Taming the Torrent

A Practical Approach to Reducing Cross-ISP Traffic in Peer-to-Peer Systems

David R. Choffnes and Fabián E. Bustamante
Department of Electrical Engineering & Computer Science, Northwestern University
Evanston, IL, USA
{drchoffnes,fabianb}@cs.northwestern.edu

ABSTRACT

Peer-to-peer (P2P) systems, which provide a variety of popular services, such as file sharing, video streaming and voice-over-IP, contribute a significant portion of today's Internet traffic. By building overlay networks that are oblivious to the underlying Internet topology and routing, these systems have become one of the greatest traffic-engineering challenges for Internet Service Providers (ISPs) and the source of costly data traffic flows. In an attempt to reduce these operational costs, ISPs have tried to shape, block or otherwise limit P2P traffic, much to the chagrin of their subscribers, who consistently finds ways to eschew these controls or simply switch providers.

In this paper, we present the design, deployment and evaluation of an approach to reducing this costly cross-ISP traffic without sacrificing system performance. Our approach recycles network views gathered at low cost from content distribution networks to drive biased neighbor selection without any path monitoring or probing. Using results collected from a deployment in BitTorrent with over 120,000 users in nearly 3,000 networks, we show that our lightweight approach significantly reduces cross-ISP traffic and, over 33% of the time, it selects peers along paths that are within a single autonomous system (AS). Further, we find that our system locates peers along paths that have two orders of magnitude lower latency and 30% lower loss rates than those picked at random, and that these high-quality paths can lead to significant improvements in transfer rates. In challenged settings where peers are overloaded in terms of available bandwidth, our approach provides 31% average download-rate improvement; in environments with large available bandwidth, it increases download rates by 207% on average (and improves median rates by 883%).

Categories and Subject Descriptors

C.2.4 [**Distributed Systems**]: Distributed Applications
C.2.3 **Network Operations** Network management

General Terms

Algorithms, Measurement, Performance, Experimentation, Management

SIGCOMM'08, August 17–22, 2008, Seattle, Washington, USA.

Keywords

P2P, ISP, cross-ISP traffic, peer selection, measurement reuse

1. INTRODUCTION

Peer-to-peer (P2P) systems use decentralization to enable a wide range of important, scalable and reliable services such as data sharing, voice-over-IP (VoIP) and video streaming. These systems are so prevalent that reports indicate they generate as much as 70% of Internet traffic worldwide [14]. Their overwhelming popularity has yielded significant revenues for Internet Service Providers (ISPs), as users upgrade to broadband for improved P2P performance [21].

Current P2P implementations, however, are oblivious to the underlying Internet topology and ISP link costs. By making peering decisions independently of these factors, P2P systems have significantly increased ISPs' operational costs, particularly in terms of cross-ISP traffic. This has driven service providers to the unfavorable solution of forcefully reducing a user's P2P traffic at the expense of unhappy subscribers and the risk of government investigations [11].

The effectiveness of ISPs' efforts at shaping, blocking or otherwise limiting P2P traffic is questionable. For example, when early P2P systems ran over a fixed range of ports (e.g., 6881-9 for BitTorrent) ISPs attempted to shape traffic directed toward those ports. In response, P2P systems have switched to non-standard ports, often selected at random. More advanced strategies such as the use of deep packet inspection to identify and shape P2P-specific flows have resulted in peer clients encrypting their connections. Recently, some ISPs have attempted to reduce P2P traffic by placing caches at the ISP's gateway to the Internet or by using network appliances (e.g., Sandvine [28]) for spoofing TCP RST messages, which trick clients into closing connections to remote peers [28, 31]. The legality of these approaches is questionable. By caching content, ISPs may become participants in illegal distribution of copyrighted material, while interfering with P2P flows in a non-transparent way may not only break the law but also lead to significant backlash [11]. In summary, history has offered no sustainable solution that relies exclusively on ISP controls for addressing traffic-engineering problems posed by P2P systems.

Two recent simulation-based studies have suggested an alternative solution in which ISPs and P2P users cooperate to reduce cross-ISP traffic. In particular, if a P2P client biases its connections to peers in the same ISP, the peer could receive near-optimal performance while significantly reducing the number of times the same data item enters the ISP [3, 8]. Both studies discuss an approach that requires an *oracle* to provide knowledge about which peers are in the same ISP. Whereas Bindal et al. [8] do not focus on any particular oracle implementation, Aggarwal et al. [3] suggest that ISPs themselves could provide such a service and demonstrate through simulation that this is an effective solution. While the basic idea of an oracle to solve the P2P conundrum is appealing, tasking ISPs with the job requires P2P users and ISPs to cooperate and to trust each other, neither of which is likely to occur.

In this paper, we propose an alternative, scalable technique to provide biased peer selection that requires no cooperation or trust between ISPs and their subscribers, no additional infrastructure and no network topology information. This technique is based on the observation that the information necessary for peer selection is already being collected by content distribution networks (CDNs). CDNs use dynamic DNS redirection to send clients to low-latency replica servers located in thousands of ISPs worldwide. We posit that if two clients are sent to a similar set of replica servers, they are likely to be close to these servers and, more importantly, to each other. In this paper we prove this to be the case, demonstrating that these CDN-based "hints" can inform a biased peer selection algorithm that significantly reduces cross-ISP traffic.

To experiment with our approach, we made it available as an extension to the Azureus BitTorrent client beginning in April 2007. We chose BitTorrent for its popularity, as it has been reported to account for over 66% of the P2P user population [14]. Today, our extension has been installed by over 120,000 subscriber peers distributed worldwide. With their help, we have performed extensive, continuous measurements, currently recording data for connections between more than 2.5 million peer IP addresses per day. In particular, we collect DNS redirection information, transfer rates, path latencies and traceroute measurements.

We use this data to show that our approach scales easily to well over one hundred thousand users, and effectively "tames" BitTorrent by significantly reducing cross-ISP traffic without sacrificing its performance or robustness. Our analysis indicates that *over 33% of the time our approach to biased selection recommends peers along paths that are within a single autonomous system (AS) and the median number of AS hops to all recommended peers is 1*. Further, we find that our system locates peers along paths that have *two orders of magnitude lower latency* and *30% lower loss rates* than those picked at random, and that these high-quality paths can lead to significant improvements in transfer rates. In challenged settings where peers are overloaded in terms of available bandwidth, our approach provides *31% average download-rate improvement*; in environments with large available bandwidth, *it increases download rates by 207% on average (and improves median rates by 883%)*. Though our data is specific to BitTorrent, we believe the results are general enough to be extended to other P2P systems.

This work provides the following significant contributions:

- A description of our scalable biased peer selection approach that recycles network views gathered, at low cost, from CDNs. Our technique does not require new infrastructure nor depend on cooperation between ISPs and their subscribers. To benefit from our approach, a peer *only* needs to have the ability to perform local DNS queries for CDN names.

- An implementation of our approach that has been deployed on over 120,000 end user systems located in more than 100 countries and over 2,800 networks.

- Detailed measurements of biased peer selection in BitTorrent comprising more than 100 million peer IPs over nine months. Our users cumulatively connect to 2.5 million peer IP addresses per day, and have reported traceroute measurements that cross over 9,700 different networks. The raw data from our ongoing measurements will be made publicly available.

- An analysis of our measurements demonstrating that CDNs are effective, low-cost oracles to help minimize P2P cross-ISP traffic. Our technique locates peers along paths that do not leave the AS of origin over one third of the time, over an order of magnitude more often than peers picked at random by BitTorrent. Our study is also the first to characterize cross-ISP traffic and per-connection bidirectional transfer rates from the perspective of a large number of BitTorrent clients.

The rest of the paper is organized as follows. In Section 2 we provide background information relevant to our work before presenting a high-level view of our approach in Section 3. Section 4 discusses how we use CDNs' network views for biased peer selection, and Section 5 describes our implementation. We present a detailed empirical analysis of the effectiveness of our approach in Section 6, discuss several key issues in Section 7, then conclude in Section 8.

2. BACKGROUND

Most P2P systems employ an arbitrary peer selection policy that ignores the underlying Internet topology and ISP link costs, establishing connections between randomly chosen subsets of cooperating peers from around the world. Such a policy results in P2P traffic that often crosses network boundaries multiple times to reach content that could have been more speedily obtained from nearby peers [2, 8, 17]. Several proposals have suggested using AS numbers in peer selection (e.g., [18, 23]) to improve performance and reduce cross-network traffic, and this approach has even been adopted by several P2P applications (e.g., Neokast and Joost). While this simple technique helps reduce cross-ISP transfers, it can both unnecessarily restrict in-network, cross-AS traffic while biasing peer connections toward distant hosts in an AS with broad geographic coverage.

Among P2P systems, BitTorrent [9] is one of the most popular [14]. BitTorrent's popularity is due to a number of factors, the most notable being the fundamental advantages of P2P over traditional approaches to content distribution, including self-scaling and resilience, and its relatively high performance in terms of user-perceived download time. The protocol has been well documented in the literature (e.g., in [13, 15, 25, 26]), thus our brief description focuses only on aspects relevant to this work.

To distribute a file using BitTorrent, a peer exchanges pieces of it with other peers that are concurrently transferring the same file. The file is described by metadata called a *torrent*; those peers sharing content described by the same metadata are said to be *connected to the same torrent* and in the same *swarm*. To locate these peers, the protocol uses *trackers* that provide each peer with a random subset of peers connected to the torrent. By default, each peer initially establishes, at random, a finite number of connections from this subset. As the transfer progresses, connections that do not benefit the transfer are dropped (i.e., *choked*) and new random connections are established (i.e., *unchoked*).

ISPs have resorted to a number of methods to control P2P traffic, in particular BitTorrent, ranging from bandwidth limiting to traffic shaping and caching. The questionable effectiveness of this one-sided approach has partially motivated some collaborative models for addressing the problem. Aggarwal et al. [3] and Bindal et al. [8] have recently suggested the idea of ISP-supported *oracles* for biased-peer selection. Rather than recommending peers for performance improvement [1, 12, 20, 30], these oracles would bias peer selection toward nodes in the same ISP to reduce service

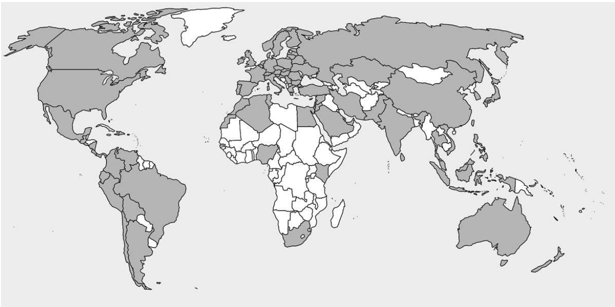


Figure 1: Map indicating countries containing users of our extension in dark gray. As of January 2008, the extension has been installed in 108 different countries.

providers' costs without affecting peers' performance. Similar in spirit to this work, the P4P [37] project attempts to address the problem through custom trackers, both for ISPs and P2P systems, using an interface based on a primal-dual decomposition of an optimization problem. This interface design simplifies the realization of traffic-engineering objectives from each parties' perspective and ensures the extensibility of the approach. Through simulation-based studies and limited experimental deployments, these collaborative approaches have been shown to effectively reduce network costs while minimally impacting application performance. A clear advantage of these proposals is that they allow ISPs to incorporate aggregated traffic policies in their tracker recommendations (e.g., a particular traffic balance ratio between peering providers). However, all of them require deployments of oracles for each participating ISP and their effectiveness is ultimately predicated on their adoption by P2P applications and a trust relationship between P2P users and their ISPs.

3. APPROACH

This paper presents a scalable approach to biased peer selection and reports on experimental results that validate its effectiveness for a popular P2P system in the wild. A key feature of our technique is that it recycles network views gathered, at low cost, from CDNs to drive biased peer selection without any path monitoring or probing.

Following from the observation that CDN redirections are driven primarily by latency [33], we base our approach on the hypothesis that if two peers exhibit similar redirection behavior, they are likely to be close to one another. Further, we expect that these peers will be mostly within the same ISP, thus avoiding cross-ISP traffic and optimizing clients' performance by avoiding most network bottlenecks [7].

Unlike previous oracle-based proposals [3, 8], our CDN-based approach does not require new infrastructure and does not depend on cooperation between ISPs and their subscribers. This work is a concrete example of the use of "recycled" information, gathered by long-running services such as CDNs, in building more efficient services — one instance of a negative feedback loop essential to Internet scalability.

To validate our approach, we made freely available an implementation of CDN-based peer selection as an extension to the popular Azureus BitTorrent client beginning in April 2007. As of January, 2008, unique users of our extension number over $120,000$, and are located in more than 100 countries (Fig. 1), with several thousand online concurrently at any point during the course of our study. In addition to implementing our scalable biased peer selection technique, the software performs network measurements and records information about file-transfer performance. We leave a detailed discussion of our implementation to Section 5.

Countries	108
IP addresses	300,000
Prefixes	15,000
ASes	2,800
Daily peers observed	$>$ 2.3 million
Daily traceroutes	1.2 million $\pm$ 200K
Total ASes traversed	$>$ 9,700

Table 1: Summary of sources for our dataset.

Table 1 presents a summary of key statistics regarding peers running our software. These peers allow us to record about $1,000,000,000$ transfer-rate samples every two weeks, and over $22,000,000$ ICMP ping measurements per day. Our dataset currently contains connection information for over $100,000,000$ BitTorrent peers running over 100 different client applications and located in well over 10,000 ASes.

4. CDNS AS ORACLES

In this section, we discuss how CDNs' network views can be recycled to drive a biased peer-selection service for P2P applications. We begin by providing a brief review of how CDNs work.

CDNs attempt to improve web performance by delivering content to end users from multiple, geographically dispersed servers located at the edge of the network [4, 19, 22]. Content providers contract with CDNs to host and distribute their content. Since most CDNs have servers in ISP points of presence, clients' requests can be dynamically forwarded, via DNS redirections or URL rewriting, to topologically proximate replicas [16, 29].

Beyond static information such as geographic location and network connectivity, CDNs rely on network measurement systems to incorporate dynamic network information in replica selection and determine high-speed Internet paths over which to transfer content within the network [5]. In previous work [33], we reported on a broad measurement study of the Akamai CDN and demonstrated that their redirections are performed frequently enough as to be useful for control, that these updates are primarily driven by network conditions and are, therefore, potentially beneficial to other applications. This work also showed that redirections for a large-scale CDN are primarily driven by latency, i.e., most of the replica servers are along low-latency paths to end hosts.

Based on these results, we hypothesize that when different hosts exhibit similar redirection behavior, they are likely close to the corresponding replica servers and, by transition, to each other. In [32], we define a way to encode redirection behavior and propose its use as the basis for relative network positioning [32]. In this work, we extend previous efforts in this area by developing a way to scalably encode and distribute redirection information. We further use this technique for biased peer selection to reduce cross-ISP traffic, and evaluate the benefit of this approach "in the wild."

We represent peer-observed DNS redirection behavior using a map of ratios, where each ratio represents the frequency with which the peer has been directed toward the corresponding replica server during the past time window. Specifically, if peer P_a is redirected toward replica server r_1 75% of the time and toward replica server r_2 25% of the time, then the corresponding ratio map is:

$$\mu_a = \langle r_1 \Rightarrow 0.75, r_2 \Rightarrow 0.25 \rangle$$

More generally, the ratio map for a peer a is a set of *(replica-*

server, ratio) tuples represented as

$$\mu_a = \langle (r_k, f_k), (r_l, f_l), ..., (r_m, f_m) \rangle$$

Note that each peer's ratio map contains only as many entries as replica servers seen by that peer (in practice, the average number of entries is 1.6 and the maximum is 31), and that the sum of the f_i's in any given ratio map equals one. For brevity, we will use $\mu_{a,i}$ to represent the ratio of time f_i that peer a is redirected to replica server r_i.

In the context of our biased peer selection service, if two peers have the same ratio map values, then the path between them should cross a small number of networks (possibly zero). Similarly, if two peers have completely different redirection behavior, it is likely that the path between them crosses a relatively large number of networks. More generally, we would like a metric that, given two peers, produces a continuum of values describing the similarity between the peers' redirection behaviors. Based on our formulation of ratio maps, each peer in a P2P network can be represented as a vertex in a general graph connected by edges labeled with the degree of overlap in their redirection frequency maps. Following from the premise that CDN redirections are primarily driven by latency, the structure of this graph can be used to locate nearby peers based on the *cosine similarity* of their ratio maps. Cosine similarity [27] is a mathematical measure of how similar two vectors are, yielding values on a scale of $[0, 1]$. Treating a redirection map as a vector and given two hosts a and b, this can be formally defined as:

$$cos\_sim(a, b) = \frac{\sum_{i \in I_a} (\mu_{a,i} \cdot \mu_{b,i})}{\sqrt{\sum_{i \in I_a} \mu_{a,i}^2 \cdot \sum_{i \in I_b} \mu_{b,i}^2}}$$

Where I_a represents the set of replica servers to which peer a has been redirected over the time window. Intuitively, the cosine similarity metric is analogous to taking the dot product of two vectors and normalizing the result. When the maps are identical, their resulting cosine similarity value is 1; when they are orthogonal (i.e., have no replica servers in common), the value is 0. Thus, to determine whether two peers a and b are likely to be in the same ISP, we can simply compute the cosine similarity of their redirection maps. If the value is greater than a certain threshold (currently 0.15 in our implementation), we recommend these peers as candidates for reducing cross-ISP traffic.

5. CDN-BASED ORACLES IN BITTORRENT

In this section, we describe key details of our CDN-based oracle implementation for BitTorrent. After a brief overview, we discuss how to support information from multiple CDNs and how our implementation efficiently locates nearby peers with low overhead.

5.1 Overview

Our implementation, named Ono, is written in Java and designed as a plugin (i.e., extension) for compatibility with the Azureus BitTorrent client. We chose Azureus because it is one of the most popular BitTorrent clients, provides cross-platform compatibility and features a powerful API for dynamically adding new functionality via plugins. Ono contains approximately 12,000 method lines of code, 3,500 of which are for the GUI and 3,000 for data collection and reporting (and thus not essential for Ono functionality). It is publicly available with source code at `http://azureus.sourceforge.net/plugin_list.php` or it can be downloaded and installed from inside the Azureus client.

The Ono plugin uses a built-in Java-based DNS client [35] to perform periodic DNS lookups on popular CDN names, which it uses to maintain ratio maps. To determine the cosine similarity value for a peer, Ono must be able to compare its ratio maps with those of other peers. The latter information can be obtained in a number of ways: through direct exchange between peers, from distributed storage and from trackers. Ono currently supports the first two options. With direct exchange, when two peers running the Ono plugin perform their connection handshake, the peers swap ratio maps directly. For the distributed storage solution, we implemented an efficient DHT-based approach for storing and retrieving ratio maps.[1] The Azureus built-in DHT, however, was not sufficiently reliable to support the functionality efficiently, so the DHT option is disabled by default.

Though Ono enjoys a large user base, it is still a small fraction of the total BitTorrent population. Thus Ono also attempts to perform DNS lookups on behalf of other peers that it encounters, to determine their ratio maps. This enables Ono to perform biased peer selection over a much larger set of peers, including those not running the Azureus client. From both direct exchange of ratio maps and DNS lookups, our Ono clients locate over $180,000$ peers per day using our CDN-based approach.

When Ono determines that a peer has similar redirection behavior, it attempts to bias traffic toward that peer by ensuring there is always a connection to it, which minimizes the time that the peer is choked. Due to limitations of the Azureus plugin API, we are currently unable to bias other aspects of peer connections, e.g., the bandwidth allocated to each connection.

Should Ono become universally adopted, we must ensure that our service does not significantly alter the appealing robustness that comes from the diversity of peers resulting from BitTorrent's random selection. To that end, Ono will bias traffic to only a fraction of the total connections established for a particular torrent.

5.2 Using CDN Names

To perform biased peer selection, Ono must maintain a ratio map for each CDN name being used for DNS lookups. As we previously showed, using different CDNs and even different names for the same CDN can lead to different results in terms of redirection behavior [33]. To study the impact of this property on Ono's recommendations, we collected ratio maps for six names spanning two large CDNs (Akamai and Limelight) as described in Table 2. In case CDN behavior changes for these names in ways that are not useful for peer selection, Ono checks a list of CDN names that we can update at any time.

Ono performs DNS lookups for each CDN name to determine redirection behavior and encodes this information as ratio maps. Because CDNs tend to cluster their servers into groups assigned to the same class-C subnet, we build ratio maps using replica-server clusters consisting of /24 addresses instead of full IP addresses.

After performing a DNS lookup for a CDN name, we adjust the corresponding ratio map to reflect the redirection frequencies. We begin by aging the existing ratios in the map using exponential decay. The decay rate is set so that a replica server not seen during the past 24 hours will be removed from the ratio map. Finally, we increase the value of the ratio for the replica server cluster returned by the DNS lookup such that the sum of all ratios is 1.

[1]The approach exploits the key-value structure of ratio maps to store information in a DHT using a small number of operations that depends on number of entries in a ratio map (typically less than 10). Locating each nearby peer requires at most two DHT lookup operations. A detailed explanation of the approach is beyond the scope of this paper.

Abbr.	DNS name	CDN	Description
AA	e100.g.akamaiedge.net	Akamai	Air Asia (Southeast Pacific)
CN	a1921.g.akamai.net	Akamai	CNN.com (US news site)
LM	a245.g.akamai.net	Akamai	LeMonde.com (French news site)
FN	a20.g.akamai.net	Akamai	Fox News (US news site)
AB	wdig.vo.llnwd.net	Limelight	ABC Streaming Video (US television network)
PW	a1756.g.akamai.net	Akamai	Popular Web Site

Table 2: CDN names used in this study.

As we previously demonstrated [33], redirection behavior changes over time scales that vary according to the user's network location. Thus, to enable adaptation to this behavior with minimum additional load on CDN name servers, we adopt the following approach. When a client has no redirection information, it performs a DNS lookup to each CDN name at most once every thirty seconds, for two minutes, to establish a basis ratio map. After this bootstrapping phase, the interval between DNS lookups increases by one minute if the redirection information for the current CDN name does not differ from the previous lookup. If the redirection behavior changes between intervals, the redirection interval is halved (to a minimum value of 30 seconds). At the end of a BitTorrent session, Ono caches the ratio maps on persistent storage. This allows Ono to avoid the bootstrapping phase if the cached ratio maps are sufficiently fresh (i.e., less than 24 hours old by default).

CDN mapping information is not particularly useful for neighbor selection if the latency along the path between a peer and its associated replica server is large. To determine the set of useful CDN names dynamically, Ono performs ICMP pings to the replica servers returned by the DNS lookup. The smaller the RTT value, the higher priority the associated name is given with respect to biasing peer selection. Further, if the RTT to one replica server is significantly higher ($>$ 50% larger) than the average, we exclude that mapping from our ratio maps. Note that ICMP pings are not required for our approach to work — if no RTT information is available, all names are used with equal weight in peer selection. As we show in Section 6, the majority of the names that Ono currently uses offer equally high performance, so filtering is typically not required.

5.3 Efficiency

Ono's overhead is extremely small: determining each peer's proximity requires network operations that scale *independently of the number of peers in the network*. In particular, maintaining ratio maps requires periodic DNS lookups, the cost of which depends on the lookup frequency, the name being used and the number of responses. Generally, the name translation request is less than 50 bytes and the response is less than 100 bytes. Using the minimum lookup interval of 30 seconds to calculate the maximum bandwidth consumed by a Ono peer, the overhead is 18 KB upstream and 36 KB downstream *per day*. For comparison, the average peak download rates from our clients is 80 KB *per second* and average peak upload rates are 40 KB/s. Thus, even when using multiple CDN names and performing DNS lookups to obtain ratio map information for peers not running Ono, the overhead is sufficiently small as to be insignificant. Ono also consumes some bandwidth when peers exchange ratio maps, but this requires at most hundreds of bytes per peer and in practice is performed much less frequently than DNS lookups.

Likewise, the computational overhead for comparing ratio maps is negligible. Since the number of replica server clusters maintained in each ratio map is generally less than 10, computing the cosine similarity between two maps requires at most hundreds of floating-point operations. Considering that the result of the computation is valid until either ratio map significantly changes, the operation is performed infrequently, e.g., on the order of tens of minutes or more.

Finally, we note that the time required to locate nearby peers is extremely small compared to the time to completion for the vast majority of torrents. In particular, when both peers establishing a connection use Ono, the exchange of ratio maps and the determination of cosine similarity is essentially immediate. In the case where one peer is not running our service, Ono is limited only by the time required to perform DNS lookups on that peer's behalf to determine its ratio maps.

6. EMPIRICAL RESULTS

In this section, we evaluate the effectiveness of CDN-based oracles for biased peer selection in BitTorrent. One of our primary goals is to demonstrate that our approach indeed locates peers along paths that significantly reduce cross-ISP traffic compared to default BitTorrent behavior. To that end, we analyze extensive data regarding paths between BitTorrent peers in the wide area. We also show that our approach is practical in the sense that it does not sacrifice transfer performance as seen by BitTorrent clients. In fact, we demonstrate that with appropriate ISPs' bandwidth allocation policies, transfer performance using Ono can yield nearly one order of magnitude improvement in median download rates.

6.1 Dataset

We first discuss our methods for collecting data in this study. While observing downloads, our software samples transfer rates for each connection once every 5 seconds. For every connection, it continuously measures the round-trip-time latency between endpoints using the average of three ICMP pings. By default, at most 10 ping measurements are issued concurrently, though users can change the limit for performance reasons. The endpoints are selected in the same order as the corresponding peers' connections, essentially forming a random total order.

We also perform traceroute measurements to peers. Given the comparatively longer duration of traceroute invocations, we perform fewer traceroute measurements than pings. A measurement is performed for each endpoint at most twice: once upon connection establishment and once upon termination only if the duration of the connection is longer than five minutes. There is at most one traceroute issued at a time.

Traceroutes provide router-level views of paths between hosts. However, an ISP may contain many routers, so we wish to analyze the traceroute measurements using metrics that more closely correspond to ISP hops. Because the Internet is divided into separate administration domains in the form of autonomous systems (ASes), we expect that AS-level path information will provide better insight regarding cross-ISP links. Although there is no one-to-one relationship between ASes and ISPs, the number of AS-hops along a path gives us an upper-bound estimate on the number of cross-ISP hops. We generate AS-level path information from our traceroute data using the AS mappings provided by the Team Cymru group [34].

Our software performs periodic DNS lookups on CDN names using a built-in Java-based DNS client. The results of the lookups are recorded individually and cumulatively, every 30 minutes, in the form of ratio maps.

Each data item is labeled with a timestamp corresponding to the GMT time of the user's local clock. When our software is loaded by a peer, it reports the GMT time of the peer's local clock and the server records that value along with the server's current local time. This allows us to synchronize all of our users' clocks to within a small number of seconds.

In our analysis, we compare statistics from peers located by Ono (referred to as *Ono-recommended peers*) to those from *all* peers selected at random by the BitTorrent protocol, *which also includes those located by Ono*. To facilitate this process, our plugin reports whether a measured peer was recommended by Ono. However, because users can terminate data reporting at any time by force-closing their client, it is not sufficient to rely on this information alone. Instead, we use ratio-map information collected over the course of a six-hour interval to determine which peers were recommended by Ono, then examine statistics inside this interval accordingly. We chose this interval because ratio maps are relatively stable at this time scale but tend to change significantly over larger ones. Thus, each point in the following figures represents the average of the statistics recorded from one Ono peer during a six-hour interval.

The following statistics were generated from data collected between December 1, 2007 and December 16, 2007. This dataset contains approximately 960,000,000 download samples, 19,000,000 traceroute measurements and well over 350,000,000 ping measurements.

6.2 Reducing Cross-ISP Traffic

We now take up the central question of whether our CDN-based oracle can significantly reduce cross-ISP traffic. We answer this question by comparing path characteristics for Ono-recommended peers to those found through the random selection algorithm employed by BitTorrent. As described in the previous section, we performed traceroute measurements to all peers identified by Ono and to a random portion of the remaining peers found by BitTorrent. We start by presenting cumulative results when using the CDN name for LeMonde.com, the online version of a popular French newspaper. We will provide a comparison among all CDN names at the end of the section.

Figure 2 presents a cumulative distribution function (CDF) of the number of IP hops taken along paths between Ono clients and their peers. *Each value on a curve represents the average number of hops for all peers, either Ono-recommended or picked at random by BitTorrent, seen by a particular Ono client during a six-hour interval.* It is immediately clear that peers found by Ono are along shorter paths in terms of IP hops; e.g., the median number of IP hops to Ono-recommended peers is 6 whereas the median number of IP hops to peers selected by BitTorrent is nearly 14 — more than twice as large. Further, over 20% of Ono-recommended peers are only one hop away from our Ono clients; less than 2% of those picked at random are the same. Finally, we note that the quantization evident in the Ono curve is not from a dearth of data points – in fact, there were over 5,100 such points. Rather, because each Ono client sees a relatively small number of Ono-recommended peers during an observation interval and because those peers are typically along short paths, the average of those path lengths takes on a small range of values, often integers.

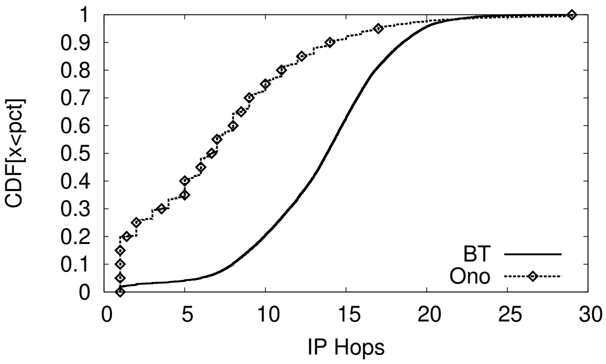


Figure 2: CDF of average number of IP hops to reach Ono-recommended peers and those from unbiased BitTorrent. The median number of IP hops to Ono-recommended peers is less than half the same value for peers picked at random by BitTorrent.

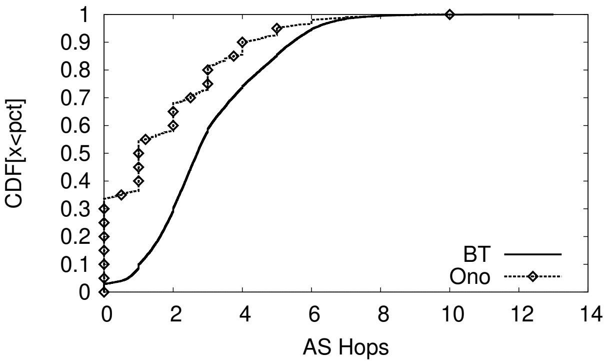


Figure 3: CDF of average number of AS hops to reach Ono-recommended peers and those from unbiased BitTorrent. Over 33% of paths to Ono-recommended peers do not leave the AS of origin and the median number of ASes crossed by paths to Ono-recommended peers is half of those picked at random by BitTorrent.

Of course, IP hop counts (greater than 1) do not necessarily tell us whether traffic crosses ISPs. To better estimate the number of ISPs crossed by a particular path, we mapped each IP address in a traceroute measurement to its corresponding AS number.

Figure 3 presents a CDF of the number of AS hops taken along paths between Ono clients and their peers. Similar to the previous graph, each value on the curve represents the average number of hops for all peers, either located by Ono or picked at random by BitTorrent, seen by a particular Ono client during a six-hour interval. The most striking property is that over 33% of the paths found by Ono do not leave the AS of origin. Further, the median number of AS hops along a path found by Ono is one, whereas this is the case for less than 10% of the paths found by BitTorrent at random. Thus, Ono significantly reduces the overall amount of cross-ISP traffic, thereby promoting "good Internet citizen" behavior that benefits not only the origin ISP but also nearby networks.

Finally, we note that the percent of observed intra-AS paths (average path length less than 1) found by unbiased BitTorrent

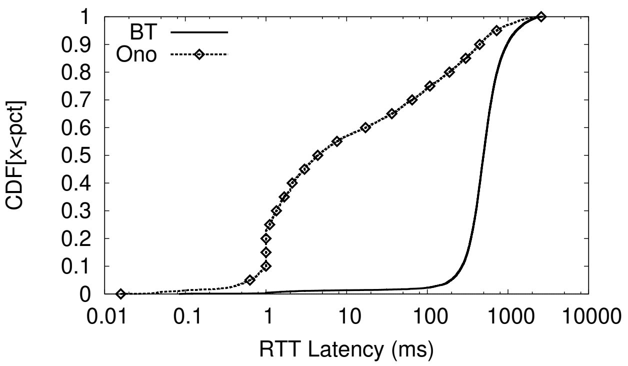


Figure 4: CDF of average ICMP ping round-trip-time latency to Ono-recommended peers and those from unbiased BitTorrent peers. The median latency to Ono-recommended peers is over two orders of magnitude smaller than that to peers picked at random by BitTorrent.

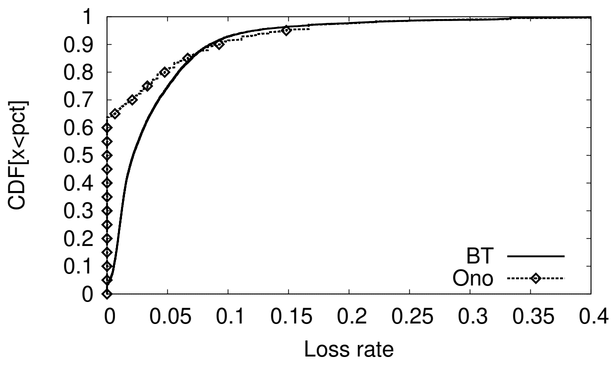


Figure 5: CDF of instantaneous loss rates for traceroutes to Ono-recommended peers and those from unbiased BitTorrent peers. On average, Ono-recommended peers exhibit 30% lower loss rates.

(8.2%) is significantly lower than those reported from simulations of P2P environments that use synthetic network graphs (14.6% [3]).

6.3 Path Characteristics

Now that we have shown that CDN-based oracles for biased peer selection significantly reduce cross-ISP traffic, we take up the question of how this biased selection impacts performance. We first evaluate the end-to-end latency between Ono clients and their peers. The CDF in Fig. 4 presents the average RTT, in milliseconds, for ICMP pings to peers found by Ono and those found at random via BitTorrent. Note the log scale of the x-axis. The figure clearly shows that peers found by Ono are extremely close to each other in terms of latency. For instance, the median latency to Ono-recommended peers is 6 ms whereas the same for peers picked at random is 530 ms – a difference of *two orders of magnitude*!

Another notable feature of the graph is the large number of values where ping latency was exactly 1 ms. The reason is that Windows returns only integer values for ping latencies and returns "$< 1 ms$" for those values less than one. For all such cases, we round the value of the RTT up to 1 ms. The integer property also explains the small step at RTT values of 2 ms.

Another important path characteristic that determines transfer performance is packet loss. In Fig. 5, we use our traceroute measurements to estimate packet loss rates along paths between peers. The graph clearly shows that paths to Ono-recommended peers have lower loss rates than those to peers selected at random by unbiased BitTorrent. On average, paths to Ono-recommended peers exhibit nearly 31% lower loss rates and their median loss rate is 0, whereas the median loss rate for paths to unbiased peers is 2.1%.

6.4 Transfer Performance

Based on the latency and packet loss data in the previous section, we expect that transfer performance from Ono-recommended peers should be higher than, or at least on par with, those picked at random from BitTorrent.

Figures 6(a) and 6(b) present CDFs of the average download and upload rates for biased and unbiased connections on a semilog scale. For this and the following figures, we use all transfer rate samples where the connection was able to sustain a 4 KB/s transfer rate at least once. Connections with lower rates tend to be choked and do not contribute meaningfully to this analysis.

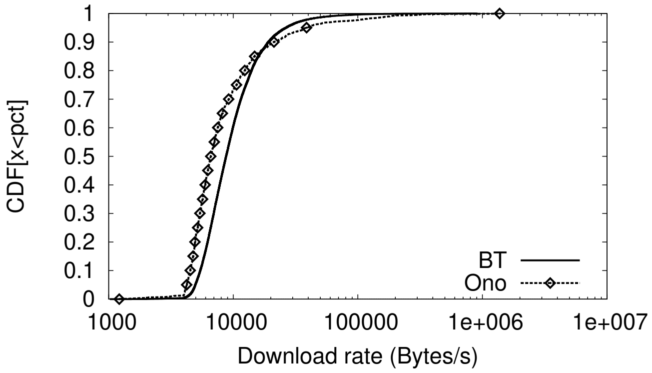
We begin by observing that peers recommended by Ono provide significantly higher peak download rates than those picked at random. In fact, this distribution features a heavy tail—although the median download rate from Ono- recommended peers is slightly lower than those picked at random by BitTorrent, the *average* download rate for Ono is 31% higher than that of unbiased BitTorrent. This seems to indicate that the relatively high quality of paths recommended by Ono also results in higher peak throughput when there is sufficient available bandwidth.

Despite the fact that Ono reduces cross-ISP traffic by proactively reconnecting to nearby peers regardless of available bandwidth, the difference between median transfer rates for Ono and unbiased BitTorrent is only 2 KB/s. Even when Ono-recommended peers do not provide higher median throughput than those picked at random, our approach does not noticeably affect time to completion for downloads. This holds because Ono-recommended peers are only a fraction of the entire set of peers connected to each client and BitTorrent generally saturates a peer's available bandwidth with the remaining connections.

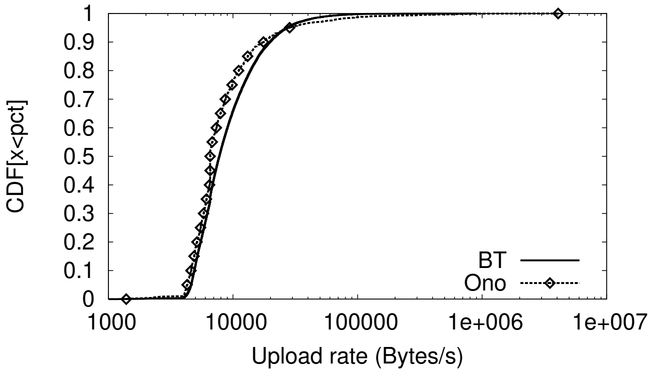
Initially, we expected higher median performance for Ono-recommended peers, given the low latencies and packet loss along paths to them. Based on the relatively low average per-connection transfer rates in both curves (around 10KB/s), we posit that performance gains for Ono-recommended peers are limited because BitTorrent peers are generally overloaded. By splitting each peer's bandwidth over a large number of peers, the BitTorrent system achieves high global transfer rates while generally providing relatively low individual transfer rates to each connection. In this case, the bottleneck for BitTorrent clients is the access link to the ISP (as opposed to the cross-ISP link) [7], so BitTorrent sees no significant performance difference between peers along paths with or without cross-ISP links. We now show that this feature is not universal; rather, it depends on ISPs' bandwidth-allocation policies.

Figure 7 shows a CDF of download rates from Ono clients located in the RDSNET ISP,[2] in Romania. The ISP is notable for offering 50Mb/s unrestricted transfer rates over fiber for *in-network traffic* (i.e., traffic inside the ISP) and 4Mb/s to connections outside the ISP, effectively pushing the bandwidth bottleneck to the edge of the network. The figure clearly shows that Ono thrives in this environment, significantly improving the download rates of Ono-

[2]http://www.rdslink.ro



(a) Download rates.



(b) Upload rates.

Figure 6: CDFs of average transfer rates from Ono-recommended peers and those from unbiased BitTorrent peers, on a semilog scale. The average download rate for Ono is 31% better than unbiased BitTorrent, and the difference in median download rates is only $\approx$ 2 KB/s. The average upload rate for Ono is 42% better than unbiased BitTorrent, and the difference in median rates is only $\approx$ 1 KB/s.

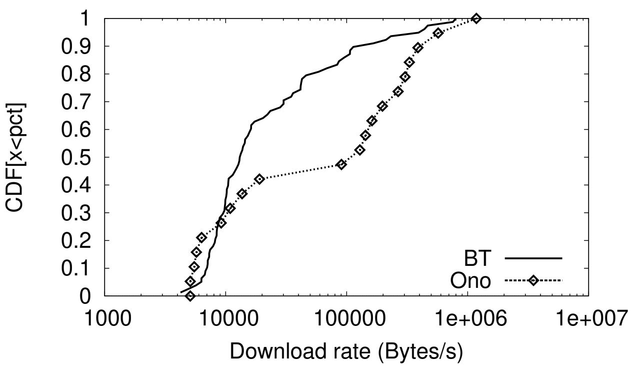


Figure 7: CDF of average download rate for an ISP that provides higher bandwidth to in-network traffic. Ono thrives in this environment.

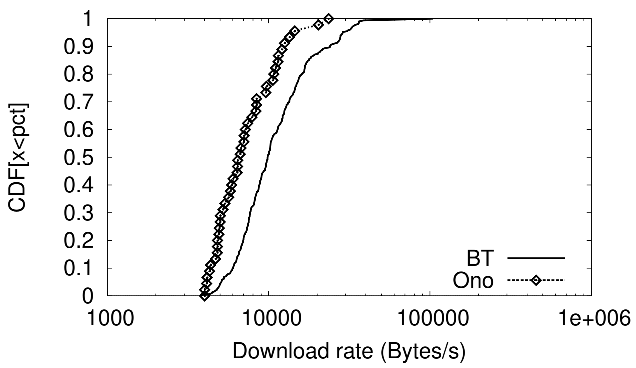


Figure 8: CDF of average download rate for an ISP with a uniform bandwidth allocation policy, which significantly constrains Ono performance.

recommended peers, by comparison with that of randomly selected nodes. In particular, we see the average download rate for Ono-recommended peers improves by 207% and their median download rate is higher by 883%.

To compare against an ISP with uniform (and relatively low) bandwidth constraints, Fig. 8 shows a CDF of download performance for Easynet,[3] an ISP located in the UK. This ISP offers 4 or 8 Mb/s downstream with only 768 Kb/s upstream. As the figure clearly shows, any performance gains that could be attained by Ono in terms of transfer rates are negated by the suboptimal bandwidth allocation. Further, we believe that the higher median performance seen by default BitTorrent peer selection comes from the ability to find peers in other networks that are less constrained by upload bandwidth allocation and therefore provide higher throughput.

Finally, we demonstrate that the bandwidth allocation model in the RDSNET ISP, when coupled with Ono, provides a mutually beneficial environment in which BitTorrent users see higher transfer performance while reducing the cost for ISPs in terms of cross-ISP traffic. The bar graph in Fig. 9 illustrates this by contrasting the AS hop count for the two example ISPs. The x-axis in this graph represents the number of AS hops along paths between peers and the y-axis represents the average of the download rates between these peers. It is clear that RDSNET, which offers higher transfer rates inside the ISP, allows users to obtain significant performance gains by reducing cross-ISP traffic. On the other hand, Easynet, which does not offer different transfer rates for in-network traffic, exhibits negligible performance differences for connections with different AS-path lengths. Consequently, performance from Ono-recommended peers will not be significantly different than those picked at random.

These results make the case for a new ISP-based approach to the problem of taming BitTorrent that is compatible with biased peer selection as implemented in this work. Rather than blocking BitTorrent flows, ISPs should change their bandwidth allocations so that it is more favorable to connect to peers inside the ISP than to those outside. Assuming that the former traffic costs are much smaller than cross-ISP traffic costs, this approach should lead to substantial savings for ISPs, higher subscriber satisfaction and fewer legal issues.

6.5 Multiple CDN Names

In the previous sections, we focused on how Ono reduces cross-ISP traffic without sacrificing download performance when using

[3]http://www.easynetconnect.co.uk

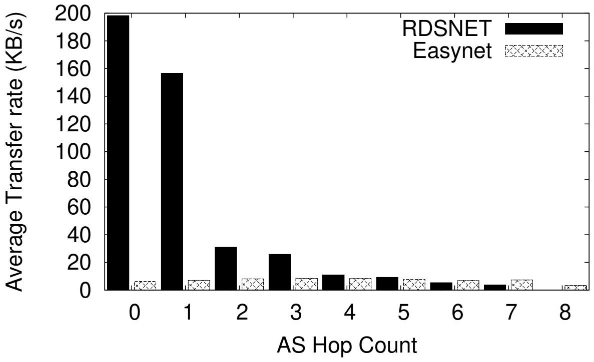


Figure 9: Bar plot relating AS hop count to transfer performance for ISPs with different bandwidth allocation policies. RDSNET gives better transfer rates to in-network traffic and Easynet does not. In the former case, Ono leads to significant performance gains.

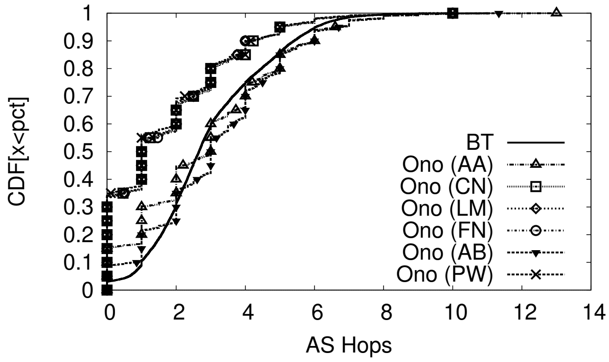


Figure 10: CDFs showing effects of different CDNs on average number of AS hops. Different CDN names lead to essentially two levels of Ono performance. The majority of CDN names lead to the best performance and those with worse performance still significantly reduce cross-ISP traffic compared to unbiased BitTorrent.

a single CDN name. Ono supports the reuse of information from multiple CDNs and through various CDN names; we now examine its performance when using other CDN names. To generate each of the following figures, we use the same methodology as in the previous sections. In addition to plotting a curve for peers found using each CDN name, we include a curve for peers found through BitTorrent's random selection algorithm for comparison. Each curve is labeled with the abbreviation for its corresponding CDN name in Table 2.

Figure 10 shows a CDF of the average number of AS hops to peers found by Ono for each CDN name. It is immediately clear that there are essentially two levels of service provided by CDN names used by Ono. The best CDN names, which include all of the Akamai CDN names except Air Asia, lead to large reductions in cross-ISP traffic with nearly half of the Ono-recommended peers being located at most one AS hop away from the source. The other CDN names, which include Air Asia (Akamai) and ABC's streaming video site (Limelight), lead to median AS-hop values triple those of the previous curves. Using these names, however, Ono can still significantly reduce cross-ISP traffic—it is over three times more likely than random to find a peer that is at most one AS hop away.

The reason for the different performance curves is that different CDNs and CDN names correspond to different approaches for providing service to their customers. For example, consider the two levels of service provided by the Akamai CDN. For the better curves, Akamai directs Web clients to one of over 10,000 replica servers worldwide. Because these servers are often located in ISP points-of-presence, the information gathered from these redirections can often be used to distinguish between peers in different ISPs. The Air Asia CDN name offers lower performance to Ono because Air Asia subscribes to a different CDN service that uses a small subset of Akamai data centers located at a number of key locations worldwide. Finally, the worst Ono performance comes from using the Limelight CDN. Limelight uses a small number (< 20) of data centers distributed worldwide (similar to the service provided by the Air Asia CDN name), thus providing much coarser proximity information. Even in this case, Ono can still significantly reduce cross-ISP traffic when compared to default BitTorrent peer selections.

The RTT latency to Ono-recommended peers is presented in Fig. 11 on a semilog scale. Similar to the previous figure, we see

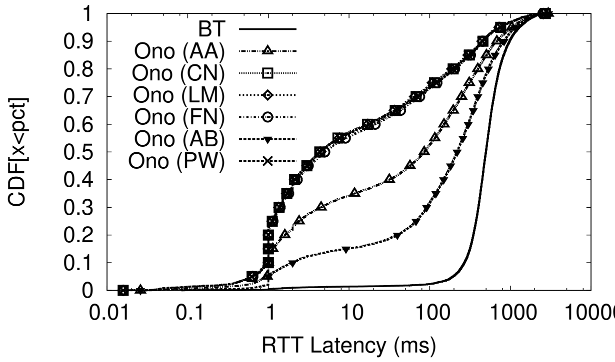


Figure 11: CDFs showing effects of different CDN names on average latency to selected peers. All CDN names result in much lower latencies than unbiased BitTorrent. The three distinct curves are influenced by the way in which CDNs provide service to their customers.

a clear separation among the curves indicating different levels of service. The best curves see two orders of magnitude improvement in median latency as described in Section 6.3. The next best curve shows larger average latencies, yet the median latency to Ono-recommended peers is still approximately an order of magnitude smaller than to peers selected at random by BitTorrent. Finally, the Limelight CDN, which offers the lowest performance of the group, still allows Ono to reduce mean latency by more than half compared to BitTorrent selection.

When considering the packet loss rates to Ono-recommended peers (Fig. 12), we find that all of the CDN names result in paths with lower loss rates on average than paths to peers picked by unbiased BitTorrent. The loss rates for the *AA* and *AB* CDN names are slightly higher than the rest, most likely because those CDN names on average produce longer paths to peers (both in terms of AS hops and router hops).

The curves for different CDN names in Fig. 13(a) and 13(b), which describe download and upload performance, are nearly identical. This lends further evidence to the claim that most BitTorrent connections are severely restricted by limited available bandwidth, which reduces the likelihood of seeing performance

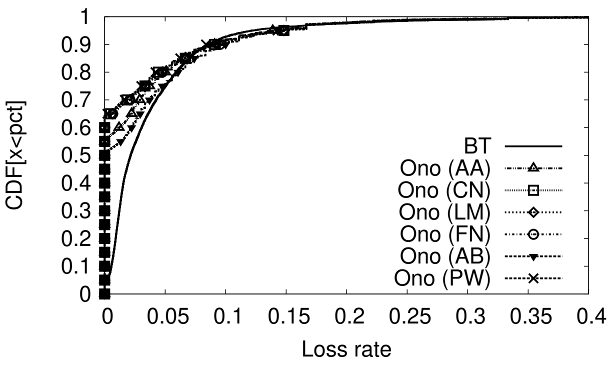


Figure 12: CDFs showing effects of different CDN names on average loss rates along paths to selected peers. The loss rates for all CDN names are lower than unbiased BitTorrent.

gains provided by high-quality paths found by Ono. We note, however, that there is noticeable separation in the curves as they approach higher throughput. We believe this occurs because at higher transfer rates, the TCP flows are much more sensitive to the latency and packet loss benefits from Ono peer selection. Regardless, the average transfer performance for all Ono curves is higher than that of unbiased BitTorrent, with improvements ranging from 6.4% to 33% in the downstream transfer rates.

7. DISCUSSION

In this paper, we demonstrated how to use CDNs as a low-cost approximation to an ideal oracle for a biased peer selection service. An alternative low-cost approach is to select only peers whose AS numbers (ASNs) are identical. Though it will indeed reduce cross-ISP traffic, this approach suffers from several limitations. On the one hand, AS-level information can be too fine-grained. For example, large ISPs have been assigned many ASNs (e.g., Comcast has over 40 different ASNs), so using these numbers can restrict cross-AS traffic that is not cross-ISP traffic. Our approach mitigates this problem because *CDN redirections reflect Internet topology and ISP policies, structural information not present when using simple ASNs.* For example, Fig. 9 shows that performance benefits in RDSNET extend beyond a single AS number, indicating that simply using AS numbers fails to capture all of the benefits from our approach.

Another limitation of fine-grained AS-level selection is that there may be few (or zero) peers in the system for a given ASN. Because CDN-based information is coarser, it allows for more opportunities to reduce cross-ISP traffic. In particular, we showed that a significant number of paths to Ono-recommended peers crosses one or more ASes, but the average number of AS hops for these paths is nevertheless much lower than the corresponding number for paths from unbiased BitTorrent. In other words, a pure AS-based approach cannot achieve the reduction in cross-ISP traffic that Ono provides in the area between the curves where $x > 0$ in Fig. 3.

On the other hand, ASNs can be too coarse, e.g., for an AS that is broadly geographically distributed. This could lead to poor performance from biased peers due to large latencies – e.g. AS7132 spreads over most of the continental US. Because CDN redirection are based primarily on latency, however, CDN-based oracles can successfully avoid these scenarios.

Another potential source for low-cost oracle information is an absolute network positioning system, e.g., Vivaldi [10] and GNP [24]. Peers could use such systems to exchange position information and bias their connections toward those with a smaller "distance." There are several limitations to this approach. For one, all peers in the system must take part in the positioning service to participate in biased selection. In contrast, Ono peers can perform DNS lookups for those peers not running the service. Even with positions for each peer, one must determine a distance threshold for including a peer in the biased set. It is unclear what, if any, threshold will lead to good performance worldwide. CDN redirections, on the other hand, offer a natural way to determine proximity between peers and we have shown that they effectively do so when encoded as ratio maps. Finally, network coordinates require potentially complex mechanisms to support large-scale systems [36], whereas the information generated by CDN redirections offers a simple, scalable and efficient way to store and retrieve ratio maps relevant to Ono recommendations, when using either centralized trackers or decentralized storage.

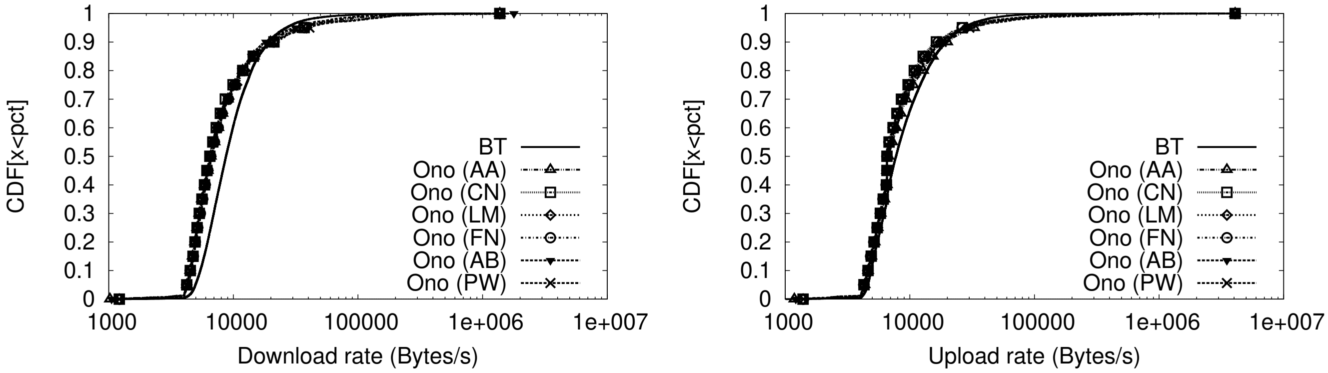
There is a number of concerns that arise from using CDN redirections in previously unanticipated ways. To begin, it is important to note that our system's interactions with CDNs in no way forces them to behave in ways that contradict their fundamental policies, nor does it access the type of information that CDNs do not already make publicly available for free [6]. Another issue is whether our service is disruptive to CDN operations. In fact, our service does *not* place a large (or even significant) burden on the CDNs from which it gathers network information. In particular, our system performs only name translations and does not actually download CDN content, so there is no additional data-traffic load placed on the CDN servers. Because our system queries its local DNS server to determine replica-server mappings, DNS lookups can be answered from the local DNS cache without contacting the CDNs' DNS servers. Further, our adaptive lookup rate mechanism can generate as few as 48 lookups from each peer per day—likely a vanishingly small fraction of those generated by web clients running in the same network.

Another important consideration is how our system will perform should CDNs change their behavior. We believe that the goal of our system (reducing cross-ISP traffic) and the policies/goals of CDNs (transferring content over high-quality paths) are directly aligned. Thus, while it is always possible for CDNs to change their behavior, we do not expect any change to interfere with our service's ability to reduce cross-ISP traffic.

8. CONCLUSION

In this paper, we presented the design, implementation and evaluation of an effective and scalable approach for reducing cross-ISP traffic in P2P applications without sacrificing performance, assuming trust between ISPs and their subscribers or requiring deployment of additional infrastructure. Our approach recycles network views collected by CDNs to inform a peer-selection algorithm that biases connections toward peers that are likely to minimize costly cross-ISP traffic. To experiment with it, we made an implementation available as an extension to the Azureus BitTorrent client beginning in April 2007. Since then, the extension has been installed by over 120,000 subscriber peers distributed worldwide. With their help, we performed extensive, continuous measurements, and are currently recording data for connections between over 2.5 million peer IP addresses per day. In particular, we collected DNS redirection information, transfer rates, path latencies and traceroute measurements.

We used this data to show that our approach scales easily to well over one hundred thousand users, and effectively "tames" BitTorrent by significantly reducing cross-ISP traffic without sac-



(a) Per-customer download rates.

(b) Per-customer upload rates.

Figure 13: CDFs showing effects of different CDN names on average transfer rates. Different customer names do not significantly affect transfer performance, except at large rates where latency and packet loss become more significant. In all cases, Ono outperforms unbiased BitTorrent on average.

rificing its performance or robustness. We show that this approach finds paths between peers that do not cross a single AS over one third of the time and significantly reduces overall cross-ISP traffic compared to unbiased peer selection. Further, we demonstrate that our biased peer selection implementation locates peers along paths with median RTT latencies that are two orders of magnitude smaller than paths to peers selected at random. Finally, we show that our biased selection does not reduce average file-transfer performance and in fact significantly increases transfer rates when peers have sufficient available bandwidth.

Based on our analysis, we suggest a new ISP-based approach to the problem of taming BitTorrent that is compatible with biased peer selection as implemented in this work. Rather than blocking BitTorrent flows, ISPs should change their bandwidth allocations so that it is more favorable to connect to peers inside the ISP than to those outside. This approach will allow ISPs to significantly reduce costs while improving the user experience for their customers.

We showed that CDN-based oracles for peer selection works in a popular P2P file-transfer application, and we expect that the high-quality paths that it finds will be useful in a variety of other P2P application contexts. As part of our future work, we plan to provide an open-source library that implements our approach in a *protocol portable* way and evaluate its effectiveness for other aspects of P2P systems. Finally, to extend its reach to an even larger portion of the BitTorrent population, we are implementing a tracker that includes a biased set of peers in each scrape result according to our approach.

9. ACKNOWLEDGMENTS

We would like to thank our shephard, Ion Stoica, and the anonymous reviewers for their helpful comments and suggestions. We thank Kobus van der Merwe (AT&T Research), Pei Cao (Google), Yan Chen (Northwestern), Robin Kravets (UIUC) and Robert Morris (Cisco Systems) for their useful feedback on early ideas and drafts of this paper. We are also grateful to Paul Gardner for his assistance in deploying Ono in Azureus. This research was supported in part by US National Science Foundation CAREER Award CNS-0644062.

10. REFERENCES

[1] ADLER, M., KUMARY, R., ROSSZ, K., RUBENSTEIN, D., SUEL, T., AND YAOK, D. D. Optimal peer selection for P2P downloading and streaming. In *Proc. of IEEE INFOCOM* (2005).

[2] AGGARWAL, V., BENDER, S., FELDMANN, A., AND WICHMANN, A. Methodology for estimating network distances of Gnutella neighbors. In *Workshop on Algorithms and Protocols for Efficient Peer-to-Peer Applications at Informatik* (2004).

[3] AGGARWAL, V., FELDMANN, A., AND SCHEIDELER, C. Can ISPs and P2P users cooperate for improved performance? *SIGCOMM Comput. Commun. Rev. 37*, 3 (2007), 29–40.

[4] AKAMAI. Akamai CDN. `http://www.akamai.com`.

[5] AKAMAI. Sureroute, May 2003. `http://www.akamai.com/dl/feature_sheets/fs_edgesuite_sureroute.pdf`.

[6] AKAMAI. Akamai introduces first-of-its-kind, real-time view into health of the Internet, June 2007. `http://www.akamai.com/html/about/press/releases/2007/press_060707.html`.

[7] AKELLA, A., SESHAN, S., AND SHAIKH, A. An empirical evaluation of wide-area internet bottlenecks. In *Proc. of the Internet Measurement Conference (IMC)* (2003).

[8] BINDAL, R., CAO, P., CHAN, W., MEDVED, J., SUWALA, G., BATES, T., AND ZHANG, A. Improving traffic locality in BitTorrent via biased neighbor selection. In *Proc. of the Int'l Conference on Distributed Computing Systems (ICDCS)* (2006).

[9] COHEN, B. Incentives build robustness in BitTorrent. In *Proc. of the Workshop on Economics of Peer-to-Peer Systems (P2PEcon)* (2003).

[10] DABEK, COX, KAASHOEK, AND MORRIS, R. Vivaldi: A decentralized network coordinate system. In *Proc. of ACM SIGCOMM* (2004).

[11] DAWSON, K. FCC seeks comment in Comcast P2P investigation, January 2007. `http://yro.slashdot.org/yro/08/01/16/0238244.shtml`.

[12] GUMMADI, K., GUMMADI, R., GRIBBLE, S., RATNASAMY, S., SHENKER, S., AND STOICA, I. The impact of DHT routing geometry on resilience and proximity. In *Proc. of ACM SIGCOMM* (2003).

[13] GUO, L., CHEN, S., XIAO, Z., TAN, E., DING, X., AND ZHANG, X. Measurements, analysis, and modeling of BitTorrent-like systems. In *Proc. of the Internet Measurement Conference (IMC)* (2005).

[14] IPOQUE. Internet Study 2007: Data about P2P, VoIP, Skype, file hosters like RapidShare and streaming services like YouTube, November 2007. `http://www.ipoque.com/media/internet_studies/internet_study_2007`.

[15] IZAL, M., URVOY-KELLER, G., BIERSACK, E., FELBER, P., HAMRA, A., AND GARCES-ERICE, L. Dissecting BitTorrent: Five months in a torrent's lifetime. In *Proc. of Passive and Active Measurement Workshop (PAM)* (2004).

[16] KANGASHARJU, J., ROSS, K., AND ROBERTS, J. Performance evaluation of redirection schemes in content distribution networks. *Computer Communications 24*, 2 (2001), 207–214.

[17] KARAGIANNIS, T., RODRIGUEZ, P., AND PAPAGIANNAKI, K. Should internet service providers fear peer-assisted content distribution? In *Proc. of the Internet Measurement Conference (IMC)* (2005).

[18] LI, J., AND SOLLINS, K. Exploiting autonomous system information in structured peer-to-peer networks. In *ICCCN* (2004).

[19] LIMELIGHT NETWORKS. Limelight networks CDN. `http://www.limelightnetworks.com`.

[20] MADHYASTHA, H. V., ISDAL, T., MICHAEL PIATEK, DIXON, C., ANDERSON, T., KIRSHNAMURTHY, A., AND VENKATARAMANI, A. iPlane: an information plane for distributed systems. In *Proc. of the USENIX Operating Systems Design and Implementation (OSDI)* (2006).

[21] MENNECKE, T. DSL broadband providers perform balancing act. `http://www.slyck.com/news.php?story=973`, November 2005.

[22] MIRROR IMAGE. Mirror image CDN. `http://www.mirror-image.net`.

[23] NAKAO, A., PETERSON, L., AND BAVIER, A. A routing underlay for overlay networks. In *Proc. of ACM SIGCOMM* (August 2003).

[24] NG, T., AND ZHANG, H. Predicting Internet network distace with coordinates-based approaches. In *Proc. of IEEE INFOCOM* (2002).

[25] POUWELSE, J. A., GARBACKI, P., EPEMA, D. H. J., AND SIPS, H. J. The Bittorrent P2P file-sharing system: Measurements and analysis. In *Proc. of the International Workshop on Peer-to-Peer Systems (IPTPS)* (Feb 2005).

[26] QIU, D., AND SRIKANT, R. Modeling and performance analysis of BitTorrent-like peer-to-peer networks. In *Proc. of ACM SIGCOMM* (2004).

[27] SALTON, G., AND MCGILL, M. J. *Introduction to modern information retrieval*. McGraw-Hill, New York, NY, 1986.

[28] SANDVINE. Sandvine incorporated: Peer-to-peer policy management, 2008. `http://www.sandvine.com/solutions/p2p_policy_mngmt.asp`.

[29] SHAIKH, A., TEWARI, R., AND AGRAWAL, M. On the effectiveness of DNS-based server selection. In *Proc. of IEEE INFOCOM* (2001).

[30] SHANAHAN, K., AND FREEDMAN, M. J. Locality prediction for oblivious clients. In *Proc. of the International Workshop on Peer-to-Peer Systems (IPTPS)* (Ithaca, NY, 2005).

[31] SHEN, G., WANG, Y., XIONG, Y., ZHAO, B. Y., AND ZHANG, Z.-L. HPTP: Relieving the tension between ISPs and P2P. In *Proc. of the International Workshop on Peer-to-Peer Systems (IPTPS)* (2007).

[32] SU, A.-J., CHOFFNES, D., BUSTAMANTE, F. E., AND KUZMANOVIC, A. Relative network positioning via CDN redirections. In *Proc. of the Int'l Conference on Distributed Computing Systems (ICDCS)* (2008).

[33] SU, A.-J., CHOFFNES, D. R., KUZMANOVIC, A., AND BUSTAMANTE, F. E. Drafting behind Akamai: Travelocity-based detouring. In *Proc. of ACM SIGCOMM* (2006).

[34] TEAM CYMRU. The Team Cymru IP to ASN lookup page. `http://www.cymru.com/BGP/asnlookup.html`.

[35] WELLINGTON, B. `dnsjava`. `http://www.dnsjava.org/`.

[36] WONG, B., SLIVKINS, A., AND SIRER, E. Meridian: A lightweight network location service without virtual coordinates. In *Proc. of ACM SIGCOMM* (2005).

[37] XIE, H., YANG, R., KRISHNAMURTHY, A., LIU, Y., AND SILBERSCHATZ, A. P4P: Provider portal for (P2P) applications. In *Proc. of ACM SIGCOMM* (2008).