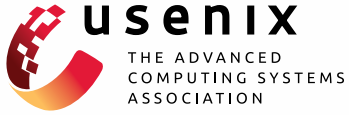

BISmark: A Testbed for Deploying Measurements and Applications in Broadband Access Networks

Srikanth Sundaresan, Sam Burnett, and Nick Feamster, ***Georgia Institute of Technology;***
Walter de Donato, ***University of Naples Federico II***

https://www.usenix.org/conference/atc14/technical-sessions/presentation/sundaresan

This paper is included in the Proceedings of USENIX ATC '14: 2014 USENIX Annual Technical Conference.

June 19–20, 2014 • Philadelphia, PA

978-1-931971-10-2

Open access to the Proceedings of USENIX ATC '14: 2014 USENIX Annual Technical Conference is sponsored by USENIX.

BISmark: A Testbed for Deploying Measurements and Applications in Broadband Access Networks

Srikanth Sundaresan,* Sam Burnett,* Nick Feamster
School of Computer Science, Georgia Tech

Walter de Donato
University of Napoli "Federico II"

http://projectbismark.net

Abstract

BISmark (Broadband Internet Service Benchmark) is a deployment of home routers running custom software, and backend infrastructure to manage experiments and collect measurements. The project began in 2010 as an attempt to better understand the characteristics of broadband access networks. We have since deployed BISmark routers in hundreds of home networks in about thirty countries. BISmark is currently used and shared by researchers at nine institutions, including commercial Internet service providers, and has enabled studies of access link performance, network connectivity, Web page load times, and user behavior and activity. Research using BISmark and its data has informed both technical and policy research. This paper describes and revisits design choices we made during the platform's evolution and lessons we have learned from the deployment effort thus far. We also explain how BISmark enables experimentation, and our efforts to make it available to the networking community. We encourage researchers to contact us if they are interested in running experiments on BISmark.

1 Introduction

A defining feature of today's Internet is the proliferation of high-speed broadband access. The United States alone has more than 245 million broadband users, and usage statistics in other regions are even more impressive: at the end of 2012, China reported more than 560 million Internet users, with a penetration rate of more than 40% [20, 21], and Africa is seeing increased penetration and plummeting costs for high-speed connectivity [22, 28].

These changes encouraged us to study the nature and evolution of Internet connectivity as many users now experience it. We aimed to deploy a platform that could support continuous measurements from long-running vantage points and allow researchers to develop, test, and deploy new systems and services for common access network environments. To support rich and accurate Internet measurements from vantage points that are characteristic of typical Internet users, researchers need a testbed that represents the perspective of the growing population of Internet users.

Our vision for such a testbed was perhaps most comparable to long-running testbeds with dedicated hardware, such as PlanetLab [2] and the RIPE Atlas Project [33]. Inspired by these projects, we decided that deploying dedicated hardware, in the form of commodity home routers, was the best way to ensure that we could perform both long-running and on-demand measurements from a consistent set of vantage points where researchers previously did not have access. Such a deployment enables measurements that are *continuous* (unlike many measurement tools, which report only a single set of measurements initiated by the user), *direct* (unlike browser or host-based measurement tools, which can often reflect the performance of the host or application rather than of the network itself) and *comprehensive* (unlike client hardware, which cannot directly measure many aspects of both home and access networks). In contrast to PlanetLab, however, our goal was to focus on broadband access networks, as opposed to research networks, thus achieving a more *diverse* set of vantage points. Moreover, in contrast to the RIPE Atlas testbed, we designed the testbed to be *extensible*, supporting custom measurements, systems, and services. We also designed the testbed with user security and privacy as a first-order concern.

To address this need, we developed BISmark, a system that allows researchers, operators, and policymakers to deploy experiments and applications that gather data about network availability, reachability, topology, security, and performance from home routers running in globally distributed access networks.

Beyond the conventional challenges of operating a long-running service in the wide-area Internet (*e.g.*, PlanetLab), we faced a unique set of challenges when deploying such a testbed in *home networks*. First, incentives do not naturally align: whereas in PlanetLab, researchers have an incentive to host machines to gain access to the testbed, BISmark explicitly targets home users, who may not necessarily be interested in networking research. Second, unlike in universities where PlanetLab nodes are deployed, technical support is not readily available, which makes system robustness, remote maintenance, and recovery even more important. Third, nodes must be small and easy-to-deploy;

*These authors contributed equally to this paper.

such nodes are typically resource-constrained. Finally, BISmark nodes are on the direct path of real Internet users; a malfunctioning BISmark device not only disrupts a normal user's Internet connectivity but also typically results in the loss of the device. (At this point, a user is likely to remove the device from the network entirely and never re-install it.) Therefore, BISmark must also ensure that an unstable device or a poorly designed experiment cannot wreak havoc on the user's Internet connection.

BISmark has enjoyed reasonable success in its first four years. It has enabled the publication of many studies from broadband access networks from around the world, and is now being adopted by major ISPs, policymakers, and researchers in several countries. Many research groups are either using the data we have collected or deploying custom experiments. Yet, enabling a broader set of experiments and scaling BISmark beyond its current size poses new challenges. Security and robustness remain paramount, and device deployment and attrition are an uphill battle, particularly in certain regions. This paper discusses the constraints we have faced (and continue to face) in the design, implementation, and deployment of BISmark, discusses lessons and things that we would have done differently (or will change in the future), and describes new challenges as the platform expands both in terms of the number of vantage points and the diversity of experiments we aim to support.

The two audiences who will find this paper most useful are (1) designers and developers of network testbeds, who can read about BISmark's architecture and deployment lessons in Sections 3 and 5, respectively; and (2) researchers who want to use BISmark to collect measurements for their own work, who can read about how other researchers have used BISmark in Section 4. Anyone who is interested in deploying user-facing testbeds or measurement systems in hardware or software may learn from our experiences. More generally, we hope that anyone who has ever grappled with building a testbed—or plans to do so in the future—will take important lessons from our experience, many of which surprised us, and still others that may seem obvious in hindsight but are nonetheless well worth codifying.

2 Related Work

Fixed server or gateway deployments. PlanetLab [31] is probably the most similar platform to BISmark in that it aims to be a fixed, large-scale deployment hosting a variety of research experiments. Because BISmark is deployed in home networks on resource-constrained devices, however, it faces additional challenges. The RIPE Atlas [33] project has deployed thousands of probing devices worldwide, but their capabilities are limited to simple measurements (*e.g.*, ping, traceroute). SamKnows [35] has deployed thousands of home routers in the US and the UK, but only supports limited performance measurements.

Host-based deployments. Dasu [36] is a host-based software client. It has a very large footprint (tens of thousands), and allows a variety of network measurements from end hosts. However, its advantages of scale comes at the cost of decreased flexibility: (1) it lacks the permissions to run certain measurements due to application restrictions, (2) it cannot run continuous measurements (*i.e.*, since hosts can be turned off, moved, etc.), and, (3) the measurements can reflect limitations of the host or the application taking the measurement and thus do not reflect the performance of a fixed network vantage point. The Grenouille project in France [17] measures the performance of access links using an agent that runs from an end host inside the home network. Neti@Home [38] and BSense [3] also use this approach, but with fewer users than Grenouille. Network Diagnostic Tool (NDT) [9] and Network Path and Application Diagnostics (NPAD) [25] send active probes to detect issues with client performance. Glasnost [16] performs active measurements to determine whether a user's ISP blocks BitTorrent traffic.

Netalzyr [23] lets users conduct a series of tests from a browser, but measurements are not continuous, and researchers cannot run custom tests from a set of hosts—the measurements collected are fixed, and the set of hosts from which measurements are collected depend on the users who decide to run the tool.

Programming frameworks. The process of vetting BISmark experiments is manual (as it was in previous testbeds such as RON [1]), which will be a limiting factor as the deployment grows. BISmark must ultimately strike a balance between flexibility (allowing researchers to specify experiments) and a constrained programming environment (limiting researchers from specifying experiments that could interfere with home users). Previous work on sandboxed, programmable measurement environments, such as Seattle [8] or ScriptRoute [39], could ultimately serve as a useful environment for specifying BISmark tests.

Other measurement studies of broadband access networks. Previous work characterizes access networks using passive traffic measurements from DSL provider networks in Japan [11], France [37], and Europe [24]. Siekkinen *et al.* [37] show that applications (*e.g.*, peer-to-peer file sharing applications) often rate limit themselves, so performance observed through passive traffic analysis may reflect application rate limiting instead of the performance of the access link. Other studies have characterized access network performance by probing access links from servers in the wide area [12, 13].

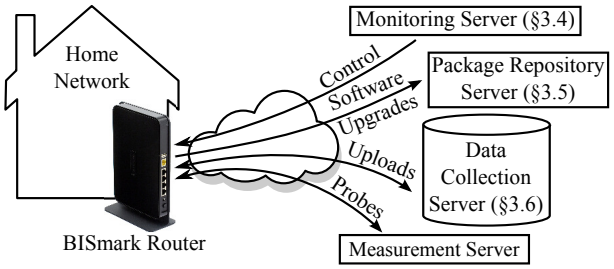

Figure 1: *BISmark architecture. The packages repository server and data collection server scale to multiple instances. The monitoring server is harder to scale, but also sees less load.*

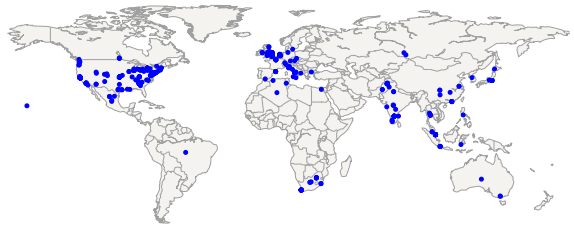

Figure 2: *Locations of the 178 BISmark routers that were online in January 2014. We have focused concentrations of routers in the US, South Africa, Pakistan, and the UK.*

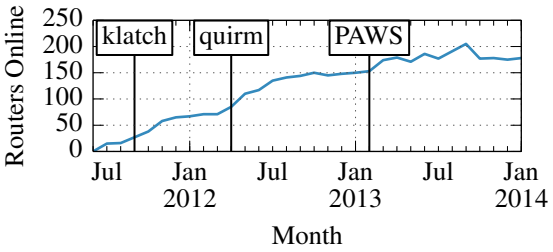

Figure 3: *The number of routers online during each month. BISmark has grown to nearly 200 routers over the past two and a half years.* klatch *and* quirm *signposts indicate two firmware releases;* PAWS *indicates a new deployment in the UK. Growth falters in some later months because we focused deployment in developing countries, where router availability is inconsistent. Numbers dropped in late 2013 when the PAWS study ended.*

3 Architecture and Implementation

BISmark aims to enable research and experimentation under constraints inherent to home routers and networks. Many of the challenges that we faced are not unique to our deployment, but they are exacerbated by operating (1) in a resource-limited setting on home routers; (2) in a setting where downtime (or general interference with users' Internet connectivity) is not acceptable.

BISmark's software fulfills four roles. First, it uniquely identifies each router and correlates it with metadata useful for conducting networking research. Second, it manages software installation and upgrades, which lets us fix bugs, issue security patches, and deploy new experiments after we have mailed the routers to participants. Third, it provides experiments a common, easy-to-use, and efficient way to upload data to a central collection server. Finally, it enables flexible and efficient remote troubleshooting. We describe BISmark's evolution path, its components, and the various roles that the BISmark software plays.

3.1 System Components

Figure 1 shows BISmark's architecture. The deployment currently comprises BISmark routers and a collection of servers that manage software, collect data, and facilitate troubleshooting.

BISmark routers. As of January 2014, the deployment has 178 active routers in over 20 countries. Figure 2 maps router locations and Figure 3 graphs the deployment's growth over the past two and a half years. We purchased, prepared, and mailed nearly half of these routers, while the rest have arisen either organically (*e.g.*, as users "flash" their own routers with BISmark firmware) or through coordinated efforts (*e.g.*, with other organizations or research groups). We currently deploy Netgear WNDR 3800 routers, which have a 450 MHz MIPS processor, 16 MB of flash storage, 128 MB of RAM, 5 gigabit Ethernet ports, and a dual-band wireless interface. This hardware is limited, even when compared to other embedded mobile devices like smartphones, yet it is powerful enough to reliably support both basic routing features and a variety of measurement experiments.

We replace the router's default software with a custom version of OpenWrt Linux [29]. OpenWrt has an active developer community, a simple and usable configuration GUI, and broad, mature hardware support. It frees us from maintaining our own firmware, but ties us to its release cycles, bugs and all. Despite a few persistent problems that have occasionally affected some users, we have been satisfied with OpenWrt. We have deployed two hardware revisions of Netgear routers and four firmware releases.

Some users download the BISmark firmware from our project page and install it on their own hardware. This enables further growth, but introduces the challenge of determining the identities of these users. To handle this scenario, our latest firmware includes a user registration system that prompts users to complete a simple registration process after configuring network settings. Registration serves two purposes: it automates the previously manual collection of metadata from users about their ISP; and it binds the user's identity to a router so we can restore the router configuration each time the router is reflashed.

Section 5.4 discusses the security implications of letting users install BISmark on their own hardware.

Management servers. To support the router deployment, BISmark has three types of servers:

1. *Package repository servers* decide which software should run on each router. Different routers can run slightly different sets of software because we've deployed several firmware versions and users have consented to run various experiments.
2. *Data collection servers* validate, store and serve data gathered by routers. We serve publicly accessible active measurements data from Amazon S3 and mirror all data in servers at our university.
3. *Monitoring servers* track availability and can initiate SSH connections to routers for troubleshooting.

Measurement servers. BISmark uses a fixed set of measurement servers against which it conducts standard performance measurements (*e.g.*, throughput). The validity of these experiments in many cases depends on having measurement servers that are geographically close to the deployed routers. We have been fortunate to obtain access to the globally distributed Measurement Lab (MLab) infrastructure [26]. In some cases where we have a critical mass of routers, we have also deployed additional measurement servers. Measurements are scheduled on the measurement servers by a central scheduler. This prevents overloading of servers by several concurrent requests from BISmark routers. We note that this infrastructure is used for intensive active tests such as throughput measurements. Other experiments that do not rely on a low-latency, globally distributed infrastructure do not use these servers.

3.2 Architectural Constraints

Like many rapidly growing systems, BISmark evolved organically in response to use. Several components written for an early pilot deployment persist. Although many design choices were sub-optimal in retrospect, the software has always addressed three practical constraints.

Constraint 1: *Severely limited client resources dominate software design decisions.*

Resource limitations preclude several conveniences. For example, we cannot run heavy scripting languages like Python or Ruby; instead, we glue together standard UNIX utilities and small C programs with shell and Lua scripts.

Constraint 2: *The basic routing functionality of BISmark routers is critical; users often place them on the home network's forwarding path.*

The router should not noticeably affect the user's networking experience (*e.g.*, by frequently saturating the uplink). Combined with limited client resources, this constraint requires us to thoroughly test software before deploying it, because the consequences of malfunctioning software may be the potentially terminal loss of a deployment site. (In our experience, most users simply unplug the router at the first annoyance and never plug it in again.)

Constraint 3: *User intervention is impractical and should be as limited as possible.*

Users expect their router to "just work". They have no desire to otherwise interact with it. After installation, attempts to interact with users via the router itself are awkward and annoying (*e.g.*, captive portals) and out-of-band communication (*e.g.*, email) is unreliable.

3.3 Naming

We assign each router a unique *router identifier*, which we use for data analysis, troubleshooting, and inventory; we correlate this identifier with all measurements we collect from the router, participant-provided details about the upstream ISP's advertised performance, the router's geographic location, and the participant's name and mailing address (used to ship the router). We do not disclose personal information except when required by law enforcement [5] (a scenario that we have not yet encountered).

Constraint 4: *Router identifiers must be (1) unique and (2) persistent across reboots and reflashes.*

Common identifiers such as manually assigned hostnames, dynamically generated tokens, or public IP addresses do not satisfy these requirements. Instead, we use the routers' LAN-facing MAC address, which is both unique and unchanging. We chose LAN (rather than WAN) addresses because they are printed on the back of the router, which simplifies technical support and inventory.

Unfortunately, LAN-facing MAC addresses pose a security risk because attackers could use them to geographically locate a router. By default, routers broadcast their MAC address to WiFi devices in the vicinity, including smartphones and collectors for Google's Street View and similar data collection projects. An attacker with access to both the router's MAC address and these databases could geolocate a router [43]. This vulnerability highlights a broader set of tradeoffs BISmark makes between privacy and transparency; Section 5 discusses these tradeoffs.

3.4 Troubleshooting

Remote access allows us to fix critical problems that would otherwise require a lengthy packaging and software update cycle to fix.

Constraint 5: *We need a fast but secure technique to log in to routers on demand.*

Every BISmark router runs an SSH server which is only exposed on the LAN interface. Exposing the server on the WAN interface has security concerns; additionally, over 60% of our routers are obscured by at least one layer of Network Address Translation (NAT), rendering the

SSH server on WAN useless. Instead, BISmark routers poll the *monitoring server* for SSH session requests by sending small UDP probes ("heartbeats") once per minute. If the server wishes to initiate an SSH session with a router, it responds with a UDP response to that effect; the router then opens an SSH tunnel forwarding a port on the monitoring server to the router's local SSH server. Administrators on the server then initiate SSH sessions to the forwarded server port. Although session requests and responses may be lost in the network, we can usually establish a tunnel within one minute and almost always within five minutes.

Tunnel creation uses restricted SSH authorization to prevent routers from executing arbitrary commands on the server. Because the server does not authenticate the UDP probes themselves (only the resulting tunnels), we rate limit requests to prevent denial-of-service or reflection attacks against the server or unsuspecting routers.

The overhead of the tunneling protocol is minimal. Each probe/response pair is 139 bytes, resulting in an overhead of 200 KB per day or 6 MB per month. The DNS time-to-live on the monitoring server is 15 minutes, resulting in an additional overhead from DNS lookups of less than 1 MB per month. Using a domain name with a reasonable TTL allows us to quickly migrate the monitoring server in an emergency. Although a DNS hijacker could direct routers' probes to a different address, such an attack will not compromise security of the routers because they use SSH to establish tunnels.

3.5 Software Upgrades

After we have deployed a router, we must be able to manage its software packages. Throughout the lifetime of the deployment, we have issued many bug and security fixes as package upgrades, deployed new measurement experiments by installing new packages, and rolled back faulty experiments via package removal. OpenWrt's built-in *opkg* lacks several features and safeguards necessary for managing software on a large deployment in the homes of non-technical users. This section illustrates several of BISmark's unique software management constraints and we overcome them.

Constraint 6: *Router state should be centrally managed and controlled.*

Exogenous events can interfere with stateful package management. For example, if a user resets their router to its original configuration, the router should automatically install and upgrade the software it had prior to the reset. Instead of executing one-time commands sent by a server, the router downloads a list of packages reflecting the current desired state of the router, and installs, removes, and upgrades its packages accordingly.

We built custom package management utilities on top of *opkg* that meet these constraints. We eschewed off-the-shelf tools like Puppet, Chef, or CFEngine because of resource and complexity constraints, which reflects a general tension between custom and commodity software that we faced throughout the project. Existing software is often both more complicated than we need and untested on non-x86 architectures. Sometimes writing a small custom package from scratch for limited functionality is easier than porting and rigorously testing an existing one.

Constraint 7: *Software package management must occur without intervention from either home network users or the BISmark administrators.*

This constraint contrasts with large-scale software administration frameworks for other platforms (*e.g.*, Android, Mac OS X, and Windows), which have at least some user interaction. Home users are often non-technical or otherwise have little desire to administer their router, and the BISmark administrators cannot manually run package commands on hundreds of routers. Therefore, software installation, upgrade, and removal must happen automatically. Routers in our deployment automatically perform these tasks at boot and approximately every twelve hours.

Constraint 8: *The consequences of accidental installation, upgrade, or removal of packages are high.*

Automating package management increases risk, because a single faulty or buggy package could cripple the entire deployment. Our management tools impose several restrictions to guard against accidental installation, upgrade, and removal of packages. First, routers only allow removal of packages that are not included with their base firmware image, which prevents us from accidentally removing critical software. For example, before instituting this restriction an errant experiment accidentally removed the `libc` package from a router during testing, rendering it unbootable. Second, packages must be explicitly tagged as "upgradable" on the server before routers will upgrade them, which helps prevent us from accidentally pushing newer, incompatible versions of packages. Finally, we require all new packages and versions be tested for several days on a small set of *canary routers* owned by members of our research group before wider deployment.

Routers usually sit on a home network's critical path and must continue functioning at all times. Fortunately, basic function only relies on a few core packages, and almost always continues to work even if our management tools fail. Even if we lose access to the router entirely, the home router's core functions are typically undisturbed, since those functions are isolated from our packages.

Constraint 9: *Routers have diverse packages and versions.*

There are three versions of the BISmark firmware currently deployed; each requires a different set of packages because of library incompatibilities. Some routers also

run additional packages because their users participate in optional BISmark measurement experiments. Our package management tools must install the correct packages and versions on each router. The consequences of mistakes range from broken functionality (*e.g.*, missing measurements) to unauthorized collection of data (*e.g.*, if we collect passive measurements from users who have not consented to it.)

3.6 Data Collection

Experiments generate data of a wide variety of sizes and rates and upload it to data collection servers for analysis. We initially used off-the-shelf file synchronization tools to upload this data, but resource, flexibility, and reliability constraints motivated us to develop custom software. *bismark-data-transmit* is our client application, which detects new files in a filesystem subtree using Linux's *inotify* filesystem monitoring interface. The client uploads these files over HTTPS to a Web application that validates and archives the data.

Constraint 10: *Bandwidth is scarce and may be capped.*

Efficient use of bandwidth is crucial because BISmark routers share uplink capacity with the home network. Experiments can generate lots of small files, often frequently, but need to upload them with minimal protocol overhead. Routers store these files on a volatile RAM disk and upload files as soon as possible; both excessive wear and extreme scarcity prevent them from storing data on persistent flash storage. To minimize the risk of data loss when they lose power or reboot, routers do not batch files for more efficient transmission. To avoid frequent and expensive handshakes while still continuously uploading files, we send all files over a single HTTPS connection with a high keep-alive timeout. Although Web services generally use keep-alive timeouts of only a few seconds, *bismark-data-transmit* sets a one hour timeout, which makes sense for communicating with a small, fixed set of clients.

Constraint 11: *Network and power are unreliable in some locations; data collection should gracefully handle uplink outages and power outages.*

Some experiments measure properties of the home network itself rather than the home's Internet connectivity and thus continue generating data even when the router's uplink is offline. These experiments can quickly accumulate a lot of data if unchecked. *bismark-data-transmit* retries failed uploads every three minutes and starts discarding data in FIFO order once it has more than 24 MB queued for upload, so even routers connected to very unreliable uplinks can contribute data without exhausting their limited onboard storage. Some routers frequently lose power, particularly those deployed in developing countries. *bismark-data-transmit* minimizes data loss in these cases by uploading data as soon as possible. Routers can lose data accumulated during a long uplink outage that immediately precedes a power outage.

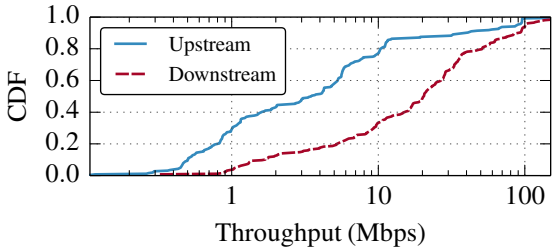

Figure 4: *Downstream and upstream throughput for routers.*

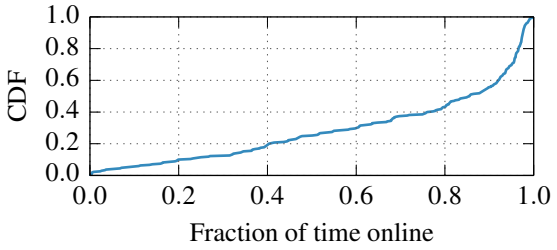

Figure 5: *Distribution of the fraction of time each router is online during its deployment. We only include routers that have been online for at least a month.*

4 Experimentation on BISmark

This section describes research projects that have used BISmark. We first describe the modes of collaboration that we have used since opening BISmark to external researchers in mid-2013. Because the platform is both resource constrained and on many users' critical path to the Internet, experiments on BISmark must cope with stricter conditions than most existing testbeds that support long-running deployments (*e.g.*, PlanetLab). For example, experiments must deal with nodes that have highly variable connectivity and availability. Figure 4 plots the 95^{th} percentile of throughput of homes in the deployment; we see ranges from basic broadband (about 1 Mbps) to fiber speeds (100 Mbps). Figure 5 shows the fraction of time a router is available and online during its lifetime; about 50% of the routers are available more than 90% of the time, but a significant fraction of routers have much patchier availability.

4.1 Modes of Collaboration

Data from many active measurements are public for anyone to use. Additionally, we have been advertising BISmark to collaborators and encouraging them to run experiments on the deployment. Most of this recruiting has been through word-of-mouth, as we build confidence that we can support a larger group of researchers (in fact, we an-

ticipate that we would be unlikely to satisfy all requests). In many of these cases, we have informally adopted a PlanetLab-like incentives model by asking the researchers to spearhead their own small deployment of BISmark routers in an ISP or region of interest. In certain research projects, the researchers want to do this anyhow because they have a specific group of users that they want to study. We have two modes of collaboration, which we outline below. In both cases, researchers must be comfortable with OpenWrt and embedded platform development.

Public deployment. Collaborators run experiments on the main deployment of routers, which we manage. We control access and schedule the experiment to run in conjunction with other experiments running on the deployment. This mode works well for researchers running lightweight experiments from the variety of vantage points that our deployment offers. We have enabled research from the University of Southern California in this way.

Private deployments. Researchers (or, in some cases, operators) purchase and deploy their own routers, while we provide the client software and manage back-end services. In these cases, the researchers retain a high degree of access to their routers, giving them an incentive to keep their deployment running. This mode is best for researchers who want to run complex or time-consuming experiments in a small geographic area. For example, University of Cambridge has deployed more than 20 BISmark routers in under-privileged communities in Nottingham to study the mechanics of Internet sharing in such communities. We have also engaged with several ISPs who have wanted to run their own autonomous deployments.

4.2 Research Projects

BISmark offers the ability to study poorly understood or understudied aspects of home networks, including access link performance, application characterization, user behavior patterns, security, and wireless performance. Table 1 summarizes several experiments we are coordinating on the deployment. In many cases, we are leading (or have led) the study ourselves; more recently, we have been collaborating with the researchers who are leading the study. The latter projects are works-in-progress. The following sections describe both sets of projects in more detail. Our discussion of experiments that have been run on BISmark is not exhaustive but is intended to highlight both the capabilities and shortcomings of the platform.

4.2.1 Performance Characterization

BISmark's location at the hub of the home network lets it gauge performance of both local devices and the access link without being affected by confounding factors from the rest of the network.

Broadband performance in the US and abroad. The home access point is ideally suited for measuring access link characteristics. We characterized access link performance and the effects of access technology and customer premise equipment in the United States [40] using data from BISmark and the similar FCC/SamKnows deployment. Our study showed, amongst other things, how the access link can have a significant impact on end-to-end performance. Research ICT Africa (RIA) reproduced our study in South Africa [10] and expanded it to include mobile devices and 3G dongles. BISmark is well suited for such studies because of its view into the last mile, and its ability to account for confounding factors and to run longitudinal experiments.

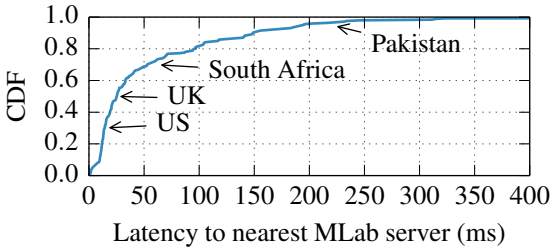

Figure 6: *Distribution of latencies to the nearest MLab server from each BISmark router, with annotations of the median latency from several countries with many routers. Latencies from Pakistan are very high because the nearest server is in Europe.*

Application performance. Because hardware limitations can prevent us from running full applications (*e.g.*, Web browsers), we aim to emulate applications' network behavior. In our work measuring network bottlenecks in Web performance [41], we used a custom browser emulator to measure one aspect of Web performance—the impact of the last mile. This study found that Web performance becomes bottlenecked on latency for broadband connections faster than 16 Mbps. Although BISmark was suitable for this particular experiment, we did not measure other aspects of Web performance, such as user perception, the effect of object ordering, or scripting on performance. This is because we cannot run a full browser on the router; it is also difficult to get such information from passive data. In such cases, we have to carefully design the experiment so that we know what we are able to study.

Wireless performance. We are developing techniques that isolate the source of performance bottlenecks to either the access link or the wireless network, as well as tools that help us understand the nature of wireless pathologies. The home access point sits between two common sources of performance issues—the access link and the wireless network—and is therefore ideally suited for identifying and isolating problems between these locations.

Lessons and caveats. As demonstrated above, BISmark is ideally suited for access link and home network characterization because it lets us probe these components and

Project	Institution(s)	Description	Publications
	Performance Characterization		
Broadband performance	Georgia Tech, University of Napoli, INRIA, FCC/SamKnows, Research ICT Africa, National University of Sciences and Technology	Study factors affecting broadband performance in the US and in developing countries.	[10, 40], WiP
Web performance	Georgia Tech, INRIA	Characterize and mitigate last-mile bottlenecks affecting Web performance.	[41]
Home wireless performance	Georgia Tech	Study home wireless pathologies and bottlenecks in home networks	WiP
	Usage and Home Network Characterization		
Home network characterization	Georgia Tech	Understand usage and connectivity.	[18]
Home Constant Guard	Comcast	Expand Constant Guard to provide information about devices infected in home networks.	WiP
PAWS	University of Cambridge	Internet sharing in underserved communities.	WiP
	Topology and Connectivity Characterization		
Google cache measurements	University of Southern California	Study effects of Google's cache deployment on performance of Web services.	[7], WiP
Network Connectivity	Georgia Tech, USC, RIA	Characterize ISP connectivity and path inflation in Africa.	[19]
Network outages and DHCP	University of Maryland	Study effects of outages on IP address allocation worldwide.	WiP
OONI/censorship	NUST, University of Napoli	Study the extent and practice of censorship in various countries (initial focus on Pakistan).	WiP

Table 1: *Summary of various experiments (and publications) that BISmark has enabled to date. "WiP" denotes work in progress.*

collect passive data from them. Application performance characterization is harder. Applications (or their emulations) must be light enough to run on the router; this might preclude certain types of applications.

Experiments that measure the access link by sending active probe traffic (*e.g.*, throughput tests) must not degrade performance of the home network while doing so. For users with bandwidth caps, probe traffic and data traffic (from uploading measurements to the server) should not constitute a significant fraction of the cap without the user's knowledge or consent. Some measurements such as TCP throughput require server deployments with low latency. Fortunately, Measurement Lab's global server infrastructure has helped: BISmark nodes automatically select the nearest MLab node for throughput measurements; Figure 6 shows that over 80% of nodes are within 100 ms of a measurement server.

4.2.2 Usage and Home Network Characterization

Several projects leverage BISmark's view of the home network behind the NAT.

Home network availability, usage, and infrastructure. We study the kinds of devices home users use to access the network, how they access the network, and their usage patterns [18]. Our study found interesting behavioral patterns (*e.g.*, users in developing countries turn off home routers when not using them) and usage patterns (*e.g.*, most traffic is exchanged with a few domains). Most prior studies are incomplete because they rely on one-time or infrequent probes from clients with limited network visibility.

Home network security. The ability to isolate traffic from different devices behind the NAT can be used to improve security. Comcast offers a security solution called Constant Guard, which captures DNS lookups to suspicious domains to inform a user when devices in their home may be compromised. We are extending BISmark to let Constant Guard identify infected devices and redirect some or all flows from suspected devices through Comcast security middleboxes via a virtual private network [4].

Internet usage in underprivileged communities. The PAWS project [30] distributes BISmark routers augmented with extra measurement tools to broadband customers who volunteer to share their high-speed broadband Internet connection for free with fellow citizens. The project studies how underprivileged communities share Internet access.

Lessons and caveats. BISmark's view into the home network and its ease of deployability allows it to run experiments that are not possible with other platforms. However, it also raises new concerns. Resource constraints limit the amount of data that can be collected and processed on the device. User privacy is a significant concern; for any experiment that studies user behavior, we must obtain informed consent from the user, which can be a slow and cumbersome process. We have conducted our own experiments that have required institutional review board (IRB) approval, but complications arise when BISmark serves as a host platform for experiments run from other universities that are sometimes in other countries. Even with permission to collect personal information, we design our experiments to collect only information necessary to answer targeted questions.

4.2.3 Connectivity Characterization

BISmark's worldwide presence lends itself to measuring many different aspects of Internet connectivity.

Measuring Internet topology and connectivity. We have looked at Internet availability in developed and developing countries [18], and researchers at USC are using BISmark to extend their study of the effects of Google's expanding cache deployment [7] on the performance of various Web services. Researchers at the University of Maryland are analyzing BISmark's UDP "heartbeat" logs (Section 3.4) to understand the effects of network outages on DHCP address allocations. Our recent work explores correlated latency spikes in ISPs [34] and the extent to which interconnectivity (or lack thereof) at Internet exchange points contributes to latency inflation and degraded application performance [19].

Global measurements of censorship. BISmark routers represent a unique opportunity to collect detailed, longitudinal data about how countries engage in censorship. Researchers in Pakistan have deployed BISmark routers in several homes to measure this phenomena; routers in other countries could also potentially collect similar measurements. We are replicating OONI [14] on BISmark.

Lessons and caveats. BISmark is well-suited for connectivity measurements because of its geographical footprint, availability, and its ability to run periodic measurements (time scale of minutes) over a long period of time (months, or even years). We could also run probes and tests between BISmark routers, but while some such experiments may be better run on platforms like Dasu which will likely always have more deployment sites, BISmark is a full-featured (if low-powered) Linux machine that offers the ability to perform a much larger variety of experiments. Experiments that measure censorship have additional ethical concerns because it is illegal in some countries, and may even place the household at risk. In these cases, we must obtain informed consent, which may be difficult with users who flashed their own hardware or don't speak English.

5 Lessons

This section summarizes lessons we have learned (often the hard way) during BISmark's development.

5.1 Recruiting Users

Convincing users to deploy routers in their homes, particularly custom hardware, is not easy. Prior to our current deployment of commodity routers, we conducted a yearlong pilot study with the NOX Box, a small form-factor PC that ran the NOX OpenFlow controller on Debian Linux. We assembled the hardware from an ALIX 2D13 6-inch by 6-inch board with a 500 MHz AMD Geode processor, 256 MB of RAM, 2 GB of flash memory, three Ethernet ports, and a wireless card. Although the NOX Box's relatively unconstrained hardware and full-featured Linux distribution helped rapid prototyping, our pilot faced several practical problems in the field.

(a) *NOX Box* **(b)** *Netgear WNDR 3800*

Figure 7: *We used the NOX Box for our pilot deployment and the Netgear WNDR3700/WNDR3800 for the second deployment. Unlike the NOX Box, the Netgear router looks like standard home networking equipment.*

Lesson 1: *Form factor matters. Users often trust commodity hardware over custom hardware simply because it is in a recognizable form.*

Figure 7 compares the NOX Box hardware from our pilot phase to the commodity Netgear hardware from our current deployment. The NOX Box doesn't look like a typical home router: it lacks familiar branding, lacks labels for status lights and Ethernet ports, and has a metal rather than plastic enclosure. These factors bred an inherent distrust of the NOX Box. People were generally more willing to deploy commodity hardware.

Lesson 2: *Users often blame BISmark for problems in their home network, whether or not the problem was caused by BISmark. Many users react by removing the router permanently from their network.*

Even with commodity hardware, users have heightened awareness of the BISmark router, particularly the experimental nature of the device, and therefore suspect it first when problems arise with their home network. In some cases, BISmark is indeed at fault. For example, a firmware bug causes unstable wireless connectivity on some devices, notably Apple MacBooks. In other cases, the router uncovered buffering problems elsewhere in the home network, temporarily degrading network conditions in the process. Many times, users misconfigured the router themselves (*e.g.*, by changing firewall settings) or incorrectly blamed BISmark for upstream ISP outages or problems with end hosts (*e.g.*, older devices that lack support for WPA2.)

Regardless of the causes of these problems, many users "solve" them by removing the BISmark router from the network. This has consequences in terms of money (the router likely will not be turned on again) and time (in flashing, packaging, and shipping the router to the user).

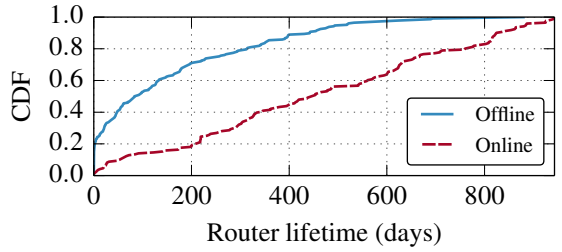

Figure 8: *Distribution of router lifetime, the difference between the first and last time we saw the router online. For 178 routers currently online, it is how long they have been online to date. 169 are now offline, while 31 never turned on at all.*

Lesson 3: *Home users and researchers have vastly divergent incentives. Home users want a working network, and researchers want to gather data and information. Care and effort must be invested to align these incentives.*

It is critical that our deployment strategy allows us to finance and maintain a large number of routers. PlanetLab's incentive structure (*i.e.*, hosting infrastructure for the right to run deployment-wide experiments) does not work in our case, because many of the most interesting vantage points will be hosted by users who are not networking researchers and have no interest in conducting their own experiments. We use two deployment strategies:

Free (or subsidized) router distribution. Our initial strategy was to ship routers to acquaintances, friends of friends, and through targeted advertising in venues such as NANOG and Internet2. It is difficult and time consuming to track routers in such cases, particularly when users turn them off. Due to the cost and effort involved, individual shipments only work at relatively small scales. About 50% of routers we distributed have either never been turned on or have since been decommissioned by their users.

Federated distribution. We are now attempting a federated deployment model to expand our geographical footprint. We work with a local contact who buys or receives a shipment of routers from us, recruits volunteers and ensures that routers stay up. This approach worked well for a deployment of approximately 15 routers in South Africa and 20 routers in the UK, and we are now attempting similar approaches in Tunisia, Pakistan, Nigeria, Cyprus, and Italy. We are also working with Comcast to deploy routers in their customers' homes in return for network analytics.

5.2 Sustaining the Deployment

Even after deploying BISmark routers in homes, it is a struggle to keep them online. Figure 8 shows attrition of the deployment. Nearly 25% of all routers go offline within three months, while another 25% have remained online for more than a year. We have learned many lessons, both about how to deploy reliable router software and how to keep users involved when unreliable software disrupts the user experience.

Lesson 4: *Users must be engaged to help keep routers online. Engagement can come in a variety of forms, and may be as simple as helping them better understand their network using the data we collect.*

If users disconnect their routers, we stand to lose both the device hardware and the data. We attempt to keep users engaged by providing useful tools like the Network Dashboard [27] to visualize ISP performance and uCap [42] to help users visualize and manage their home network usage. We conduct our development and data collection in the open; users can track BISmark development online [6, 32].

Lesson 5: *Upgrading critical software in the field is risky, but the ability to upgrade other software is essential for sustainable deployment.*

The ability to upgrade non-critical software after deployment has enabled us to pursue "good-enough" software development by deploying systems that are not fully ready. We can fix bugs and deploy new features by upgrading software in the field. This helps us reconcile the need for deploying systems that work for end users with constraints that include the need for extensibility and experimentation and limited time and manpower for testing and support.

We do not update certain critical software when there is a chance, however small, that a critical functionality (*e.g.*, the wireless network) could break. Our approach has been to minimize such cases by using a well-tested base platform that can maintain basic functionality even when higher-level client and backend software malfunctions.

Lesson 6: *Every home network has unique conditions and usage patterns, making comprehensive testing before deployment nearly impossible; bugs arise in practice.*

We aim to ensure that BISmark is foremost a stable access point, and that our custom software and experiments do not degrade user experience. The BISmark gateway is on the critical path of Internet access for at least one device in 92% of the homes; a malfunctioning router will disrupt network connectivity and, in the worst case, even completely take those devices offline. We have one window of opportunity per user to ensure that a router is installed and working correctly. Most people have no desire to troubleshoot home networks and will readily disconnect their BISmark router if it stops working as intended.

Lesson 7: *Community support is crucial; we rely heavily on commodity hardware manufacturers and open source software developers to build reliable, usable home routers.*

Commodity hardware solves many problems we faced with custom hardware because manufacturers (*e.g.*, Netgear) design hardware specifically to deploy it in the homes of non-technical users, which exactly matches BISmark's deployment scenario. Besides appearance, com-

modity hardware addresses many of the NOX Box's reliability and usability problems: (a) flash storage cards failed after only 3–4 months in the field, far sooner than expected for our workload; (b) we assembled each NOX Box from components, a laborious process; and (c) the component cost for each device was approximately $250 USD, or 2–3 times the cost of a commodity wireless router.

Similarly, OpenWrt's global community ensures that it is much more comprehensively tested on a variety of home routers than Debian. Although we occasionally found ourselves "held hostage" by bugs that were not fixed on our timescales, becoming embedded in OpenWrt's developer community was generally helpful.

Lesson 8: *Users may want to customize router settings, but doing so may introduce security vulnerabilities.*

BISmark routers have a flexible administrative interface to help users configure the router; this is a potential security vulnerability. One household accidentally disabled the router's firewall, opening its DNS resolver to the Internet. Attackers eventually recruited the router for amplification attacks over a period of many months, which we only discovered when the ISP notified the user of the problem. Although the disabled firewall was the culprit in this case, it led to a wholesale audit of the deployment and a spirited email exchange with the affected user. It is still unclear exactly how the firewall was disabled.

5.3 Experimentation

Designing and deploying measurements on BISmark has highlighted several nuances of supporting experimentation in production home networks.

Lesson 9: *It is difficult to reconcile the need for open data with that of user privacy.*

To encourage open data, we publish measurements collected from BISmark, but only if doing so doesn't threaten user privacy. Sometimes this decision isn't obvious. It is unclear when active data measurements can yield insight into user behavior; for example, patterns in router availability or throughput and latency measurements could indicate when users are home and using the network.

Lesson 10: *Vetting experiments is challenging, and a poorly designed (or controlled) experiment can cripple a user's Internet connection.*

Enabling a wide range of experiments introduces management and security concerns, specifically with reviewing code, controlling access, and ensuring that experiments do not disrupt user experience by making the device unstable or consuming too much network resources.

One household had comparatively slow upstream connectivity (512 Kbps upload) and an old modem with a large buffer, where even short throughput tests can induce bufferbloat pathologies [15]. Although the household's typical workload did not stress the network often enough to expose bufferbloat in their typical usage, BISmark's periodic throughput tests saturated the buffer and rendered the Internet unusable for the duration of the test (a few seconds). The degradation was bad enough for the user to complain and stop using the device after a few weeks.

5.4 Security

BISmark routers should compromise neither home network security nor the integrity of the platform. Although we try to minimize the possibility of security vulnerabilities by adopting industry-standard software and protocols wherever possible, some attacks against BISmark's backend infrastructure are still possible.

Lesson 11: *Users have access to the hardware and can modify firmware; this imposes new security challenges.*

BISmark's backend is subject to two broad security threats. The first is denial-of-service attacks, where malicious users could attempt to exhaust server resources for processing legitimate routers or measurements. Attackers could impersonate other users or even mount Sybil attacks to create many fake routers. Several backend components employ rate limiting, but these limits generally only protect against errant behavior of non-malicious clients. Thus far, we have deliberately chosen *not* to fix this class of vulnerabilities to make it easier for people to install BISmark on their routers without our involvement.

Other attacks could contribute malicious data to influence conclusions. Mitigating such attacks requires instrumenting routers with Trusted Platform Modules running signed executables to generate signed data. Attackers have physical access to router hardware and the software source code, so we rely on social measures: we try to deploy to trusted users and assume that they won't collude. Anyone can install BISmark on their own hardware, so we treat measurements from such routers with greater suspicion.

6 Conclusion

Although we did not initially plan to build (and maintain) such a large testbed, we realized the need for it 2009 when we began a study of access network performance. We recognized the variety of uses for a programmable testbed in home networks, and we also discovered that other researchers and operators share our interest. As BISmark continues to expand in terms of size and the diversity of experiments that it hosts, we will need to continually re-evaluate many of our design decisions. We believe our experiences thus far offer a unique perspective in comparison to existing long-running testbeds and useful lessons for others who perform research in broadband access networks.

Acknowledgments

BISmark would not have been possible without its countless contributors. Stephen Woodrow was instrumental in improving the underlying management infrastructure, integrating active measurements with MLab, and managing releases and project outreach. Alfred Roberts and Abhishek Jain led development of Network Dashboard. Thomas Copeland, Adam Allred, Aman Jain, Craig Balfour, and Brian Poole provided excellent backend support. Dave Täht helped bootstrap OpenWrt development. Hyojoon Kim, Abhinav Narain, Sarthak Grover, Andrew Kahn, and Andrew Kim contributed to the codebase. Marshini Chetty, Saad Qaisar, and the PAWS project have supported local deployments in various countries. We thank Giuseppe Aceto, Errol Arkilic, Jonathan Brier, Marc Brown, Enrico Calandro, kc Claffy, Russ Clark, Chip Elliott, Merrick Furst, Ethan Katz-Bassett, Dave Levin, Yogesh Mundada, Michael O'Reirdan, Antonio Pescapé, Bharath Ravi, Tiziana Refice, Glenn Ricart, Swati Roy, Stephen Soltesz, Stephen Stuart, Renata Teixeira, Andy Warner, Meredith Whittaker, and Yiannis Yiakoumis for valuable feedback. This research has been supported by NSF Award CNS-1059350, a Google Focused Research Award, and logistical support from Measurement Lab.

References

[1] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris. Resilient Overlay Networks. In *Proc. 18th ACM Symposium on Operating Systems Principles (SOSP)*, Banff, Canada, Oct. 2001.

[2] A. Bavier, M. Bowman, D. Culler, B. Chun, S. Karlin, S. Muir, L. Peterson, T. Roscoe, T. Spalink, and M. Wawrzoniak. Operating System Support for Planetary-Scale Network Services. In *Proc. First Symposium on Networked Systems Design and Implementation (NSDI)*, San Francisco, CA, Mar. 2004.

[3] G. Bernardi and M. K. Marina. BSense: a system for enabling automated broadband census. In *Proceedings of the 4th ACM Workshop on Networked Systems for Developing Regions*, 2010.

[4] BISmark Project Partners with Comcast. `http://noise-lab.net/2013/05/19/bismark-project-partners`.

[5] BISmark privacy statement. `http://projectbismark.net/participant/privacy`.

[6] BISmark uploads. `http://uploads.projectbismark.net`.

[7] M. Calder, X. Fan, Z. Hu, E. Katz-Basset, J. Heidemann, and R. Govindan. Mapping the expansion of Google's serving infrastructure. In *ACM SIGCOMM IMC*, IMC '13, 2013.

[8] J. Cappos, I. Beschastnikh, A. Krishnamurthy, and T. Anderson. Seattle: a platform for educational cloud computing. In *ACM SIGCSE Bulletin*, volume 41, pages 111–115. ACM, 2009.

[9] R. Carlson. Network Diagnostic Tool. `http://e2epi.internet2.edu/ndt/`.

[10] M. Chetty, S. Sundaresan, S. Muckaden, N. Feamster, and E. Calandro. Measuring broadband performance in south africa. In *Proceedings of the 4th ACM Annual Symposium on Computing for Development*, DEV 4, 2013.

[11] K. Cho, K. Fukuda, H. Esaki, and A. Kato. The impact and implications of the growth in residential user-to-user traffic. In *ACM SIGCOMM Computer Communication Review*, volume 36, pages 207–218. ACM, 2006.

[12] D. Croce, T. En-Najjary, G. Urvoy-Keller, and E. Biersack. Capacity Estimation of ADSL links. In *Proc. CoNEXT*, Dec. 2008.

[13] M. Dischinger, A. Haeberlen, K. P. Gummadi, and S. Saroiu. Characterizing residential broadband networks. In *Proc. ACM SIGCOMM IMC*, San Diego, CA, Oct. 2007.

[14] A. Filastò and J. Appelbaum. Ooni: Open observatory of network interference. In *USENIX FOCI*, Aug. 2012.

[15] J. Gettys. Bufferbloat. `http://www.bufferbloat.net/`.

[16] Glasnost: Bringing Transparency to the Internet. `http://broadband.mpi-sws.mpg.de/transparency`.

[17] Grenouille. `http://www.grenouille.com/`.

[18] S. Grover, S. Sundaresan, M. S. Park, S. Burnett, H. Kim, B. Ravi, and N. Feamster. Peeking behind the nat: An empirical study of home networks. In *Proceedings of the 13nd ACM SIGCOMM conference on Internet measurement*, IMC '13, 2013.

[19] A. Gupta, M. Calder, N. Feamster, M. Chetty, E. Calandro, and E. Katz-Bassett. Peering at the Internet's frontier: A first look at ISP interconnectivity in Africa. In *Passive and Active Measurement Conference*, 2014.

[20] Internet Usage for all the Americas. `http://www.internetworldstats.com/stats2.htm`.

[21] Internet Usage in Asia. `http://www.internetworldstats.com/stats3.htm`.

[22] ITU. Ict facts and figures, Jan. 2012. `http://www.itu.int/ITU-D/ict/facts/2011/material/ICTFactsFigures2011.pdf`.

[23] C. Kreibich, N. Weaver, B. Nechaev, and V. Paxson. Netalyzr: illuminating the edge network. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, 2010.

[24] G. Maier, A. Feldmann, V. Paxson, and M. Allman. On dominant characteristics of residential broadband internet traffic. In *Proc. Internet Measurement Conference*, Chicago, Illinois, Oct. 2009.

[25] M. Mathis, J. Heffner, and R. Reddy. Network Path and Application Diagnosis. `http://www.psc.edu/networking/projects/pathdiag/`.

[26] Measurement Lab. `http://measurementlab.net`, Jan. 2009.

[27] Network Dashboard. `http://networkdashboard.org/`.

[28] B. Norton. Peering in africa, Aug. 2012. `http://drpeering.net/AskDrPeering/blog/articles/Ask_DrPeering/Entries/2012/8/29_Peering_in_Africa.html`.

[29] OpenWrt. `https://openwrt.org`, Sept. 2013.

[30] Public access wifi service. `http://publicaccesswifi.org/`. Retrieved: September 2013.

[31] L. Peterson, A. Bavier, M. E. Fiuczynski, and S. Muir. Experiences building PlanetLab. In *Proceedings of the 7th symposium on Operating systems design and implementation*, pages 351–366. USENIX Association, 2006.

[32] Project BISmark: Open development portal. `http://projectbismark.github.io`.

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

[34] S. Roy and N. Feamster. Characterizing correlated latency anomalies in broadband access networks. In *Proceedings of ACM SIGCOMM*, pages 525–526. ACM, 2013.

[35] SamKnows. `http://samknows.com/`.

[36] M. A. Sánchez, J. S. Otto, Z. S. Bischof, D. R. Choffnes, F. E. Bustamante, B. Krishnamurthy, and W. Willinger. Dasu: Pushing experiments to the internets edge. In *Proc. of USENIX NSDI*, 2013.

[37] M. Siekkinen, D. Collange, G. Urvoy-Keller, and E. Biersack. Performance limitations of ADSL users: A case study. In *Passive and Active Measurement Conference (PAM)*, 2007.

[38] C. R. Simpson, Jr. and G. F. Riley. NETI@home: A distributed approach to collecting end-to-end network performance measurements. In *Passive & Active Measurement (PAM)*, Apr. 2004.

[39] N. T. Spring, D. Wetherall, and T. Anderson. Scriptroute: A public internet measurement facility. In *Proc. 4th USENIX Symposium on Internet Technologies and Systems (USITS)*, Mar. 2003.

[40] S. Sundaresan, W. de Donato, N. Feamster, R. Teixeira, S. Crawford, and A. Pescapè. Broadband internet performance: A view from the gateway. In *Proc. ACM SIGCOMM*, Toronto, Ontario, Canada, Aug. 2011.

[41] S. Sundaresan, N. Feamster, R. Teixeira, and N. Magharei. Measuring and mitigating web performance bottlenecks in broadband access networks. In *Proc. ACM SIGCOMM IMC*, 2013.

[42] uCap: Your data usage assistant. `http://ucap.projectbismark.net/promo/`.

[43] Web attack knows where you live. `http://www.bbc.co.uk/news/technology-10850875`, Aug. 2010.