Instrumenting Home Networks

Kenneth L. Calvert
Lab for Advanced Networking
University of Kentucky
Lexington, Kentucky, USA
calvert@netlab.uky.edu

W. Keith Edwards
College of Computing
Georgia Institute of Technology
Atlanta, Georgia, USA
keith@cc.gatech.edu

Nick Feamster
College of Computing
Georgia Institute of Technology
Atlanta, Georgia, USA
feamster@cc.gatech.edu

Rebecca E. Grinter
College of Computing
Georgia Institute of Technology
Atlanta, Georgia, USA
beki@cc.gatech.edu

Ye Deng
Lab for Advanced Networking
University of Kentucky
Lexington, Kentucky, USA
deng@netlab.uky.edu

Xuzi Zhou
Lab for Advanced Networking
University of Kentucky
Lexington, Kentucky, USA
xuzi@netlab.uky.edu

ABSTRACT

In managing and troubleshooting home networks, one of the challenges is in knowing what is actually happening. Availability of a record of events that occurred on the home network *before* trouble appeared would go a long way toward addressing that challenge. In this position/work-in-progress paper, we consider requirements for a general-purpose logging facility for home networks. Such a facility, if properly designed, would potentially have other uses. We describe several such uses and discuss requirements to be considered in the design of a logging platform that would be widely supported and accepted. We also report on our initial experience deploying such a facility.

Categories and Subject Descriptors

C.2.3 [**Network Operations**]: Network Monitoring; H.3.1 [**Content Analysis and Indexing**]: Abstracting Methods

General Terms

Experimentation, Management, Measurement

Keywords

home network management, home network troubleshooting

1. INTRODUCTION

As networking technology penetrates ever more deeply into end-user-managed environments, particularly the home, the problems are becoming well known. Recent data, as well as our own experiences, suggest that user experience problems plague home networking and are a source of much consumer confusion and frustration. In addition, management of home networks is a global security issue, due to home users' inability to properly configure and secure their networks. (A great deal of spam is sent from compromised home computers, for example.) The causes of these problems are manifold, and will necessarily require a multifaceted solution. It seems clear, however, that tools for helping with recognition, diagnosis, and correction of home network problems should be part of that solution.

A significant impediment to the development of such tools is the problem of knowing what is going on (or what happened in the recent past) in a home network. The ability to answer questions like "What changed around the time the problem appeared?" would be extremely useful in diagnosing failures and misconfigurations—whether by an expert or an expert system designed to help naive users.

Unfortunately, the answers to such quesions today are difficult to come by. Consumer-grade routers, switches, and access points provide only rudimentary logging facilities, with limited capacity—if they provide anything at all. Moreover, access to such logs is generally only available via a web interface, making it difficult to build tools that leverage the log data they do have. Host-based tools like `tcpdump` provide useful information, but only for knowledgeable users; also, they are typically used *after* something is perceived as being broken, in an attempt to diagnose network problems. Moreover, as a host-based tool, `tcpdump` can only observe traffic on a single local link; in most homes, the presence of a router/gateway that performs network address translation renders `tcpdump` useless for determining what traffic is actually *leaving* the home network.

In this position/work-in-progress paper, we argue that an autonomous, comprehensive, and scalable logging platform can be a key component for a number of interesting services related to home networks, including some that could help mitigate the usability problems described above. By *autonomous*, we mean that the facility should operate nonstop without user input, and not just when a problem is reported or a user attempts diagnosis. By *comprehensive*, we mean that the facility should provide an overview of the *entire* home network, for some specified period into the past (if not indefinitely). By *scalable*, we mean that the amount

HomeNets 2010, September 3, 2010, New Delhi, India.

of data should scale in both time and space. In other words, the system should collect data from each instrumented home over a long time, and data should be collected across *many* individual home networks, permitting analysis of aggregated data. And finally, by *platform* we mean that the facility should support programmatic access to collected data in a variety of forms and levels of detail, to support the creation of applications that leverage this data. By analogy with the flight data recorders carried by some aircraft, we call this proposed system *Home Network Data Recorder*, or HNDR.

Our contributions in this paper are as follows. In the next section we describe, at a high level, an architecture for a HNDR service. Then we outline a number of applications enabled by a properly-designed HNDR platform (Section 3). In Section 4, we discuss requirements and challenges for such a platform, and some possible solution approaches. Finally (Section 5), we describe our initial foray into HNDR deployment, as well as our plans for future work in this space. Section 6 covers related work.

Note: throughout this paper, the term "user" refers to the person(s) living in the home.

2. HOME NETWORK DATA RECORDER

The primary motivation for the proposed system is to aid in troubleshooting home networks. As such, a fundamental requirement is to support answering questions like "Are packets from this particular application making it out of the house?" and "What changed around the time the problem became apparent?" A general-purpose black box service could be useful whether those questions are being asked by a resident of the home, by the home's network service provider, or by a third-party support service. In Section 3 we also consider the value of such a service for several other uses.

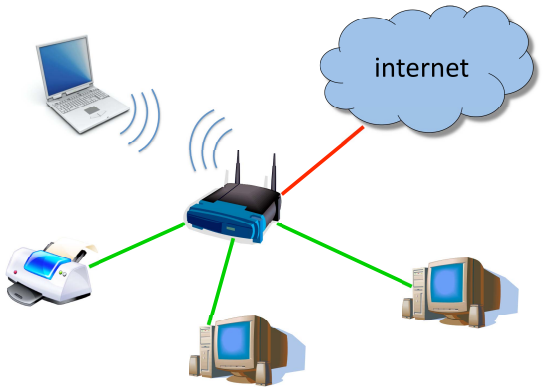

Figure 1: Target Home Network Configuration

Our target environment is a home network structured as depicted in Figure 1. The network comprises a wired segment and a wireless segment, both mediated by a single box that acts as a switch for the wired segment and an access point for the wireless segment. We assume that all traffic on the home network passes through the switch box, which also has an "upstream" connection (DSL or cable modem) to the Internet service provider's network. The switch box has a single routable address on the upstream interface; private-space IP addresses are used on the inside wired and wireless segments, and the box performs both bridge and network address translation (NAT) functions.

Although homes with more complex networks certainly exist, we believe this general configuration applies to a substantial fraction of homes with broadband Internet connections. Indeed, we have argued that this configuration—an intelligent central component that mediates all communication in the home—offers certain architectural advantages [5]. In any case, the design generalizes in a straightforward way to networks with more than one switch.

Figure 2 depicts the architecture of the HNDR system. Sensor functions housed on the central switch collect the raw data, recording events on the network: headers and partial payloads of each packet sent or received on each interface; events associated with the wireless interface such as associations and authentications; and changes to the network configuration (e.g., loss of "link" connectivity on a wired segment). These events are stored in low-level form, and also pass through an abstraction mechanism that produces higher-level summaries (e.g., Netflow-like records from packet-level traces).

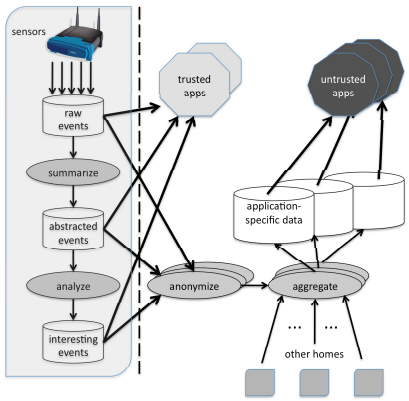

Figure 2: System architecture

Summary events are parsed and processed by an adaptive module that tags *interesting* events—where "interesting" means some combination of importance (to network functionality) and rarity. This analysis in general will involve state, and adapt over time to the specifics of each home environment. We expect that some fairly straightforward techniques can suffice to identify "unusual" events. (Examples of unusual events might include a new MAC address appearing for the first time, or modification of a configuration setting or file that is rarely touched.)

The primary application for the HNDR data is support for troubleshooting and diagnosis when something fails on the home network. Given sufficiently widespread support for HNDR, such services might be offered by ISPs as a means of differentiation, or by third parties on a subscription basis—similar to the maintenance and repair subscription services

offered to homeowners in some areas for water and/or sewer infrastructure. Such services will almost certainly require access to the original data, which includes sensitive per-user information that should not be made public. Such applications, labeled "trusted" in Figure 2, get access to the log data through a "push" interface (dotted line in the figure), which exports the data to a trusted collection point, over a secure channel under user control. Other applications (labeled "untrusted" in the figure) are not permitted access to the full data, but can see it after it is anonymized and aggregated with other homes' data by trusted components.

Like an airplane's black box, the HNDR platform is passive, and is intended to remain unnoticed until it is needed. It does have some limitations. For example, no direct record is kept of configuration changes to individual devices (e.g., PCs) on the home network. Also, if the upstream channel fails, a troubleshooting application running outside the home may be useless if the log data it needs is stored inside the home.

To be successful, the HNDR platform needs to be supported by service providers and by manufacturers of home network equipment, in particular gateway/access points. (Note that support by manufacturers of networked devices is not required by our design, though it might increase the scope of available information.) We therefore consider additional incentives for such support, in the form of other applications that it might enable.

3. POTENTIAL APPLICATIONS

The existence of a standard platform for logging events on a home network enables other interesting applications that have wider scope than that of a single home. We describe three, and consider the form of data required for each.

Internet Performance Measurement. Users of home networks occasionally observe performance problems. Sometimes these performance problems are a result of the way that they have their own network configured; other times performance degradations result from policies or configuration in the access ISP. To detect cases where a user's access ISP is the cause of performance degradation, bin Tariq et al have developed the *Network Access Neutrality Observatory (NANO)*, which collects network flow statistics—as well as metadata about the users' connections themselves—from a large number of user access networks and attempts to isolate the cause of the performance degradations [3]. Although NANO was designed specifically to detect cases where the access ISP was causing the performance degradations, it can, in principle, identify any enumerated feature as the common cause for performance degradation. One possible application of HNDR, then, is to deploy NANO collection agents at home network gateways and aggregate traffic and performance statistics to help users (or other interested parties) identify (1) whether the performance they are seeing compares well with that of other users; (2) if performance is suffering, what the likely causes might be.

This application requires access to performance data for individual homes, but only in *comparative* form. It is not necessary to identify endpoints of particular flows, only to compare their performance with other flows going to the same or "similar" endpoints.

Network Security. Many intrusion detection, spam filtering, botnet detection, and other network security algorithms depend on the ability to collect and aggregate information from a wide variety of data sources [17, 11, 16]. Unfortunately, because each home network operates in isolation, and home networks typically are connected to the Internet via many different access ISPs, it is difficult to collect information that might expose coordinated activity across home networks (e.g., a collection of bots that all belong to the same botnet). HNDR could make this task easier, as well: each HNDR collection point could serve as a "sensor" in a larger wide-area network monitoring system. Because many bots do, in fact, reside in home networks, HNDR might be able to help a security service provider achieve a much broader view of activity across home networks and detect coordinated activity. This application requires data to be collected from many home networks to succeed. However, it is not clear whether indivual homes' information needs to be identified.

Network Troubleshooting and Auto-Configuration. The NetPrints project observed that users might be able to more quickly isolate problems on their own home networks by comparing their network configurations with those of other users [2]. Similarly, Sheehan *et al* observed that users often seek help online to troubleshoot problems that they are observing with their home network configurations [8]. HNDR could support automation of these troubleshooting-related activities by collecting critical information about home network topology, device configuration, and network performance, aggregating networks with similar configurations, and determining the potential cause of performance problems by comparing network topologies and configurations. Such a system might suggest, for example, that a particular brand of network-connected digital music device would be an appropriate—and compatible—addition to the home network, based on the behaviors of other users. As another example, after examining aggregate data from many homes, the system might determine that a particular version of a patch causes problems for household users with a particular model of router, and warn others against installing the patch.

A recommender system such as this would need access to anonymized data from the HNDR that describes device type and traffic type information, as well as possible problem-related "interesting" events.

4. REQUIREMENTS AND CHALLENGES

Here we consider some of the requirements on the design of a general-purpose home network logging platform, and ways to deal with some of the challenges they present.

4.1 Privacy and Security

The prospect of collecting, sharing and using detailed information about events occurring in peoples' home networks is perhaps even more fraught with challenges than the same prospect in enterprise or provider networks [1]. While it is obvious that HNDR logs contain information that users might not want to share—for example, identities of web sites they have visited—it is also clear that such information in aggregate could be a valuable asset. Thus, there is a "tussle" between the individual users and third parties who would like to have access to the data. To further complicate matters, a variety of legal requirements may apply depending on the locality. In the United States there is no single privacy statute, but the federal Wiretap Act prohibits anyone who is not "a participating party to a private communication"

(including network communication) from intercepting such communication using an "electronic, mechanical or other device," unless one of several statutory exemptions applies.

An absolute requirement is that no personal information is revealed to a third party without the user's consent. In the case of an *automated* support system based on HNDR logs and local to the home, a straightforward solution is to simply keep all personally-identifiable information and the automated support system *inside* the home network. Information can be revealed to the user as needed for troubleshooting, but need not be seen by any third party.

In the case of an outside network troubleshooting support service, the HNDR information needs to be revealed to the service provider in order to implement the service. In this case, the user would presumably give explicit permission for access to the data when subscribing to the service. At the same time, the service provider would need to formulate and publish to the user a *privacy policy* constraining its use of the user's personal information. The network service provider has access to much of the information anyway; presumably the benefits of access to the troubleshooting service outweigh any concerns about loss of privacy from.

The use of HNDR-collected data for purposes such as those described in Section 3 must be approached with care. Two primary requirements [1]: First, users must be asked to give explicit consent for use of their information for each specific purpose or application. This consent might be tied to some consideration or compensation for the user—say, discounted fees for enhanced support services, or access to the system-produced advice in the case of recommender systems.

Second, only the information necessary to implement the application should be revealed. Other information should be anonymized or otherwise obscured to hide users' identities. Unfortunately, it is now pretty well-known that it is very difficult to effectively anonymize network trace data of the kind considered here while preserving its utility for research purposes [4]. In general there is an inverse relationship between the difficulty of of mapping logged addresses and applications to actual real-world counterparts, and the usefulness of the log information for various purposes.[1] However, if the service supports a large number of households, it may be possible to combine the data from many homes in such a way that the identities of individul households remain hidden. These techniques would need to be combined in a large-scale deployment, to ensure privacy.

Another potential issue is control: Some users may be reluctant to allow an outside entity to exert any control over their home network. Given the widespread acceptance of auto-update functions in popular operating systems and software, however, a similar model would likely work here. That is: the home gateway periodically contacts a well-known site, perhaps run by its manufacturer, to obtain software updates. This delivery mechanism avoids the need to open inbound ports on the router/gateway for control traffic.

4.2 Scale

Scale is an issue in multiple dimensions. One is the amount of data that can reasonably be stored within the home itself. Our design assumes an inexpensive router/gateway built from commodity hardware; at best it may have a few megabytes of stable storage, and much less writable storage. The NOXboxes in our current prototype (Section 5) have 8GB of compact flash storage. In an era of streaming video, packet traces can grow large very quickly. An experimental installation of our prototype system in a home shared by five graduate students generated about 3 GB/day of trace data. Even without video, collecting every event can generate data at a surprising rate. In another test installation, the upstream interface, which was connected to a cable network, generates about a megabyte of captured packet headers about every 15 minutes—just from ARP traffic—even when no communication with the outside world is happening.

The problem is compounded by the paucity of computing power available on commodity consumer gateways for filtering and/or generic compression. A straightforward solution—the one we are using in the current deployment—is to ship the data off to a collection site during periods of reduced activity. This requires the existence of such a (trusted) site and can potentially interfere with normal user traffic. Moreover, there is a tradeoff between the potential for interference and the availability of the log data when it is needed after an incident. Our present solution is a compromise: we store a day's worth of log data on the box itself, and ship it to a collection point in the middle of the night.

A second dimension of scale is the "horizontal" dimension. In order to enable some of the applications discussed in Section 3, HNDR must be deployed across a significant number of homes. Depending on how the system is structured administratively, data from millions of homes might be collected in a single system. The challenge here is not physical storage, but administrative management—ensuring that individuals' data is kept private and is treated in accordance with the users' wishes.

Finally, the utility of the HNDR platform for a variety of applications might be considered another dimension of scale. The interfaces, particularly the access provided to "sanitizing" components (lower part of the dotted line in Figure 2), must be designed for flexibility. For example, the latter interface should allow access to be controlled at high resolution, so that prohibiting access to some parts of a record does not prevent access to other parts of a record.

4.3 Resilience

A crucial aspect of the HNDR's usefulness is robustness. It must, like the hardware it is designed to work with, be robust to power failures, frequent connection and disconnection, and configuration changes. It needs to be self-maintaining and self-updating. A very important requirement is that component failure should not cause the loss of significant amounts of log data. (Moreover, each failure should be recorded in the event logs themselves.) Our current storage strategy limits the amount of data vulnerable to catastrophic switch failure to one day's worth.

5. STATUS AND FUTURE WORK

We have implemented an initial prototype HNDR data-collection system as a proof-of-concept and a vehicle for understanding what actually goes on in home networks. At the time of writing, the prototype is deployed in several of the authors' homes, with further deployments expected.

Our prototype design is based on the "NOX Box" [15], a

[1]Typically a sophisticated attacker, who may have access to side information about the household, its usage patterns, etc., is assumed.

small form-factor computer resembling an off-the-shelf home router/gateway, which runs the Linux kernel, OpenFlow [13] and NOX [10]. OpenFlow is a protocol for establishing flow state in switches. NOX is a "network operating system," designed to manage networks of switches via OpenFlow. While our current prototype makes minimal use of the capabilities of NOX/OpenFlow—the NOX Box acts like a standard home router/gateway, i.e., like a bridge at layer 2 and a NAPT box at layer 3 for flows between inside and outside—they add a dimension of flexibility that we expect will be handy in the future.

The NOX Box hardware has a 500MHz AMD Geode processor, three Ethernet ports and an 802.11b/g access point. One of the Ethernet ports can be (and is, in our deployment) designated as the "upstream" port, and network address translation is performed on traffic to and from that port. The default configuration restricts communication between the home network and the rest of the Internet so it can be initiated from "inside" only. Our NOX Boxes have 8G of flash memory installed. Because flash can sustain only a finite number of erase/write cycles before losing integrity (currently somewhere between 10^5 and 10^6 cycles [14]), the standard NOXBox configuration mounts its filesystems read-only, and runs from memory-based copies. The filesystem must be remounted read-write in order to write to it.

In our initial prototype, `tcpdump` runs on each of the wired Ethernet and 802.11 interfaces. The output (raw packet headers) is sent to a rotating collection of files in the memory-based filesystem. Periodic background jobs move these files to a directory in the persistent file system. Once a day, in the wee hours of the morning, the raw files are uploaded from the NOXBox to an offsite system then deleted. Any distillation and filtering occurs on the outside system, and not on the NOX Box itself. To prevent these potentially lengthy file transfers from distorting the logs themselves, the `tcpdump` running on the upstream interface is stopped before and restarted after the transfer.

In addition to collecting packet events with `tcpdump`, we log wireless (layer 2)-related events via the `iwevent` facility. The output of `iwevent` captures both configuration changes and hardware-generated events. Examples of the former include changes in the network ID or encryption mode; examples of the latter include registering or expiring a new client node, and periodic output of statistics regarding packets transmitted, received, and lost.

While the above store-and-upload approach worked well for a single-family household, in the home of five graduate students it proved inadequate. In particular, heavy downloading of streaming video caused the file system to fill up before the nightly upload. To get around this, the platform was modified so that trace files are written directly to a 500GB USB drive attached to the NOX Box. This unfortunately requires that the drive be swapped out periodically by hand.

Because the utility of the HNDR for troubleshooting depends on the accuracy of the recorded data, we investigated the reliability of using `tcpdump` on the NOX Box to capture packet events. For the experiments, three hosts were connected to the "inside" networks of the NOX Box. The "medium" load consisted of two videos downloaded from Hulu on a PC, a video downloaded fom YouTube on a laptop, and another video downloaded from YouTube to an iPhone. The "heavy" load consisted of one BitTorrent hot file download, one eMule hot file download, and one Hulu streaming video running on a PC, and two YouTube video downloads on a laptop. Both P2P applications reach a large number (thousands) of active peers. The "medium" load tests ran for 150 seconds—enough time for the streaming applications to reach a steady download sped; the "heavy" load tests lasted for 300 seconds.

We found that under the default NOX Box setup, `tcpdump` occasionally dropped packets under heavy load. We observed loss rates as high above 10% under the heaviest load with P2P applications running. However, we found that setting the default socket buffer size larger makes `tcpdump` reliable—with that change, no packet drops are observed, even under the heaviest load. Table 1 shows the numbers of packets captured in each run; in all cases, the number of packets dropped was zero.

We also investigated the effect of a background compression job on the loss rate in `tcpdump`. We found that running "`gzip`" in the background had no effect on performance, or the `tcpdump` loss rate.

Table 1: Packets captured under load on NOX Box

Run #	1	2	3	4
Load	med	med	heavy	heavy
Packets Captured	94574	89976	495466	263578

Our experience suggests that the NOX Box makes a satisfactory platform for experimenting with HNDR. Our work in the near term will focus on three areas. First, we will extend the capabilities of our prototype to collect the full range of network events described earlier. Second, we will develop analysis techniques to identify important/unusual events, as well as those relevant to particular kinds of trouble. We expect that techniques used by designers of intrusion detection systems may be useful here. Third, we want to learn from the data itself: how do things go wrong in home networks? What mistakes do users make that disrupt their networks? How frequent are network outages?

6. RELATED WORK

A number of user studies of home networking systems have pointed to the need for improved diagnosis and troubleshooting tools, of the sort that the HNDR system described here is intended to support. For example, [7], [12], [9] all describe home users' current frustrations with the network management tools available to them.

Some systems have begun to emerge that demonstrate the role that ongoing monitoring can play in home (and other) networks. For example, the Home Watcher system [6] uses a modified home router to collect data that is used to drive an interactive visualization system that allows home users to monitor and control bandwidth usage. The Eden system [19] similarly uses a modified router to collect a range of data used to visualize and manage a host of home networking functions. In the enterprise context, tools such as NetMedic [18] have demonstrated the power of network instrumentation to support diagnosis.

Systems such as NetPrints [2] also illustrate the utility of aggregation of data across multiple households. We see our platform as being a step toward facilitating the creation

of other systems with similar functionality, while aiming to preserve user privacy.

7. CONCLUSION

In addition to supporting troubleshooting services for home network users, a widely-supported Home Network Data Recorder platform would enable other interesting applications, although there are significant privacy and scalability challenges to be overcome. We have deployed an initial prototype based on a NOX/OpenFlow design, and are collecting comprehensive packet and event data in several home networks. The collected data will help us not only understand home networks better, but also extend and refine our design.

8. REFERENCES

[1] M. Allman and V. Paxson. Issues and Etiquette Concerning Use of Shared Measurement Data. In *Proceedings of 2007 ACM Internet Measurement Conference*, pages 135–140, San Diego, October 2007.

[2] B. Aggarwal and R. Bhagwan and T. Das and S. Eswaran and V. N. Padmanabhan and G. Voelker. NetPrints: Diagnosing home network misconfigurations using shared knowledge. In *Proc. 6th USENIX NSDI*, Boston, MA, Apr. 2009.

[3] M. bin Tariq, M. Motiwala, N. Feamster, and M. Ammar. Detecting Network Neutrality Violations with Causal Inference. In *Proc. CoNEXT*, Dec. 2009.

[4] M. Burkhart, D. Schatzman, B. Trammel, E. Boschi, and B. Plattner. The Role of Network Trace Anonymization under Attack. *ACM Computer Communications Review*, 40(1):6–11, January 2010.

[5] K. Calvert, W. Edwards, and R. Grinter. Moving Toward the Middle: The Case Against the End-to-End Argument in Home Networking. In *Proc. 6th ACM Workshop on Hot Topics in Networks (Hotnets-VI)*, Atlanta, GA, Nov. 2007.

[6] M. Chetty, R. Banks, R. Harper, T. Reagan, A. Sellen, C. Gkantsidis, T. Karagiannis, and P. Key. Who's Hogging the Bandwidth? The Consequences of Revealing the Invisible in the Home. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, Atlanta, GA, April 2010.

[7] Erika Sheehan Poole and Marshini Chetty and Rebecca E. Grinter and Warren Keith Edwards. More Than Meets the Eye: Transforming the User Experience of Home Network Management. In *Proceedings of the ACM Conference on Designing Interactive Systems (DIS 2008)*, Cape Town, South Africa, Feb. 2008.

[8] Erika Sheehan Poole and Marshini Chetty and Tom Morgan and Rebecca E. Grinter and W. Keith Edwards. Computer help at home: Methods and motivations for informal technical support. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2009)*, Boston, MA, Apr. 2009.

[9] R. Grinter, W. Edwards, M. Newman, and N. Ducheneaut. The work to make a home network work. In *European Conference on Computer-Supported Cooperative Work*, volume 18, page 22. Springer, 2005.

[10] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker. NOX: towards an operating system for networks. *ACM SIGCOMM Computer Communication Review*, 38(3):105–110, July 2008.

[11] S. Hao, N. Syed, N. Feamster, A. Gray, and S. Krasser. Detecting Spammers with SNARE: Spatio-temporal Network-level Automatic Reputation Engine. In *Proc. 18th USENIX Security Symposium*, Montreal, Quebec, Canada, Aug. 2009.

[12] J.-Y. S. Marshini Chetty and R. E. Grinter. How smart homes learn: The evolution of the networked home and household. In *Proceedings of Ubicomp 2007*, Innsbruck, Austria, 2007.

[13] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. OpenFlow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74, 2008.

[14] Micron Technology, Inc. Micron Collaborates with Sun Microsystems to Extend Lifespan of Flash-Based Storage, Achieves One Million Write Cycles. `http://www.micron.com/about/news/pressrelease.aspx?id=5F432D92EFA2B68E`, December 2008.

[15] NOX Box: a form-factor PC running OpenFlow and NOX . `http://noxrepo.org/manual/noxbox.html`.

[16] R. Perdisci, W. Lee, and N. Feamster. Behavioral Clustering of HTTP-Based Malware. In *Proc. 7th USENIX NSDI*, San Jose, CA, Apr. 2010.

[17] A. Ramachandran, N. Feamster, and S. Vempala. Filtering spam with behavioral blacklisting. In *Proc. 14th ACM Conference on Computer and Communications Security*, Alexandria, VA, Oct. 2007.

[18] S. Kandula and R. Mahajan and P. Verkaik and S. Agarwal and J. Padhye and P. Bahl. Detailed Diagnosis in Enterprise Networks. In *Proc. ACM SIGCOMM*, Barcelona, Spain, Aug. 2009.

[19] J. Yang. Eden: An Interactive Home Network Management System. Ph.D. dissertation, Georgia Tech, 2009.