PEERING: An AS for Us

Brandon Schlinker[1], Kyriakos Zarifis[1], Italo Cunha[2], Nick Feamster[3], and Ethan Katz-Bassett[1]

[1]*University of Southern California* — [2]*Universidade Federal de Minas Gerais* — [3]*Georgia Institute of Technology*
{*bschlink, kyriakos, ethan.kb*}*@usc.edu* — *cunha@dcc.ufmg.br* — *feamster@cc.gatech.edu*

ABSTRACT

Internet routing suffers from persistent and transient failures, circuitous routes, oscillations, and prefix hijacks. A major impediment to progress is the lack of ways to conduct impactful interdomain research. Most research is based either on passive observation of existing routes, keeping researchers from assessing how the Internet will respond to route or policy changes; or simulations, which are restricted by limitations in our understanding of topology and policy.

We propose a new class of interdomain research: researchers can instantiate an AS of their choice, including its intradomain topology and interdomain interconnectivity, and connect it with the "live" Internet to exchange routes and traffic with real interdomain neighbors. Instead of being observers of the Internet ecosystem, researchers become members. Towards this end, we present the *PEERING* testbed. In its nascent stage, the testbed has proven extremely useful, resulting in a series of studies that were nearly impossible for researchers to conduct in the past. In this paper, we present a vision of what the testbed can provide. We sketch how to extend the testbed to enable future innovation, taking advantage of the rise of IXPs to expand our testbed.

Categories and Subject Descriptors: C.2.6 [Computer - Communication Networks] *Internetworking*

General Terms: Design; Experimentation; Measurement

1. INTRODUCTION

Interdomain routing suffers from a range of problems. ISPs lack effective mechanisms to coordinate across boundaries [36], leading to congestion and geographically circuitous paths [51, 57]. They lack the visibility to troubleshoot effectively, contributing to long-lasting outages [28, 29, 45]. BGP, the Internet's interdomain routing protocol, can experience slow convergence [30] and persistent route oscillations [17, 54]. It lacks mechanisms to prevent spoofing [5, 27] and prefix hijacks [24, 32, 58]. Despite known problems, little has changed with interdomain routing, and there has been little impactful research in recent years.

This stagnancy in the face of known problems is in stark contrast to the rapid innovation in other areas of networking. We are in an era of remarkable changes in networking and its role in our lives, as mobile connectivity and streaming video change how we use the Internet, and advances in software defined networking and data centers change how we run networks. Even the Internet's topology has changed tremendously, with the rise of IXPs [1] and content delivery networks [11]. Yet despite these substantial changes, the underlying mechanisms and approaches of interdomain routing remain unchanged, interconnecting islands of innovation but stuck with problems we have known for years.

Progress is impeded because we have only very limited means to conduct interdomain research and lack ways to incrementally deploy experimental approaches. Most research on interdomain routing is either based on measurements of existing routes, which keeps researchers from directly observing how the Internet will respond to changes in protocols or policies, or based on simulations, the realism of which is severely restricted by the limitations in our understanding of the Internet's topology [40] and policies [26, 51]. We have such limited visibility into Internet routing that a lot of recent work on it simply reports the details of privileged datasets [1, 31] or shows how to scale measurement approaches up to the entire Internet [9, 28, 45].

To bypass the lack of control of passive route measurements and the lack of realism of simulations, we argue that we have to move from researchers being fundamentally outside of interdomain routing—and therefore focusing on measuring, modeling, and simulating it—to being full and active participants in the interdomain routing ecosystem. Previous examples of researchers participating in interdomain routing proved useful [7, 14, 37], but these researchers interacted directly with only one or a few real ISPs. This limited interconnectivity is suitable only for certain types of experiments, since it does not match the widespread peering on

HotNets-XIII, October 27–28, 2014, Los Angeles, CA, USA.
Copyright is held by owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3256-9/14/10…$15.00
http://dx.doi.org/10.1145/2670518.2673887.

the Internet today. Even this small-scale interconnectivity is beyond the easy reach of most researchers since it requires both an IP address prefix and BGP peers to announce it to. We can enable a new class of interdomain research by giving researchers the ability to design an AS, including its intradomain topology and policies and its interdomain interconnectivity, and connect it to the live Internet, exchanging routes and traffic with actual interdomain neighbors.

Towards this end, we present the *PEERING* testbed, for *Pairing Emulated Experiments with Real Interdomain Network Gateways*. It couples an emulated intradomain experiment with real interdomain peering and connectivity. In its nascent stage, the testbed has facilitated many studies that were nearly impossible for academic researchers to achieve in the past [19, 26, 29, 42, 53]. To encourage others to take advantage of *PEERING*'s unique capabilities, we present a vision of what the testbed can provide, as a step towards encouraging the innovation we believe that *PEERING* can help deliver. In addition, we sketch how to extend the testbed to enable future innovation. The key insight enabling this expansion is that the widespread peering prevalent on today's Internet makes it amenable to our testbed.

2. GOALS AND REQUIREMENTS

Our approach is to let researchers run their own experimental ASes, peer them with the real Internet, and observe the results of experiments. While this model is not suitable for some research–one cannot upgrade every Internet router to a new protocol–it models the backwards compatibility that new ideas will need to be deployed.

This section describes capabilities the testbed should offer, research that each capability enables, and existing research that would have benefited. Some of the research uses our initial version of *PEERING*.

Control of interdomain topology and routing. Researchers should have flexibility in deciding which ASes to peer with and where, and what announcements to make.
Example research. This type of route injection was the basis for influential work on BGP convergence [30]. *LIFEGUARD* used route injection to route around failures [29].

The ability to inject routes makes it possible to observe how ASes react to different routing changes. Without this ability, iPlane had to infer routing policies from preferred paths [35]. With *PEERING*, *PoiRoot* made announcements to expose ASes' routing preferences and find causes of path changes [26]. *PoiRoot* also used *PEERING* to make controlled path changes, to use as ground truth for evaluation which is hard to achieve on the Internet and was unavailable to previous work [8, 15].

Realistic, rich connectivity. Transit Portal [52] and BGP beacons [37] let researchers inject routes but only connected to a few ASes. Transit Portal had only universities as upstream providers, which made it difficult to perform experiments that required commercial Internet connectivity.

Driven by the rise of IXPs, open peering policies, and route servers [12], as well as the rise of video and corresponding desire to reduce transit costs, the Internet has moved from the simplistic hierarchy of the past to a rich peering mesh [1]. To realistically represent this setting, *PEERING* must provide widespread interconnectivity at locations around the world, so researchers can experiment from the perspective of a large AS with thousands of peers, not just a small one.
Example research. The ultimate benefit of secure BGP depends on which ASes adopt it and what policies they use; our understanding of partial deployment relies on theoretical analysis and simulations [34]. A researcher recently submitted a proposal to use *PEERING* announcements to assess adoption. BGP security depends on where announcements propagate, so a thorough study requires rich connectivity.

Control of traffic. In addition to announcing routes, researchers should be able to exchange traffic between their experiment and the real Internet.
Example research. Whereas normally one can only measure the performance of preferred paths or end-host overlays [2, 18, 49], PECAN used *PEERING* announcements to uncover alternate paths in the Internet and traffic to measure their performance [53].

Ability to deploy real services. *PEERING* should let researchers run (prototypes of) services that attract traffic. The networking research community has learned a lot from deployed systems, such as PlanetLab-based CDNs [16, 41].
Example research. Given the ability to attract and forward traffic, outage detection systems [28] could have served as the basis for outage avoidance services. Two systems used early versions of *PEERING* as the basis for real-world prototypes. `ARROW` demonstrated an incrementally deployable solution to black holes, denial of service attacks, and prefix hijacking [42]. SDX proposed an architecture for a software-defined Internet exchange, and the prototype used *PEERING* to route traffic to and from the actual Internet [19].

Control of intradomain topology and routing. In addition to control of interdomain routes and traffic, researchers should be able to define the topology, routing protocols, and policies of emulated ASes they design. Existing testbeds generally focus only on control of entire domains [4, 33, 56] or only on interacting with other ASes [37, 52].
Example research. Without a testbed with both capabilities, earlier work on the interplay between interdomain and intradomain routing used simulations [20]. In contrast, a researcher is using *PEERING* to study man-in-the-middle hijacks, in which an attacker uses BGP to intercept traffic to inspect before forwarding it to the destination [44]. Emulating an attack requires rich interdomain connectivity to successfully divert traffic, then intradomain control to experiment with approaches to return it to the destination.

Support safe research. In addition to providing researchers these various aspects of realism and control, *PEERING* should make it *easy for researchers to deploy experiments*, it should support *multiple simultaneous experiments*, and it should provide *safety*. *PEERING* should isolate experiments, so that

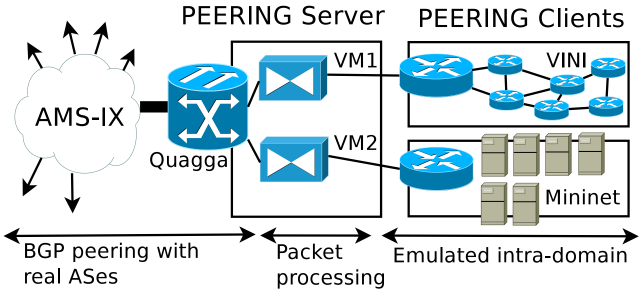

Figure 1: Overview of *PEERING* architecture.

each one can make independent routing decisions as well as send traffic and announcements without interfering with others. The testbed should protect the rest of the Internet from experiments by enforcing best practices (for example, no prefix hijacking or leaks, and only carefully controlled source address spoofing [27–29]). It should present stability to the rest of the Internet, without requiring peers to reconfigure for the coming and going of individual experiments. Careful consideration of safety will encourage acceptance and support of the testbed among network operators, which will help us attain the rich connectivity necessary for realistic experiments.

Empower research. Although previous systems achieve some of these goals (see Sec. 5), our contribution is to combine *all* of them into one platform. This combination can improve existing work—rich connectivity would give ARROW more flexibility in routing around problems and increase accuracy in *PoiRoot*—and enable new research, such as the BGP security and hijacking studies mentioned above.

3. ARCHITECTURE AND DESIGN

PEERING lets researchers experiment with interdomain routing on a global scale, running their own ASes that peer with hundreds—eventually thousands—of real ASes around the world. Our approach is to (1) deploy a real AS on the Internet; (2) rely on today's trends towards massive peering to help connect our AS to thousands of ASes around the world; (3) allow researchers to use existing tools to emulate intradomain networks and services of their choice; (4) connect these emulated networks to our global AS, allowing researchers to use it to exchange routes and traffic between their emulated networks and the real Internet. *PEERING* comprises two types of devices: *PEERING servers* exchange routes with real ASes, and *PEERING clients* connect to servers to execute experiments. *PEERING* staff operate the servers. Researchers operate clients. Fig. 1 shows an overview of *PEERING*. Below, we describe the components in more detail, focusing on how they meet the goals above.

Controlling interdomain topology and routing. *PEERING* extends the Transit Portal system [52] to provide interdomain control. *PEERING* operates an AS number and owns an IPv4 /19 prefix. *PEERING* servers run the Quagga software router to establish BGP sessions with these peers, but provide hosted experiments with full control over route announcements. A normal BGP router with multiple peers runs the BGP route selection process, chooses a best route, then exports this route to other peers. To provide researchers with control, *PEERING* servers do not run the BGP route selection process; instead, they establish one BGP session per peer with each client. These multiple sessions serve three goals: first, clients receive routes exported by each peer (instead of just the best route) and can make choices independent of other clients; second, clients can control which of their announcements go to each provider or peer; third, by "ignoring" particular peers, clients can pick and choose peers in order to emulate a particular topology. Combined, it is as if clients connect directly to peers. As always with BGP, clients cannot directly influence ASes they do not peer with.

Although we will not provide transit for non-*PEERING* destinations, clients can emulate multiple domains. Each emulated domain uses a private ASN "behind" *PEERING*, which will strip these off and present only the public *PEERING* ASN to the rest of the Internet. The emulated domains can exchange routes and traffic with each other directly or across the real Internet. An emulated domain can provide transit for real (non-*PEERING*) ASes towards a prefix announced by another emulated domain, even selecting routes that traverse real ASes in between the two emulated domains. With only our current single ASN, the configuration for certain scenarios is complex. We plan to acquire multiple public ASNs in the future to ease the deployment and flexibility of these and other experiments.

While Quagga suffices in our current deployment, it requires a single connection between client and server for each upstream peer and thus cannot support large IXPs with many peers [52]. We plan to substitute a more streamlined solution for multiplexing upstream sessions using the BIRD software router, which enables lightweight multiplexing by using BGP Additional Paths [6].

Achieving rich connectivity. We take advantage of the increasing role of IXPs [1] to provide *PEERING* clients with rich connectivity, even as this trend renders growing fractions of links invisible to traceroute and route collectors [40]. Many IXPs now offer route servers, which offer a central point for multilateral peering, sidestepping the need to establish bilateral agreements and configuration. By peering with the route server at the Amsterdam Internet Exchange (AMS-IX), *PEERING* instantly established peering with hundreds of ASes (see Section 4.1). Even among ASes that do not connect to route servers, or at IXPs that do not offer route servers, many ASes have open peering policies, meaning they are willing to peer with any other AS without restrictions. This open policy contradicts the common view that many researchers seem to hold that peering agreements usually dictate balanced traffic ratios and other requirements. Section 4.1 shows that open peering is the most prevalent policy at AMS-IX, and we will discuss how easy it is to establish these peerings. Content providers in particular tend to offer open peering. With more and more traffic coming from a few CDNs and cloud providers–YouTube and Netflix alone account for 47% of North American traffic [48]–we are able to connect *PEERING* directly to these important networks. With the growing number of IXPs and ASes present at those IXPs and the prevalence of open peering and multilateral peering, we can deploy *PEERING* routers at these IXPs and expect to attain peering with many commercial ASes.

However, maintaining a globally distributed testbed may require quite a bit of operational attention. Conveniently, the growing role of remote peering providers means that we need not maintain such a large physical deployment. Hibernia Networks offered us virtualized layer 2 connectivity from our AMS-IX server to tens of IXPs around the world.

PEERING has nine servers on three continents, dozens of indirect providers through universities, and hundreds of peers AMS-IX. Other European IXPs also provide route servers with open peering, and the largest such IXPs have opened similar facilities in the US and Asia. Some IXPs in other locations are adopting the model, and we added a server at the Phoenix-IX in Arizona in September 2014. We are expanding, with a goal of deploying servers at major IXPs and remotely peering at smaller IXPs.

Controlling intradomain topology and routing. Since we are not deploying an actual wide-area network with dedicated routers and bandwidth, we strive instead to offer flexible options such that researchers can find one that fits their needs. Researchers can use existing testbeds for intradomain experiments that we can couple with *PEERING*'s interdomain connectivity, configuring clients inside the testbed to speak to *PEERING* servers. The intradomain network can be emulated or real and could be, for instance, a software-defined network, a data center, or a wide-area network. It can include custom routing protocols and middleboxes. Some experiments may use VINI to experiment with a geo-distributed virtualized WAN [4] interconnecting *PEERING* PoPs.

For other experiments, Mininet's lightweight container-based emulation environment [33] may be appropriate, allowing fine-grained control over arbitrary topologies without the memory overhead of a virtual machine-based emulation. We developed a set of Mininet extensions to enable *PEERING* experiments. Our extension layer, Minine*X*t, makes it possible to build highly-scalable *PEERING* experiments with ease [38]. Minine*X*t offers greater isolation between containers than Mininet, and it includes building blocks for common networking infrastructure (such as Quagga) and for connecting to *PEERING*'s interdomain servers.

Controlling traffic. *PEERING* servers forward traffic sent by clients to peers and forward traffic from peers to clients via OpenVPN tunnels. Once traffic reaches clients, they can, for example, forward it through switches in a Minine*X*t emulation or across the VINI WAN. Researchers can also run lightweight code in VMs on *PEERING* servers to process packets. They can rewrite, rate-limit, or DPI traffic; coordinate with an SDN controller; or deploy services.

There are limits to the traffic *PEERING* will support. Primarily, we do not want the responsibility of carrying non-experimental traffic, and so we will only carry traffic that is coming from or destined to an experiment. Secondarily, we only support low traffic volumes, as all our traffic goes across the public Internet, and some university sites may not want us originating high volumes. This second limitation is not fundamental; in the future, we plan to interconnect some server locations with dedicated bandwidth via VINI [4], CloudLab [13], and Internet2.

Deploying real services. *PEERING* delegates control of its AS number and prefixes to clients. With access to these resources, researchers can advertise services on real IP addresses and potentially attract traffic to them, e.g., by anycasting a prefix from all *PEERING* providers and peers. Virtual machines on *PEERING* servers also let services process traffic arbitrarily. A decoy routing service [23] could take traffic at an IXP, rewrite packets, and send the modified packet back to the IXP fabric towards its new destination. The virtual machines allow flexibility but incur high overhead. Going forward, we plan to expose a lightweight packet processing API (e.g., running an OpenFlow software switch or extending Linux's iptables) to provide common packet processing capabilities to clients at lower overhead. This would free up processing power and allow execution of more services at the server.

Enforcing safety. Because servers interpose between researchers' clients and the actual Internet on both the control and data planes, they are ideally positioned to enforce safety [52]. From each client's perspective, it essentially has direct connections to the upstream and peer ASes. From the perspective of each upstream AS, the AS only connects to *PEERING*, which maintains a stable BGP session across experiments. By applying outbound filters on prefixes and origin AS and by route-flap dampening, *PEERING* prevents experiments from impacting routing for prefixes outside *PEERING* control. Clients cannot hijack or leak prefixes, and they cannot spoof traffic in uncontrolled ways.

Supporting multiple simultaneous experiments. Each experiment receives its own prefixes out of *PEERING*'s supply, isolating them from each other. *PEERING* scalability depends on the number of available prefixes. Some researchers have offered to donate IPv4 prefixes to *PEERING*'s pool, and we also plan to add support for IPv6.

Easing management and experiment deployment. *PEERING* saves researchers the burden of registering an AS, getting an IP prefix, and establishing worldwide peerings, removing obstacles that could prevent evaluation or deployment of new techniques and services in a realistic scenario. We implemented a prototype web service that lets users schedule announcements without setting up a client software router and configure connections with *PEERING*. The system will then notify researchers when their announcements will be executed so they can perform any necessary measurements. We also automatically collect regular control and data plane measurements towards *Peering* prefixes. These mechanisms lower the bar for simple experiments.

To support easy deployment of multiple simultaneous experiments, the testbed itself must also be easy to for us to operate. We have hired operational staff and are putting in place monitoring and maintenance software. We are automating many aspects of processes such as deploying new clients (allocating prefixes and establishing data and control plane connectivity to our servers), configuring new peerings, and deploying new server sites, with all the relevant data tracked in a database. Ultimately, we plan a web portal by which a researcher can request an account. We (via an advisory board) will vet experiments, at which point the provisioning will be automated, configuring servers and giving researchers the configuration they need for their clients.

4. EVALUATION

We demonstrate that our proposed approach to interdomain research is reasonable: our plan to obtain interdomain peering will feasibly result in rich connectivity, and our intradomain emulation scales to reasonably-sized networks.

4.1 Rich interdomain peering

In this section, we describe our first IXP deployment, at the Amsterdam Internet Exchange (AMS-IX). We believe that this experience is representative of peering at the large IXPs that dominate Europe and are expanding into other areas, including the US. To obtain initial peering, we sent a brief proposal asking AMS-IX to host *Peering*. AMS-IX agreed to provide us with free hosting and connectivity. We deployed a server running our Quagga-based software router at an Amsterdam colocation facility.

Obtaining peers. AMS-IX is one of the largest IXPs in the world. It has 669 member ASes and operates public route servers which any member AS can connect to in order to enable multilateral peering. *Peering* has both multilateral peers via the route servers and bilateral peers via direct peerings. Overall, 554 members peer with the route servers; we immediately obtained peering with them when our router established a BGP session with the route server. Because route servers are so popular, the connectivity *Peering* obtains is similar to that of large numbers of ASes.

We also peer with some of the 115 ASes that do not use the route server. Of these, 48 have open peering, meaning that they will peer with anyone who sends a request.[1] Establishing peering just requires a simple configuration update. We have sent requests to a few dozen ASes, and the vast majority accepted our request and configured a session, even though our requests do not include information on the project, we have not yet established a strong web presence describing *Peering*, and we do not send or receive significant volumes of traffic (so there is not a strong economic case for peering with us). One AS replied with questions about why we wanted to peer given the lack of traffic, and a handful of ASes have not responded to our requests.

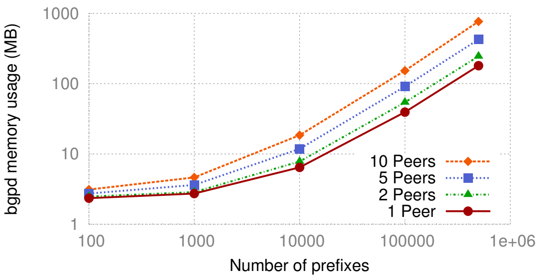

Figure 2: BGP table memory usage as # of prefixes and peers increases.

Who do we peer with? *Peering* already peers with hundreds of commercial ISPs, including such important networks as Airtel, Akamai, GoDaddy, Google, Hurricane Electric, Microsoft, Netflix, Pacnet, RETN, Terremark, and TransTeleCom. Many of the peers are from the Netherlands and nearby countries, but we have peers based in 59 countries, covering Europe, Africa, North and South America, Asia, and Australia. We peer with at least 13 of the 50 largest ASes and 27 of the largest 100, as ranked by the size of their customer cones [10]. These numbers will grow as we peer at more IXPs and establish more bilateral peerings.

Which destinations can we reach via peerings? Ignoring transit, *Peering* has AMS-IX routes to over 131,000 prefixes, one quarter of the Internet. To characterize whether we can reach important destinations, we performed DNS lookups for the Alexa Top 500 URLs. We have peer routes to 157 of them. When fetched, those 500 pages included 49,776 resources from 4,182 distinct FQDNs. We ran DNS lookups for these domain names from our AMS-IX server, resulting in 2,757 distinct IP addresses. Reflecting the fact that we peer with major CDNs and content providers, we have peer routes to 1,055 of the 2,757 addresses.

Overall, we find that it is feasible to build a testbed AS with real connectivity to many important ASes and destinations. Given the ease of attaining hosting and peering at AMS-IX, and the rich interconnectivity it provides, we are optimistic that we will obtain similar peering in other IXPs. In fact, we have already been invited to deploy in a number of IXPs and just installed a server at Phoenix-IX.

4.2 Scalable intradomain emulation

We next demonstrate that Mininе*X*t, our *Peering*-focused extensions to Mininet, can emulate reasonably-sized intradomains. We emulated the PoP-level global backbone of Hurricane Electric (HE), using data from Topology Zoo [25]. We set up a Quagga routing engine for each of the 24 PoPs, configured each PoP to originate a prefix, and configured sessions between adjacent PoPs. We then connected the emulated Amsterdam PoP to peer at AMS-IX via *Peering*, similar to how the actual HE PoP peers there. Routes from AMS-IX propagated through the emulated HE topology, and Mininе*X*t forwarded routes from emulated PoPs out to the

[1] In addition, 12 have closed policies, 40 consider peering on a case-by-case basis and 15 do not list their policies.

	PL	VN	EM	MN	RC	BC	TP	**PR**
Interdomain	✗	✗	✗	✗	✗	≈	✓	✓
Rich conn.	✓	✗	✗	✗	✓	✗	✗	✓
Traffic	✓	✓	✓	✓	✗	✗	≈	✓
Real services	✓	✓	✗	✗	✗	✗	✓	✓
Intradomain	✗	✓	✓	✓	✗	✗	✗	✓
Open/Simult. experiments	✓	✓	✓	✓	✓	✗	✗	✓

Table 1: Whether testbeds meet §2 goals (≈ means limited support). (PL=PlanetLab, VN=VINI, EM=EmuLab, MN=MiniNet, RC=Route Collectors, BC=Beacons, TP=TransitPortal, PR=*PEERING*). No two other systems can be combined to provide the set of goals *PEERING* achieves.

Internet via AMS-IX, enabling traffic to flow between emulated PoPs and Internet destinations. The emulation ran on a commodity desktop using 8GB RAM. To run even larger topologies beyond the limitations of a single host, we can connect Minine*X*t containers across multiple physical hosts.

Mininet's container-based emulation has low overhead, so the applications that run within Minine*X*t containers determine resource use. We built example topologies consisting of Quagga routers in which N peers each sent X routes to a single router. Figure 2 shows the amount of memory consumed by that single Quagga router. While the memory footprint of Internet-scale (500K) tables may seem high, (a) route reflectors and MPLS backbones mean that many internal routers do not carry multiple copies of the full table, and (b) peers do not typically export full tables. For example, at AMS-IX, only our 5 largest peers give us more than 10K routes, and 307 give us fewer than 100 routes.

5. RELATED TESTBEDS

Prior successes demonstrate the empowering impact of new testbed capabilities. RON and PlanetLab led to a flurry of work on overlays [2, 18, 35], Emulab enabled work on protocols and measurement tools [3, 23, 50], DETER created a means for security work [39,55], and Mininet inspired SDN work [21, 22]. Unfortunately, existing systems do not achieve all our goals, as summarized in Table 1 and below.

Control over interdomain routes. Most research platforms cannot interact with the Internet's control plane. For example, networks emulated in Emulab [56] or VINI [4] cannot exchange routes with the real Internet. Virtualization tools like Mininet [33] are similar. Transit Portal [52] and, to a limited extent, BGP beacons [37] provide control over interdomain routes. *PEERING* extends this type of control to richer connectivity, including supporting many peers at IXPs and more fine-grained control over announcements.

Rich connectivity. Emulation-based platforms such as VINI, Emulab, and Mininet can exchange traffic with the Internet only at emulation sites, which are one (or a few) with limited upstream connectivity. End-host platforms such as PlanetLab [43] or RIPE Atlas [46] and route collectors like RouteViews [47] have more sites and better connectivity. *PEERING* achieves rich connectivity through large IXPs and remote peering.

Control over traffic. Most platforms allow researchers to control traffic. For example, researchers can run any measurement technique on Emulab, VINI, and Mininet, or modify packets in transit. BGP beacons and collectors do not provide any support for sending active measurements on the data-plane. *PEERING* allows control of traffic at *PEERING* clients and servers.

Support for real services. Services have specific requirements, but require at least running code and available resources. Platforms that do not allow user code, like RIPE Atlas or route collectors, or ones that cannot commit resources permanently, like Emulab, ultimately cannot support real services. PlanetLab supports services such as CDNs [16,41]. *PEERING* can support real services on clients and servers.

Control over intradomain routes. End-host platforms like PlanetLab and RIPE Atlas cannot (sensibly) emulate intradomain routes. Transit Portal is similar as it forwards packets between upstream providers and downstream clients without support for intradomain research. *PEERING* allows control over intradomain routes by interfacing with intradomain emulation platforms like VINI, Emulab, and Mininet.

Openness and simultaneous experiments. Openness is a trade-off between concurrency and realism. Dedicated resources allow complete control over an experiment but prevent resource sharing and reduce scalability. Most successful testbeds (even Emulab, which provides dedicated resources) share resources and execute experiments concurrently, with different trade-offs. *PEERING* supports a client per /24 prefix and virtualizes server resources among clients.

6. CONCLUSION

Despite long-standing problems with interdomain routing, solutions have been slow to come, hindered by the inability to deploy and test alternate mechanisms in an environment that is realistic enough to be credible, controllable enough to be interesting, and safe enough to experiment. Our *PEERING* testbed lets researchers run experiments as *part of* the Internet's interdomain routing ecosystem, rather than just as passive observers or external participants. *PEERING* pairs controlled emulation with real-world interaction and deployment and, for the first time, offers researchers a means to deploy and test new interdomain routing approaches within the context of the current routing infrastructure.

Acknowledgments. We gratefully acknowledge Harsha Madhyastha, Ramesh Govindan, Minlan Yu, Valas Valancius, and the HotNets reviewers. *PEERING* would not be available without the researchers and network administrators at the sites, and the generous hosting provided by AMS-IX and Phoenix-IX. Google, NSF (CNS-1351100, CNS-1405754), and CNPq partially funded this work. We are thankful for their support.

7. REFERENCES

[1] B. Ager, N. Chatzis, A. Feldmann, N. Sarrar, S. Uhlig, and W. Willinger. Anatomy of a Large European IXP. In *SIGCOMM*, 2012.

[2] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris. Resilient Overlay Networks. In *SOSP*, 2001.

[3] M. Balakrishnan, K. Birman, A. Phanishayee, and S. Pleisch. Ricochet: Lateral Error Correction for Time-Critical Multicast. In *NSDI*, 2007.

[4] A. C. Bavier, N. Feamster, M. Huang, L. L. Peterson, and J. Rexford. In VINI Veritas: Realistic and Controlled Network Experimentation. In *SIGCOMM*, 2006.

[5] R. Beverly, A. Berger, Y. Hyun, and K. Claffy. Understanding the Efficacy of Deployed Internet Source Address Validation Filtering. In *IMC*, 2009.

[6] The BIRD Internet Routing Daemon. `http://bird.network.cz/`.

[7] R. Bush, O. Maennel, M. Roughan, and S. Uhlig. Internet Optometry: Assessing the Broken Glasses in Internet Reachability. In *IMC*, 2009.

[8] M. Caesar. Towards Localizing Root Causes of BGP Dynamics. Technical report, University of California, Berkeley, 2003.

[9] X. Cai and J. S. Heidemann. Understanding Block-level Address Usage in the Visible Internet. In *SIGCOMM*, 2010.

[10] CAIDA-ASRank. `http://as-rank.caida.org/`.

[11] M. Calder, X. Fan, Z. Hu, E. Katz-Bassett, J. S. Heidemann, and R. Govindan. Mapping the Expansion of Google's Serving Infrastructure. In *IMC*, 2013.

[12] N. Chatzis, G. Smaragdakis, A. Feldmann, and W. Willinger. There is More to IXPs than Meets the Eye. *CCR*, 2013.

[13] CloudLab. `http://www.cloudlab.us/`.

[14] L. Colliti. *Internet Topology Discovery Using Active Probing*. PhD thesis, Universita degli Studi Roma Tre, 2006.

[15] A. Feldmann, O. Maennel, Z. M. Mao, A. W. Berger, and B. M. Maggs. Locating Internet Routing Instabilities. In *SIGCOMM*, 2004.

[16] M. J. Freedman, E. Freudenthal, and D. Mazieres. Democratizing Content Publication with Coral. In *NSDI*, 2004.

[17] T. Griffin, F. B. Shepherd, and G. T. Wilfong. The Stable Paths Problem and Interdomain Routing. *IEEE/ACM Trans. Netw.*, 2002.

[18] P. K. Gummadi, H. V. Madhyastha, S. D. Gribble, H. M. Levy, and D. Wetherall. Improving the Reliability of Internet Paths with One-hop Source Routing. In *OSDI*, 2004.

[19] A. Gupta, L. Vanbever, M. Shahbaz, S. P. Donovan, B. Schlinker, N. Feamster, J. Rexford, S. Shenker, R. Clark, and E. Katz-Bassett. SDX: A Software Defined Internet Exchange. In *SIGCOMM*, 2014.

[20] N. Gvozdiev, B. Karp, and M. Handley. LOUP: The Principles and Practice of Intra-Domain Route Dissemination. In *NSDI*, 2013.

[21] B. Heller, D. Erickson, N. McKeown, R. Griffith, I. Ganichev, S. Whyte, K. Zarifis, D. Moon, S. Shenker, and S. Stuart. Ripcord: a Modular Platform for Data Center Networking. In *SIGCOMM*, 2010.

[22] B. Heller, C. Scott, N. McKeown, S. Shenker, A. Wundsam, H. Zeng, S. Whitlock, V. Jeyakumar, N. Handigol, J. McCauley, K. Zarifis, and P. Kazemian. Leveraging SDN Layering to Systematically Troubleshoot Networks. In *HotSDN*, 2013.

[23] A. Houmansadr, G. T. K. Nguyen, M. Caesar, and N. Borisov. Cirripede: Circumvention Infrastructure Using Router Redirection with Plausible Deniability. In *CCS*, 2011.

[24] X. Hu and Z. M. Mao. Accurate Real-time Identification of IP Prefix Hijacking. In *IEEE Security and Privacy*, 2007.

[25] The Internet Topology Zoo. `http://www.topology-zoo.org/`.

[26] U. Javed, I. Cunha, D. R. Choffnes, E. Katz-Bassett, T. E. Anderson, and A. Krishnamurthy. PoiRoot: Investigating the Root Cause of Interdomain Path Changes. In *SIGCOMM*, 2013.

[27] E. Katz-Bassett, H. V. Madhyastha, V. K. Adhikari, C. Scott, J. Sherry, P. van Wesep, T. E. Anderson, and A. Krishnamurthy. Reverse Traceroute. In *NSDI*, 2010.

[28] E. Katz-Bassett, H. V. Madhyastha, J. P. John, A. Krishnamurthy, D. Wetherall, and T. E. Anderson. Studying Black Holes in the Internet with Hubble. In *NSDI*, 2008.

[29] E. Katz-Bassett, C. Scott, D. R. Choffnes, I. Cunha, V. Valancius, N. Feamster, H. V. Madhyastha, T. E. Anderson, and A. Krishnamurthy. LIFEGUARD: Practical Repair of Persistent Route Failures. In *SIGCOMM*, 2012.

[30] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed Internet Routing Convergence. *IEEE/ACM Trans. Netw.*, 2001.

[31] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian. Internet Inter-domain Traffic. In *SIGCOMM*, 2010.

[32] M. Lad, D. Massey, D. Pei, Y. Wu, B. Zhang, and L. Zhang. PHAS: A Prefix Hijack Alert System. In *USENIX Security*, 2006.

[33] B. Lantz, B. Heller, and N. McKeown. A Network in a Laptop: Rapid Prototyping for Software Defined Networks. In *HotNets*, 2010.

[34] R. Lychev, S. Goldberg, and M. Schapira. Is the Juice Worth the Squeeze? BGP Security in Partial Deployment. In *SIGCOMM*, 2013.

[35] H. V. Madhyastha, E. Katz-Bassett, T. Anderson, A. Krishnamurthy, and A. Venkataramani. iPlane Nano: Path Prediction for Peer-to-Peer Applications. In *NSDI*, 2009.

[36] R. Mahajan, D. Wetherall, and T. E. Anderson. Mutually Controlled Routing with Independent ISPs. In *NSDI*, 2007.

[37] Z. M. Mao, R. Bush, T. Griffin, and M. Roughan. BGP Beacons. In *IMC*, 2003.

[38] MinineXt — Mininet eXtended. `http://mininext.uscnsl.net/`.

[39] J. Mirkovic, H. Shi, and A. Hussain. Reducing Allocation Errors in Network Testbeds. In *IMC*, 2012.

[40] R. V. Oliveira, D. Pei, W. Willinger, B. Zhang, and L. Zhang. The (In)completeness of the Observed Internet AS-level Structure. *IEEE/ACM Trans. Netw.*, 2010.

[41] V. S. Pai, L. Wang, K. Park, R. Pang, and L. L. Peterson. The Dark Side of the Web: An Open Proxy's View. *CCR*, 2004.

[42] S. Peter, U. Javed, Q. Zhang, D. Woos, A. Krishnamurthy, and T. Anderson. One Tunnel is (Often) Enough. In *SIGCOMM*, 2014.

[43] L. Peterson, A. Bavier, M. Fiuczynski, and S. Muir. Experiences Building PlanetLab. In *OSDI*, 2006.

[44] A. Pilosov and T. Kapela. Stealing The Internet: An Internet-Scale Man In The Middle Attack. In *DEFCON 16*, 2008.

[45] L. Quan, J. Heidemann, and Y. Pradkin. Trinocular: Understanding Internet Reliability through Adaptive Probing. In *SIGCOMM*, 2013.

[46] RIPE Atlas. `https://atlas.ripe.net/`.

[47] The University of Oregon Routeviews Project. `http://www.routeviews.org`.

[48] Sandvine Global Internet Phenomena Report: 1H 2014. Technical report, Sandvine, 2014.

[49] S. Savage, A. Collins, E. Hoffman, J. Snell, and T. E. Anderson. The End-to-End Effects of Internet Path Selection. In *SIGCOMM*, 1999.

[50] V. Sekar, M. K. Reiter, W. Willinger, H. Zhang, R. R. Kompella, and D. G. Andersen. cSamp: A System for Network-Wide Flow Monitoring. In *NSDI*, 2008.

[51] N. T. Spring, R. Mahajan, and T. E. Anderson. The Causes of Path Inflation. In *SIGCOMM*, 2003.

[52] V. Valancius, N. Feamster, J. Rexford, and A. Nakao. Wide-Area Route Control for Distributed Services. In *USENIX ATC*, 2010.

[53] V. Valancius, B. Ravi, N. Feamster, and A. C. Snoeren. Quantifying the Benefits of Joint Content and Network Routing. In *SIGMETRICS*, 2013.

[54] K. Varadhan, R. Govindan, and D. Estrin. Persistent Route Oscillations in Inter-domain Routing. *Computer Networks*, 2000.

[55] A. Viswanathan, A. Hussain, J. Mirkovic, S. Schwab, and J. Wroclawski. A Semantic Framework for Data Analysis in Networked Systems. In *NSDI*, 2011.

[56] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar. An Integrated Experimental Environment for Distributed Systems and Networks. In *OSDI*, 2002.

[57] K. Zarifis, T. Flach, S. Nori, D. R. Choffnes, R. Govindan, E. Katz-Bassett, Z. M. Mao, and M. Welsh. Diagnosing Path Inflation of Mobile Client Traffic. In *PAM*, 2014.

[58] C. Zheng, L. Ji, D. Pei, J. Wang, and P. Francis. A Light-Weight Distributed Scheme for Detecting IP Prefix Hijacks in Realtime. In *SIGCOMM*, 2007.