Quantifying the Benefits of Joint Content and Network Routing

Vytautas Valancius, Bharath Ravi, Nick Feamster, and Alex C. Snoeren†

Georgia Tech and †UC San Diego

ABSTRACT

Online service providers aim to provide good performance for an increasingly diverse set of applications and services. One of the most effective ways to improve service performance is to replicate the service closer to the end users. Replication alone, however, has its limits: while operators can replicate static content, wide-scale replication of dynamic content is not always feasible or cost effective. To improve the latency of such services many operators turn to Internet traffic engineering. In this paper, we study the benefits of performing replica-to-end-user mappings in conjunction with active Internet traffic engineering. We present the design of PECAN, a system that controls both the selection of replicas ("content routing") and the routes between the clients and their associated replicas ("network routing"). We emulate a replicated service that can perform both content and network routing by deploying PECAN on a distributed testbed. In our testbed, we see that jointly performing content and network routing can reduce round-trip latency by 4.3% on average over performing content routing alone (potentially reducing service response times by tens of milliseconds or more) and that most of these gains can be realized with no more than five alternate routes at each replica.

Categories and Subject Descriptors

D.2.0 [**Computer-Communication Networks**]: General

General Terms

Measurement, Performance

Keywords

Wide-area routing, CDN

1. INTRODUCTION

Online service providers (OSPs) such as Facebook and Google are offering an increasingly diverse set of interactive online services ranging from social networking services to online productivity tools. Consumers expect these services to be responsive, and OSPs are continually implementing optimizations to improve their performance with significant impact to their bottom lines. Google research showed that a 500-millisecond increase in latency caused a 20% traffic drop [8], while it was reported that a latency increase of 100 milliseconds can produce a 1% drop in revenue for Amazon [28]. Accordingly, recent years have seen many optimizations to accelerate the delivery of such services, ranging from better transport protocols [40] to browser enhancements [4, 7] to new compression and site optimization algorithms [1, 17].

One of the most common and effective ways to improve the performance of an online service is to decrease the path latency between the service and its clients by replicating the service at many geographic locations. Operators of replicated online services continually map clients to the data center that offers the best end-to-end service performance; this process is called *content routing*. Thanks to content distribution networks (*e.g.*, Akamai and Limelight), replication and content routing are prevalent for static content, but replicating full-featured Web-service logic that generates dynamic content can be difficult and costly, and, as a result, large-scale replication is not always feasible. Indeed, even the most popular Web service providers often host back-end logic at only a handful of sites. For example, Facebook serves all dynamic content from only four data centers, and Amazon serves its EC2-based properties from seven.

On the other hand, past work on detour routing [37], overlay routing [13], and multi-homing [10–12] suggests that the default wide-area Internet path between a client and any given service replica may be suboptimal: Network operators may be able to optimize the *network routing* at each replica site to improve performance. In practice, however, network operators have little visibility into the performance that a given replica would offer to a particular client or set of clients. The operations teams that perform network and content routing are often distinct, and frequently do not coordinate with one another [26]. The operators of major OSPs that we surveyed stated that their service-replica operators and network operators have only limited cooperation, and they do not attempt to reap the benefits of jointly optimizing content and network routing. Client performance suffers from this lack of coordination: operators of service replicas currently have no visibility into the performance or cost of alternate network paths between a service replica and its clients, so they optimize replica mapping based upon the current network paths that have been exposed as a result of network operators' traffic engineering optimizations. On the other hand, network operators, who do have access to alternate wide-area paths, have little insight into the application-level performance these alternate paths might provide.

To bridge this divide, we design and evaluate PECAN (Performance Enhancements with Content And Network routing), a sys-

SIGMETRICS'13, June 17-21, 2013, Pittsburgh, PA, USA.

tem that performs joint content and network routing for dynamic online services. PECAN enables joint content and network routing for online services by augmenting an OSP's existing content routing framework to provide a diverse set of wide-area routes between each interactive service replica and its clients. To ensure that PECAN does not harm the performance of any existing service, it explores alternate wide-area routes using separate IP prefixes; clients can always reach the online service either via the default wide-area Internet routes or via PECAN's routes.

We measure the performance benefits that PECAN can achieve in practice by emulating an online service provider's infrastructure. We place service replicas at the Transit Portal (TP) locations [6] and clients at nodes in the PlanetLab testbed. TP allows us to emulate an OSP with a five geographically diverse, U.S.-based points-of-presence (PoPs), each of which provides access to many alternate wide-area paths to clients. There are many ways to measure performance. One metric is Web page load time, but accurately measuring page load time is challenging, as it requires instrumenting each client with browser software—a difficult task for a large-scale measurement study. Instead, we focus on network latency (i.e., round trip time), because many online service providers have identified latency as a key factor governing a user's experience [18, 34].

Using three months of data from our testbed, we find that, when compared to performing content routing alone, using *joint routing* improves performance for about 35% of clients. Moreover, over 20% benefit directly from joint content and network routing: they achieve better performance by employing an alternate route to a different replica than they would have selected if only generically optimized default routes were available. In our experiments, we find that applying content routing alone decreases service latency by 16.75% on average relative to an optimally placed non-replicated service. Joint routing delivers an additional 4.3% (or about 5 ms) average round-trip latency reduction over performing content routing alone, which may translate to at least tens of milliseconds of reduction in Web page load time [41]. Of course, the performance benefits from replication will depend on the replicas' locations, but our results show that—especially for services that are difficult to replicate widely—PECAN can offer tangible benefits. The ability of an OSP to extract these gains will vary according to the techniques employed to explore alternate network routes.

We make several contributions. First, we perform (and publicly release) millions of performance measurements over three months on a globally distributed testbed that emulates an online service provider network, to evaluate the benefits that joint content and network routing could offer to online services in practice. Although we focus on latency, we also quantify PECAN's benefits for throughput and jitter. Second, we decompose the performance results by studying how content routing and network routing alone reduce network latency on our testbed. Finally, we have developed and deployed a prototype implementation of PECAN, which we describe in Section 3.

2. STATE OF THE ART

We begin by providing an overview of both content routing and network routing as employed by online service providers today. We first describe the state of the art, as best as we can determine through discussions with network operators and online service providers. We then explain in more detail the challenges operators face in jointly performing wide-area network routing and content routing.

2.1 Content Routing

Content routing refers to the process of selecting which replica among a geographically distributed set should service a particular request. Content in the context of this work refers to both static content and interactive services. Content routing systems have been heavily influenced by academic research, which we overview in Section 7. Today's content routing systems deployed in practice perform three major functions: 1) collecting performance information, 2) mapping clients to replicas, and 3) directing clients to replicas according to the client-replica map.

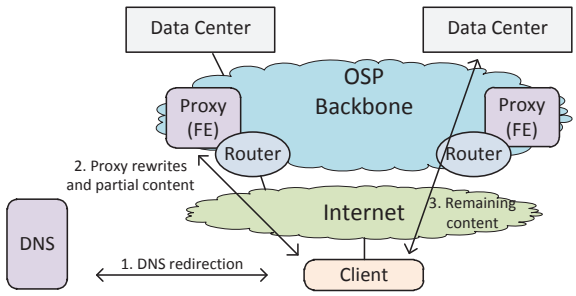

Figure 1: *Directing clients to replicas. Some services rely only on DNS; other services can use proxies for HTTP redirects and client-specific HTML rewriting.*

Step 1: Collecting performance information. OSPs use many technologies to measure performance between online service replicas and their clients. Such measurements can be classified into *active*, *passive*, and *indirect*. Active measurements usually involve sending probes from replicas to clients (or *v.v.*), which provide direct information about the network performance to the client. Active measurements are problematic in at least two ways: 1) the probes might not be handled by the network in the same way the actual traffic is handled, and 2) active measurements do not scale to a large number of replicas and customers [22]. For these two reasons, OSPs typically prefer to use a combination of passive and indirect measurements.

Passive measurements record the performance of the actual online service traffic between the replica and the clients. Passive measurements scale well and can provide direct insight into the performance of a service over the network by recording information such as TCP round-trip times and packet loss. To measure performance for all <client, replica> pairs, OSPs randomly redirect a small fraction of their clients' requests to alternate replicas [38].

To map clients to replicas when no active or passive measurements are available to inform selection, OSPs often resort to indirect inference of the likely performance between replicas and clients. Commercial IP geo-location databases [29] are often augmented with historical information to estimate performance between replicas and clients. As new clients begin to use the system, the OSP can update indirectly inferred performance estimates with passive measurements.

Step 2: Mapping clients to replicas. OSPs use a variety of proprietary algorithms to map clients to replicas. Client-to-replica performance is important, but there are other inputs to the algorithms that produce these mappings as well, including service availability, servicing costs, desired load, and regulatory restrictions [45]. We focus strictly on the performance aspects of the mapping and leave potential interactions with such policy constraints to future work.

Step 3: Directing clients to replicas. OSPs use three main techniques to implement their client-to-replica mapping: 1) DNS-based redirection, 2) HTTP redirection, and 3) client-tailored HTML rewriting. DNS mapping uses DNS servers to respond to clients with IP addresses of best replicas. ("Clients" in this case most of-

ten are the DNS resolvers resolving names on behalf of the end-systems.) DNS mapping is most useful to improve the performance of initial resource requests. HTTP redirection can further redirect clients to a better replica (at a cost of initial request latency.) When the requested resource has multiple sub-components (*e.g.*, a Web page with images), the OSP can use client-tailored HTML rewriting to direct clients to retrieve sub-components from disparate replicas. Figure 1 shows an example a system which can perform content routing with both DNS-based direction and proxy-based redirection (*e.g.*, with a front end (FE) that rewrites URLs). In this study, we do not distinguish between the different ways that a network operator might perform content routing; we merely assume that an OSP can map clients to replicas in at least one of these ways.

2.2 Network Routing

Network routing refers to the process of selecting a network path that service replica will use to communicate a client. In this work we focus on the inter-domain routing where OSP has to choose wide-area paths between a service replicas at its edge PoPs and remote clients. We identify three types of approaches OSPs take to optimize wide-area routing: 1) long-term capacity planning and network build-out; 2) medium-term planning and execution of routing policy changes; and, in rare occasions, 3) deployment of off-the-shelf commercial platforms or services that enable near-real-time adjustments to inter-domain routing policies in reaction to changes in wide-area network conditions. We discuss each of these approaches below.

Long-term network planning. Large OSPs improve the performance of their interactive services by deploying their own networks and increasing the richness of their peering connections. OSPs must pick locations to build new PoPs and choose the strategy in pursuing peering, paid-peering, and provider connectivity. The variables OSPs consider include: cost of a new PoP, cost of backbone connectivity to the new PoP, the choice of peers and providers that can connect in the PoP, the customer base that will be served by the PoP, the benefit in terms of latency, throughput, or traffic cost reduction that the PoP will bring, *etc.*.

Medium-term planning and execution of routing policy changes. OSPs monitor the trends of their traffic patterns and adjust their routing policies over the existing infrastructure. Routing policy is adjusted due to changing peering and customer-provider agreements or due to shifts in traffic demand patterns. Such changes must be well thought out: changes in the egress routing policy might saturate upstream networks, while changes in ingress routing might unpredictably shift traffic loads among ingress links.

Commercial platforms and services. Avaya and Cisco offer the PathControl [36] and Performance Routing (PfR) [2] route management platforms, respectively. These platforms perform continual performance measurements to online services and adjust inter-domain routes between the services and their clients based on these performance measurements. Similarly, Internap provides route optimization services for their clients [3] by performing measurements along alternate paths and redirecting traffic between services and clients by adjusting inter-domain routing policy. These platforms and services primarily target large enterprises with multiple upstream providers. PECAN applies similar types of inter-domain route control to adjust routes between clients and replicas, and it is possible that some variant of these systems could be used to implement aspects of PECAN's network routing subsystem. Our evaluation also hints at how these services might scale to large OSPs who have have millions of clients and many replicas.

2.3 Joint Content and Network Routing

Uncoordinated content and network routing as described above exhibits the following problems. First, network operators have little insight into application-level wide-area path performance between a service replica and all of the potential clients for that replica. As described in Section 2.1, only passive measurements can scale to collect application-level performance data to all of the OSPs clients. Second, the content routing subsystem that collects passive measurements operates only on the paths that are already selected by network operators. As a result, potential alternate paths to the clients are never explored by content routing system.

PECAN enables joint content and network routing and solves the problem described above. Joint routing in this context is the ability for content routing system to explore wide-area network path diversity to select the best path between an end-user and a service. At a high-level, PECAN operates in three steps: 1) the network routing subsystem exposes a diversity of wide-area network paths; 2) the content routing subsystem collects application-level performance over the exposed paths; and 3) each client is content-routed to the best service replica over the best path to that replica.

When designing PECAN, we assume that services are replicated at the edge of the OSP's network. This assumption holds for at least some major OSPs and their services [42]. If services are located deeper in an OSP network, the OSP would have to jointly select not only replica and wide-area paths, but intra-domain paths as well.

3. PECAN

As we discuss in Section 2, modern OSPs use sophisticated content routing systems to load balance requests between replicated data centers in an attempt to improve client performance. In this section, we describe how operators can extend their existing content routing systems to support network routing as well. In particular, we present the design of PECAN (Performance Enhancements with Content And Network routing), a system that enables seamless integration of content and network routing.

3.1 Exposing Routing Choices

The basic idea behind PECAN is to extend an OSP's current client-replica mapping infrastructure to instead map clients to <replica, route> pairs. To do so, PECAN breaks each replica into a set of virtual replicas, where each virtual replica corresponds to a different choice of routes to the replica (i.e., a single <replica, route> tuple). Figure 2 shows how PECAN allows a content routing system to tap into network route diversity. The router in the figure has a separate routing slice dedicated to each set of alternate routes (virtual replica). For example, in today's routers such a slice can be implemented using virtual routing and forwarding (VRF) instances or a host of alternative technologies.

3.1.1 Egress routes

There are a wide variety of mechanisms available to employ alternate egress routes from a given virtual replica. For example, conventional BGP multi-homing can increase route diversity; operators can use BGP's local preference setting to adjust the choice of egress routes to each client prefix. PECAN could also benefit from protocols such as Detour [37], RON [13], Platypus [35], Deflections [47], and Path Splicing [32], all of which increase an end system's choice of (and control over) egress paths. Similarly, many practitioners see industry proposals like Locator/ID Separation Protocol (LISP) [19] as a feasible improvement to BGP. LISP separates the endpoint identifier (EID) (*i.e.*, a host IP address) information from routing locator (RLOC) (*i.e.*, the information that encodes the location of the EID in the wide-area Internet.) LISP

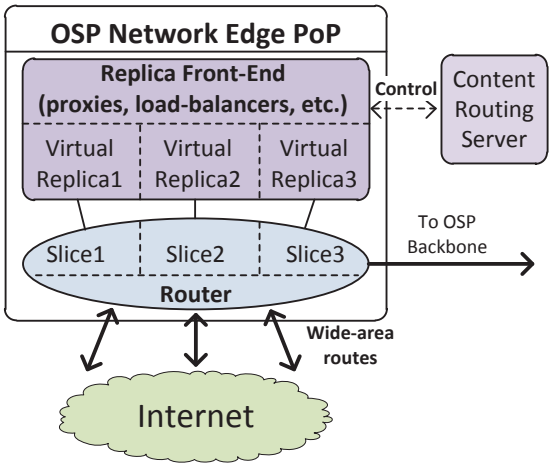

Figure 2: *Content and network routing subsystems allocate isolated resources to explore new network routes.*

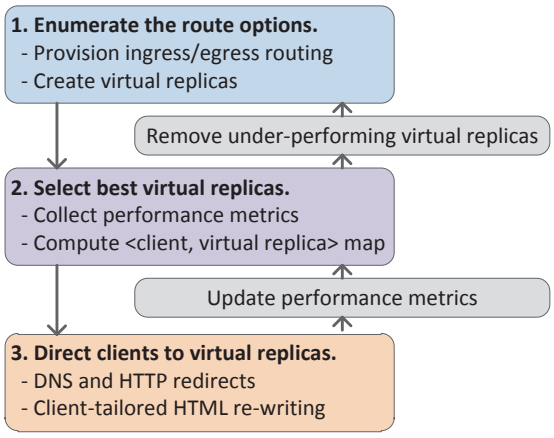

Figure 3: *Joint network and content routing selection with PECAN.*

could allow an OSP to explore egress routes by selecting the entry points to a remote network as encoded with RLOCs.

3.1.2 Ingress routes

Affecting a virtual replica's ingress routes is more challenging since it requires changing the way other networks forward packets. The key to enabling distinct route sets for each virtual replica is to separate these routing decisions. In PECAN, an OSP allocates a distinct IP address prefix to each virtual replica. Hence, to map clients to a particular virtual replica, PECAN need only point them to an IP address within the virtual replica's prefix.

Today's Internet supports a number of ways to impact route selection, including selective prefix announcement (*i.e.*, announcing a prefix only to a subset of neighbors), prepending `AS_PATH` attributes, setting BGP communities or MED attributes, and BGP `AS_PATH` poisoning. We evaluate employing AS path poisoning in PECAN extensively in the following sections. Future technologies, such as LISP, might provide even more elegant alternatives.

Critically, by maintaining one virtual replica (address prefix) at each physical replica that always uses the default network paths, PECAN does no harm: clients can always obtain the performance provided by content routing alone if none of the joint routing options provide superior performance.

3.2 Selecting a Virtual Replica

We now describe how PECAN's virtual replicas enable the process of joint content and network routing. The joint optimization could happen in many ways; we take an iterative approach, as shown in in Figure 3. First, PECAN optimizes network routing between each client/replica pair: for each replica, PECAN identifies the network path to each client that yields the best performance, and establishes a virtual replica with that path preference. Then, for each client, PECAN selects the virtual replica that offers the best performance among the available options at each physical replica.

This process proceeds in three steps, which could be either automated or manually performed by the operators of the network and online service.

1. **Enumerate the route options.** OSP network operators must enumerate the alternate routes from each replica to clients that the system should explore. Depending on the route selection technique employed, the operator may wish to enumerate egress (*e.g.*, a choice of a next-hop neighbor) routes and ingress (*e.g.*, selective route announcement) routes separately, or jointly. Evaluating all possible alternate routes to each client is unlikely to scale, but our evaluation (Section 6) shows considering just five virtual replicas (i.e., sets of alternate ingress routes) at each replica can realize performance improvements that are 60% of the maximum possible improvement.
2. **Select the best virtual replica for each client.** PECAN evaluates the performance of each virtual replica for each client. To evaluate a new virtual replica (route selection) for a given client and physical replica, PECAN redirects a small fraction of client requests to the virtual replicas and evaluates the performance that the client sees. PECAN gradually increases the number of clients mapped to a virtual replica to avoid overloading any network path or physical replica. Isolating test measurements from the bulk of the traffic requires a set of dedicated load-balancing proxies, as shown in Figure 2, As long as the evaluated route offers improved performance for enough clients and is reliable over the test period, PECAN maintains the virtual replica in the set of virtual replicas that can be used for joint routing.
3. **Direct clients to virtual replicas.** Once PECAN has selected the best virtual replica for each client, it implements the mapping. To implement this mapping, PECAN uses DNS load balancing to map each client to a virtual replica IP address, where the BGP prefix for that IP address corresponds to the route that the PECAN has selected for that client and replica using the previous steps. Because PECAN maps each virtual replica to its own prefix, a client always has the option of using either default content routing (*i.e.*, the route in today's CDN) or the PECAN-provided route.

4. THE CASE FOR JOINT ROUTING

We now summarize the potential gains of a joint content and network routing system; we expand on our methodology and findings in subsequent sections. We emulate an OSP setup using a globally distributed testbed that allows us to both replicate services across sites and control inbound routes to these sites. This testbed, which we describe in detail in Section 5.1, emulates an OSP with replicas in a number of geographically distinct locations with a diversity of wide-area network routes at each location.

Routing type	RTT	BW	Jitter
Best replica	107.35 ms	212.47 Mbps	5.95 ms
Network	4.35%	0.87%	9.32%
Content	16.75%	8.11%	11.82%
Joint	**20.44%**	**11.29%**	**17.57%**

Table 1: *Average improvement to latency (RTT), throughput (BW), and jitter. The baseline, over which improvement is measured for each technique, is the performance to the single best replica.*

Client percentile	RTT (%)	BW (%)	Jitter (%)
Most-improved 5%	16.94	10.25	41.20
Most-improved 10%	13.43	1.42	29.24
Most-improved 20%	9.91	0.47	13.48

Table 2: *Marginal gains of joint routing over content routing. Clients in each percentile improve by at least the amount indicated.*

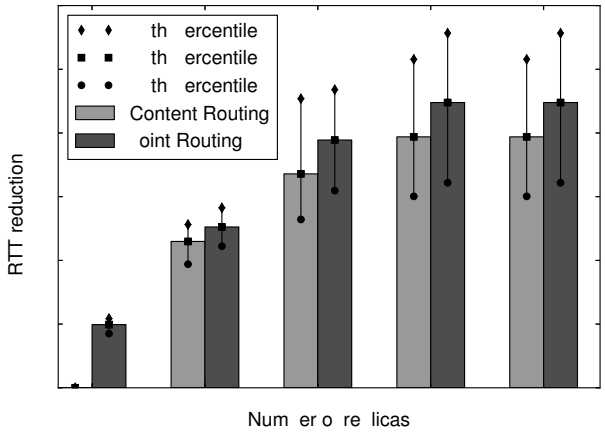

Figure 4: *Latency reduction as a function of the number of replicas; the baseline performance is the average performance to a single best replica.*

Overview of measurements. The testbed has five replicas distributed across the United States; from each replica, we explore about 250 alternate routing choices to 174 globally distributed clients. For three months (from October to December 2011), we collected a comprehensive <client, replica, route> performance map consisting of millions of measurements. Our raw dataset available for the public to download [5]. Section 5 explains our experimental setup in more detail.

Improvement over the best replica. When OSPs roll out a new online service, it often starts at a single replica and then expands to more sites. It is interesting to know how expanding the set of replicas and/or adding joint content and network routing improves the service performance. Table 1 compares how network routing, content routing and joint routing (PECAN) each improve over a single best replica. (We formally define each metric in Section 6.)

The "network" routing row in Table 1 shows the gains if the OSP chooses to explore alternate routes only for that single best replica. Conversely, the "content" routing line shows the improvement if the OSP chooses to replicate the service to all five locations available in our testbed. Content routing provides greater performance gains than simply applying network routing for one site. Finally, the "joint" routing row shows the gains attained when the OSP chooses to both replicate the service to all five locations and perform joint content and network routing.

In practice, in addition to network-level performance, the effectiveness of both replica and network path selection depend on traffic acquisition costs, replica loads, and other variables. Unfortunately, it is hard to obtain data to model the effect of such variables. Hence, we consider only latency, throughput, and jitter. While this choice might bias our results, it similarly impacts both content routing and joint routing; thus, we can still compare the two.

Figure 4 shows that the benefits from joint routing are largely independent of the size of the replica set in our testbed: adding more replicas to an OSP yields latency improvements for both content and joint routing. Hence—at least at the scales we study—an OSP can improve its performance using joint routing regardless of the number of replicas it currently employs. The figure shows the 80th, 85th and 90th-percentile gains over the performance of a single best replica.

Improvement over content routing. While joint routing unquestionably provides greater gains than network or content routing alone, most OSPs already deploy content routing. Hence, its practical impact is often governed by the marginal gain over content routing. Table 2 provides a breakdown of joint routing performance gains over content routing alone for various client percentiles. For example, when compared to content routing, the most-improved 10% of clients see their latency reduced by 13.43% or more, an increase of 1.42% or greater in throughput, and at least a 29.24% reduction in jitter. For online services such as search or online gaming, such latency savings are significant.

Improvement with limited route diversity. In some circumstances, it may not be practical to support all possible alternate routes to each replica. Fortunately, in our testbed, most of the improvements we observe are obtained by avoiding a few underperforming default routes. Hence, it may be possible to achieve the lion's share of the benefit with only a small set of alternate routes. In particular, for our testbed, Figure 12 (found in Section 6) shows that exploiting just five alternate routes at each replica yields 60% of the possible improvement across the hundreds of alternate routes that we consider in our evaluation.

Taken together, these results suggest that an OSP can provide tangible improvement over the state of the art by employing joint content and network routing. In the next two sections, we discuss PECAN's benefits in detail.

5. EVALUATION SETUP

This section describes our evaluation methodology. We describe the testbed infrastructure and our measurement procedures.

5.1 Infrastructure

We use PlanetLab to emulate a set of clients, from which we perform measurements to the replicas over many different sets of routes, and Transit Portal to deploy an ersatz Web service with both replica and route diversity.

5.1.1 Clients: PlanetLab

We use 174 PlanetLab nodes as our client set. From the full list of approximately 600 PlanetLab nodes, we select nodes with which we can establish sessions. We further filter the set of these "live" nodes to include only a single node per PlanetLab site. In the end, we have a client pool with 38% of the nodes in North America, 36% in Europe, 21% in Asia, and 5% in South America.

It is well known that PlanetLab nodes are not the best representation of the Internet. It is hard to quantify how much PlanetLab

Service Replica	Location	# of routes (poisons)	# of measurements (RTT)		# of measurements (BW)	
			Default route	Alternate routes	Default route	Alternate routes
1	Atlanta, GA	259	292,806	1,453,137	9,535	14,679
2	Clemson, SC	253	19,401	1,442,832	14,853	22,021
3	Princeton, NJ	261	224,457	1,438,588	5,595	6,243
4	Seattle, WA	247	366,357	347,302	14,844	9,651
5	Madison, WI	247	67,473	1,389,266	7,321	14,032

Table 3: *The Transit Portal deployments that we use to emulate a replicated online service with route control. At each Transit Portal location, we host a replica of an online service; from each of these locations, we explore approximately 250 alternate routes (poisons) between each replica and the set of clients that it could reach.*

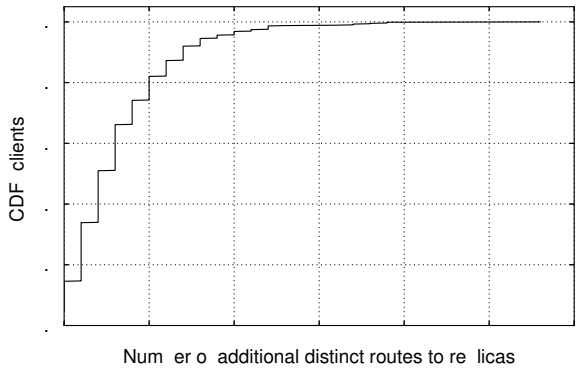

Figure 5: *Number of additional paths to replicas.*

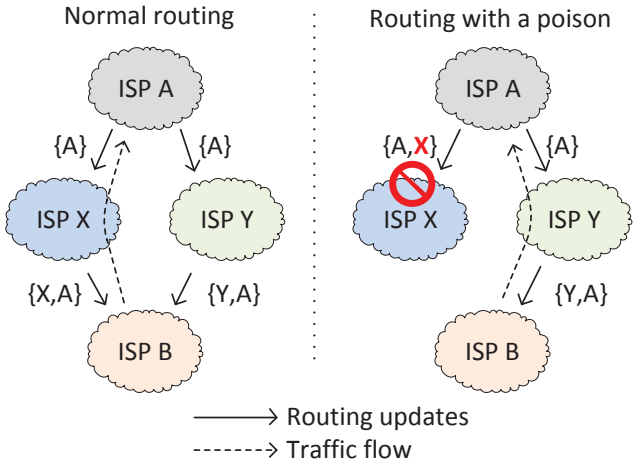

Figure 6: *Obtaining alternate paths between ISP A and ISP B with poisoning. The left-hand figure shows that, by default, ISP B prefers to route traffic for ISP A through ISP X. In the figure on the right, ISP A poisons AS X, leaving the path through ISP Y as the only viable path from ISP B to ISP A.*

biases our measurements. On one hand, PlanetLab nodes are better provisioned and have better "last mile" connections to the immediate provider than their residential counterparts. On the other hand, we focus our measurements on the performance we can gain by exploiting replica and route diversity in the network core, and not on the network edge. Our results will be more affected by how well a PlanetLab node's provider is connected to the Internet relative to an average Internet user. Most PlanetLab nodes' immediate access providers are academic institutions, whose connectivity to the Internet is often comparable to the connectivity of smaller ISPs or medium enterprises.

5.1.2 Replicas: Transit Portal

Transit Portal (TP) is a platform that enables researchers to perform experiments that require altering wide-area Internet routes [6]. There are five TP sites; each site has a functional Internet router, connecting to an upstream ISP, receiving a full Internet routing table. Each node is able to participate in BGP routing by issuing BGP updates from the IP address space and AS numbers allocated to Transit Portal. TP nodes allow multiple researchers to use these routing resources concurrently. We obtained access to the five TP sites described in Table 3, each of which acts as a replica in our testbed. In terms of Internet topology, the nodes in Atlanta and Seattle are very close to major Internet exchanges; the nodes in Clemson and Wisconsin are one AS-hop away from a Tier-1 provider, while the node in Princeton is two AS-hops away from a Tier-1 provider.

As explained below, TP sites can advertise routes using BGP `AS_PATH` poisoning to alter the routes that PlanetLab clients use to reach them. Using path poisoning, we discover approximately 250 different route advertisements from TP sites that result in alternate paths to at least one of the clients in our client set. Figure 5 shows a CDF of number of alternate paths per client; we find that on average, in addition to the default path, poisoning yields 3.4 paths per client to our replica set.

5.2 Experimental Procedure

We describe how we use our testbed to explore alternate routes between clients and replicas measure the performance improvements that result from these alternate routes.

5.2.1 Obtaining route diversity: poisoning

We explore wide-area route diversity by using BGP `AS_PATH` poisoning [16, 24]. BGP `AS_PATH` poisoning is an unconventional technique: it finds alternate, policy-compliant wide-area ingress routes to the network that advertises the poisoned route (in our case, the replicas). Although it has a number of known drawbacks in practice, BGP `AS_PATH` poisoning enables us to affect wide-area route selection without requiring access to the BGP routers that control inbound and outbound traffic to our replica sites. In practice, a real OSP could control both ingress and egress routes in a variety of ways. For example, OSPs could also use BGP `AS_PATH` prepending, BGP community attributes, and selective advertisements; to control egress traffic, OSPs can often select among multiple neighbors to send traffic. Given the wider variety of techniques at their disposal, it is plausible that OSP operators might see even greater performance gains than our experiments suggest.

BGP `AS_PATH` poisoning leverages the BGP-loop prevention algorithm to explore alternate routes on the Internet. As BGP

route advertisements propagate through the Internet, each router attaches its own AS number to the AS_PATH attribute. BGP's loop prevention algorithm, which is implemented on all BGP-speaking routers, says that a router must drop a BGP route update if the AS_PATH attribute of the route contains the AS number of said router. Dropping these updates prevents the router from accepting updates that the router has already received, thus preventing loops. As shown in Figure 6, an Internet Service Provider (ISP) can use BGP AS_PATH poisoning to exploit this algorithm by inserting a target ISP's AS number in the AS_PATH attribute before the update is originated. The target AS, in turn, will drop the update and its clients will likely choose alternate routes to the route originator.

We use the traceroute tool to identify the ISP networks (and their AS numbers) on the default paths from our clients to each replica. We then poison these AS numbers one by one to reveal alternate paths from clients to replicas. Not every client moves to an alternate path after we issue a poisoned update: some updates affect just a few clients, while some affect a great many. To find which clients are affected by a poison we, again, use traceroute from every client to the poisoned IP prefix. There is a possibility that a some fraction of these alternate paths are a result of network topology changes not under our control—i.e., not because of our poisoning. We repeatedly snapshot the default path between each <client, replica> pair to see how often the AS path changes to determine the "noise floor" of the Internet topology churn. We find that an average <client, replica> pair observes approximately 0.35 paths in addition to the default (un-poisoned) path during our study period. This low churn estimate gives us confidence that most of the 3.4 alternate paths that we observe (not counting the default) are due to poisoning.

5.2.2 Measuring performance improvements

There are many ways to measure network performance improvement; page load time is one of the most popular measures of OSP performance. We do not measure page load times due to the complexity of such measurements. Instead, we considered measuring median and average HTTP object download times, but find that latency is a reliable predictor for such times. (During preliminary testing we discovered that round-trip-time correlates to download time for a median-sized—400-byte [17]—object with a Pearson correlation coefficient of approximately 0.83, for both a default path as well as a poisoned path that affected 30% of the replica clients.) Hence, we measure three basic network performance primitives: latency, throughput, and jitter.

As explained in Section 3, each of our replicas advertises an un-poisoned prefix at all times. This un-poisoned prefix can always be reached to perform a measurement over the default path that rarely changes. For poisoned routes, we use the following sequence for each replica to collect a client/replica path performance map:

1. **Announce a prefix with a poisoned update.** The poison will propagate the prefix to some client networks over the alternate paths.
2. **Perform measurements to the poisoned prefix.** From every client in our client set, collect measurements to the replica using the poisoned prefix. Clients for which the poison did not affect the end-to-end path will see no improvement. Clients for which the prefix affected the end-to-end path will see either improved or reduced performance.
3. **Perform measurements to an un-poisoned prefix.** Conduct the same set of measurements over the default path (*i.e.*, using the un-poisoned prefix) to the replica to collect a contemporaneous baseline to which we will compare our poisoned path.

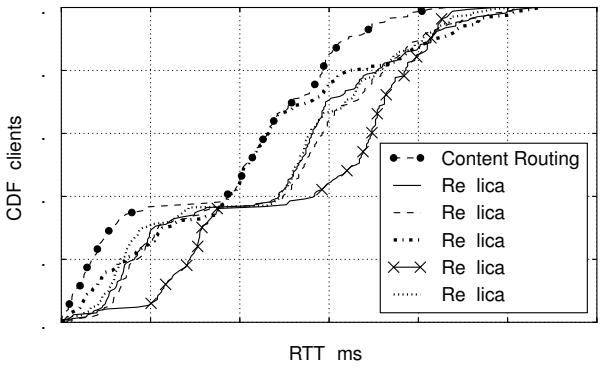

Figure 7: *Minimum latency over the default path.*

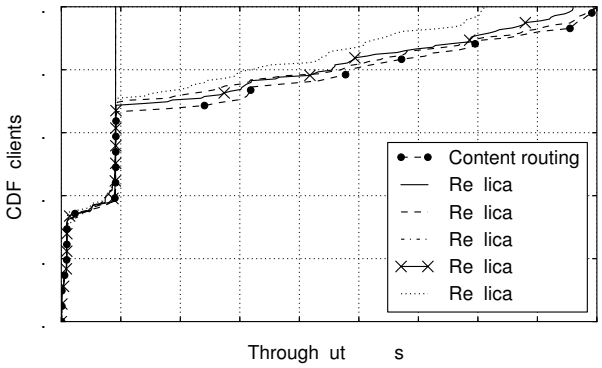

Figure 8: *Maximum throughput over the default path.*

As shown in Table 3, the dataset resulting from these measurements contains many more measurements of the default path than the poisoned paths. The abundance of default path measurements allows us to establish confidence in our baseline performance measurements. We consider a poisoned path between a client and a replica to improve latency over the client's default path only if the poisoned path shows latency smaller than the minimum latency ever recorded over the default path between the client and the replica. We apply a similar litmus test for jitter measurements. For throughput, we record an improvement only if a poisoned path produced higher throughput than any throughput measurement we ever observe on a default path.

6. EVALUATION

We evaluate the benefits of joint routing with respect to latency, throughput and jitter. We also show how well joint routing performs compared to traditional content routing.

6.1 Baseline Performance

When considering the performance improvements that different routing approaches induce, we must establish a baseline to compare them against. In this section, we use two baselines for comparison: 1) a best replica baseline, and 2) a content routing baseline. Before formally defining these baseline metrics, we describe how we perform the measurements that help us establish these baselines.

Measurements of default path performance. Figure 7 shows the CDF of minimum latencies clients experience to each replica over a

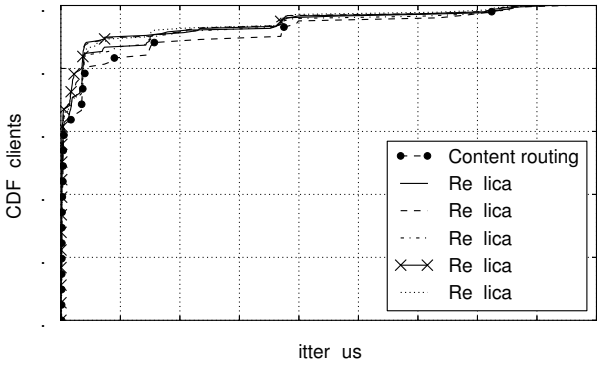

Figure 9: *Minimum jitter over the default path.*

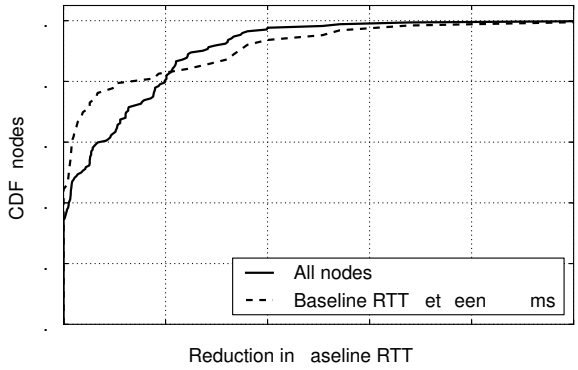

Figure 10: *Latency reduction when using joint routing.*

Replica	Latency	Throughput	Jitter
1	6.93%	7.77%	12.62%
2	1.15%	35.92%	35.92%
3	56.64%	6.77%	8.74%
4	22.54%	48.54%	18.45%
5	12.71%	0.97%	24.27%

Table 4: *Percentage of clients for which each replica is best.*

default path. The minimum latency for each client is obtained from a large set of measurements: On average, each client measures the default path to a replica 6,692 times. The figure shows two major groups of clients: About 40% of the clients have latencies between 0–50 ms, and about 60% of the clients see latencies of 90 ms or larger, with just a few in between. These modalities reflect the client geographic distribution: About 38% of clients are in the U.S and Canada and see lower latencies, while the rest are overseas.

Similarly, Figures 8 and 9 show the CDFs of maximum throughput and the minimum jitter, respectively, as observed by clients to each replica. As with latencies, the maximums and minimums are computed over a set of measurements for the default path between each <client, replica> tuple. For each such tuple, we have, on average, 189 jitter and throughput measurements. Figure 8 highlights why it is difficult to measure capacity using the PlanetLab nodes as clients. The clients in the figure form three distinct groups: 1) those with 10-Mbps links, 2) those with 100-Mbps links, and 3) those with speeds above 100 Mbps. The 10-Mbps and 100-Mbps groups identify cases where the PlanetLab nodes are directly connected to a bottleneck link; in these cases the bottleneck is at the client itself and no type of routing can improve its throughput.

Best replica baseline. When a new online service is launched, it often starts with a single replica. We want to know how much the network performance improves over that single replica when the OSPs start adding more replicas and implement content routing or joint routing. We define the *best replica* for some performance metric as the replica that the largest fraction of clients would select, given that each client can select its own best replica based on that performance metric. Table 4 shows the breakdown of popularity of different replicas when each client selects a replica based on the performance of the default paths for each <client, replica> tuple. The average performance to the best replica across all clients yields the *average best replica performance*. For example, as shown previously in Table 1, the average latency that clients experience to the best replica (which, as indicated in Table 4, is replica 3 in Princeton, NJ) is 107.35 milliseconds.

Content routing baseline. PECAN extends content routing to also incorporate the benefits of network routing; to quantify the additional benefit of PECAN relative to content routing, we compare the performance of PECAN against the performance that content routing alone provides. Recall that a content routing system maps each client to its own best replica. Formally, for a client i, the content routing latency $RTT_{content,i}$ is

$$RTT_{content,i} = \min_{j \in (1 \ldots M)} \widehat{RTT}_{ij0},$$

where M is number of replicas and $\widehat{RTT}_{ij0}$ is the minimum latency that we measured between client i and replica j over the default path (noted as path 0). Taking the average across all clients yields *average content routing performance*. In addition to showing the performance of clients to each replica, Figures 7–9 show the performance of a content routing system when each client is directed to its own best replica (labeled "content routing"). Note that the content routing system we implement uses only network performance as the basis for selecting a replica for each client; in practice, OSPs also use replica loads and costs to inform content routing decisions.

6.2 How Well Does Joint Routing Work?

We quantify the benefit that PECAN provides when compared to content routing. When PECAN is in use, we formally define client i's latency as

$$RTT_{PECAN,i} = \min_{j \in (1 \ldots M)} \min_{l \in (0 \ldots K_{ij})} \widehat{RTT}_{ijl}.$$

where M is the number of replicas, K_{ij} is the number of paths between client i and replica j, and $\widehat{RTT}_{ijl}$ is the minimum latency we recorded over the path l between client i and replica j. Recall that path $l = 0$ corresponds to the default path. With measurements of both content routing and PECAN performance, we can compute the percentage improvement as

$$100 \cdot (RTT_{content,i} - RTT_{PECAN,i}) / RTT_{content,i}.$$

We use the same approach for jitter; for throughput, we take maximums instead of minimums. Below we provide a breakdown of the average performance improvements that presented in Table 1.

Latency. Joint routing delivers an additional 4.3% (or about 5-ms) round-trip latency reduction on average beyond performing content routing alone, which may translate to significant improve-

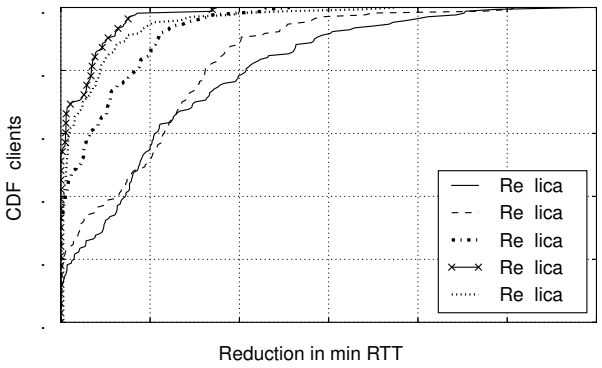

Figure 11: *Latency reduction with network routing.*

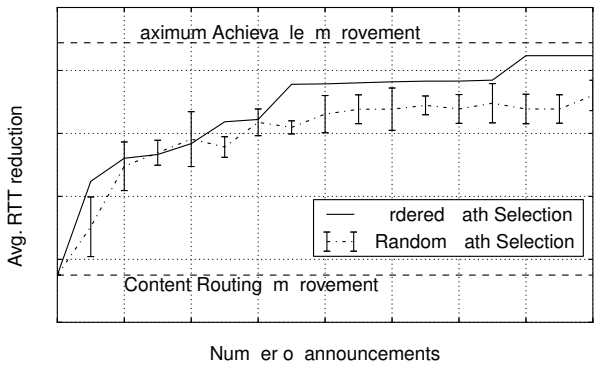

Figure 12: *Average latency reduction with increasing number of route announcements per replica.*

ments in service response times [41]. Figure 10 shows the percentage improvement in latency over the content routing baseline. The solid line in the figure shows improvement for all clients, while the dashed line shows shows the latency reduction for clients that had a baseline latency of 0–50 ms. We find that 20% of our clients see a reduction in latency of at least 10%. We also observe that clients with baseline latencies of 0–50 ms see similar improvements, with 18% of these clients improving by 10% or more. This result is significant, since content replication can only reduce latency by placing content closer to the users. At some point, however, placing replicas close to all clients might become prohibitively expensive; in such cases, an OSP might rely on PECAN to improve routing between the closest replica and the clients nearby.

As discussed earlier, Figure 4 plots content and joint routing benefit over the best replica baseline as we increase the number of replicas. Adding replicas provides higher improvements with joint routing, but with decreasing marginal improvement at every addition. Content routing behaves similarly, albeit with a lower improvement at each step.

Throughput and Jitter. When compared to the baseline of content routing, only 5% of clients that are using PECAN experience a throughput improvement of 20% or more, while almost 90% of clients see no improvement. We attribute this limited improvement to the inability of most PlanetLab nodes, which mostly use 10-Mbps and 100-Mbps interfaces, to saturate potential Internet path bottlenecks. In case of jitter, 10% of the clients see approximately 30% less jitter, while 60% of clients see no improvement at all. As with throughput, we see that PlanetLab nodes' inability to saturate router buffers along Internet paths limits the observed jitter.

6.3 Why Does Joint Routing Work?

Joint routing improves performance over content routing alone because it provides multiple alternate network paths for each replica. As explained in Section 5, PECAN finds about 3.4 alternate paths on average for each <client, replica> tuple. Even when a client cannot improve its performance by switching replicas, network routing can often improve performance to one of the replicas. Figure 11 shows latency improvements from network routing alone for each replica. For each replica, the baseline over which we compute the latency reduction is the minimum latency over the default path to that replica. We find that 20% of the clients experience improvements of 5–20%, depending on which replica we choose to evaluate. Some replicas (*e.g.*, Replicas 1 and 2) have relatively poor default paths, so approximately 80% of clients achieve some benefit from alternate paths to these replicas.

6.4 Scalability and Stability

To be practical, PECAN must judiciously limit the number of route changes it broadcasts to the Internet; it also should limit oscillations induced by large numbers of clients changing replicas. In this section we seek to assess these requirements by analyzing two questions: (1) How many routes must the OSP explore to achieve the benefits of joint routing? And (2) how many clients change their preferred replica after joint routing is applied?

6.4.1 Scaling route selection

OSPs that tweak egress routes can do so without affecting the Internet routing system. To explore the alternate ingress routes, however, OSPs must issue additional routing updates. All of the experiments presented so far evaluate the improvements that PECAN can provide by using approximately 250 ingress routes (achieved via poisoned BGP advertisements) at each geographic location. Over time, OSPs might be able to evaluate all of these 250 configurations, but doing so frequently may not be practical. Hence, we consider how many routes an OSP needs to explore to see improvement from joint content and network routing.

When an OSP uses a limited set of routes to feed traffic to virtual replicas, it must decide which routes to use, but selecting the optimal subset of routes for each replica is computationally intractable. To avoid exploring all possible route combinations of route advertisements from each replica, we devise a simple heuristic: for each replica, we sort, in descending order, the routes based on the average improvement they provide to all clients when compared to the performance over the default path. We then take the routes sequentially from that ordering and announce them from the replica. For example, if OSP decides to announce three of the alternate routes, it will pick three top routes from the ordered list. We compare this heuristic against selecting sets of routes at random.

Figure 12 shows how performance improves as we increase the number of announced routes. The gains in the figure are shown over the best replica baseline. The figure contains four plots: "maximum", "ordered", "random", and "content". The "maximum" line represents the maximum gain that an OSP can get with joint routing (see Table 1). The "ordered" line shows the gains as we increase number of alternate routes from 0 to 16. The routes here are picked using the heuristic described above. The "random" line shows the gains when we add additional routes at random. To generate the "random" plot, we run 10 iterations and report average and standard error values. Finally, the "content" line shows the content routing gains over the best replica (also from Table 1). In most

cases, the ordered heuristic outperforms random route selection. It is also encouraging that only five additional virtual replicas per physical replica can obtain approximately 60% of the performance gains that are possible with all 250 alternate routes.

6.4.2 Stability of replica selection

Content providers typically take loads at each replica into account when assigning clients to replicas. With PECAN, however, clients that previously had a suboptimal replica will shift their traffic to other replicas. The OSPs must quantify such shifting and plan for it accordingly (*e.g.* by provisioning necessary capacity or by directing clients to second-best replicas).

We quantify client shift in our testbed. We find that, when optimizing for latency, 8% of clients change the choice of replica when joint routing is enabled. In other words, 92% of clients use the same replica (but perhaps over a better network path) as the one they were using when only default paths were available.

7. RELATED WORK

We first survey prior work on content routing methods; then we cover prior work on enhancing inter-domain routing; finally, we overview prior research on joint content and network routing, most of which assumes intra-domain routing context.

7.1 Content Routing

The late 1990s witnessed the first efforts in optimizing mapping of end users to content or service replicas. Bhattacharjee *et al.* [15] presented a seminal paper describing a client-to-replica mapping system. This system used IP anycast to reach a directory service (*e.g.*, DNS) which then routed the clients to the best service replica based on the client-replica performance map. Seshan *et al.* [38] invented a new way to collect a comprehensive client-replica performance map: a small fraction of clients would be directed to randomly selected replicas to estimate the performance. Andrews *et al.* [14] presented a system called Webmapper that used algorithms for performing approximate client-to-best-replica matching.

The initial step in client-to-replica mapping is usually performed using the Domain Name System (DNS). Pang *et al.* [33] evaluated the responsiveness of DNS to changes in client-to-replica mapping and found that in many cases DNS is sluggish to respond. Nonetheless, DNS is still the primary method for directing initial client requests to the best replicas. Huang *et al.* [22] introduced a DNS reflection method for client-to-replica performance map generation. Instead of usual actual client traffic, DNS reflection forces local DNS (LDNS) servers to use iterative queries to remote replicas to estimate the delay between the LDNS servers and the replicas. The LDNS performance information is then used as a proxy metric for client-to-replica performance. Most recently, Wendell *et al.* [45] describe a system called DONAR that allows DNS servers to make mapping decisions with only partial global information.

7.2 Network Routing

In the last decade there has been a lot of research on improving performance of inter-domain routing. A lot of of such research was focused on the effectiveness of multi-homing enterprise networks. Our work builds upon these efforts and extends them to include scenarios where an OSP has a choice not only of diverse network paths (network routing) but also a choice of replicas (content routing). For example, Akella *et al.* [9–12] explore the effects of multi-homing on the performance of a site that either sends or receives Internet traffic. The authors study 68 Akamai nodes in 17 cities as a testbed: A city often contains multiple nodes, each with a different upstream ISP; the authors connect to all the Akamai nodes in one city to estimate performance of each ISP in that city, effectively emulating a multi-homing setup in that city. The authors found that route optimization produces greater benefits in peak time intervals. Unfortunately, the study was limited in the number of alternative Internet paths (only upstreams) and it did not consider joint network and content routing.

Others also study multi-homing routing but stop short of quantifying its benefits to a multi-replica configuration. Goldenberg *et al.* [20] assess the benefit of single-site multi-homing and also consider cost in the analysis. Guo *et al.* [21] analyze a commercial solution for multi-homed enterprises, which focuses on possible performance gain with two upstream ISPs. Lee *et al.* [27] explore ways to scale active measurements for multi-homed enterprises. Uhlig *et al.* [43] and Wang *et al.* [44] propose formalizing upstream ISP selection as an optimization problem. Most of the efforts mentioned above focus on the enterprise setting and do not compare content routing with network routing.

Overlay networks provide yet another way to improve inter-domain routing performance. Savage *et al.* [37] explore the benefits of an overlay routing system, Detour, that selects best performing alternative paths. In addition to a conventional routing system underneath, Detour requires an active network of overlay nodes to improve user experience. Andersen *et al.* [13] deploy and evaluate RON, a similar overlay system across diverse locations in the Internet. Commercial content distribution networks apply similar techniques for finding the best paths to pre-cached content [31].

Less conventional proposals to improve inter-domain routing include works by Yang, Xu, and Motiwala. Yang *et al.* [47] presents a system that can increase path diversity with routing deflections. End hosts in such systems can set bits that instruct routers on the path to perform deflections over better paths. Xu and Rexford [46] introduce MIRO: a system that provides increased diversity of paths choices for inter-domain routing. Motiwala *et al.* [32] present a routing algorithm for Internet routers that enables scalable exploration of Internet path diversity. Unfortunately, utilizing such systems requires changes in Internet routers and in end hosts.

7.3 Joint Routing

A sizable body of work explores joint content and network routing in the context of intra-domain routing and overlay network routing. Killian *et al.* [25] formalize a joint optimization problem to solve content placement, content routing, and network routing in an overlay network. The authors prove that computing the optimal solution is NP-complete and introduce several heuristics that perform well in realistic overlay network topologies. Jiang *et al.* [23] compare formal models of joint routing by combining the ISP's intra-domain traffic engineering problem and a content provider's server selection problem. They use concepts of cooperative game theory and simulations of their models to conclude that giving content providers some control over network routing can yield more optimal performance for both the ISP and the content provider. Yu *et al.* [48] explore the tradeoffs between using multi-path routing and deploying more replicas in an OSP network to maximize throughput between replicas and clients. Unlike our work, the authors did not explore benefits of wide-area path diversity between replicas and clients. Concurrently with our work, Sharma *et al.* [39] describe a joint optimization problem that tackles content placement, content routing, and network traffic engineering in an intra-domain routing scenario. Using real demand traces and network topologies the authors show that unplanned strategies such as IverseCap and LRU often perform better than sophisticated optimization using historical demand data.

8. LIMITATIONS AND FUTURE WORK

In this section, we discuss some limitations of our study and directions for future work. We focus in particular on how our dataset might be made more representative.

Size of the testbed. One of the most serious challenges in evaluating the performance gains that an OSPs can attain is to have a replica set that matches that of real OSPs. This equivalence entails two primary components: (1) diversity of replica locations and (2) diversity of route choice in each location. In terms of geographic diversity, our set of replicas is comparable to the set of North American Amazon EC2 data centers, although it is much smaller than the infrastructure of a large commercial OSP such as Google or Microsoft. In future work, we plan to perform similar experiments with replicas hosted across a Tier-1 ISP backbone network; we are in the process of deploying this measurement infrastructure to allow for more comprehensive studies.

Route exploration technique. Our testbed setup limits us to exploring the routing diversity at each of our replicas that BGP `AS_PATH` poisoning can provide. Moreover, poisoning introduces undesirable routing churn. In practice, however, OSPs have a much wider variety of techniques at their disposal for controlling both egress and ingress Internet routes as described in Section 3.1.

Client set. Another challenge in emulating real-world OSP performance is obtaining a representative client set. In our case the clients are PlanetLab nodes, which are hardly a representative set of Internet end hosts. Many of the nodes we use are housed in well-connected university campuses. It does bias our client set, but it is not clear whether performance improvements—especially latency improvements—would differ with a more representative set of clients. On one hand, PlanetLab nodes might be better connected than average Internet nodes, providing greater route diversity to and from such nodes. On the other hand, PlanetLab nodes might be better-provisioned in general than typical end hosts, and, thus, hard to improve on. Less well-connected and more remote networks might see more performance improvements from joint routing.

Measurement method. A future study might also attempt to measure or approximate the overall user experience of using a particular replica and network route, perhaps approximating user experience by page load times, as has been done in previous work on Web performance [42]. Because our clients are run from PlanetLab nodes, it is not practical to instrument a browser and record the performance from each client. A promising direction for future work would be to conduct a more comprehensive study of how systems such as PECAN can improve user-perceived performance.

PECAN deployment costs. To put the performance improvement PECAN achieves into perspective, one must measure the cost needed to build a functioning PECAN system such as the one described in Section 3. These costs depend on the ease of integration between content routing systems and inter-domain traffic engineering systems. Although existing content routing systems, such as those described in Section 2, are highly programmable, the same cannot be said about today's routers. Recent advances in software defined networking (SDN) [30] indicate that this type of programmatic integration might be less costly in the future.

9. CONCLUSION

Online service providers currently perform replica selection (content routing) and wide-area route selection (network routing) independently. We introduce PECAN (Performance Enhancements with Content And Network routing), a system that performs joint content and network routing for an OSP that wishes to improve end-to-end performance to clients. We design PECAN as an extension to the content routing systems that OSPs currently use.

We evaluate the performance of PECAN on a globally distributed testbed that emulates a modern OSP by running the replicated online service on five Transit Portal (TP) sites, each offering a large choice of network paths to clients. We use 174 PlanetLab nodes as clients to estimate network performance to our replicated service. Our experiments show that PECAN reduces round trip latency by 4.3% (or nearly 5 ms) on average over simply performing content routing alone, which may translate to at least tens of milliseconds of reduction in Web page load time [41]. Finally, we find that PECAN can provide most of the potential benefit with only a few judiciously selected network paths: exploring just five sets of alternate network routes between clients and replicas in our testbed can yield 60% of the maximum possible benefit of joint routing to an online service provider. Given the increasing reliance of today's online services on even small improvements in latency, the improvements that PECAN yields over standard content routing may be warranted for certain latency-sensitive services, particularly in cases when replication itself is costly.

10. ACKNOWLEDGMENTS

This research was supported by NSF award CNS-1261357. We gratefully acknowledge our shepherd Adam Wierman and the SIGMETRICS reviewers. This work could not have been done without an army of network operators who helped us to establish and operate Transit Portal [6] testbed. We specifically want to thank Schyler Batey, Larry Billado, Michael Blodgett, Jeff Fitzwater, Scott Friedrich, Brian Parker, and Kit Patterson.

11. REFERENCES

[1] Chrome software updates: Courgette. `http://dev.chromium.org/developers/design-documents/software-updates-courgette`. URL retrieved April 2013.

[2] Cisco Performance Routing (PfR). `http://www.cisco.com/en/US/products/ps8787/products_ios_protocol_option_home.html`. URL retrieved April 2013.

[3] Internap. `http://www.internap.com/`. URL retrieved April 2013.

[4] node.js. `http://nodejs.org`. URL retrieved April 2013.

[5] PECAN measurement dataset. `https://sites.google.com/site/pecanrouting/`. URL retrieved April 2013.

[6] Transit Portal. `http://tp.gtnoise.net`. URL retrieved April 2013.

[7] V8 JavaScript engine. `http://code.google.com/p/v8/`. URL retrieved April 2013.

[8] Marissa Mayer at Web 2.0. `http://glinden.blogspot.com/2006/11/marissa-mayer-at-web-20.html`, Nov. 2006. URL retrieved April 2013.

[9] A. Akella, B. Maggs, S. Seshan, and A. Shaikh. On the performance benefits of multihoming route control. *IEEE/ACM Transactions on Networking*, 16(1), Feb. 2008.

[10] A. Akella, B. Maggs, S. Seshan, A. Shaikh, and R. Sitaraman. A measurement-based analysis of multihoming. In *Proc. ACM SIGCOMM*, Karlsruhe, Germany, Aug. 2003.

[11] A. Akella, J. Pang, B. Maggs, S. Seshan, and A. Shaikh. A comparison of overlay routing and multihoming route control. In *Proc. ACM SIGCOMM*, Portland, OR, Aug. 2004.

[12] A. Akella, S. Seshan, and A. Shaikh. Multihoming performance benefits: An experimental evaluation of practical enterprise strategies. In *Proc. USENIX Annual Technical Conference*, Boston, MA, June 2004.

[13] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris. Resilient Overlay Networks. In *Proc. 18th ACM Symposium on*

Operating Systems Principles (SOSP), pages 131–145, Banff, Canada, Oct. 2001.

[14] M. Andrews, B. Shepherd, A. Srinivasan, P. Winkler, and F. Zane. Clustering and server selection using passive monitoring. In *Proc. IEEE INFOCOM*, New York, NY, June 2002.

[15] S. Bhattacharjee, M. H. Ammar, E. W. Zegura, V. Shah, and Z. Fei. Application layer anycasting. In *Proc. IEEE INFOCOM*, Kobe, Japan, Apr. 1997.

[16] R. Bush, O. Maennel, M. Roughan, and S. Uhlig. Internet optometry: Assessing the broken glasses in Internet reachability. In *Proc. Internet Measurement Conference*, Chicago, Illinois, Oct. 2009.

[17] T. Callahan, M. Allman, and V. Paxson. A longitudinal view of HTTP traffic. In *Passive & Active Measurement (PAM)*, Zurich, Switzerland, Apr. 2010.

[18] J. Chu et al. *Increasing TCP's Initial Window*. Internet Engineering Task Force, Feb. 2013. `http://tools.ietf.org/pdf/draft-ietf-tcpm-initcwnd-08.pdf`. URL retrieved April 2013.

[19] D. Farinacci, V. Fuller, D. Oran, and D. Meyer. *Locator/ID Separation Protocol (LISP)*. Internet Engineering Task Force, Apr. 2008. Internet Draft (`http://tools.ietf.org/html/draft-farinacci-lisp-07`). Work in progress, expires October 2008.

[20] D. K. Goldenberg, L. Qiu, H. Xie, Y. R. Yang, and Y. Zhang. Optimizing cost and performance for multihoming. In *Proc. ACM SIGCOMM*, pages 79–92, Portland, OR, Aug. 2004.

[21] F. Guo, J. Chen, W. Li, and T. Chiueh. Experiences in building a multihoming load balancing system. In *Proc. IEEE INFOCOM*, Hong Kong, Mar. 2004.

[22] C. Huang, N. Holt, A. Wang, A. Greenberg, jin Li, and K. W. Ross. A DNS reflection method for global traffic management. In *Proc. USENIX Annual Technical Conference*, Boston, MA, June 2010.

[23] W. Jiang, R. Zhang-Shen, J. Rexford, and M. Chiang. Cooperative content distribution and traffic engineering in an ISP network. In *Proc. ACM SIGMETRICS*, Seattle, WA, June 2009.

[24] E. Katz-Bassett, C. Scott, D. R. Choffnes, I. Cunha, V. Valancius, N. Feamster, H. V. Madhyastha, T. Anderson, and A. Krishnamurthy. LIFEGUARD: practical repair of persistent route failures. In *Proc. ACM SIGCOMM*, Helsinki, Finland, Aug. 2012.

[25] C. Killian, M. Vrable, A. C. Snoeren, A. Vahdat, and J. Pasquale. Brief announcement: The overlay network content distribution problem. In *Proc. ACM PODC*, Las Vegas, NV, 2005.

[26] R. Krishnan, H. V. Madhyastha, S. Jain, S. Srinivasan, A. Krishnamurthy, T. Anderson, and J. Gao. Moving beyond end-to-end path information to optimize CDN performance. In *Proc. Internet Measurement Conference*, 2009.

[27] S. Lee, Z.-L. Zhang, and S. Nelakuditi. Exploiting as hierarchy for scalable route selection in multi-homed stub networks. In *Proc. Internet Measurement Conference*, Taormina, Italy, Oct. 2004.

[28] G. Linden. Make data useful. `https://sites.google.com/site/glinden/Home/StanfordDataMining.2006-11-28.ppt`. URL retrieved April 2013.

[29] MaxMind GeoIP Country. `http://www.maxmind.com/app/geolitecountry`. URL retrieved April 2013.

[30] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. OpenFlow: Enabling innovation in campus networks. *ACM Computer Communications Review*, Apr. 2008.

[31] G. Miller. Overlay routing networks (Akarouting). `http://www-math.mit.edu/~steng/18.996/lecture9.ps`. URL retrieved April 2013.

[32] M. Motiwala, M. Elmore, N. Feamster, and S. Vempala. Path Splicing. In *Proc. ACM SIGCOMM*, Seattle, WA, Aug. 2008.

[33] J. Pang, A. Akella, A. Shaikh, E. Krishnamurthy, and S. Seshan. On the responsiveness of DNS-based network control. In *Proc. Internet Measurement Conference*, Taormina, Italy, Oct. 2004.

[34] S. Radhakrishnan, Y. Cheng, J. Chu, A. Jain, and B. Raghavan. TCP fast open. In *Proc. CoNEXT*, Dec. 2011.

[35] B. Raghavan and A. C. Snoeren. A system for authenticated policy-compliant routing. In *Proc. ACM SIGCOMM*, Portland, OR, Aug. 2004.

[36] Adaptive Networking Software. `http://198.152.212.23/css/Products/P0345`. URL retrieved April 2013.

[37] S. Savage, T. Anderson, et al. Detour: A Case for Informed Internet Routing and Transport. *IEEE Micro*, 19(1):50–59, Jan. 1999.

[38] S. Seshan, M. Stemm, and R. Katz. A Network Measurement Architecture for Adaptive Applications. In *Proc. IEEE INFOCOM*, Tel-Aviv, Israel, Mar. 2000.

[39] A. Sharma, A. Venkataramani, and R. Sitaraman. Distributing content simplifies ISP traffic engineering. In *Proc. ACM SIGMETRICS*, Pittsburgh, PA, June 2013.

[40] SPDY: An experimental protocol for a faster web. `http://www.chromium.org/spdy/spdy-whitepaper`. URL retrieved April 2013.

[41] S. Sundaresan, W. de Donato, N. Feamster, R. Teixeira, S. Crawford, and A. Pescape. Broadband Internet Performance: A View from the Gateway. `http://www.ietf.org/proceedings/85/slides/slides-85-irtfopen-2.pptx`. URL retrieved April 2013.

[42] M. B. Tariq, A. Zeitoun, V. Valancius, N. Feamster, and M. Ammar. Answering "What-if" Deployment and Configuration Questions with WISE. In *Proc. ACM SIGCOMM*, Seattle, WA, Aug. 2008.

[43] S. Uhlig and O. Bonaventure. Designing bgp-based outbound traffic engineering techniques for stub ases. *ACM Computer Communications Review*, 34:89–106, Oct. 2004.

[44] H. Wang, H. Xie, L. Qiu, A. Silberschatz, and Y. Yang. Optimal ISP subscription for Internet multihoming: Algorithm design and implication analysis. In *Proc. IEEE INFOCOM*, Miami, FL, Mar. 2005.

[45] P. Wendell, J. Jiang, J. Rexford, and M. Freedman. DONAR: Decentralized server selection for cloud services. In *Proc. ACM SIGCOMM*, August/September 2010.

[46] W. Xu and J. Rexford. MIRO: Multi-path Interdomain ROuting. In *Proc. ACM SIGCOMM*, Pisa, Italy, Aug. 2006.

[47] X. Yang, D. Wetherall, and T. Anderson. Source selectable path diversity via routing deflections. In *Proc. ACM SIGCOMM*, Pisa, Italy, Aug. 2006.

[48] M. Yu, W. Jiang, H. Li, and I. Stoica. Tradeoffs in CDN designs for throughput oriented traffic. In *Proc. CoNEXT*, Dec. 2012.