DAVID CHOFFNES AND
FABIÁN E. BUSTAMANTE

taming the Torrent

David R. Choffnes will receive his PhD this year from the Department of Electrical Engineering and Computer Science at Northwestern University. Not surprisingly, his research interests align with those of his advisor (below). In particular, he currently focuses on designing, building, and evaluating large-scale, practical distributed systems.

drchoffnes@eecs.northwestern.edu

Fabián E. Bustamante is an associate professor in the Department of Electrical Engineering and Computer Science at Northwestern University. His research interests include operating systems and distributed systems, and networking in both wired and wireless settings.

fabianb@eecs.northwestern.edu

OVER THE PAST DECADE, THE PEER-TO- peer (P2P) model for building distributed systems has enjoyed incredible success and popularity, forming the basis for a wide variety of important Internet applications such as file sharing, voice-over-IP (VoIP), and video streaming. This success has not been universally welcomed. Internet Service Providers (ISPs) and P2P systems, for example, have developed a complicated relationship that has been the focus of much media attention. While P2P bandwidth demands have yielded significant revenues for ISPs as users upgrade to broadband for improved P2P performance, P2P systems are one of their greatest and costliest traffic engineering challenges, because peers establish connections largely independent of the Internet routing. Ono [4] is an extension to a popular BitTorrent client that biases P2P connections to avoid much of these costs without sacrificing, and potentially improving, BitTorrent performance.

Most P2P systems rely on application-level routing through an overlay topology built on top of the Internet. Peers in such overlays are typically connected in a manner oblivious to the underlying network topology and routing. These random connections can result in nonsensical outcomes where a peer—let's say, in the authors' own campus network in the Chicago suburbs—downloads content from a host on the West Coast even if the content is available from a much closer one in the Chicago area. This can not only lead to suboptimal performance for P2P users, but can also incur significantly larger ISP costs resulting from the increased interdomain (cross-ISP) traffic.

The situation has driven ISPs to the unfavorable solution of interfering with users' P2P traffic—shaping, blocking, or otherwise limiting it—all with questionable effectiveness. For instance, when early P2P systems ran over a fixed range of ports (e.g., 6881–89 for BitTorrent), ISPs attempted to shape traffic directed toward those ports. In response, P2P systems have switched to nonstandard ports, often selected at random. More advanced ISP strategies, such as deep packet inspection to identify and shape P2P-specific flows, have resulted in P2P clients that encrypt their connections. Recently, some

ISPs have attempted to reduce P2P traffic by placing caches at ISP network edges or by using network appliances for spoofing TCP RST messages, which trick clients into closing connections to remote peers. 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 not only may break the law but also can lead to significant backlash. Given this context, it is clear that any general and sustainable solution requires P2P users to buy in.

One possible approach would be to enable some form of cooperation between P2P users and ISPs. ISPs could offer an oracle service [2] that P2P users rely on for selecting among candidate neighbor peers, thus allowing P2P systems to satisfy their own goals while providing ISPs with a mechanism to manage their traffic [7]. However, we have seen that P2P users and ISPs historically have little reason to trust each other. Beyond this, supporting such an oracle requires every participating ISP to deploy and maintain infrastructure that participates in P2P protocols.

To drive peer selection, Ono adopts a new approach based on recycled network views gathered at low cost from content distribution networks (CDNs) without additional path monitoring or probing. Biased peer selection addresses a key network management issue for ISPs, obviating controversial practices such as traffic shaping or blocking. By relying on third-party CDNs to guide peer selection, Ono ensures well-informed recommendations (thus facilitating and encouraging adoption) while bypassing the potential trust issues of direct cooperation between P2P users and ISPs.

We have shown that peers selected based on this information are along high-quality paths to each other, offering the necessary performance incentive for large-scale adoption. At the end of July 2009, Ono had been installed more than 630,000 times by users in 200 countries.

BitTorrent Basics

Before describing Ono, we discuss how BitTorrent peers select neighbors for transferring content. A more complete description of the BitTorrent protocol can be found in the article by Piatek et al. [5].

BitTorrent distributes a file by splitting it into fixed-size blocks, called pieces, that are exchanged among the set of peers participating in a swarm. After receiving any full piece, a peer can upload it to other directly connected peers in the same swarm.

To locate other peers sharing the same content, peers contact a tracker, implemented as a centralized or distributed service, that returns a random subset of available peers. By default, each peer initially establishes a number of random connections from the subset returned by the tracker. As the transfer progresses, downloading peers drop low-throughput connections and replace them with new random ones.

CDNs as Oracles

Ono biases peer connections to reduce cross-ISP traffic without negatively impacting, but indeed potentially improving, system performance. Ono's peer recommendations are driven by recycled network views gathered at low cost from CDNs.

CDNs such as Akamai or Limelight attempt to improve Web performance by delivering content to end users from multiple, geographically dispersed servers located at the edge of the network. Content providers (e.g., CNN.com and ABC.com) contract with CDNs to host and distribute their content. When a client attempts to download content hosted by a CDN, the client performs a DNS request for the associated URL. By redirecting the DNS lookup into the CDN DNS infrastructure, the CDN can dynamically respond with IP addresses of replica servers that will provide good performance (see Figure 1).

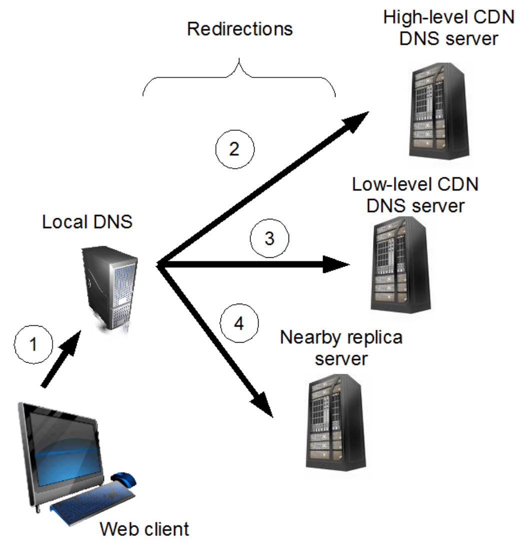

FIGURE 1: DIAGRAM OF CDN DNS REDIRECTION. THE WEB CLIENT PERFORMS A DNS LOOKUP (1), WHICH IS REDIRECTED TO A HIERARCHY OF CDN DNS SERVERS (2 AND 3) THAT EVENTUALLY RETURN THE IP ADDRESSES OF REPLICA SERVERS (4).

CDNs have all the attributes of an ideal oracle: they are pervasive, with an essentially global presence, have a comprehensive and dynamic view of network conditions [6], and, through DNS redirections, offer a scalable approach to access these views. Ono relies on CDNs as oracles, building on the hypothesis that when different hosts are sent to a similar set of replica servers, they are likely near the corresponding replica servers and by transition, also close to each other, possibly within the same ISP.

To use redirection information, we first need to encode it in a way that allows a client to compare the redirections that each peer witnesses. To this end, 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:

$$\mu_a = \{r_1 \rightarrow 0.75, r_2 \rightarrow 0.25\}$$

More generally, the ratio map for a peer P_a is a set of (replica-server, ratio) tuples represented as

$$\mu_a = \{(r_k, f_k), (r_l, f_l), \ldots, (r_m, f_m)\}$$

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 biased peer selection, 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. For cases in between, we use the cosine similarity metric to produce a continuum of similarity values between 0 (no similarity) and 1 (identical). This metric is analogous to taking the dot product of two vectors and normalizing the result.

In the next section, we show how Ono collects and distributes CDN redirection information in a scalable way and how it exploits this information to reduce cross-ISP traffic in BitTorrent.

Biasing P2P Connections with Ono

To perform biased peer selection, Ono must maintain a ratio map for each CDN-associated URL used for DNS lookups (e.g., a1921.g.akamai.net). As described in [6], using different CDNs and even different URLs for the same CDN can lead to different results in terms of redirection behavior. For the purpose of reducing cross-ISP traffic, these different URLs provide different granularity for proximity information.

The Ono plugin uses a built-in DNS client to perform periodic DNS lookups on popular URLs, which it uses to maintain ratio maps. 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*. 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: (1) through direct exchange between peers, (2) from trackers, and (3) from some form of distributed storage. Ono currently supports all of these options. With direct exchange, when two peers running the Ono plugin perform their connection handshake, the peers swap ratio maps directly. Our implementation for the opentracker software allows Ono peers to report ratio map information to a tracker and receive a set of nearby peers in response. The last option uses the Vuze built-in distributed hash table (DHT) to distribute ratio maps, but due to its relatively large latencies this technique is currently disabled.

When Ono determines that another 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 (i.e., waiting for data transfer to start). To maintain the appealing robustness that comes from the diversity of peers provided by BitTorrent's random selection, Ono biases traffic to only a fraction of the total connections established for a particular torrent (currently, at most, half).

Ono is written in Java and designed as a plugin (i.e., extension) for compatibility with the Vuze BitTorrent client. We chose Vuze 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. Our plugin 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 Vuze client.

Experience in the Wild

To evaluate the effectiveness of Ono, we instrumented the plugin to measure and report performance information for participating users. As our results show, Ono indeed reduces cross-ISP traffic without reducing transfer performance—in fact, Ono-selected peers on average *improve* performance compared to default peer selection. Thus, Ono not only reduces cross-ISP traffic but also provides an incentive for users to engage in ISP-friendly behavior. This incentive is the main reason behind Ono's large-scale adoption.

In our evaluation, we use traceroute measurements from Ono subscribers to connected peers to infer whether Ono-selected peers reduce cross-ISP traffic compared to ones selected by the standard (random selection) BitTorrent protocol. Traceroutes provide router-level views of paths between hosts. Since an ISP may contain many routers, we need 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 (http://www.cymru.com/).

For brevity, we present cumulative results when using the URL for LeMonde.com, the online version of a popular French newspaper. A comparison among all CDN-associated URLs can be found in [4].

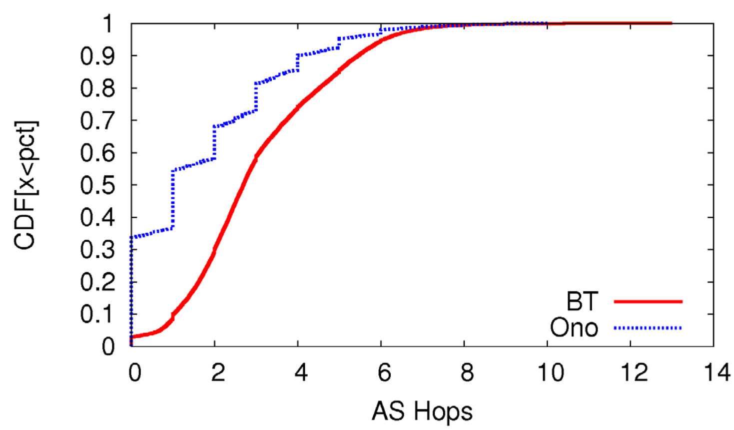

FIGURE 2: CDF OF AVERAGE NUMBER OF AS HOPS TO REACH ONO-RECOMMENDED PEERS AND THOSE FROM UNBIASED BITTORRENT. OVER 33% OF PATHS TO ONO-RECOMMENDED PEERS DO NOT LEAVE THE AS OF ORIGIN, AND THE MEDIAN NUMBER OF ASES CROSSED BY PATHS TO ONO-RECOMMENDED PEERS IS HALF OF THOSE PICKED AT RANDOM BY BITTORRENT.

Figure 2 presents a CDF of the number of AS hops taken along paths between Ono clients and their peers. 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.

Not only does Ono reduce cross-ISP traffic, but it does so while locating high-quality paths to biased peers. For instance, the median latency to Ono-recommended peers is 6 ms, whereas the same for peers picked at random is 530 ms—two orders of magnitude difference. We also saw improvements for traceroute-inferred loss: 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%.

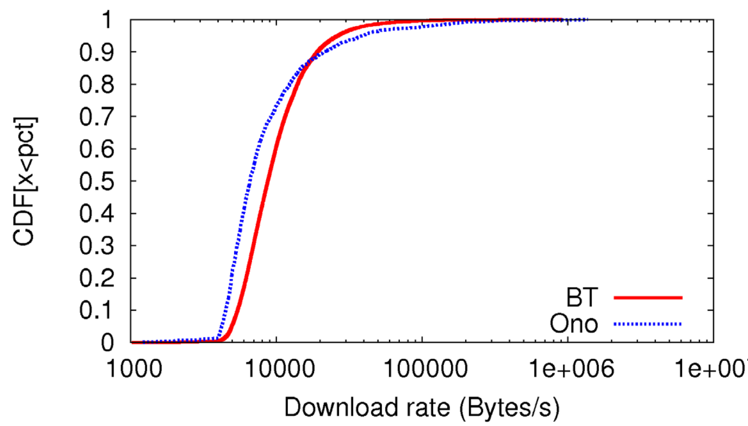

FIGURE 3: CDFS OF AVERAGE RATES FROM ONO-RECOMMENDED PEERS AND THOSE FROM UNBIASED BITTORRENT PEERS, ON A SEMI-LOG SCALE. THE AVERAGE DOWNLOAD RATE FOR ONO IS 31% BETTER THAN UNBIASED BITTORRENT AND THE DIFFERENCE IN MEDIAN DOWNLOAD RATES IS ONLY 2 KB/S.

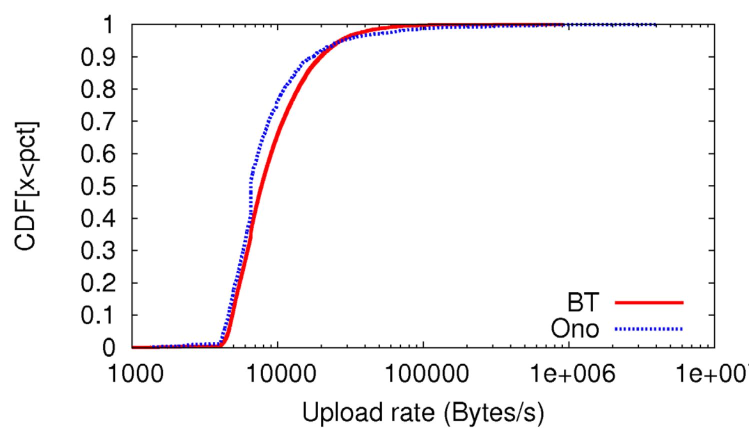

FIGURE 4: CDFS OF AVERAGE UPLOAD RATES FROM ONO-RECOMMENDED PEERS AND THOSE FROM UNBIASED BITTORRENT PEERS, ON A SEMILOG SCALE. THE AVERAGE UPLOAD RATE FOR ONO IS 42% BETTER THAN UNBIASED BITTORRENT AND THE DIFFERENCE IN MEDIAN RATES IS ONLY 1 KB/S.

In the end, however, it is safe to assume that most P2P users care how these paths impact *throughput* for their BitTorrent transfers. To evaluate this property, Figures 3 and 4 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 dropped 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 result 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. Note, however, that even when Ono-recommended peers do not provide higher median throughput than those picked at random, our approach does not noticeably affect completion time 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.

That said, 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 10 KB/s), we believe that performance gains for Ono-recommended peers are most likely 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 rather than the cross-ISP link [1]. While this situation occurs frequently, it is not universal and, as we now show, depends on ISPs' bandwidth-allocation policies.

Figure 5 plots a CDF of download rates from Ono clients located in the RDSNET ISP (http://www.rdslink.ro) in Romania. At the time of this study, the ISP offered 50 Mb/s unrestricted transfer rates over fiber for *in-network traffic* (i.e., traffic inside the ISP) and 4 Mb/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-recommended peers in 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%.

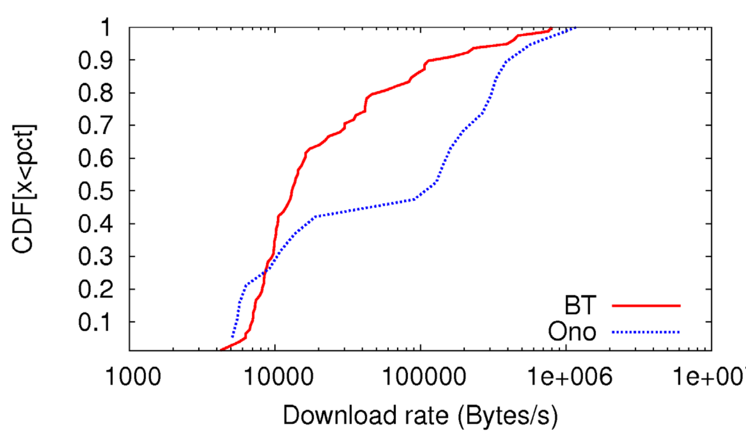

FIGURE 5: CDF OF AVERAGE DOWNLOAD RATE FOR AN ISP THAT PROVIDES HIGHER BANDWIDTH TO IN-NETWORK TRAFFIC. ONO THRIVES IN THIS ENVIRONMENT.

To compare against an ISP with uniform (and relatively low) bandwidth constraints, Figure 6 shows a CDF of download performance for Easynet (http://www.easynetconnect.co.uk), 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.

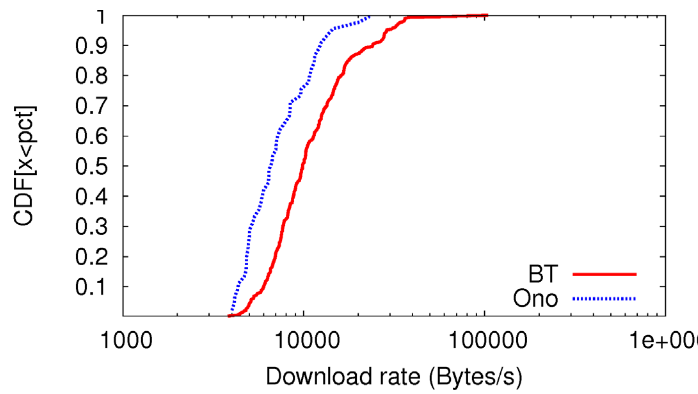

FIGURE 6: CDF OF AVERAGE DOWNLOAD RATE FOR AN ISP WITH AN ASYMMETRIC BANDWIDTH ALLOCATION POLICY, WHICH SIGNIFICANTLY CONSTRAINS ONO PERFORMANCE.

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. Figure 7 illustrates this using a bar graph, with the x-axis representing the number of AS hops along paths between peers and the y-axis representing 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 from those picked at random.

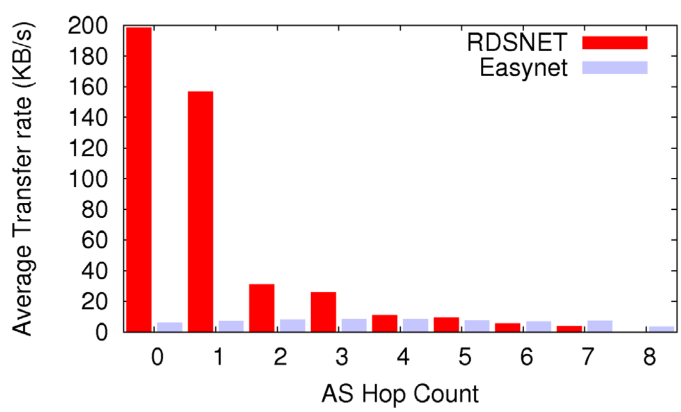

FIGURE 7: BAR PLOT RELATING AS HOP COUNT TO TRANSFER PERFORMANCE FOR ISPS WITH DIFFERENT BANDWIDTH ALLOCATION POLICIES. RDSNET GIVES BETTER TRANSFER RATES TO IN-NETWORK TRAFFIC AND EASYNET DOES NOT. IN THE FORMER CASE, ONO LEADS TO SIGNIFICANT PERFORMANCE GAINS.

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.

Conclusion

In this article, we briefly described how Ono reuses CDN redirection information to bias P2P connections, thus reducing ISPs' costs associated with P2P traffic without sacrificing system performance. While our current implementation is targeted specifically at the BitTorrent P2P protocol, the approach is being adopted for other services, including video streaming in the Goalbit project and P2P file transfer in Gnutella (in the Limewire client).

The Ono project is one piece in a broader research agenda that explores the potential for strategic reuse and recycling of network information made available by long-running services for building large-scale distributed systems. In that vein, we have reused CDN redirections to drive a high-performance detouring service [3] and reused passively gathered P2P performance information to automatically detect network problems. For more information, visit our Web site at http://aqualab.cs.northwestern.edu.

ACKNOWLEDGMENTS

We would like to thank Rik Farrow for his interest in our work. We are also grateful to Paul Gardner for his assistance in deploying Ono in Vuze. This work was supported by NSF CAREER Award Grant No. CNS-0644062.

REFERENCES

[1] A. Akella, S. Seshan, and A. Shaikh, "An Empirical Evaluation of Wide-Area Internet Bottlenecks," *Proceedings of the Internet Measurement Conference (IMC)* (2003).

[2] R. Bindal, P. Cao, W. Chan, J. Medved, G. Suwala, T. Bates, and A. Zhang, "Improving Traffic Locality in BitTorrent via Biased Neighbor Selection," *Proceedings of the Int'l Conference on Distributed Computing Systems (ICDCS)* (2006).

[3] D. Choffnes and F. Bustamante, "On the Effectiveness of Measurement Reuse for Performance-Based Detouring," *Proceedings of IEEE INFOCOM* (April 2009).

[4] D.R. Choffnes and F.E. Bustamante, "Taming the Torrent: A Practical Approach to Reducing Cross-ISP Traffic in P2P Systems," *Proceedings of ACM SIGCOMM* (2008).

[5] M. Piatek, T. Isdal, T. Andrewson, A. Krishnamurthy, and A. Venkataramani, "Building BitTyrant, a (More) Strategic BitTorrent Client," *;login:* 32, 4 (August 2007).

[6] A.-J. Su, D.R. Choffnes, A. Kuzmanovic, and F.E. Bustamante, "Drafting behind Akamai: Travelocity-Based Detouring," *Proceedings of ACM SIGCOMM* (2006).

[7] H. Xie, R. Yang, A. Krishnamurthy, Y. Liu, and A. Silberschatz, "P4P: Provider Portal for (P2P) Applications," *Proceedings of ACM SIGCOMM* (2008).