections that reach a tier-1 AS is higher than the number of paths reaching tier-1, indicating that a larger fraction of connections are mapped to these paths. The percentage of connections that top in a tier-2 AS is around 16% to 37% which is smaller than the percentage of paths reaching tier-2. Interestingly, the percentage of connections that top in tier-1 ASes decreases over time while the percentage of connections that top in a tier-2 ASes is increasing. These trends suggest the increasing connectivity between ASes in lower tiers which reduces the fraction of connections that have to climb the hierarchy up to tier-1 ASes.

V. Related Work

During the past few years, the ever-increasing portion of Internet traffic resulting from P2P applications has become a concern for commercial ISPs. Due to the symmetric nature and steady pattern of P2P traffic, these ISPs find it costly to accommodate it and try to limit it by simply blocking certain P2P applications or regulating the traffic rate, which in turn can cause other problems.[3]. On a more technical level, a number of studies have aimed at quantifying the impact of P2P applications on the ISPs. Karagiannis *et al.* [18] compare the load on the ISPs for the cases of traditional client-server, P2P, local caching and locality-aware peer-assisted content delivery and concludes that the last type provides the most benefits in terms of both ISP load and the observed performance by the users. They suggest that if the amount of locally available content is large, the applications can make use of the locally available content to reduce the load on the external links of the ISP. Another study by Gummadi *et al.* [19] presents an analysis of the traffic associated with the KazaA P2P file sharing application at the University of Washington campus network. They also show that there is a substantial percentage of requests that can be resolved locally, effectively reducing the traffic on the external link. Both studies are based on packet traces captured locally at the ISP gateways. Although they were successful in inspiring works on localizing the P2P traffic, their results are only valid for specific types of networks and can hardly be extended to other networks of other types and sizes. Furthermore, the P2P traffic pattern between ISPs is not part of these studies.

In response to the ISPs concerns and actions against P2P traffic, several research papers and Internet drafts have been published in the recent years. In the first set of papers, the common approach is designing ISP-friendly P2P applications. Such P2P applications use a variety of techniques to enable each peer to discover other local peers and effectively form an overlay that is "locality-aware" without any explicit assistance from the network layer. A few examples for such *independent locality aware overlays* are [20]–[23] where the first two use landmarks, the third uses DNS domain names and the last one uses LDAP protocol to detect locality among peers.

Subsequent papers in this area propose methods that rely on a cooperation interface between the network layer (ISP) and the application layer (P2P) through which the ISP provides necessary information to the application with the goal of making the overlay load less "costly" for the ISP and also more desirable (based on the applications goal) for the application. Oracle [3], is a server run by the ISP with the underlay information to which the local peers send their list of candidate peers. It is responsible for sorting them according to the ISP's preference and returning them to the requesting peer. In P4P [2], the network information is provided to the application's

[3]In the summer of 2008, the FCC issued a ruling against a major ISP that deployed such practices and asked the ISP to stop them.

local *tracker* (*e.g.*, BitTorrent) which then implements those information in assisting peers with neighbor selection.

To the best of our knowledge no previous work has been done on the *global* impact of P2P (*i.e.*, overlays) on the AS-level underlay. In particular the following questions have not been answered by previous studies: (i) How does the traffic generated by a given P2P application in the overlay impact the AS-level underlay? (ii) What is the pattern of P2P traffic flow among ISPs at different levels of the Internet's hierarchy (tiers)? (iii) How similar (or dissimilar) are the patterns of P2P traffic in different geographical regions? (iv) Which ASes are affected the most by the traffic imposed by an overlay?

VI. Conclusion and Future work

In this paper, we explored the problem of quantifying the load that a particular overlay contributes to the AS level underlay. We identified the challenging components of this problem and described existing techniques to address each component. We presented a methodology for mapping the load of an application-level overlay on the AS-level underlay that incorporates a collection of best existing practices. We applied our methodology to characterize the load of real world P2P overlays on the AS-level underlay. Our study deepens the understanding of interactions between application level overlay and the underlay.

As part of our future work, we plan to investigate how changing the geographical location of peers and their connectivity change the imposed load on the AS-level underlay. Furthermore, we plan to derive and incorporate traffic model for different P2P application into our methodology. Finally, we leverage pricing models that are used by ISPs to determine how structure and workload of an overlay determine revenue of ISPs in the AS hierarchy of the underlay.

References

[1] A. H. Rasti, R. Rejaie, and W. Willinger, "Characterizing the Global Impact of P2P Overlays on the AS-Level Underlay," in *Passive and Active Measurement Conference*, Zurich, Switzerland, Apr. 2010.

[2] H. Xie, Y. R. Yang, A. Krishnamurthy, Y. Liu, and A. Silberschatz, "P4P: Provider Portal for Applications," in *SIGCOMM*, 2008.

[3] V. Aggarwal, A. Feldmann, and C. Schneideler, "Can ISPs and P2P systems co-operate for improved performance?" *CCR*, vol. 37, no. 3, pp. 29–40, Jul. 2007.

[4] D. R. Choffnes and F. E. Bustamante, "Taming the torrent: A practical approach to reducing cross-ISP traffic in P2P systems," in *ACM SIGCOMM*, Aug. 2008.

[5] L. Gao, "On Inferring Autonomous System Relationships in the Internet," *IEEE/ACM Transactions on Networking*, vol. 9, pp. 733–745, 2000.

[6] Z. Ge, D. R. Figueiredo, S. Jaiswal, and L. Gao, "On the Hierarchical Structure of the Logical Internet Graph," in *SPIE ITCom*, Denver, Colorado, USA, Nov. 2001.

[7] U. of Oregon, "RouteViews." [Online]. Available: http://www.routeviews.org/

[8] B. Quoitin and S. Uhlig, "Modeling the Routing of an Autonomous System with C-BGP," *IEEE Network*, vol. 19, no. 6, Nov. 2005.

[9] D. Stutzbach, R. Rejaie, N. Duffield, S. Sen, and W. Willinger, "On Unbiased Sampling for Unstructured Peer-to-Peer Networks," in *IMC*, Rio de Janeiro, Brazil, Oct. 2006.

[10] B. Augustin, X. Cuvellier, B. Orgogozo, F. Viger, T. Friedman, M. Latapy, C. Magnien, and R. Teixeira, "Avoiding traceroute anomalies with Paris traceroute," in *IMC*, 2006.

[11] H. Chang, S. Jamin, and W. Willinger, "Inferring AS-level internet topology from router-level path traces," in *SPIE ITCom*, 2001.

[12] R. Oliveira, D. Pei, W. Willinger, B. Zhang, and L. Zhang, "In search of the elusive ground truth: the internet's as-level connectivity structure," in *SIGMETRICS*, 2008.

[13] M. Roughan, S. J. Tuke, and O. Maennel, "Bigfoot, Sasquatch, the Yeti and other missing links: what we don't know about the AS graph," in *IMC*, Vouliagmeni, Greece, Oct. 2008.

[14] X. Dimitropoulos, D. Krioukov, M. Fomenkov, B. Huffaker, Y. Hyun, K. Claffy, and G. Riley, "AS Relationships: Inference and Validation," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 1, pp. 29–40, 2007.

[15] D. Stutzbach and R. Rejaie, "Capturing Accurate Snapshots of the Gnutella Network," in *Global Internet Symposium*, Miami, FL, Mar. 2005, pp. 127–132.

[16] CAIDA, "Cooperative Association for Internet Data Analysis." [Online]. Available: www.caida.org

[17] W. Muhlbauer, A. Feldmann, O. Maennel, M. Roughan, and S. Uhlig, "Building an AS-topology model that captures route diversity," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4, pp. 195–206, Oct. 2006.

[18] T. Karagiannis, P. Rodriguez, and K. Papagiannaki, "Should Internet Service Providers Fear Peer-Assisted Content Distribution?" in *Internet Measurement Conference*, Berkeley, CA, Oct. 2005, pp. 63–76.

[19] K. P. Gummadi, R. J. Dunn, S. Saroiu, S. D. Gribble, H. M. Levy, and J. Zahorjan, "Measurement, Modeling, and Analysis of a Peer-to-Peer File-Sharing Workload," *ACM SIGOPS Operating Systems Review*, vol. 37, no. 5, pp. 314–329, Dec. 2003.

[20] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, "Topologically-Aware Overlay Construction and Server Selection," in *INFOCOM*, Jun. 2002.

[21] Y. Kim and K. Chon, "Scalable and Topologically-aware Application-layer Multicast," in *Globecom*, Dallas, TX, Nov. 2004.

[22] N. J. Harvey, M. B. Jones, S. Saroiu, M. Theimer, and A. Wolman, "SkipNet: A Scalable Overlay Network with Practical Locality Properties," in *USENIX Symposium on Internet Technologies and Systems*, 2003.

[23] J. Togashi, A. Inomata, K. Fujikawa, and H. Sunahara, "Proposal and Evaluation of an Effective Method for Underlay-Aware Overlay Networks using LDAP," in *Parallel and Distributed Computing and Networks*, 2009.