University of Kentucky
UKnowledge

Theses and Dissertations--Computer Science | Computer Science

2016

Understanding Home Networks with Lightweight Privacy-Preserving Passive Measurement

Xuzi Zhou
University of Kentucky, xuzi.zhou@icloud.com
Digital Object Identifier: http://dx.doi.org/10.13023/ETD.2016.382

Right click to open a feedback form in a new tab to let us know how this document benefits you.

Recommended Citation

Zhou, Xuzi, "Understanding Home Networks with Lightweight Privacy-Preserving Passive Measurement" (2016). *Theses and Dissertations--Computer Science*. 50.
https://uknowledge.uky.edu/cs_etds/50

This Doctoral Dissertation is brought to you for free and open access by the Computer Science at UKnowledge. It has been accepted for inclusion in Theses and Dissertations--Computer Science by an authorized administrator of UKnowledge. For more information, please contact UKnowledge@lsv.uky.edu.

STUDENT AGREEMENT:

I represent that my thesis or dissertation and abstract are my original work. Proper attribution has been given to all outside sources. I understand that I am solely responsible for obtaining any needed copyright permissions. I have obtained needed written permission statement(s) from the owner(s) of each third-party copyrighted matter to be included in my work, allowing electronic distribution (if such use is not permitted by the fair use doctrine) which will be submitted to UKnowledge as Additional File.

I hereby grant to The University of Kentucky and its agents the irrevocable, non-exclusive, and royalty-free license to archive and make accessible my work in whole or in part in all forms of media, now or hereafter known. I agree that the document mentioned above may be made available immediately for worldwide access unless an embargo applies.

I retain all other ownership rights to the copyright of my work. I also retain the right to use in future works (such as articles or books) all or part of my work. I understand that I am free to register the copyright to my work.

REVIEW, APPROVAL AND ACCEPTANCE

The document mentioned above has been reviewed and accepted by the student's advisor, on behalf of the advisory committee, and by the Director of Graduate Studies (DGS), on behalf of the program; we verify that this is the final, approved version of the student's thesis including all changes required by the advisory committee. The undersigned agree to abide by the statements above.

Xuzi Zhou, Student

Dr. Kenneth L. Calvert, Major Professor

Dr. Miroslaw Truszczynski, Director of Graduate Studies

UNDERSTANDING HOME NETWORKS WITH LIGHTWEIGHT PRIVACY-PRESERVING PASSIVE MEASUREMENT

DISSERTATION

A dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy in the
College of Engineering
at the University of Kentucky

By
Xuzi Zhou

Lexington, Kentucky

Director: Dr. Kenneth L. Calvert, Professor of Computer Science

Lexington, Kentucky

2016

Copyright © Xuzi Zhou 2016

ABSTRACT OF DISSERTATION

UNDERSTANDING HOME NETWORKS WITH LIGHTWEIGHT PRIVACY-PRESERVING PASSIVE MEASUREMENT

Homes are involved in a significant fraction of Internet traffic. However, meaningful and comprehensive information on the structure and use of home networks is still hard to obtain. The two main challenges in collecting such information are the lack of measurement infrastructure in the home network environment and individuals' concerns about information privacy.

To tackle these challenges, the dissertation introduces *Home Network Flow Logger (HNFL)* to bring the lightweight privacy-preserving passive measurement to home networks. The core of HNFL is a Linux kernel module that runs on resource-constrained commodity home routers to collect network traffic data from raw packets. Unlike prior passive measurement tools, HNFL is shown to work without harming either data accuracy or router performance.

This dissertation also includes a months-long field study to collect passive measurement data from home network gateways where network traffic is not mixed by NAT (Network Address Translation) in a non-intrusive way. The comprehensive data collected from over fifty households are analyzed to learn the characteristics of home networks such as number and distribution of connected devices, traffic distribution among internal devices, network availability, downlink/uplink bandwidth, data usage patterns, and application traffic distribution.

KEYWORDS: Home Network, Lightweight Measurement, Passive Measurement, Privacy Preservation, Traffic Analysis

Xuzi Zhou

August 25, 2016

UNDERSTANDING HOME NETWORKS WITH LIGHTWEIGHT PRIVACY-PRESERVING PASSIVE MEASUREMENT

By

Xuzi Zhou

Dr. Kenneth L. Calvert

Director of Dissertation

Dr. Miroslaw Truszczynski

Director of Graduate Studies

August 25, 2016

ACKNOWLEDGMENTS

First, I would like to express the deepest appreciation to my advisor Prof. Kenneth Calvert. From him, I have learned to be a rigorous researcher and rational thinker. I would never succeed in my Ph.D. study without his guidence and supervision along the way.

I would like to thank Prof. James Griffioen, Prof. Zongming Fei, Prof. Hank Dietz for their valuable insights and help in my Ph.D. study and Prof. Sujin Kim for acting as the outside examiner.

I'm also grateful for the assistance of the following, all of the University of Kentucky: Mr. Hussamuddin Nasir, Mr. William Marvel, and Mr. Lowell Pike for timely assistance and maintenance of all lab equipments; Ms. Michelle Sublette and Prof. Melody Carswell for help navigating the IRB process, and recruiting and screening participants for the study; Mr. Paul S. Eberhart, Mr. Jacob Chappell, and Mr. Jerzy Jaromczyk for help with deployment and troubleshooting in the project.

Thanks also go to the members in the Laboratory for Advanced Networking during my study: Dr. Shufeng Huang, Dr. Xiongqi Wu, Dr. Onur Ascigil, Dr. Yinfang Zhuang, Mr. Song Yuan, and Mr. Ye Deng. I want to thank them all for their helpful suggestions and creating a friendly working environment.

Most importantly, I would like to thank my mother and father for their selfless support and thank my wife, Chia-Cheng, for her supportive and joyful company during my all these years.

Finally, I thank the National Science Foundation for their generous support under grants NSF-0904350 and NSF-1058977.

Table of Contents

Acknowledgments **iii**

List of Tables **vii**

List of Figures **ix**

1 Introduction **1**
1.1 Challenges in Measuring Home Networks 1
1.2 Contributions of the Dissertation 4
1.3 Dissertation Organization 4

2 Related Work **6**
2.1 Home Network Measurement Approaches 6
2.1.1 Outside-in Home Network Measurement 7
2.1.2 Endpoint-based Home Network Measurement 10
2.1.3 Gateway-based Home Network Measurement. 12
2.2 Applications of Measurement Results 16
2.2.1 Visualized Network Management 16
2.2.2 Crowdsourced Management and Troubleshooting 17

3 Measurement Infrastructure **19**
3.1 System Overview 19
3.2 Hardware and Firmware 22
3.3 Home Network Flow Logger (HNFL) 25
3.3.1 Forms of Traffic Data 26
3.3.2 Processing Pipeline 27
3.3.3 *hnfl*: Collect Traffic Data 28
3.3.4 *hnflc*: Process Traffic Data 31
3.3.5 Anonymizing Endpoints 34
3.4 HNFL Performance Evaluation 34
3.4.1 Data Collection Correctness 36
3.4.2 Data Collection Efficiency 38
3.4.3 Data Processing Efficiency 40
3.5 Data Upload Subsystem 43
3.5.1 Sender 43
3.5.2 Receiver 44

3.5.3 Secure Transmission . . . 44
3.6 Remote Update Subsystem . . . 45
3.6.1 Daily Update Routine . . . 45
3.6.2 Server Operations . . . 46

4 In-home Applications **48**
4.1 Home Network Traffic Dashboard . . . 48
4.1.1 Dashboard Interface . . . 49
4.1.2 Source of Data . . . 53
4.1.3 Authentication and Restricted Access . . . 54
4.2 iOS App for Basic Home Network Monitoring and Management . . . 55
4.2.1 User Interface . . . 55
4.2.2 Application Server on Router . . . 58
4.2.3 Authentication and Secure Communication . . . 59

5 Deployment and Data Collection **61**
5.1 User Recruitment and Router Deployment . . . 61
5.2 Data Management . . . 62
5.2.1 Format of Uploaded Data . . . 63
5.2.2 Data Storage and Backup in File System . . . 64
5.2.3 Data Organization in Database . . . 65
5.3 Maintenance of Deployed Routers . . . 66
5.3.1 Router Status Dashboard . . . 66
5.3.2 Email/Phone/Onsite Services . . . 67

6 Data Analysis **68**
6.1 Summary of Dataset . . . 68
6.2 Home Network Devices . . . 72
6.2.1 Number of Devices and Activities . . . 72
6.2.2 Device Vendors . . . 75
6.3 Internet Destinations . . . 77
6.4 Households . . . 80
6.4.1 Achievable Transmission Rate . . . 80
6.4.2 Diurnal Usage Patterns . . . 81
6.5 Internet Applications . . . 86
6.5.1 HTTP vs HTTPS . . . 86
6.5.2 Download vs Upload . . . 87
6.5.3 Edges vs Packets vs Bytes . . . 88
6.5.4 Devices and Applications . . . 91
6.6 Use of Home Network Traffic Dashboard . . . 91

7 Conclusions and Future Work **93**
7.1 Implications of Data Analysis . . . 95
7.2 Future Work . . . 96

Appendices **99**

Appendix A Understanding Network Usage Via Dashboard: the Instructions Given to Wildcat Home Router Users **100**
A.1 Open Dashboard 100
A.2 Understanding the Dashboard 101

Appendix B MySQL Database Tables **105**

Appendix C MySQL Queries for Data Analysis **108**

Appendix D Diurnal Network Usage Patterns of All Households **112**

Bibliography **116**

Vita **122**

List of Tables

3.1 Technical specification of routers 23
3.2 Result of correctness test 37
3.3 DNS query time 40

6.1 Summary of the dataset 70
6.2 Protocols observed in the dataset 71
6.3 Manufacturers of daily active devices 74
6.4 Manufacturers of devices 77
6.5 The data transmitted by applications. 88
6.6 Device/application affinity 90

List of Figures

2.1 Architecture of outside-in measurement platform 7
2.2 Architecture of Dasu measurement platform. 11
2.3 Architecture of gateway-based home network measurement platforms. 13

3.1 Overview of the measurement system. 19
3.2 Selection of routers in the project. Photographs by the author. 24
3.3 Linux NetFilter IPv4 hooks 26
3.4 Components of HNFL. 28
3.5 Use of NetFilter hook functions 29
3.6 HNFL kernel modules: *hnfl* and *hnfl_dns*. 30
3.7 HNFL user space daemon: *hnflc*. 33
3.8 Evaluation environment for correctness and efficiency. 35
3.9 Evaluation environment for DNS resolution. 36
3.10 Performance of *hnfl* 39
3.11 Performance of *hnflc* 41
3.12 Relationship between flows and edges 42

4.1 Dashboard: overall connections 50
4.2 Dashboard: incoming number of bytes 51
4.3 Homenet Control: icon and login view 55
4.4 Homenet Control: main function views 57

5.1 The deployed measurement router. Photograph by the author. 62
5.2 Color codes for the router status 67

6.1 The period of data contribution from all participating households . . 69
6.2 The number of networked devices in each household. 72
6.3 The traffic distribution among home network devices. 73
6.4 Activeness of devices. 74
6.5 The lifespan of edges with longest duration in households. 75
6.6 The number of devices and edges from different manufacturers or devices types. 76
6.7 The traffic distribution in edges among Internet destinations. 78
6.8 The traffic distribution in download bytes among Internet destinations. 78
6.9 How Internet destinations are shared among devices in a household. 79
6.10 Popular Internet destinations. 80
6.11 Highest upload/download rate observed from 52 households. 81
6.12 Diurnal network activities across all households. 82

6.13 Data transmitted (combined upload and download) on weekdays and weekends across all households. 83
6.14 Representative traces for six types of diurnal network activity on weekdays. 84
6.15 The trend in increasing HTTPS traffic. 87
6.16 The traffic distribution among Internet applications for all households. 89
6.17 The usage of Home Network Traffic Dashboard. 92

Chapter 1

Introduction

Home networks now constitute much of the "edge" of the Internet. Globally, as of 2015 about 45% of households have broadband coverage, while the penetration in developed countries is much higher—over 80% [1]. Unfortunately, home networks remain the least measured part of the Internet.

All three relevant parties, broadband users, Internet Service Providers (ISPs), and regulators can benefit from measurement results of home networks. Users can use the measurement results of different ISPs as the reference before they shop for a certain service and also as the benchmark to compare with their received services. Apart from end users, as stated by M. Linsner et al. in [2], ISPs and regulators are also the beneficiaries of efforts in home network measurement. ISPs could use the dataset to evaluate newly deployed devices and technologies. The measurement data across all customers can also be used to identify, isolate, and fix problems. As for regulators, they can use the measurement data to monitor the enforcement of regulatory policies, check the alignment of broadband deployment and the strategic goal, and facilitate the decision of new policies.

1.1 Challenges in Measuring Home Networks

The network technologies used in today's home networks, such as firewalls and Network Address Translation (NAT), make it very challenging to study home

networks from the outside (e.g., from the provider perspective). A study by Maier et al. [3] shows that roughly 90% of more than 20,000 Digital Subscriber Line (DSL) lines from a large European Internet Service Provider (ISP) were using a "NAT-enabled" gateway back in 2010. In recent years, researchers have been trying to study home network performance from the inside. However, such studies typically focus on active measurement of the "last mile" channel [4, 5, 6, 7]; such an approach generates traffic that can interfere with the normal home traffic.

In general, getting access to individual home networks for measurement purposes is the most challenging part. Indeed, users have few incentives to take part in measurement studies on their home networks, especially if there is any risk of service degradation. Privacy-oriented users, especially, may feel uncomfortable with their network activities being explicitly monitored by a third party (although it is noting that such monitoring by service providers seems to be an unavoidable condition of network access).

Measurement in home networks generally takes one of two forms: instrumenting one or more endpoints (host/devices), or instrumenting the gateway that connects the home network to the Internet. Endpoint-based measurement is generally favored for—and limited to—active measurements, since it is easy for a single host to send probes, but more difficult for it to observe traffic from all hosts on the inside network. Active measurements can be carried out by inexpensive dedicated devices [7] and rate-controlled, thus avoiding common difficulties, which include installing software on heterogeneous host platforms and interfering with the user's normal traffic (not to mention potentially consuming a portion of the user's bandwidth cap).

Home router/gateways also exhibit a lot of heterogeneity, but since the router is a generic appliance that is (i) mostly transparent to users, and (ii) typically not highly customized—unlike, say, a laptop or smartphone—that issue can be overcome by providing a custom-built gateway with built-in measurement capabilities. Some

prior gateway-based studies have involved users re-flashing their existing router; this may bias the study toward homes with more technically proficient residents.

Another challenge with gateway-based studies is that most home router hardware platforms have very limited compute and storage resources. This can be overcome by installing a more powerful system to conduct measurements (as in Homework [8]), but then the researcher is faced with the choice of retrieving the system at the end of the study (thus inconveniencing the user twice), or absorbing the cost of leaving the gateway with the user.

Prior gateway-based passive measurement studies using commodity gateway routers [5, 8] have used heavyweight methods of data collection, and involved anonymization methods that precluded releasing their data. For example, BISmark [5, 6]) exported every packet to user space via a packet filter for analysis. According to the previously-reported measurements [9], that method began to interfere with end-to-end performance (i.e., caused packets to be dropped at the router) at around 16 Mbps. Although that data transmission rate was reasonably high a few years ago, a significant fraction of homes today have Internet connections capable of higher speeds (cf. Figure 6.11).

This dissertation describes a lightweight privacy-preserving passive measurement system using specifically designed software [9] and present collected measurement data and some analysis results. The platform is designed to overcome the home network measurement challenges described above. The resulting measurement system has a negligible impact on performance and user experience, and runs on commodity hardware. It also preserves privacy by removing identifying information from collected data (in a non-prefix-preserving way). These characteristics help in recruiting subjects because the custom device provides very good performance, yet is inexpensive enough that it can be left on the field at the conclusion of the measurement study.

1.2 Contributions of the Dissertation

The contributions in this dissertation include:

- Design and implement a lightweight, privacy-preserving passive-measurement system, which can run on an inexpensive commodity platform without interfering with user quality of experience. The source code of the measurement system is available online.

- Collect measurement data from more than 50 U.S. residential households using the measurement system and add to the corpus of home network information by making collected data available to researchers.

- Analyze *anonymized* data about flows crossing the home router, both to reappraise previous studies and to obtain novel insights regarding home networks.

Unlike previous router-based work (specifically, BISmark [6] and Homework [8]), the traffic study described in this dissertation is *not* primarily concerned with either performance or enabling network management. The *passively-collected* dataset contains more than twice as many households as the passive dataset reported in BISmark [5], and the collection covers a longer time (six months on average vs. two weeks).

1.3 Dissertation Organization

The dissertation starts with the description of different approaches to measuring home networks and the applications using various measurement results of home networks in Chapter 2. Chapter 3 discusses the approach to understanding home networks and describes the measurement infrastructure in detail. Chapter 4 illustrates the applications designed for home network users based on measurement results. Details

about the deployment of the measurement infrastructure and the collection and management of measurement data are available in Chapter 5. In the following Chapter 6, details about collected data and interesting results derived from the analysis are elaborated. At last, Chapter 7 concludes with thoughts of future work and research possibilities.

Chapter 2

Related Work

Home networks constitute a large and important part of today's global Internet. Researchers have explored many facets of the home network. This chapter discusses previous and current related work in the area of measuring home networks and applications using measurement results.

2.1 Home Network Measurement Approaches

Unlike data centers, corporate networks, or university networks, home networks are usually closed and unwatched. The majority of home network users do not have the knowledge and skillset to measure and monitor their own home networks. Maybe some savvy users and networking experts might measure and monitor their own home networks using stand-alone tools such as ping [10], traceroute [11], iperf [12], tcpdump [13], wireshark [14] and so forth. However, these single-shot tests and unpublished private measurement data are not beneficial to the industry, regulators, or the research community.

Thus, in recent years, a number of studies have emerged to investigate home networks. These studies can generally be classified according to whether they are outside-in, endpoint-based, or gateway-based, and whether they use active or passive measurement techniques. This chapter discusses outside-in, endpoint-based, and gateway-based approaches separately.

2.1.1 Outside-in Home Network Measurement

Measurement data from outside-in approaches are usually collected by ISPs from passively scanning network packets at their backbone facilities or by researchers using specifically developed techniques to actively probe selected home networks from other hosts via the Internet. One advantage of outside-in approaches is that such approaches do not require end users to participate directly. However, due to the closed nature of home networks, these approaches are limited in what they can measure.

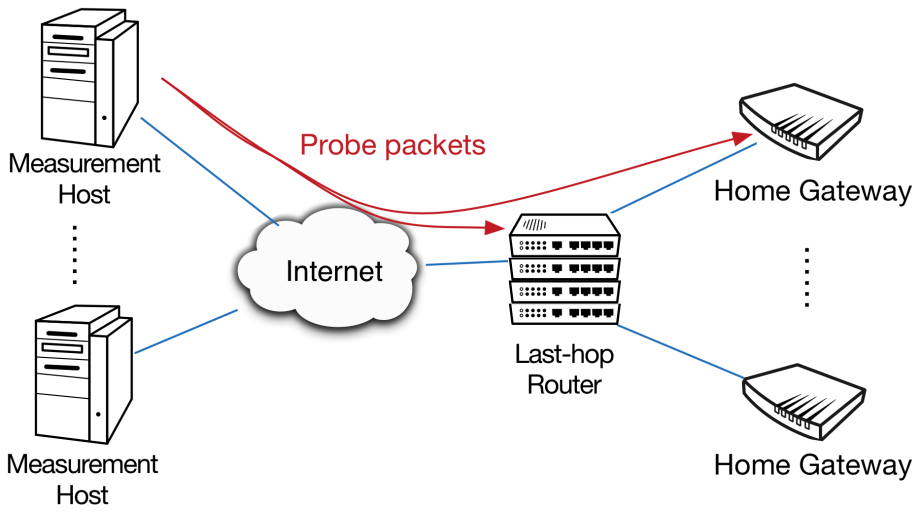


Figure 2.1: Architecture of outside-in measurement platform[1]

A study by Dischinger et al. [15] presents a unique technique to probe home networks by sending different types of packet trains to residential gateways and inferring the characteristics of the corresponding broadband. A packet train is a series of packets sent by the same source and targeting the same destination. As shown in Figure 2.1, the authors use a set of measurement hosts to send probe packets to selected target gateways. A valid target gateway needs to respond to Internet Control Message Protocol (ICMP) echo requests with ICMP echo responses and respond to Transmission Control Protocol (TCP) acknowledgement (ACK) packets, which do not belong to any open TCP connection, with TCP reset (RST) packets. The study

[1] All clipart used in this dissertation is from OmniGraffle 6.5.3.

covers 1,894 broadband hosts from 11 DSL and cable ISPs in North America and Europe that conform to requirements. The authors send large TCP ACK packets at 10 Mbps (megabits per second) to estimate the downlink bandwidth of target gateways by counting the proportion of TCP ACK packets answered by gateways. The 10 Mbps send rate is determined by the fact that all broadband plans offered by selected ISPs have advertised bandwidths under 10 Mbps at the time of the study (year 2007). Regarding uplink bandwidth, the authors send large ICMP echo packets instead at the rate of 10 Mbps. The uplink bandwidths can be estimated from the number of ICMP echo packets returned from gateways because the target gateways respond with ICMP echo packets of the same size and the number of ICMP echo reply packet is limited by the uplink capacity, which is usually a fraction of the downlink capacity. Meanwhile, the last-hop packet latencies and jitter could be estimated by sending small TCP RST packets to both target gateways and their corresponding last-hop routers, which are discovered using traceroute tool, and comparing the latencies of responses.

Similarly, Schulman et al. developed the ThunderPing tool [16], which sends ICMP echo packets to selected home networks under different weather conditions to measure the influence of severe weather conditions to the connectivity of home networks . Using the same architecture as presented in Figure 2.1, ThunderPing simultaneously sends ICMP echo packets from several PlanetLab [17] machines to millions of IP addresses from 11 ISPs in the United States using different access technologies (cable, DSL, Satellite, and Fiber) during 66 days. The host at an IP address is considered suffering from network failures if the host turns from responding to most ICMP echo packets to not responding to any. The results show that the same host are two times more likely to experience network failure during rain and four times more likely under thunderstorms.

Besides the active measurement efforts discussed above, some researchers study

home networks by analyzing network traces obtained directly from ISPs. Maier et al. [18] share their observations from the analysis of packet-level data from a major European ISP covering more than 20,000 home networks connected by DSL lines. The findings in the study are: 1) DSL sessions have very short durations as the median duration is between 20 to 30 minutes; 2) using IP addresses as identifiers may be misleading because IP addresses are frequently reassigned; 3) Hypertext Transfer Protocol (HTTP) applications have replaced peer-to-peer (P2P) applications as the dominating Internet application due to the popularity of multimedia streaming services such as youtube.com; 4) most clients have employed new TCP options like window scaling and selective acknowledgment (SACK) to boost data transmission efficiency; 5) most DSL subscribers do not fully use their available bandwidth; 6) the latencies between home networks and their first-hop routers dominate the packet round trip time (RTT) probably due to the interleaving mechanism of Asymmetric Digital Subscriber Line (ADSL). However, the authors cannot accurately distinguish individual hosts from home networks due to the existence of NAT.

Sargent et al. [19] study behaviors of fiber-to-the-home network users using the dataset collected in the campus network of a U.S. university over a 23 month period. The dataset includes transport-level connection logs and packet-level traces. According to the analysis results, Sargent et al. find out that 1) households use the network at transmission rates lower than commercial available ones even though fiber-to-the-home network provides ten times higher capacity; 2) HTTP is the top receiving application, while BitTorrent is the top sending application; 3) although end hosts should achieve higher performance according to TCP theory, certain TCP implementations may artificially limit the performance.

Although the outside-in measurement approaches have many limitations, the measurement results still have great reference value and are comparable to the results presented in Chapter 6.

2.1.2 Endpoint-based Home Network Measurement

Endpoint-based measurement approaches are usually software-based. They require participants to download and install a copy of the software to run on their own network devices. Some researchers prefer the software-based approach due to its lower deployment cost and adoption barrier. But the measurement results from endpoint-based approaches cannot accurately represent the performance and behaviors of the entire home network if there are multiple active devices within the same home network.

Netalyzr [20] by Kreibich et al. is distributed in the form of Java applets[2], which run within web browsers. In order to initiate the tests, a user has to open the Netalyzr website and start the tests. The Netalyzr architecture includes a suite of servers to measure the performance and diagnosis parameters of the client's network: 1) echo servers: test the reachability of measurement services; 2) Domain Name System (DNS) servers: test DNS and NAT behaviors; 3) bandwidth measurement servers: measure network-layer performance parameters such as bandwidth, latency, uplink buffer, packet loss, packet reordering, and packet duplication; 4) path Maximum Transmission Unit (MTU) measurement server: measure the network path behaviors related to MTU. Two years after the deployment of Netalyzr, Dhawan et al. introduced Fathom [21], which is a Firefox[3] extension that ports the Java applet-based Netalyzr into Javascript[4].

Dasu by Sánchez et al. [22] makes use of Vuze [23] BitTorrent[5] clients by providing a custom software plugin. Besides the distributed clients, the Dasu architecture also involves a set of services in the measurement controller and collector infrastructure as presented in Figure 2.2. Researchers can coordinate hosts with the Dasu plugin

[2]Java applet: a small application written in Java, which is a popular programming language.
[3]Firefox: a popular web browser with a large user base globally.
[4]Javascript: a high-level programming language mainly used in the web environment.
[5]BitTorrent: a peer-to-peer (P2P) communications protocol for file sharing.

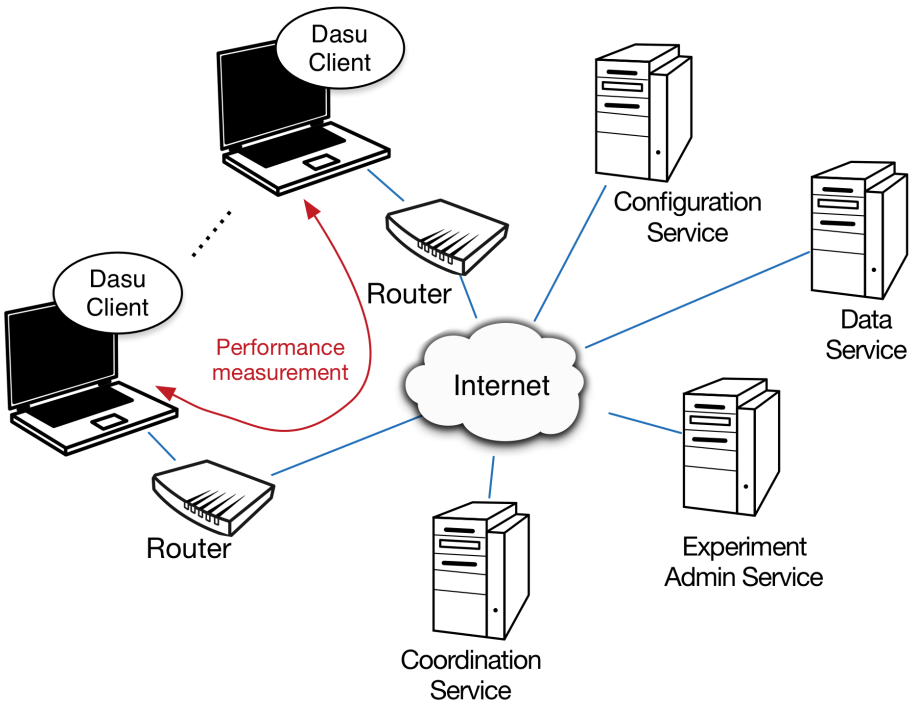


Figure 2.2: Architecture of Dasu measurement platform.

to conduct various active measurement tests across its large user base. The plugin module can also perform active measurements with other active clients. The active measurement tasks collect data about downlink and uplink throughput, end-to-end latency, forwarding path, and DNS resolution. Since most active measurement tasks involve multiple clients, Dasu modules employ a scheduler to synchronize tasks among participating clients. Meanwhile, the Dasu plugin can passively collect performance and behaviors data of the BitTorrent client and system-wide statistics such as the number of active and closed TCP connections.

DiCioccio et al. [24] study home networks using Universal Plug and Play (UPnP) [25] information obtained using HomeNet Profiler [26] and Netalyzr [20] from end hosts. According to the observation of DiCioccio et al., only 35% of all homes have UPnP enabled. From homes with UPnP enabled, the following measurements can be conducted: 1) achievable uplink and downlink capacity at the home network gateway, 2) data transmission rates of local hosts within the home network, 3) estimating

packet loss during active Netalyzr's active capacity tests, and 4) inferring buffer sizes on gateway router devices from different manufacturers. DiCioccio et al. prove that UPnP can be used to extend the capacity of endpoint-based home network measurement approaches.

The Archipelago (Ark) measurement infrastructure of Cooperative Association for Internet Data Analysis (CAIDA) [7], is not software-based. Each participant connects a small measurement node, inexpensive Raspberry Pi, to the home network just like a normal networked device. The Ark platform does not test the performance of the home network. Instead, it makes use of the 158 nodes deployed in both residential and institutional networks around the world to conduct distributed active measurement of reachability and topology of global network infrastructure.

2.1.3 Gateway-based Home Network Measurement.

There is a better observation point than individual endpoints in home networks considering the comprehensive measurement—the home network gateway. From the view of home network gateways, researchers can measure traffic from all active network devices behind the NAT.

SamKnows [4] deployed a measurement platform that continuously and actively measures the broadband performance of home networks. The measurement platform is composed of measurement routers (Whiteboxes), data collection infrastructure, and measurement servers (see Figure 2.3). SamKnows Whiteboxes are TP-Link routers flashed with custom programmable OpenWrt firmware [27] and aimed to measure global broadband performance by running active measurement tests from participating households [28]. The test results are sent by the Whiteboxes to the geographically distributed data collection infrastructure and viewable from the SamKnows performance monitoring dashboard [29] for participating users. The active measurement conducted on Whiteboxes covers a range of properties: downlink and

uplink throughput, end-to-end and last-mile latencies, network availability, forward path, and performance of various applications such as HTTP, Voice over IP (VoIP), P2P, DNS, email, File Transfer Protocol (FTP), and video streaming.

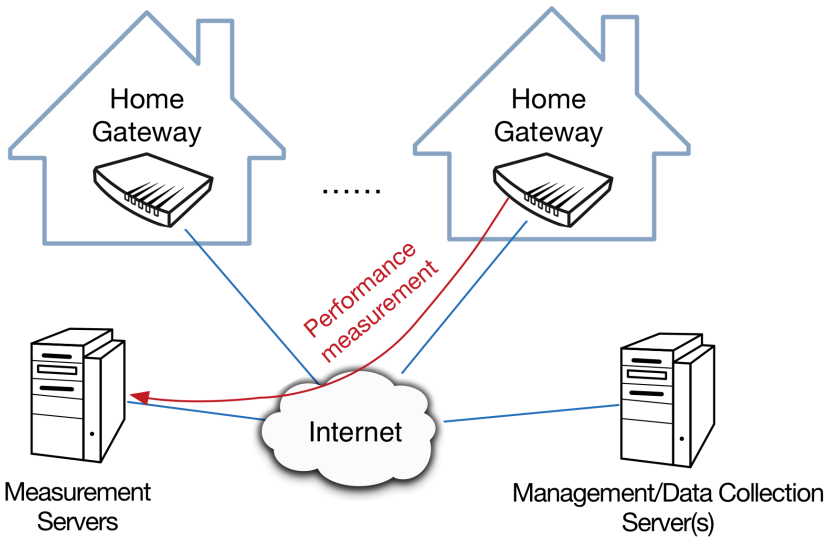


Figure 2.3: Architecture of gateway-based home network measurement platforms.

Bischof et al. [30] use the publicly shared SamKnows residential gateway data from U.S. users [31], endpoint-based dataset from Dasu [22] and a survey of international commercial broadband connectivity plans [32] over a 23-month period. This study focuses on the relationship between broadband service prices, user demands, and broadband connection characteristics. According to the study, user demands of moving up to higher service tiers have less influence on the increase in broadband traffic than the increasing broadband subscriptions and service capacities. Also, there is a strong correlation between user demands and service quality, including transmission rates, latency, and packet loss rates.

The Broadband Internet Service Benchmark Project (BISmark) [6] by Georgia Tech has deployed several hundreds of programmable routers in homes from North America, Europe, Africa, and Asia. The architecture of BISmark platform is similar to the one of SamKnows (see Figure 2.3). BISmark also uses OpenWrt firmware

with NetGear and TP-Link routers to act as home network gateways and perform measurements in participating households. Although BISmark has a much smaller user base compared with SamKnows, BISmark enables researchers to deploy both active and passive measurement projects in home networks [5, 33]. Since BISmark routers are programmable, researchers can instruct routers to install different tools for specialized measurement tasks. Some of the tools used in BISmark projects are 1) ShaperProbe [34]: measure the broadband link capacity with Measurement Lab (M-Lab) servers; 2) Distributed Internet Traffic Generator (D-ITG) [35]: measure packet latency, loss, and jitter; 3) Paris traceroute [36]: measure the forward and reverse path between BISmark routers and M-Lab servers.

The most recent BISmark study [5] explores similar aspects of home networks to the ones discussed in this dissertation. This dissertation reproduces several of their passive-measurement experiments. Grover et al. present a set of attributes of home networks using both active and passive measurement approaches, including: 1) Availability of Internet connectivity: households in developed countries tend to keep the gateways routers up while households in developing countries switch off their gateway router frequently; 2) the number and types of devices that are used in home networks; 3) the utilization ratio of available bandwidth; 4) diurnal usage patterns of users; 5) the amount of traffic flow towards popular domains such as google.com, youtube.com, and so forth. However, their dataset of passive measurement is relatively small (25 homes) and short (two weeks); much of their focus was on external-facing measurements and device availability. Meanwhile, libpcap-based passive measurement facility in BISmark imposed considerable overhead and can affect user-perceived performance on inexpensive commodity router platforms [9].

Among the earliest work was Home Network Data Recorder (HNDR) [37], proposed as an infrastructure to collect comprehensive packet and event data passively in home networks. The HNDR has three components: 1) *recorder* collects raw

data including headers and partial payloads of packets sent and received, wireless connection events, and network configuration changes; 2) *summarizer* provides aggregated data from raw data; 3) *parser* processes collected data and tags important and rare events. The prototype implementation of HNDR makes use of the powerful kit-based NOX Boxes [38], which have more memory and storage resources than commodity routers. However, HNDR uses tcpdump to capture raw packet data, which is not efficient enough to collect all network traffic even with the powerful NOX Box hardware. HNDR reports about 10% loss rate during measurement under heavy traffic load with active P2P applications [37].

Even earlier, the Homework project [8] developed a router-based platform focused on making network control and troubleshooting easier for users by giving them a comprehensive view into what was happening on their networks. In order to deal with the overhead of libpcap-based flow data collection, Homework used laptops as gateway routers. They recorded events including flows, wireless association and Dynamic Host Configuration Protocol (DHCP) lease transitions; a STREAM database [39] was used to manage the storage and processing overhead.

A large-scale passive measurement platform using home network gateways can generally yield better understanding of home networks than outside-in and endpoint-based approaches. However, one major obstacle preventing researchers from developing such platform is the high cost of deploying powerful gateway devices that support accurate passive measurement. Thus, this dissertation attempts to solve the problem by designing and implementing a lightweight passive measurement tool that runs on cheap commodity routers and still measures the whole home network accurately.

2.2 Applications of Measurement Results

Researchers have been using results from home network measurement in aiding areas like home network management and troubleshooting.

2.2.1 Visualized Network Management

Chetty et al. explore the impact and meaning of graphic network monitoring and management tools for home network users in a series of studies [40, 41, 42, 43]. The Home Watcher system [40] was designed to help users track real-time device level broadband data usage and limit the data rate for all devices within the home network. In order to use the Home Watcher, the household needs to set up a separate laptop or personal computer (PC) to act as the centralized display and controller and install the client software in each network device if the user wants to view and control the data usage of the device. In the following study [41], Chetty et al. introduce a new visual tool, Kermit, for home network management. The design of Kermit is different from the Home Watcher. Kermit uses routers flashed with DD-WRT [44], a open-source Linux based firmware, to collect bandwidth usage data for all devices using the home network. Instead of the centralized display and controller of the Home Watcher, Kermit users can use the visual tool via web browsers. In addition to the functions provided by the Home Watcher, the Kermit tool also allows users to test the network speed with an outside server and provides historical usage data to users as well. Both Home Watcher and Kermit were deployed in the United States. From the feedback of users, such visual network monitoring and management tools are welcomed by home network users because the visible usage data give users the edge to talk with their service providers for network issues or settle internal disputes around data usage within the household.

Unlike ISPs in the United States that usually offer unlimited data plans, other ISPs, especially the ones in developing countries, apply monthly data limit on their

broadband subscribers. The studies [42, 43] by Chetty et al. introduce the specified tool, uCap, for users with a bandwidth cap to better understand and also impose limit on the data usage for the whole household and among all devices.

The home network traffic dashboard discussed in Chapter 4.1 shares the same goal—uncovering hidden network usage with visualization technologies.

2.2.2 Crowdsourced Management and Troubleshooting

Rana et al. [45] use TShark [46] on a Linux router to capture traffic flows and feed the flows to a traffic policy server. In return, the policy server sends the IPTables rules back to the router to control the traffic of the home network. The preliminary results from the study show the possibility of using traffic measurement data to feed dynamic and automatic traffic management in home networks. However, due to the limited compute and storage resource available on commodity home routers, a dedicated third-party system is possibly needed to provide such advanced services to distributed home networks. Feamster [47] proposes a home network management system. The proposed system requires a centralized controller and distributed programmable gateway routers. The controller collects network data from gateways and runs security inference algorithms, such as spam filtering and detection of botnet[6] and malware. The centralized controller can use the inference results to generate traffic policies and configuration commands and push them back to home network gateways.

There are other studies trying to troubleshoot by latitudinal comparisons across home networks. Network Access Neutrality Observatory (NANO) [48, 49] is designed to discover if ISPs degrade performance or connectivity for certain users or applications. NANO collects system information and conducts performance measurement from a large user base and aggregates the measurement results according to user groups. Since the users in a group usually share similar attributes, such as ISP,

[6]botnet: a network of computers infected with malicious software that may be used to conduct malicious tasks such as attacking other computer systems and sending spam messages.

geographical location, and operating system, NANO compares the performance from hosts in the same group to determine the cause of degraded network performance.

Otherwise, Agarwal et al. introduce NetPrints [50] (short for Network Problem Fingerprints) to help network users find out the cause of network problems by comparing network configurations with other users using NetPrints. The NetPrints client software, which runs on end hosts such as personal computers, collects gateway configuration (e.g. MTU value, NAT table, and DHCP settings), local host configuration (e.g. firewall rules and TCP parameters), and remote configuration if the remote host has installed NetPrints software as well. The configuration uploaded to NetPrints server is labeled as "good" or "bad" depending on whether network applications are running successfully. When a user experiences application failure and runs NetPrints diagnosis, NetPrints software marks the user's current configuration as "bad" and suggests a "good" configuration that is close to the current "bad" configuration.

Chapter 3

Measurement Infrastructure

The home network environment differs from enterprise networks or data centers, in that home networks are usually smaller in scale but no less heterogeneous in types of devices. They are generally unadministered, but are more privacy-sensitive. Perhaps most importantly, the vast majority have a single uplink to the Internet; this makes it possible to collect data about all communication between the household and the outside world by placing the monitoring facility at the gateway between the home network and the Internet. This chapter gives an overview of the measurement system, which is specially designed to be unobtrusive and privacy-preserving.

3.1 System Overview

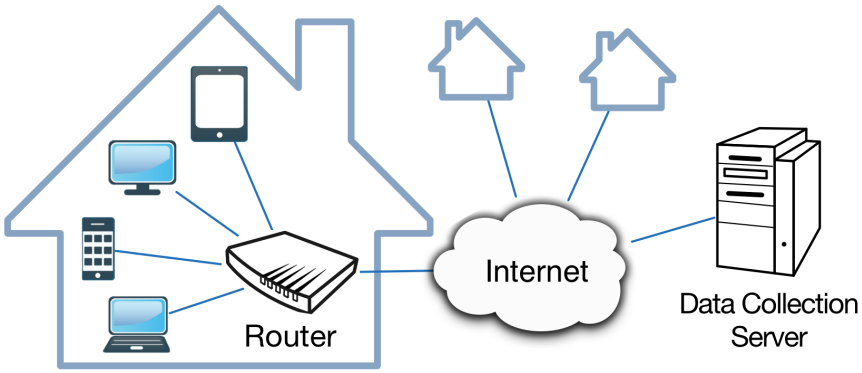


Figure 3.1: Overview of the measurement system.

Figure 3.1 provides an overview of the data collection system. Each participating home has installed a custom home router, which was the first endpoint behind the cable or Digital Subscriber Line (DSL) modem connecting the home to the Internet (Chapter 5). Traffic data were periodically uploaded to a server in the lab located in University of Kentucky. Unlike other gateway-based measurement platforms, such as SamKnows Whiteboxes [4] and BISmark [6], the measurement system does not involve measurement servers as illustrated in Figure 2.3 because the measurement system does not conduct active measurements that require external servers.

The measurement infrastructure is designed and implemented with the following goals in mind:

Transparency: The system should not interfere with normal operation and should not affect the user's experience in any observable way. This implies that the overhead of collecting measurements—both per-packet, and at upload time—should be small.

Privacy and Security: It must be possible to assure participants in the study that their personal information will remain confidential, even if collected data are made available to researchers. In practice, any information that can reveal the identity of a household, or the browsing habits of its inhabitants—including external Internet Protocol (IP) addresses or Domain Name System (DNS) names contacted, and internal Media Access Control (MAC) addresses—must never leave the home.

Comprehensiveness and Consistency: The system should collect enough data to support interesting conclusions, and should enable both latitudinal comparisons across households (modulo the privacy requirement), and longitudinal comparisons within households over time.

To some extent, these goals conflict. In particular, there is a tension between the amount of data collected/uploaded and preservation of the user experience. A collection of too fine-grained data can result in a low-level, unpredictable background load on the uplink.

The deployed home router hardware is a TP-Link WDR3600 N600 dual-band commodity box with gigabit ethernet and dual radios supporting 802.11 a,b,g and n standard (more details about hardware selection in Section 3.2). To ensure that the data are preserved across network outages and other interruptions (even long ones), each router is equipped with an 8GB USB drive, which is mounted as a separate file system.

The data collection software runs on the open-source OpenWrt (Attitude Adjustment) operating system, and consists of custom kernel modules and user-space programs and scripts (Section 3.3). A kernel facility collects flow information in a kernel data structure as each packet is forwarded. Every five minutes, a user-space process wakes up and reads this data via the proc filesystem[1], processes it, and places a file containing the anonymized and aggregated flow information for that interval in the "upload" directory of the USB filesystem. Periodically, these files are uploaded to the collection server in the lab by a cron[2] job (Section 3.5). To secure the data transfer, the upload program and the server use TLS with mutual authentication (each knows the other's public key). To ensure that software can be updated, every router also "calls home" each night to check for the availability of new or modified scripts; such scripts can be designated for installation either on a specific router, or on all routers. (See Section 3.6 for details.)

The system also includes, as part of the web interface, a dashboard that displays collected data in a graphical format (Section 4.1). The web display shows the

[1]proc filesystem: a control and information centre for the kernel, which is often used to move data between kernel space and user space in Linux

[2]cron: a job scheduler software in Unix-like computer operating systems

most-active (inside, outside) host pairs in terms of bytes and packets exchanged; graphs for the most recent five-minute interval and an exponentially-weighted moving average statistic can be displayed. Unlike the uploaded data, the user-visible display also shows names for inside and outside endpoints when they can be determined (most of the time). These names are obtained by snooping Dynamic Host Configuration Protocol (DHCP) and incoming DNS response messages, respectively. The dashboard—which enables users to see which devices are responsible for the most traffic—is designed to be an additional incentive to participate, and it was demonstrated as part of the installation process. Unlike the results reported by Grover et al [5], most users rarely if ever made use of the dashboard according to the measurement data. This may be due in part to the fact that virtually no household in the study had a usage cap imposed by its service provider.

3.2 Hardware and Firmware

A programmable gateway router is the core hardware of the passive measurement infrastructure in home networks. Several different platforms are tested throughout the lifetime of the home network project as shown in Figure 3.2. The specifications of the four router platforms are listed in Table 3.1.

Figure 3.2 presents the four routers used in different stages of the project. During the early stage of the project, a conceptual home network was built in the lab using the NOX Box. The NOX box platform runs standard Debian GNU/Linux and has plenty of computing resource to support experiments. However, the NOX box platform is not suitable for large scale deployment due to the following three reasons:

- Support for the NOX system software is discontinued. There is no update to the NOX from the developer since Oct. 2011[3].
- The NOX box requires assembly.

[3]The online repository of the NOX: https://github.com/noxrepo/nox-classic

Table 3.1: Technical specification of routers

Device	Nox Box	NetGear WNDR3700v2	NetGear WNDR3800	TP-Link WDR3600
CPU	AMD 500MHz	MIPS 680 MHz	MIPS 680 MHz	MIPS 560 MHz
Flash	2 GB	16 MB	16 MB	8 MB
RAM	256 MB	64 MB	128 MB	128 MB
Wired	3x 10/100E	5x GbE	5x GbE	5x GbE
Wireless	2.4 GHz 802.11 a/b/g	2.4/5 GHz 802.11 a/b/g/n	2.4/5 GHz 802.11 a/b/g/n	2.4/5 GHz 802.11 a/b/g/n
USB	2x	1x	1x	2x
Price	$285.00[1]	$74.99[2]	$96.99 [2]	$72.50[3]

[1] The price of NOX Box is calculated from parts sold on Netgate.com as of June 2011.
[2] The price of NetGear WNDR3700v2/WNDR3800 routers is quoted on Amazon.com as of May 2013.
[3] The price of TP-Link WDR3600 router is quoted on Amazon.com as of May 2014.

- The total cost of all NOX box parts is too expensive.

Thereafter, during the prototype design and implementation stage, software is tested on NetGear WNDR3700v2 and WNDR3800 routers. The hardware has limited yet enough computing power to reliably host the passive measurement infrastructure.

NetGear WNDR3700v2/WNDR3800 routers do not run the programmable operating system off the shelf. Thus, the measurement system uses a customized system on top of the OpenWrt Linux "Attitude Adjustment" release [27] to replace the default static firmware. The OpenWrt system comes with a fully writable file system, convenient package management, and an active developer community. Meanwhile, OpenWrt has consistent support for a large list of commodity routers [51].

During the course of our research, the sale of NetGear WNDR3700v2/WNDR3800 routers were discontinued. TP-Link WDR3600 routers are selected as the replacement platform. The passive measurement infrastructure is also compatible on the TP-Link WDR3600 router since its architecture is similar to the NetGear routers.

It is worth noting that OpenWrt had already announced its new "Barrier Breaker" release before the planned deployment. The new system is not used in the study

Figure 3.2: Selection of routers in the project. Photographs by the author.

because all of the system design and implementation are done under "Attitude Adjustment" release. Without apparent problems with the old Attitude Adjustment release, there were no incentives to take the risk of losing system stability or postponing the deployment due to any problems raised by the new OpenWrt release.

3.3 Home Network Flow Logger (HNFL)

The target environment of the study is a "typical" home network, with one consumer-grade gateway router connecting a single internal subnet to an access network, and various network devices that connect to the router through wired or wireless connections to generate network traffic. The following design goals guided development of the passive measurement infrastructure:

- **Increase measurement performance.** The focus of the study is cross-boundary traffic that can be represented by bipartite graphs. The study does not adopt the traditional libpcap-based[4] approach because: a) *libpcap* causes frequent expensive context switches between kernel and user space, which cost a lot of computing resource on the router and lead to inaccurate measurement when libpcap cannot keep up with the transmission rate of packets; and b) duplicate packets are captured in order to cover all cross-boundary traffic. Therefore, HNFL is based on the NetFilter subsystem in the Linux kernel [52], which is included in versions 2.4 and higher. By choosing NetFilter hooks carefully (Figure 3.3, discussed further in Section 3.3.3), the software captures all and only desired packets exactly once. Meanwhile, the entire traffic data collection operation takes place within the kernel space.
- **Reduce overhead.** Flow data are further aggregated into bipartite graphs to save storage space before data transmission and bandwidth during transmission.

[4]libpcap: a portable library for user-level network packet capture using C/C++ programming language.

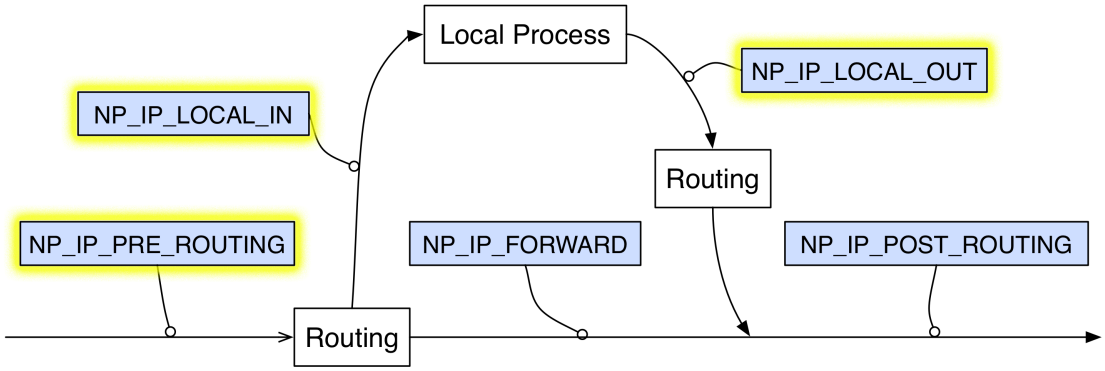


Figure 3.3: Linux NetFilter IPv4 hooks

- **Preserve privacy and produce comparable data.** The software use locally defined node IDs instead of real addresses to identify both local and foreign nodes in the bipartite graphs, so that no sensitive data leaves the router. Bipartite graphs generated from the same home network are always comparable since the mapping from node address to node ID is consistent over all time for each home network.
- **Ensure data consistency.** All important HNFL data are backed up locally on permanent storage (an external USB drive) right after the periodic traffic data processing. Generally a router crash will not cause inconsistency between collected data and previously uploaded data stored on the remote server.
- **Scale.** The amount of data that can be stored within the router is limited. The adopted approaches should not cause immoderate storage usages while maximizing the utility of the collected measurements.

3.3.1 Forms of Traffic Data

HNFL stores network traffic data in two forms: flows and bipartite graphs.

- **Flow:** A flow is a series of network packets traveling in the same direction and sharing the same set of identities: (i) source and destination IP addresses, (ii) packet protocol type (TCP/UDP/ICMP), and (iii) TCP/UDP port numbers or ICMP type

and code. For instance, an established TCP connection generates two flows, which have reversed source and destination parameters (IP addresses and port numbers) with each other. In addition, a flow records the number of packets and total number of bytes observed. A flow expires after being inactive for a period of time (e.g., five minutes). Packets with the same identity parameters as a recently-expired flow are considered to be a new flow (independent of any transport-level semantics).

- **Edge:** An edge connects one local node and one foreign node and aggregates the the flow data (number of flows, number of packets, and number of bytes) for both directions.
- **Bipartite graph:** A bipartite graph has two disjoint node sets. The node sets represent local nodes and foreign nodes. There are edges connecting nodes from the two sets.

Bipartite graphs provide a much more compact representation of network traffic data than individual flow records, drastically reducing the bandwidth and storage overhead of HNFL on storage and bandwidth while still maintaining the ability to compare device-level traffic data. More precisely, as described in Section 3.4.3), bipartite graphs save about 80% of storage space compared to plain flow records in most cases.

3.3.2 Processing Pipeline

Figure 3.4 illustrates the processing pipeline of HNFL. HNFL has two main components, kernel module *hnfl* and user space daemon *hnflc*. In kernel space, *hnfl* obtains access to raw network packets from specific NetFilter hooks, extracts network flow data and DNS resolution information, and makes the data available to user space through the proc filesystem. The kernel module uses the *seq_file* facility with *procfs*, to facilitate exporting of data items larger than one page from kernel space. In user space, *hnflc* periodically reads traffic data from kernel space (via *procfs*) and

processes it to generate bipartite graphs and update the local data store. Afterward, *hnflc* transmits bipartite graphs to a remote server for archiving, further analysis and research.

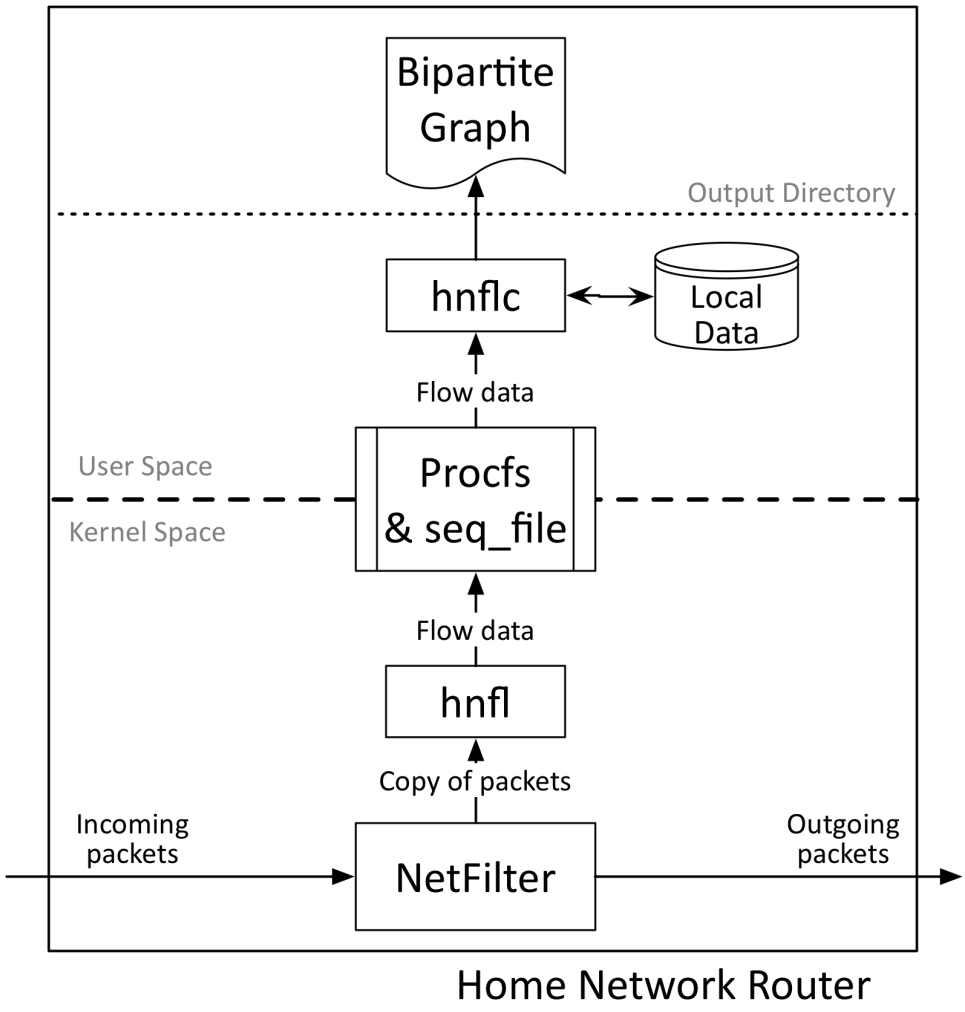


Figure 3.4: Components of HNFL.

3.3.3 *hnfl*: Collect Traffic Data

Figure 3.6 shows the flow of information through the traffic data collector kernel module *hnfl*. The *hnfl* module registers callback function at the following NetFilter

hooks (highlighted in Figure 3.3 and illustrated in Figure 3.5) in order to capture and process desired packets:

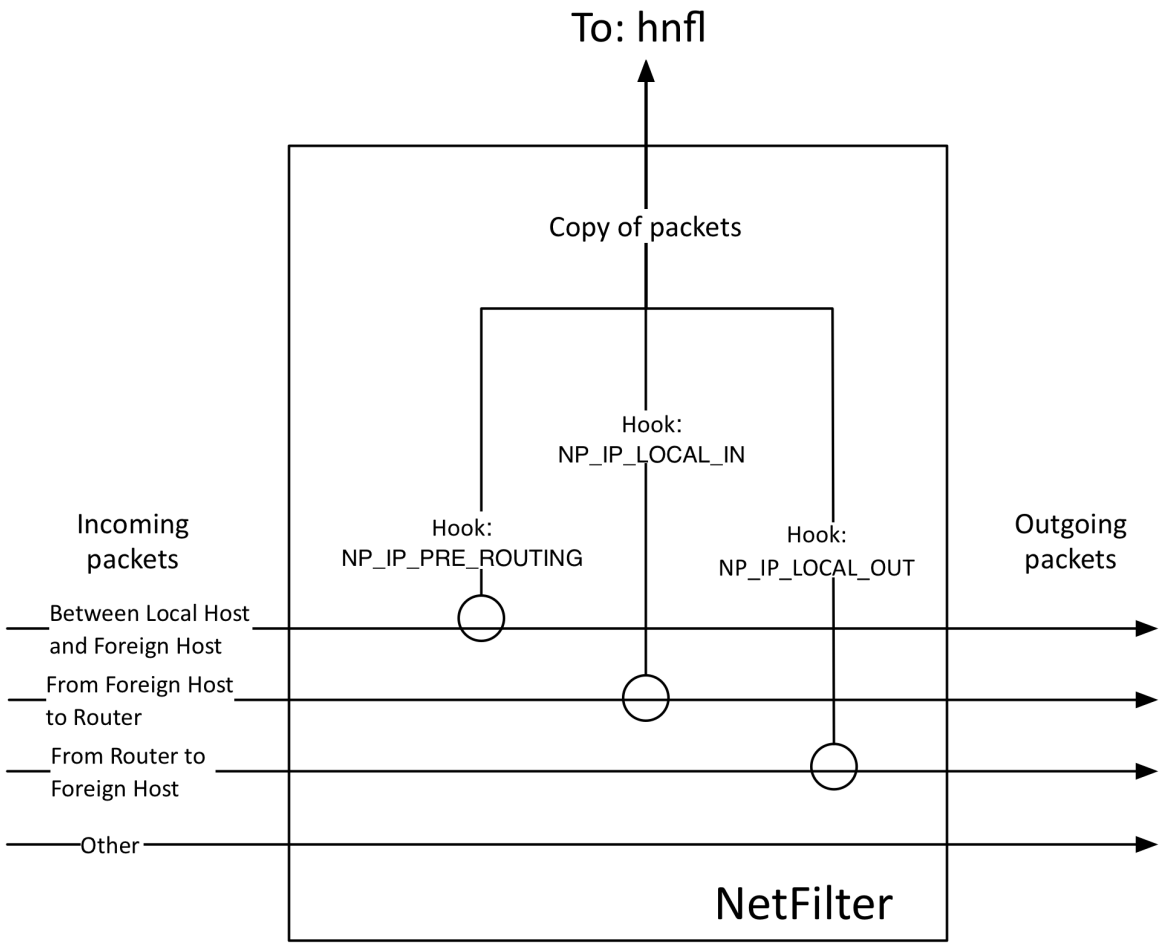


Figure 3.5: Use of NetFilter hook functions

- **NF_IP_PRE_ROUTING.** This hook catches all packets originating from external networks or local hosts within the home network. The *hnfl*'s callback function at the hook is set with the lowest priority, because the NAT module's callback function has a higher priority and modifies packets with destination NAT (DNAT) before *hnfl* sees them. As a result, *hnfl* always sees the true internal IP address of local hosts in packets from external networks. Otherwise, the source

NAT (SNAT) for outgoing packets happens at hook NF_IP_POST_ROUTING, after the processing of packets at hook NF_IP_PRE_ROUTING.

- **NF_IP_LOCAL_OUT.** This hook catches all the packets originating from the router itself.
- **NF_IP_LOCAL_IN.** At this hook, *hnfl* looks for valid DNS packets containing responses to DNS host address queries so as to obtain DNS resource data.

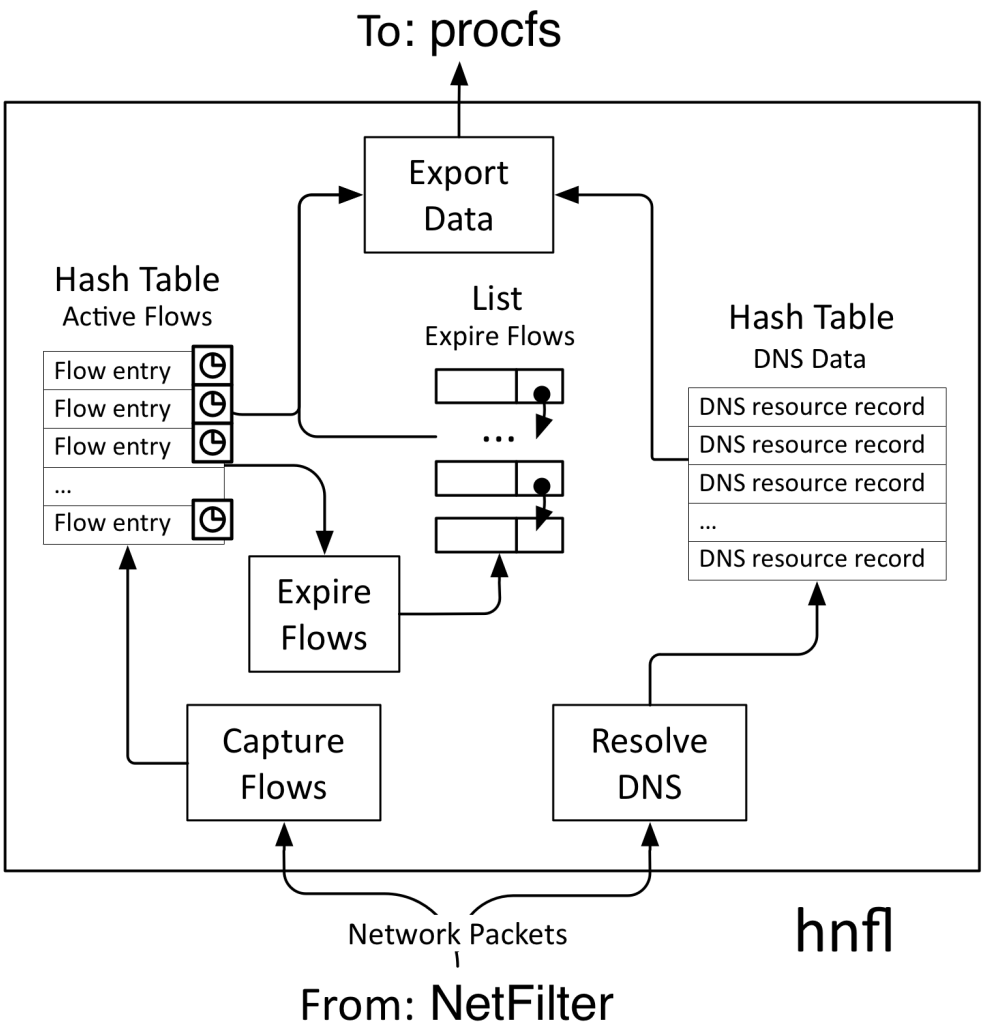


Figure 3.6: HNFL kernel modules: *hnfl* and *hnfl_ dns*.

Figure 3.6 illustrates how network traffic data are processed in *hnfl* before exporting to user space. Information obtained from individual packets are used to

update two kernel hashtables:

- **Active Flow Table** is a hashtable for uni-directional flows. The key of a flow entry is the five-tuple (source/destination IP address/port and protocol number). The flow entry in the active flow table maintains the number of bytes and packets related to the flow for its lifetime and current interval. Each flow entry also has its own timer so that *hnfl* can expire inactive flows after timeout (five minutes) and move them from the active flow table to the list of expired flows.

- **DNS Resource Table** is a hashtable containing DNS resource records obtained from DNS packets responding to domain name queries. Each DNS resource record has two fields: 1) the IP address obtained from the "Answer" part of a DNS packet and 2) the domain name obtained from the "Question" part of a DNS packet. The DNS resource table uses the IP address as the key. For DNS packets with multiple IP addresses in the "Answer" part, the *hnfl* creates or updates a DNS resource record in the DNS resource table for each IP address.

The *hnflc* requests flow data and DNS resource data separately. When *hnfl* receives the pull request for flow data from the proc filesystem, *hnfl* returns data from both the active flow table and the expire flow list. Afterward, *hnfl* clears all data from the expire flow list and resets interval data in each flow within the active flow table to zero. As for DNS resource data, *hnfl* exports all DNS resource record to *hnflc* on request and clears the DNS resource table right after the export. The next section (Section 3.3.4) describes how *hnflc* deals with the flow data and DNS resource data obtained from *hnfl*.

3.3.4 *hnflc*: Process Traffic Data

Figure 3.7 illustrates the user-space daemon, *hnflc*. The *hnflc* periodically checks for router-local network information, including Address Resolution Protocol (ARP) table

and DHCP leases, and pulls flow data and DNS resource data from kernel module *hnfl* through *procfs*. After processing all the data, *hnflc* updates the following four hash tables:

- **Local Host Statistics (LHSTAT).** LHSTAT maintains the following traffic statistics for each observed local host: (i) incremental local ID, IP address, MAC address, and host name; (ii) number of flows, packets, and bytes for the current interval and cumulative over all time; (iii) the lifetime exponential weighted moving average (α=0.1) of flows, packets, and bytes for every interval, and d) the lifetime total number of flows, packets, and bytes.
- **Flows from Current Export (CF).** CF keeps all the flows of the latest flow export from *hnfl*.
- **Recent Flows (RF).** RF keeps the last thirty minutes of flow records. Each exported flow has a sequence number and a version number as unique identifiers. On receiving the latest flow export from *hnfl*, *hnflc* updates an existing flow record if *hnflc* finds a match of the sequence numbers between a new exported flow and an existing flow and if the version number of the new exported flow is higher. Otherwise, *hnflc* adds the flow with a sequence number into RF table.
- **DNS Records (DNSR).** DNSR holds DNS resource records from *hnfl* for 24 hours. For each DNS resource record obtained from *hnfl*, *hnflc* updates the DNSR table with data from the current interval if existing entry for the same IP address is found in the DNSR table. Otherwise, *hnflc* creates a new entry for the DNS resource record. If a DNS resource record is not updated within 24 hours after creation or the lastest update, *hnflc* removes the DNS resource record from the DNSR table.

The user-space program *hnflc* generates two versions of bipartite graph files, by aggregating data from CF, DNSR, and LHSTAT tables: one for anonymized output and another, in the form of JSON files, for the local home network traffic dashboard

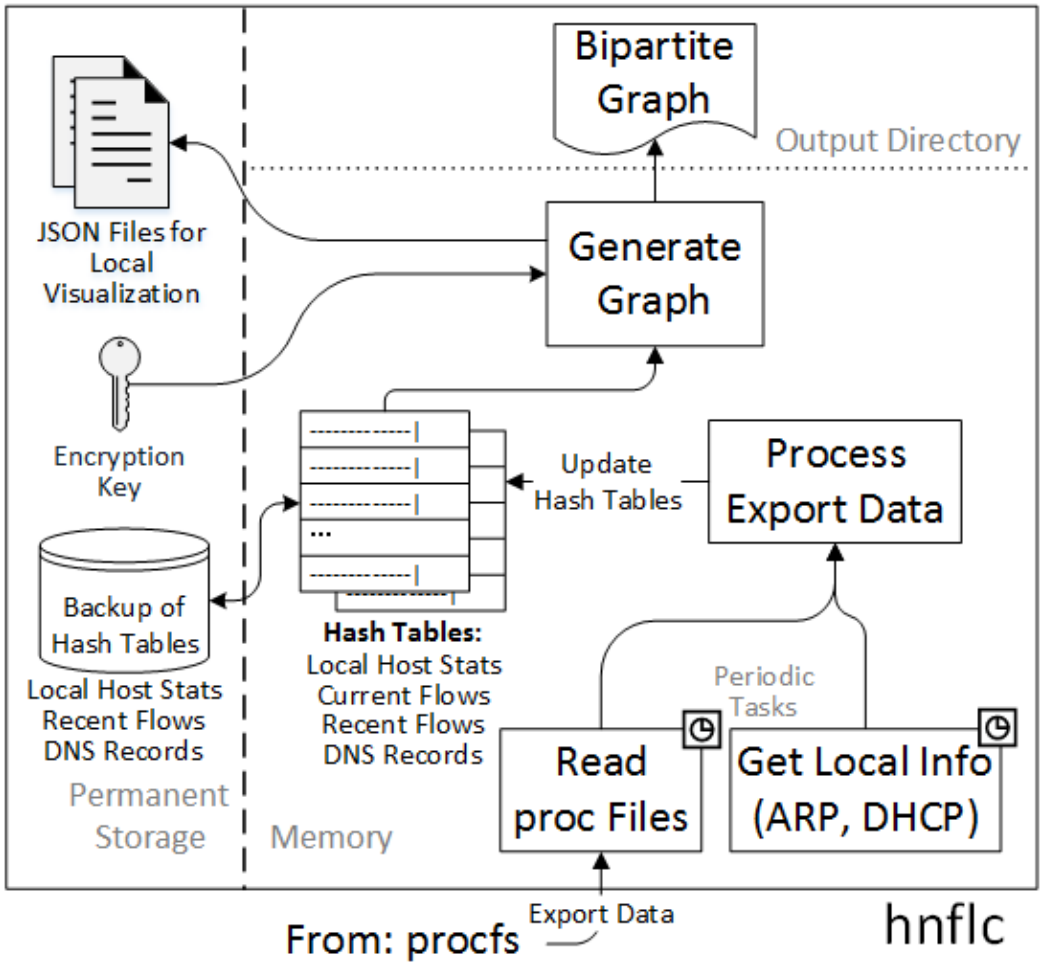


Figure 3.7: HNFL user space daemon: *hnflc*.

(Section 4.1). After finishing each round of processing, *hnflc* cleans CF table, removes expired records from RF and DNSR tables, and backs up LHSTAT, RF, and DNSR tables to the USB drive before sleeping for the next interval.

3.3.5 Anonymizing Endpoints

In bipartite graphs for output, *hnflc* replaces IP addresses of local hosts with local IDs, and encrypts foreign IP addresses using the Advanced Encryption Standard encryption with 128-bit keys (AES-128) [53]. Upon first bootup, *hnflc* creates a random AES encryption key, using entropy from */dev/random*; the key is stored in the flash memory of the router (and nowhere else).

The use of AES-128 encryption ensures a consistent, router-specific, irreversible mapping from IP addresses to opaque identifiers. Initially, 32-bit incremental IDs were used to replace the IP addresses of outside hosts. The mappings from IP addresses to 32-bit IDs are kept locally on the USB drive. When a new flow was seen, *hnflc* checked the mapping store to ensure that the same mapping was always used. However, one of the pilot deployment households had installed a CAIDA "archipelago" measurement box on the home network [7]; about every three days it performed a traceroute to every routed IPv4 /24 prefix. This drastically increased the size of IP-to-ID mapping table and filled up kernel memory. Although this behavior is not typical, the final version of the software uses the stateless AES-128 mapping to ensure a robust framework. Doing so slightly increases the size of bipartite graphs, but eliminates any concerns about scalability of the IP-to-ID mapping. In addition, it has the pleasant side effect of making future support for IPv6 measurement easier.

3.4 HNFL Performance Evaluation

The *hnfl* and *hnflc* software implementations are tested on a NetGear WNDR3800 router with OpenWrt "Attitude Adjustment" release.

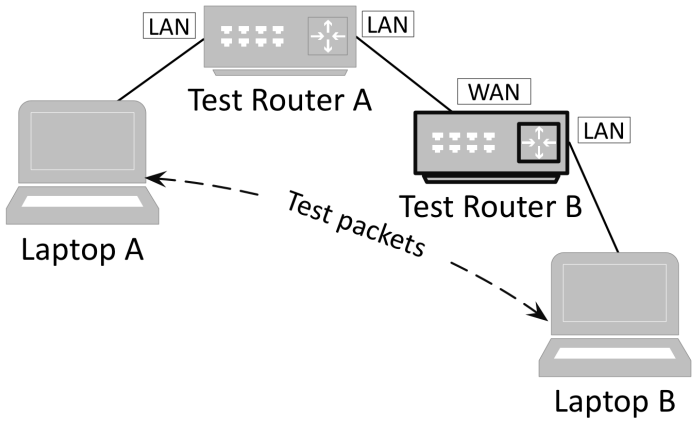


Figure 3.8: Evaluation environment for correctness and efficiency.

The correctness and efficiency of *hnfl* is evaluated in a test environment presented in Figure 3.8. The test environment consists of two Macbook Pro laptops and two NetGear WNDR3800 routers. All devices within the test environment are connected using network cables through Gigabit Ethernet ports. The two NetGear routers run the same OpenWrt operating system. Test router B in Figure 3.8 is the main test platform, which is used to run *hnfl*/*hnflc* and other passive measurement software. Test router A acts as a normal switch. The test router B's Wide Area Network (WAN) interface is connected to one of the Local Area Network (LAN) ports of test router A. Meanwhile, laptop A is connected to another LAN port of test router A. Laptop B is connected to test router B's LAN port. With this setup, laptop B can reach laptop A with the IP address assigned by test router A. The kernel module *hnfl* and other passive measurement tools are evaluated on test router B to compare their performance.

Moreover, the overhead of DNS packet inspection in *hnfl* is evaluated with a simpler test environment as shown in Figure 3.9. In this simpler environment, the test router running *hnfl* is connected to the Internet via its WAN interface. Only one Macbook Pro laptop is connected to the test router's LAN interface using a

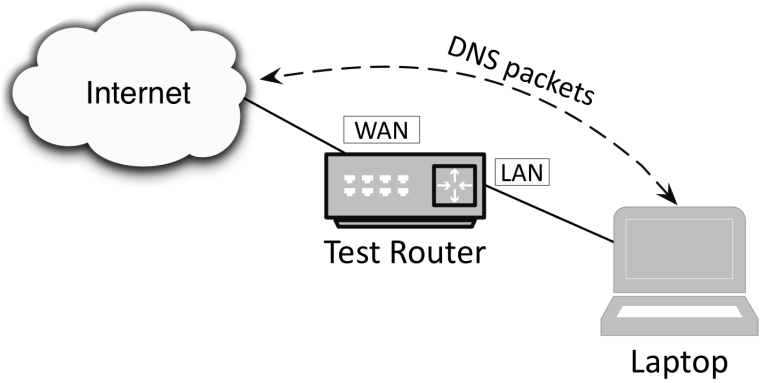


Figure 3.9: Evaluation environment for DNS resolution.

network cable. In addition, the performance of *hnflc* is measured in one of the pilot deployments with real-world home network traffic. The following sections describe the evaluation setup and results.

3.4.1 Data Collection Correctness

Theoretically, NetFilter catches every packet going through the network stack. Thus, *hnfl* should also be able to scan all these packets by correctly using NetFilter hooks. The correctness of *hnfl*'s packet capturing is evaluated by using *iperf* [12] to send UDP packets with different payloads and send rates and comparing the number of packets received on laptop A with the number of packets captured by *hnfl*. If the two numbers are identical, it means that *hnfl* captured all the packets correctly. The *iperf* tool is commonly used for active measurements of the achievable bandwidth on a network path using various protocols (TCP, UDP, and SCTP). The hosts on both side of the network path in test need to run the *iperf* in server mode and client mode respectively before starting actual measurement tests.

Table 3.2 lists the test results collected from iperf report, Wireshark on laptop A, and flow export from *hnfl*. For each test, Table 3.2 shows the number of dropped and received packets from iperf test report, number of forwarded packets from *hnfl*, and

Table 3.2: Result of correctness test

Mode	Send Rate	iperf report		Wireshark	*hnfl*
		# of pkt Received	# of pkts Dropped	# of pkts Received	# of pkt Forwarded
with *hnfl*	5 Mbps	72,818	0	72,818	72,818
forward-only	5 Mbps	72,816	0	72,816	—
with *hnfl*	20 Mbps	291,264	0	291,264	291,264
forward-only	20 Mbps	291,262	0	291,262	—
with *hnfl*	50 Mbps	731,665	0	731,665	731,665
forward-only	50 Mbps	731,652	0	731,652	—
with *hnfl*	100 Mbps	945,195	542,511	945,195	945,195
forward-only	100 Mbps	1,014,143	470,713	1,014,143	—
with *hnfl*	130 Mbps	748,234	1,132,885	748,234	748,234
forward-only	130 Mbps	773,031	1,109,281	773,031	—

Note: The calculation of send rate does not include packet headers (Ethernet header, IP header, and UDP header). The packets sent by *iperf* have the same data payload of 258 bytes.

number of received packets at laptop B from Wireshark. The result from Wireshark is obtained by using rules to filter out UDP test packets. The result from *hnfl* is extracted from the flow export file by looking for data from flows with the IP address pair used in the test. The results from the three different sources are identical to each other. This suggests that the NetFilter hook system of Linux is working correctly and so is *hnfl*.

Packet drop is observed while *iperf* sends packets at a rate higher than the router's packet processing rate (e.g. 100 Mbps and 130 Mbps tests). As shown in Table 3.2, *hnfl* causes about 2%-5% higher packet drop rate in the tests in comparison to forward-only tests. The limiting factor on the router is packet processing speed. When the router cannot process packets fast enough, some packets will be dropped. If the Ethernet Maximum Transmission Unit (MTU) size of 1,500 bytes is fully used, which is about five times larger in size comparing to the packets sent during the test, *iperf* can reach a much higher transmission rate (over 300 Mbps) without packet dropping. Section 3.4.2 further discusses the highest achievable throughput and the impact of the passive measurement infrastructure on router performance.

3.4.2 Data Collection Efficiency

In Figure 3.10, we compare the *hnfl*-enabled router's performance, in terms of achievable TCP bandwidth and latency, with routers using other passive measurement facilities under different conditions:

- **openwrt**: unmodified OpenWrt code
- **hnfl-0**: *hnfl*-enabled router, without initial flow entries
- **hnfl-2K**: *hnfl*-enabled router, with 2,000 random preloaded flow entries
- **hnfl-6K**: *hnfl*-enabled router, with 6,000 preloaded flow entries
- **tcpdump**: OpenWrt router running *tcpdump* configured to capture the first 64 bytes of each packet
- **bismark-p**: the same hardware running the BISmark passive measurement software

The WNDR3800 router has a CPU primary data cache of 32 KB. Each flow entry in *hnfl* takes 144 bytes. Thus, about 227 flow entries can fit in the data cache. In order to simulate real-world traffic and eliminate the performance boost from the data cache, the *nping* [54] tool is used to generate various numbers (100, 200, 400, or 800) of active flows into *hnfl* while doing *iperf* TCP bandwidth tests and *ping* latency tests. Thus, AF-100 in Figure 3.10 indicates 100 active flows in background, and so on. During the test, *nping* inserts one packet for each active background flow per four seconds.

The plain OpenWrt router provides TCP throughput of around 418 Mbps (TCP payload data only) and an average latency of around 2.4 ms. Using *hnfl* decreases the achievable TCP throughput to around 360 Mbps and adds about 0.4 ms to the latency. The performance of *hnfl* drops slightly with more flow entries and more

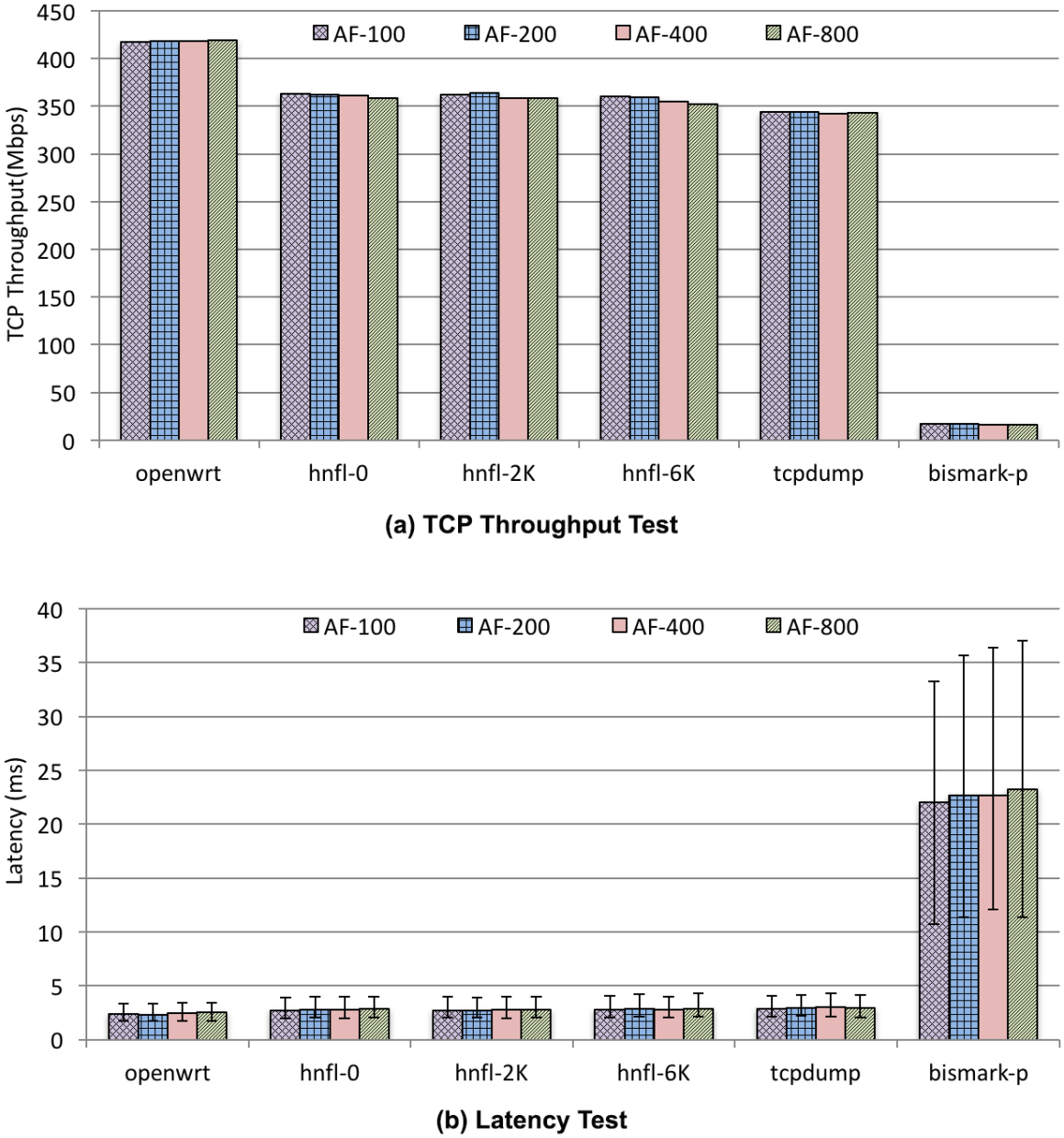


Figure 3.10: Performance of *hnfl*

Table 3.3: DNS query time

	10th Percentile	**Average**	**90th** Percentile
OpenWrt	0.347 ms	0.405 ms	0.463 ms
OpenWrt+hnfl	0.376 ms	0.433 ms	0.501 ms

active flows. However, the reduced TCP throughput on the *hnfl*-enabled router is still adequate to handle the demand of almost all home networks. The same platform using *tcpdump*—that is, capturing packets but not analyzing them—performs slightly worse than *hnfl* while *tcpdump* drops over 80% of captured packets. Contrastingly, the router running the BISmark passive measurement software can only achieve the TCP throughput of 16 Mbps, due to its heavy real-time flow analysis.

Furthermore, the simpler test environment (Figure 3.9) is used to measure the performance of DNS resolving in *hnfl*. OpenWrt routers act as caching DNS servers and give instant response to DNS queries matching the 150 cached queries. Thus, in order to minimize the latency variation in the test, the *hnfl* is modified to resolve DNS responses generated by the router at hook NF_IP_LOCAL_OUT instead of the hook NF_IP_LOCAL_IN in design and connect the testing laptop to the router using wired link. The *dig* [55] tool is used to query ten popular domain names each for 100 times, with and without *hnfl* loaded, separately. Meanwhile, the *Wireshark* software is running on the testing laptop to capture DNS packets and calculate the latency between each pair of DNS query and response. According to the test result, the latency cost of *hnfl*'s DNS snooping—which stores mappings between foreign IP addresses and DNS names, for (local) presentation to the user, see Chapter 4.1—is usually less than 0.04 ms (see Table 3.3).

3.4.3 Data Processing Efficiency

During a consecutive seven-day period, we recorded *hnflc*'s time cost for each separate processing step: i) process data from *hnfl*, ii) clean an back up data, and iii) generate graph files. The test includes data from 2,016 consecutive five-minute intervals.

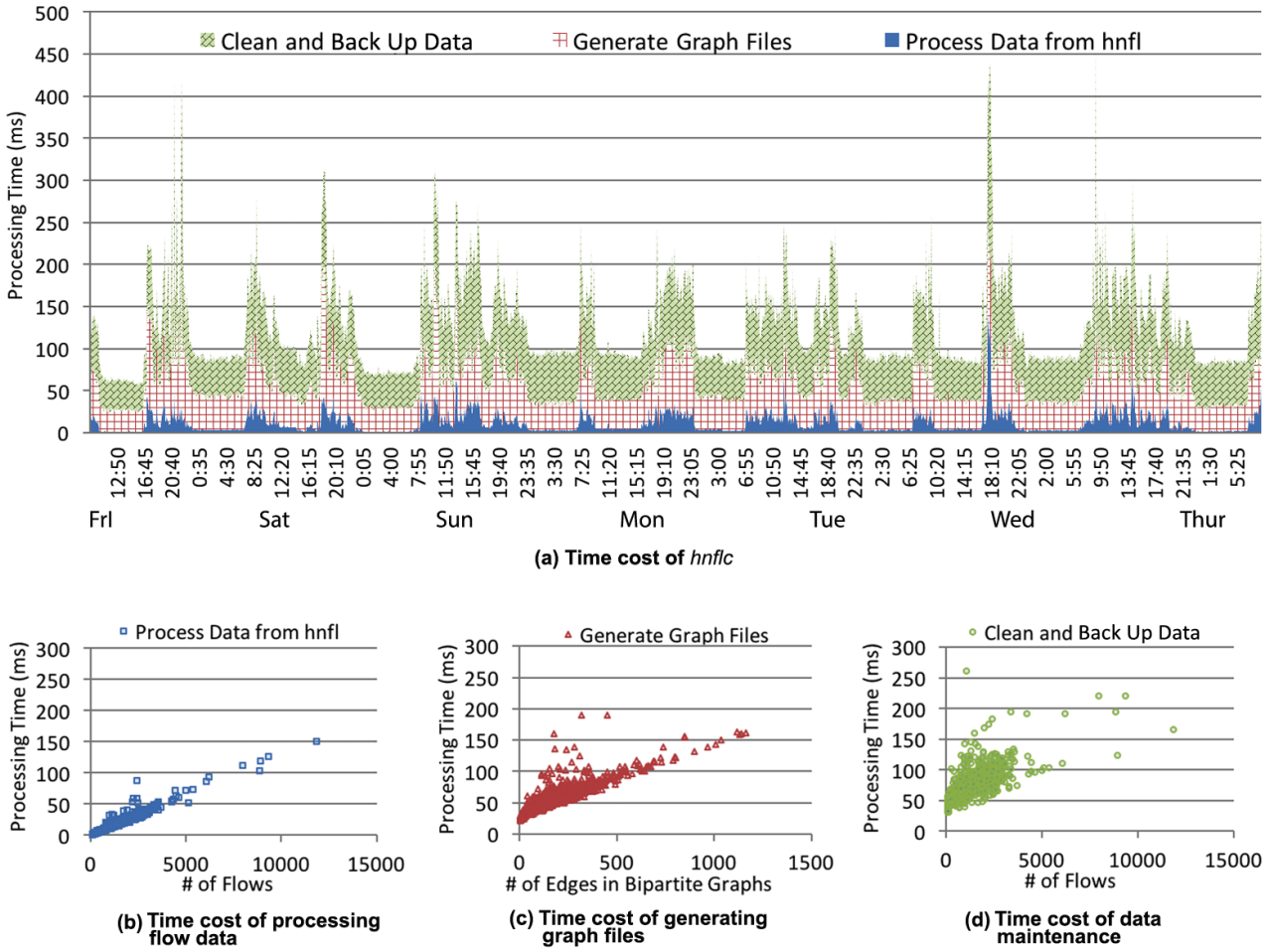


Figure 3.11: Performance of *hnflc*

Figure 3.11 presents test results. The *hnflc* finishes most rounds within 200 ms (Figure 3.11-a). The total processing time can be divided into three parts: processing data from *hnfl*, generating bipartite graph files, and maintaining data in local hash tables. The time cost of processing traffic data has a close-to-linear relationship with number of flows exported from the kernel, while the time cost of reading DNS records from *hnfl* is negligible. As shown in Figure 3.11-b, it takes about 50 ms to process 4,000 flows into edges of a bipartite graph. However, according to Figure 3.11-c, it is more expensive for *hnflc* to generate the final bipartite graph files—including anonymized edges for uploading and a set of JSON files presenting different graph parameters for local visualization. It takes about 100 ms to arrange 600 edges into different graph files. Different graph topologies also cause a variation in the time cost of generating graph files. The *hnflc* spends another large part of processing time to maintain local hash tables Figure 3.11-d. The time costs of generating graph files and maintaining hash tables are higher and more variable due to expensive disk I/O operations on the USB drive.

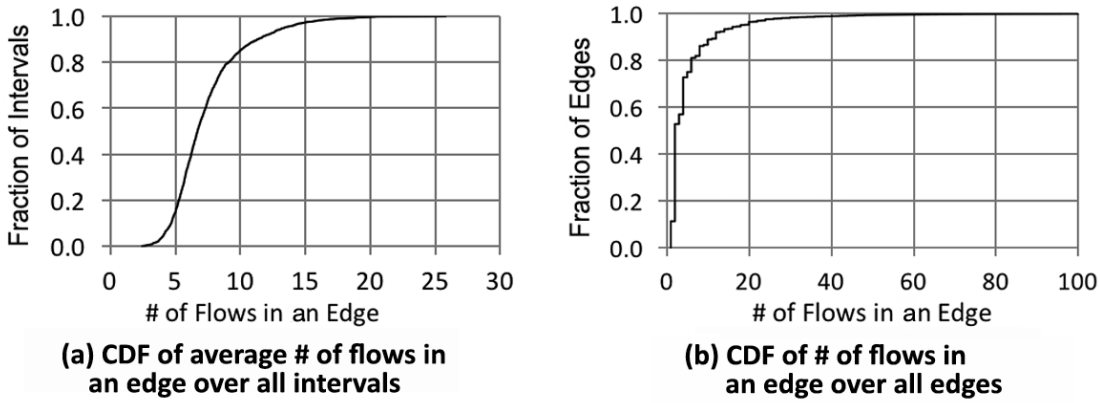


Figure 3.12: Relationship between flows and edges

Furthermore, Figure 3.12-a plots the average number of flows aggregated in an edge over 2,016 *hnflc* intervals. In almost 85% of intervals, the number of active flows is more than five times of the number of active edges. Figure 3.12-b illustrates the

detailed per-edge flow counts with data from all 268,417 collected edges. About 53% of all edges have one or two flows; 44% of edges have three to twenty flows; 3.6% of edges have more than twenty flows, ranging from 21 to 1,410 flows. The edges with dozens of flows may be the result of using multiple separate flows to download resources from large websites, such as images and videos from amazon.com or facebook.com.

3.5 Data Upload Subsystem

While HNFL modules are working hard to generate network measurement data in home networks, the data upload subsystem is also running once a minute to send the data over to the collection server in the lab. Otherwise, the passive measurement data will eventually overflow the data storage on each router or negatively affect network experience of the corresponding household by sending a large amount of measurement data at once.

The data upload subsystem is composed of the centralized collection server as the receiver, all active measurement routers as senders, and the secure transmission channel.

3.5.1 Sender

On measurement routers, the data files generated by *hnflc* are temporarily stored in a staging directory. The sender (or uploader) module is a set of scripts that check the staging directory periodically and send any file found in the directory to the remote server. The data files are removed from the step directory after successful uploading. If the sender script failed to upload due to reasons like server unreachable or network outage, the sender will leave the file in the directory and check back at a later time.

A cron job is scheduled to activate sender scripts at the start of every minute. To avoid the situation that all the routers in the field send data files to the collection

server at the same time, the sender scripts will wait for a random amount of time (from 0 to 45 seconds) before sending the files.

Every router is equipped with an 8GB USB drive. A routine system maintenance script checks the existence of the USB drive. If the USB drive is found, the staging directory will be mounted on the USB drive. Otherwise, the staging directory will be mounted on the temporary directory (/tmp) of the router, which has limited storage size and loses its data once the router is powered off.

The files generated by *hnflc* differ in size. However, they have an average size of 10 KB in general. Thus, as *hnflc* generating files at the rate of one per five minutes, the 8GB USB drive can hold more than seven years of data before its storage is full.

3.5.2 Receiver

The receiver is a PHP[5] application running on the collection server, which is publicly accessible from the Internet. Before running the PHP code to accept the file from the sender, the server and client authenticate each other using mutual Transport Layer Security (TLS) authentication, which is discussed in Section 3.5.3. If the authentications failed, the sender will stop the transaction immediately and try to authenticate with the receiver and upload measurement data one minute later.

On receiving any incoming files, the receiver checks the format of the file name and decodes the router ID and time stamp information from the file name. If the file name is badly formatted, the receiver will discard the file immediately. Otherwise, the receiver saves the file to its corresponding directory on the server's storage according to the router ID and time stamp. Section 5.2 describes the storage of data.

3.5.3 Secure Transmission

The data transmitted between the senders (routers) and the receiver (the data collection server) are sensitive network traffic data. Even though the data are already

[5]PHP: a server-side scripting language designed for web development.

anonymized, malicious parties may still unveil data that could violate the privacy of participants with some background knowledge. Therefore, a secure transmission channel between sender and receiver is required.

To ensure the transmission security, the connection between the sender and the receiver is secured using Transport Layer Security (TLS) with mutual authentication. Every deployed router carries a pair of key and certificate generated by the same Certificate Authority (CA), which is operated and maintained on an offline computer located in a locked room in the lab. The web server on the collection server is also configured with a pair of key and certificate from the same CA. A successful data file upload requires that both sender and receiver verify each other's certificate during TLS handshake.

3.6 Remote Update Subsystem

User privacy and security are primary considerations in the design and implementation of the data collection system. Once a router is deployed in a participant's household, it is no longer under direct control of the experimenter. The routers in the field run standard firewall software to protect themselves from unauthorized access from the Internet. Since the use of remote access protocols like SSH are restricted to connections from local devices within the home networks, a method to remotely instruct the router in trouble to recover is required. Therefore, the system includes a remote update subsystem to establish contact with deployed routers and initiate package updates and maintenance routines on target routers.

3.6.1 Daily Update Routine

Every deployed router runs a daily update routine, which is a set of scripts scheduled to run from 05:00 a.m. to 08:00 a.m. (Coordinated Universal Time or UTC). As almost all of our deployed routers are in the Eastern Time Zone (ETC), the scheduled

update routine runs at midnight. For the one participating household located in the Central Time Zone (CST), the operating time of the daily update routine is still during usual off-peak hours of Internet usage.

The update routine has three steps:

1. **Status Check-in:** At the hour of 05:00 UTC, the router creates a file containing a list of installed packages (package name and package version) in the upload directory for check-in files. The specific minute in that hour for status check-in is set before deployment to disperse the check-in time of all deployed routers. After that, the data upload subsystem (see Section 3.5) periodically checks the upload directory and transmits the files found to the corresponding interface on the collection server.

2. **Update Check-out:** One hour after the status check-in, the router proactively requests a list of available updates from the server. The response file is saved in a directory for pending updates. In the response file from the server, the router can find the name and download URL (Uniform Resource Locator) of any available update.

3. **Update Installation:** One hour after the update check-out, the router downloads the updates appearing in the update list using the provided URL to the temporary directory. Afterward, the router installs the update packages and removes the file containing the update list.

3.6.2 Server Operations

There are three types of updates available to routers during the measurement study:

Software Update: Although all essential features of the measurement system are complete before deployment, it may still be necessary to push a fix of bugs to all deployed routers. Software updates are pushed to all active routers when

available. When an update check-out request is received on the server, the request handler will push the new update package to the router if it finds out that the router is running an older version of the package according to the package information received during the corresponding router's latest status check-in.

Configuration Change: Some features of our software are configurable. For example, the experimenter can turn on the optional collection of Organizationally Unique Identifier (OUI) information from MAC addresses upon receiving the user's agreement. For configuration change updates, the experimenter needs to manually assign a list of target routers of a specific package to the update handler on the server. The update handler will include the information of the update package in the response message to target routers.

Maintenance Routine: The "health" condition of all deployed routers is monitored by the centralized collection server. Once a router is found acting abnormally, the experimenter instructs the update handler to push the corresponding maintenance package to the affected router. More details about the maintenance routine are discussed in Section 5.3.

Chapter 4

In-home Applications

Along with the HNFL measurement modules, user-facing applications were developed to improve home networking experience. The OpenWrt system comes with a Lua Configuration Interface (LuCI) [56] package to enable users to manage their home networks through a web interface hosted locally on the router. However, while LuCI is very useful to conduct configuration and management tasks using a computer, it lacks the ability to provide advanced visualization of network usage. Also, while smartphone users may access the web interface, it is not designed for small screens, and does not provide much security.

This chapter introduces the two applications designed for the users of measurement routers: 1) a web dashboard presenting per-device usage information about the home network and 2) an iPhone application enabling users to conduct basic configuration operations on the home network.

4.1 Home Network Traffic Dashboard

The web dashboard is a tool for users to see the usage of the local home network. Users can check essential information considering active network devices, device level upload/download bandwidth usage, and top Internet destinations.

4.1.1 Dashboard Interface

The dashboard is implemented in the form of a web page using the Data-Driven Documents (D3) JavaScript library [57] to visualize the resulting bipartite graphs generated by *hnflc* to router users. The dashboard is hosted on the router using a lightweight web server package on OpenWrt–uHTTPd [58]. For the convenience of users, a local domain name (http://myrouter.home) is configured on the router by adding one entry in the dnsmasq[1] configuration. Users can use the domain name instead of the IP address of the router to access the dashboard.

The dashboard is composed of five tabs (see Figure 4.1 and Figure 4.2):

- **Connected Hosts.** There are two graphic sections in this tab showing the number of connected Internet hosts for each active home network device for both the latest measurement interval (Current) and the exponential moving average values (Average) with $\alpha = 0.1$.

- **Outgoing Packets.** This tab presents the usage data regarding the uplink by showing the number of packets transmitted. In addition to *Current* and *Average* sections, this tab displays an extra section, named *Top 5 Links*, which illustrates usage data of individual links between local devices and Internet hosts. The links are ranked by numbers of packets transmitted during the five-minute measurement interval. Usage data of links other than the top five are concatenated to a link connecting to the "# other hosts" node.

- **Outgoing Bytes.** This tab has the same structure as tab *Outgoing Packets*. However, the measurement parameter used in this tab is the number of bytes transmitted in uplink instead.

[1]dnsmasq: a lightweight Domain Name System (DNS) forwarder and Dynamic Host Configuration Protocol (DHCP) server designed for resource constrained routers

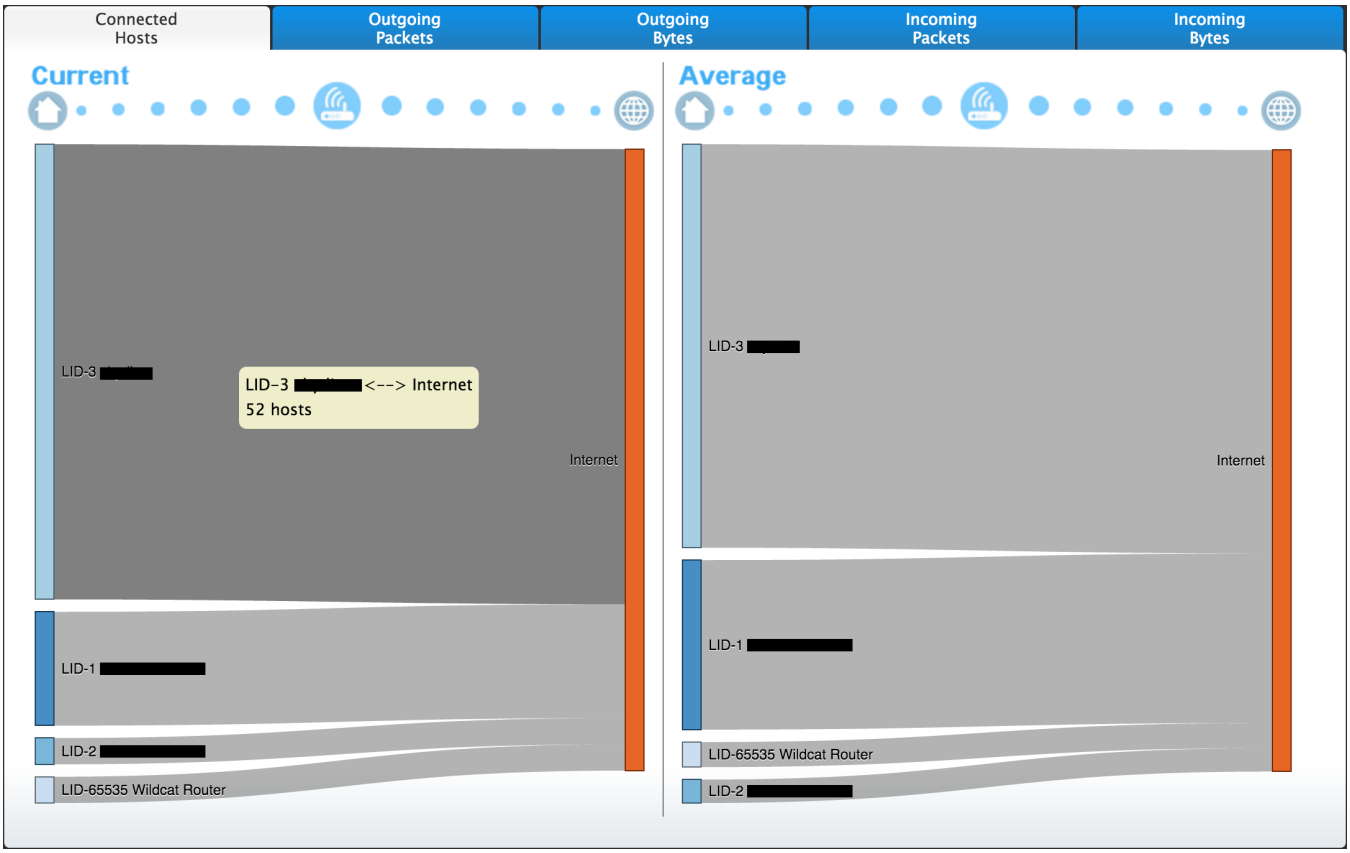


Figure 4.1: Dashboard: overall connections

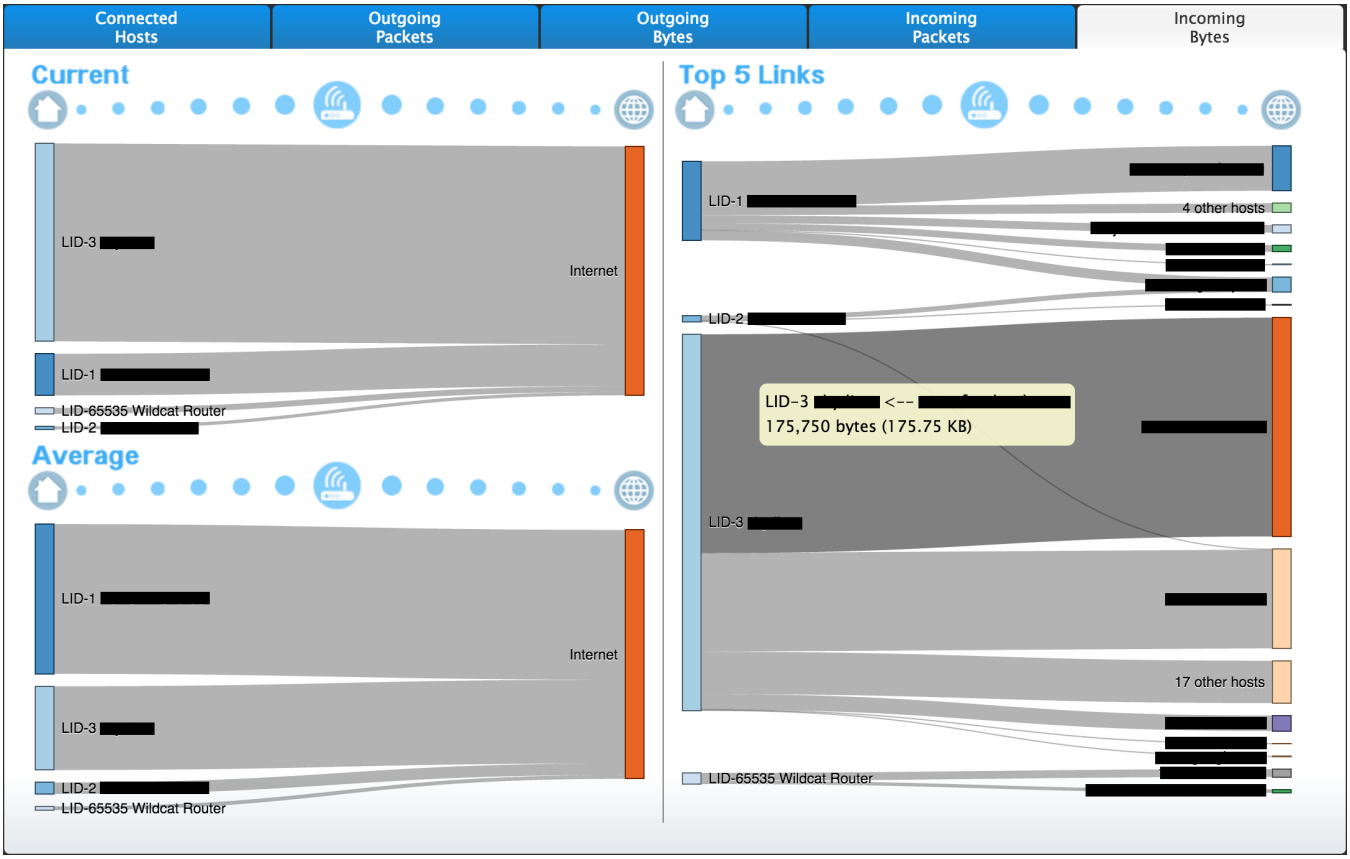


Figure 4.2: Dashboard: incoming number of bytes

- **Incoming Packets.** The same three-section display structure is used in this tab to present network usage by the number of packets transmitted in downlink.
- **Incoming Bytes**. Illustrate the network usage by the number of bytes transmitted in downlink.

There are two to three graphic sections in each tab. Each graphic section has four components:

- **The top bar** carries the name of the section and graphic badges indicating the rest of the components in the graphic section.
- **Local device nodes** are located on the left side of the graphic section. The text label attached to the node is the name of the local device.
- **Internet host nodes** are located on the left side of the graphic section. The text label attached to the node is used to identify an Internet host. The text can be the domain name, IP address, or an anonymized site ID for the Internet host depending on the user's configuration and if HNFL possesses the domain name information of the Internet host.
- **Edges** connect local device nodes to Internet host nodes.

To interact with the dashboard, the user can click on one of the five tabs to check the corresponding measurements. In each graphic section, the user can move the mouse over the node and path to activate a pop-up information box showing the usage details regarding the selected node or path.

The dashboard automatically refreshes itself with latest data from *hnflc* by default. However, users have the ability to disable the auto-refresh function through an on-screen checkbox. Otherwise, users can choose to display the names of Internet destinations by their domain names/IP address or by anonymized site ID numbers.

By default, All Internet destinations on the dashboard are displayed by anonymized site ID numbers due to the privacy concerns from users of the pilot deployment. More details about using the dashboard are explained in Appendix A.1.

4.1.2 Source of Data

As discussed in Chapter 3.3.2, *hnflc* generates a new bipartite graph file per five minutes according to the measurement data obtained from kernel module *hnfl*. The *hnflc* further process the bipartite graph files into smaller separate JavaScript Object Notation (JSON) files for each graphic sections in the dashboard. The content in JSON file is organized in the following format so that the visualization module using D3 library can render interactive traffic graphs efficiently:

```
{"nodes":[
  {"name":"LID-1 Xuzi-iPhone",
   "mac":"aa:aa:aa:aa:aa:aa",
   "ip":"10.40.162.214"},
  {"name":"LID-65535 Wildcat Router",
   "mac":"bb:bb:bb:bb:bb:bb",
   "ip":"192.168.0.1"},
  {"name":"Internet",
   "mac":"N/A",
   "ip":"N/A"}],

 "links":[
  {"source":0,
   "target":2,
   "value":623124},
  {"source":1,
   "target":2,
   "value":15878}]
}
```

The JSON file example listed above contains the data for a "Current" graphic section similar to the one shown in Figure 4.2, which consists of nodes and links. A node has three parameters: 1) the name, 2) the MAC address, and 3) the IP address. A link also has three parameters: 1) the index of the inside node, 2) the index of the outside node, and 3) a value for a specific traffic measurement. The node indexes

are integers indicating the position of corresponding node listed in "nodes" starting from 0. In the example, there are three nodes, an iPhone device, the router, and the Internet. An inside node is located on the left side of a graphic section while a target node is located on the right side. In a graphic section, a node can only appear as an inside node once and as an outside node one or multiple times. However, it is important to make sure that no one node is both an inside node and an outside node.

The resulting JSON files are stored in a fixed directory on the attached USB drive. Meanwhile, a symbolic link to the directory is created in the web server's directory for the dashboard to access the JSON files.

4.1.3 Authentication and Restricted Access

The web server is configured to be accessible only from the inside home network. In other words, a user can only access the dashboard web interface when his/her network device (e.g. computer or smartphone) has an IP address on the home network using wired or wireless connections.

Furthermore, the dashboard interface asks the user for proper credentials before granting access to the user, which is the administrator account and password randomly generated for the router at the time of deployment. The credentials are printed on a label attached to the back of the router. The user can modify the password through the included configuration web interface.

4.2 iOS App for Basic Home Network Monitoring and Management

The iOS[2] mobile app, myHomeNet, is designed to prove that a mobile app can be beneficial to home network users. The myHomeNet app, which was implemented in 2012 under iOS 5 system, is a proof-of-concept application demonstrating that home network users can monitor and manage some basic part of the router. At that time, there was no mobile app available on the market for users to manage their routers. Using a web browser to access the router-hosted web interface was virtually the only choice to manage home networks. Later on, major home network router manufacturers rolled out their mobile app to accompany their router products, such as NetGear's Genie [59] (2012) and TP-Link's Tether [60] (2013).

4.2.1 User Interface

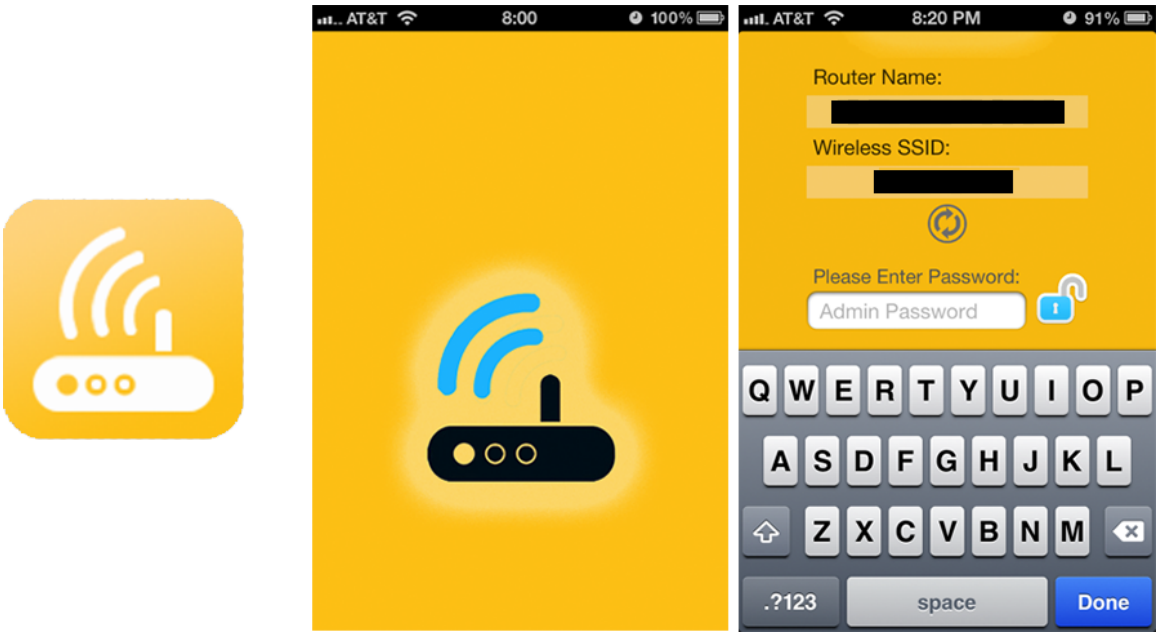


Figure 4.3: Homenet Control: icon and login view

[2]iOS: is a mobile operating system created and developed by Apple Inc.

The myHomeNet app has a simple tab view as presented in Figure 4.3 and Figure 4.4. Each tab shows some information about the home network router, while some tabs also allow the user to modify the corresponding configurations of the home network. The four tabs are:

- **Addresses** lists the essential network addresses about the home network:
    1. Local Area Network (LAN) address of the router
    2. Wide Area Network (WAN) address of the router
    3. Gateway address of the access network
    4. Domain Name Service (DNS) server address
- **Traffic** shows the overall network usage of the whole home network
    1. Number of bytes that the router has received and sent since its latest boot-up
    2. Current overall upload and download transmission rate of the home network
- **Wireless** is the place where the user can check and modify the wireless settings of the network, including both Service Set Identifier (SSID) and password of any available wireless Access Point (AP).
- **QoS** provides the following interfaces to the user:
    1. Enable or disable QoS on home network router
    2. View the current QoS rules
    3. Check and modify the speed limits for the whole home network

Figure 4.4: Homenet Control: main function views

4.2.2 Application Server on Router

The application server, iServer-ssl, is implemented in the C programming language. Once installed, the iServer-ssl program is always running on the router in order to handle the requests from the myHomeNet app. Most of the work of iServer-ssl is accomplished by interacting with the Unified Configuration Interfaces (UCI) of OpenWrt [61]. UCI is a very useful utility originated in OpenWrt and is intended to centralize the whole configuration of a device running OpenWrt system.

The iServer-ssl uses the following UCI commands to respond to the requests from myHomeNet app:

- Requests from the "Addresses" tab:

```
// LAN IP address
uci show network.lan.ipaddr
// WAN IP address
uci show network.wan.ipaddr
// WAN gateway address
uci show network.wan.gateway
// DNS server address
uci show network.wan.dns
```

- Requests from the "Wireless" tab:

```
// check wireless SSID
uci show wireless.wifi-iface.ssid
// check wireless password
uci show wireless.wifi-iface.key
// set new SSID
uci set wireless.wifi-iface.ssid = [NEW SSID]
// set new wireless password
uci set wireless.wifi-iface.key = [NEW PASSWORD]
```

- Requests from the "QoS" tab:

```
// check QoS status
uci show qos.wan.enabled
// check QoS upload speed limit
uci show qos.wan.upload
// check QoS download speed limit
uci show qos.wan.download
// enable or disable QoS
uci set qos.wan.enabled=[0|1]
// set QoS download speed limit
uci set qos.wan.upload = [A POSITIVE INTEGER]
// set QoS upload speed limit
uci set qos.wan.download = [A POSITIVE INTEGER]
```

Regarding the requests from the "Traffic" tab, iServer-ssl responds to the requests by reading and processing traffic statistics files from the router system. The location of files may change depending on the router platform and OpenWrt version. For example, on the NetGear WNDR3700v2 router running OpenWrt Attitude Adjustment release, the following two files are used to calculate data transmission rates:

- Bytes received:

```
/sys/devices/platform/ag71xx.1/net/eth1/statistics/rx_bytes
```

- Bytes sent:

```
/sys/devices/platform/ag71xx.1/net/eth1/statistics/tx_bytes
```

4.2.3 Authentication and Secure Communication

The communication between the app client on smartphone and the app server on the router is using TCP sockets with TLS authentication.

As mentioned in Chapter 3.5.3, a public-key certification authority (CA) is used to issue and manage certificates used to create secure connections. Both the myHomeNet app and the application server use the key and certificate issued by the CA to secure

the communication. All communication traffic between the application client and the server is secured by mutual TLS. The application client and server need to present their certificates, signed by the same CA, when they establish the TLS session. Furthermore, the app asks the user for the administrator password when the user opens the app or brings the app to the main screen from background mode (see Figure 4.3).

Chapter 5

Deployment and Data Collection

Home networks are individual semi-closed networks living on the edges of the global Internet. Currently, it is hard to find a central place to look for detailed data about home networks. To obtain first-hand measurement data about home networks, researchers have to reach out to individuals. This Chapter describes the details on recruiting measurement participants, collecting measurement data from residential households, and organizing data from all participating households in a way to facilitate further analysis. This chapter also discusses the approach to keeping deployed routers up-to-date and healthy.

5.1 User Recruitment and Router Deployment

Participants were recruited through emails sent to university and local mailing lists, and through word-of-mouth. All prospective participants completed a screening questionnaire to ensure that their home networks were technically suitable. Project personnel contacted qualified participants who then provided informed consent. Participants received $50 cash (and later another $50 to extend the study) and were allowed to keep the router at the end of the study. The informed consent materials assured participants that no personal information would be made public. The study eventually recruited 53 households, most of which were in the Lexington, Kentucky area. (A few were out of state.) Project personnel—i.e., students from the University

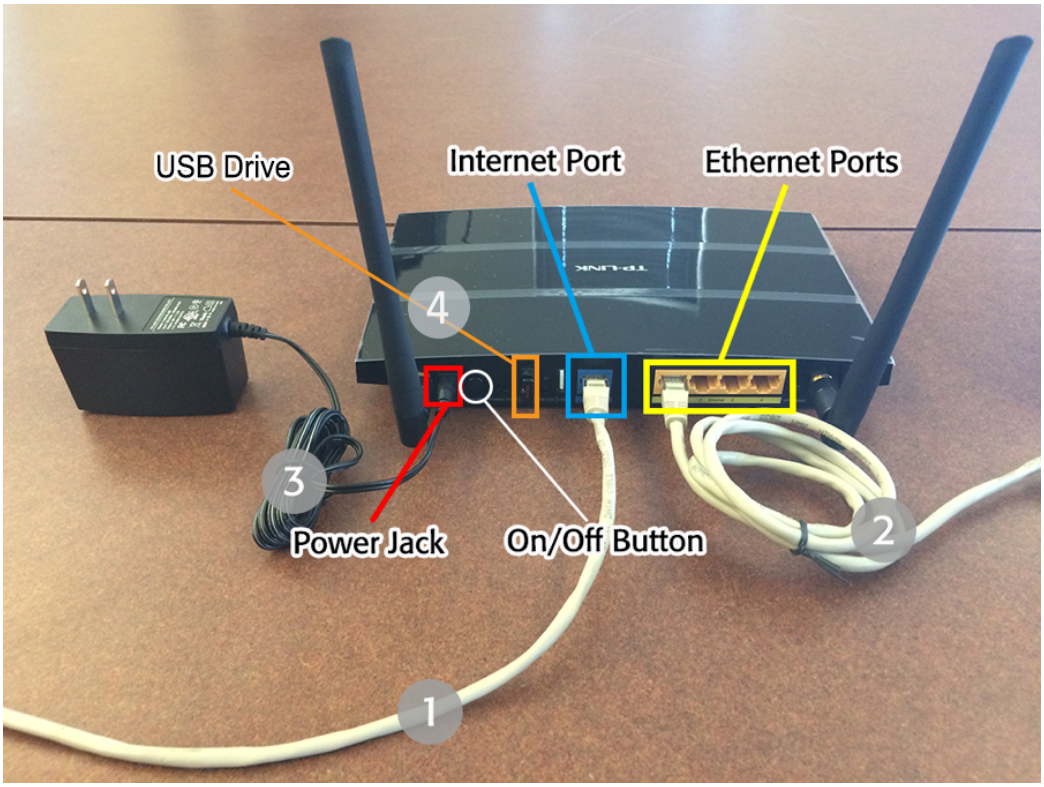


Figure 5.1: The deployed measurement router. Photograph by the author.

of Kentucky—installed and configured most of the routers in homes, to ensure that the new router duplicated the user's existing setup exactly. Figure 5.1 shows the hardware components in the deployment packages. The marked components in Figure 5.1 were explained to users upon installation.

Routers were installed over a five-month period from April to September 2015; the study was concluded at the end of March 2016. The study included households with a range of demographics, from multi-student apartments to families with young children to working couples with no children at home. (Household demographics were optionally provided at the time of informed consent.)

5.2 Data Management

On the collection server, all received measurement data are organized in a manner that meets the following requirements are met:

- Prevent data loss or damage
- Restrict data access to certain project personnel with permission.
- Facilitate fast and agile data analysis

Accordingly, two forms of the measurement data are maintained: individual files in the file system and organized data in the database.

5.2.1 Format of Uploaded Data

The first line of an uploaded data file summarizes the data contained in the file. Each following line in the file describes an *edge* in the interval. Each *edge* in the uploaded data is labeled with:

- Inside host identifier
- Outside host identifier
- # of flows, packets, and bytes between the hosts.
- The outside port # used by the most flows.
- The number of flows using that port.
- The outside port # used by the second-most flows.
- The number of flows using that port.
- The outside port used by the third-most flows.
- The number of flows using that port.
- The protocols of flows belonging to the edge.

- The number of intervals over which the edge has been continuously active. If this edge was not in the graph for the last interval, this value is 1; otherwise, this value is 1 plus the value of this parameter in the graph for the last interval.

Outside hosts are identified by a 128-bit identifier, which is the result of encrypting the endpoint's IP address using AES with a router-specific key. This key is randomly generated at setup time and never leaves the router. Inside hosts are identified by their unique MAC address, but the router replaces the MAC address with an ordinal number in the uploaded data. An *edge* corresponds to the existence of one or more flows in either direction between a unique inside host and a unique outside host in a particular five-minute interval.

For router deployments that occurred after August 2015, the experimenter asked for (and mostly provided) a separate informed consent form to collect the Organizationally Unique Identifier (OUI) of the IEEE 802 MAC addresses used by devices in the household. Each OUI is assigned by the Institute of Electrical and Electronics Engineers to a manufacturer and provides useful information about the types of devices in the household. If a household gave this consent, uploaded information also included the OUI of each edge's inside host in the inside host identifier.

5.2.2 Data Storage and Backup in File System

The measurement data are uploaded to the collection server from deployed routers in the form of text files. The files are stored on the local disk drive of the collection server. In the data directory, the files are organized in subdirectories. Each router has its own directory in the primary data storage directory. Furthermore, data files are separated into smaller directories by calendar days according to the timestamp encoded in the file name. The layered structure accelerates the access to individual data files.

A cron job checks the data directory for any newly added files every minute. Once a new file is found in the data directory, the scheduled task will copy the file to Network Attached Storage (NAS), which is connected to the same local network, via Network File System (NFS) protocol. The files are organized in the same directory structure in the NAS.

In addition, another copy of all data files is stored on an external hard drive, which is stored securely.

5.2.3 Data Organization in Database

While text files are suitable for permanent data storage and access of particular measurement values, it is very hard to conduct data analysis based on data in raw text files. For this reason, a MySQL database was created to promote the efficiency of data analysis.

All data files are processed to fill in the following database tables:

- **Router Information**: stores basic information about specific routers, such as time zone, measurement start and end time, and the number of uploaded files.
- **Router Edges**: stores traffic data regarding network edges relevant to the router itself. The router-specific data, the edges with the router as the inside node, are organized separately in order to avoid "pollution" from irrelevant network traffic such as measurement data upload.
- **User Edges**: stores all network edges generated by devices connected to participating home networks.
- **Devices**: stores the identifiers and Organizationally Unique Identifier (OUI) part of MAC addresses of devices that have network activity during the measurement period.

- **MAC OUI**: maintains a map from MAC OUI to manufacturer names, which is obtained from IEEE [62].

More details about the database tables mentioned above are available in Appendix B.

5.3 Maintenance of Deployed Routers

All computer systems may fail. The routers deployed are no exception. When a problem comes along, the experimenter needs to detect the problem and provide a fix. This section describes the details about detecting and reacting to problems that surfaced during the measurement study.

5.3.1 Router Status Dashboard

Before deploying the routers to participating households, a server-side web dashboard is created to monitor running status of deployed routers. Only authenticated project personnel have the credentials to access the dashboard. The data received from the router are used to infer the running status of both measurement software and router itself:

- **Router Health**. The network forwarding and routing functions of the router are considered as healthy if the server receives the router's measurement data on time for each five-minute interval. Otherwise, a router is in "critical" condition if the server does not receive data from the router for an extended period. The severeness of the critical condition is classified into six health levels depending on the duration of connection loss (see Figure 5.2).

- **Software Health**. The measurement software on a router is considered as healthy if received data files contains valid data. The same standard (Figure 5.2) applies to the level of software health.

Figure 5.2: Color codes for the router status

There are many situations that would cause the router or measurement software to enter a critical condition. The experimenter reacts to the situations when an orange or higher level (to the right end of the health color code bar as shown in Figure 5.2) is observed through the server-side dashboard.

5.3.2 Email/Phone/Onsite Services

Because most of the participants of the home network measurement study have limited knowledge about networking, users were given both a telephone number for a "help line" and several email addresses to contact in the event of a problem.

The "help line" telephone number was configured to ring one of the project personnel's phone at all times. Only a handful of calls were received over the duration of the study (including one after the study had ended and routers were disconnected). Some users prefer using emails to ask for help. Except for the problems with using the configuration interface, virtually all other problems were due to hardware issues related to the router or modem.

For the limited number of cases related to hardware problems, study personnel provided onsite services for households located nearby. If there was a problem with the router hardware that could not be fixed on site, a new router would be provided to replace the router in trouble. Otherwise, if it was confirmed that the problem was with the modem or a broken uplink connection, users were suggested to contact their Internet service provider (ISP) for further help.

Chapter 6

Data Analysis

This chapter discusses the findings from the pioneer study on the aggregated and anonymized passive measurement dataset. Firstly, Section 6.1 summarizes the dataset collected from all deployed home network routers in the home network measurement study. Subsequent sections organize the data analysis part of this chapter in a bottom-up fashion, starting from the data considering individual devices within home networks. The analysis results are compared with previous work or other public datasets if applicable.

6.1 Summary of Dataset

During the period of the measurement study, the data collection server received passive measurement data from 53 participating households. Figure 6.1 shows the intervals of activity for each of the 53 households. The curve shows the total number of households with an active router on each day. A router was considered "active" on a day (and its horizontal bar includes that day) if it contacted the data collection server at some point during that day. Some routers occasionally uploaded empty files; they were still considered to be "active" in that interval. One router (number 26 in the figure) only contacted the server during a brief window (a few days), and for reasons unknown sent only empty files. Thus, the router #26 is omitted from the analyses presented later. During the period from mid-August to the beginning

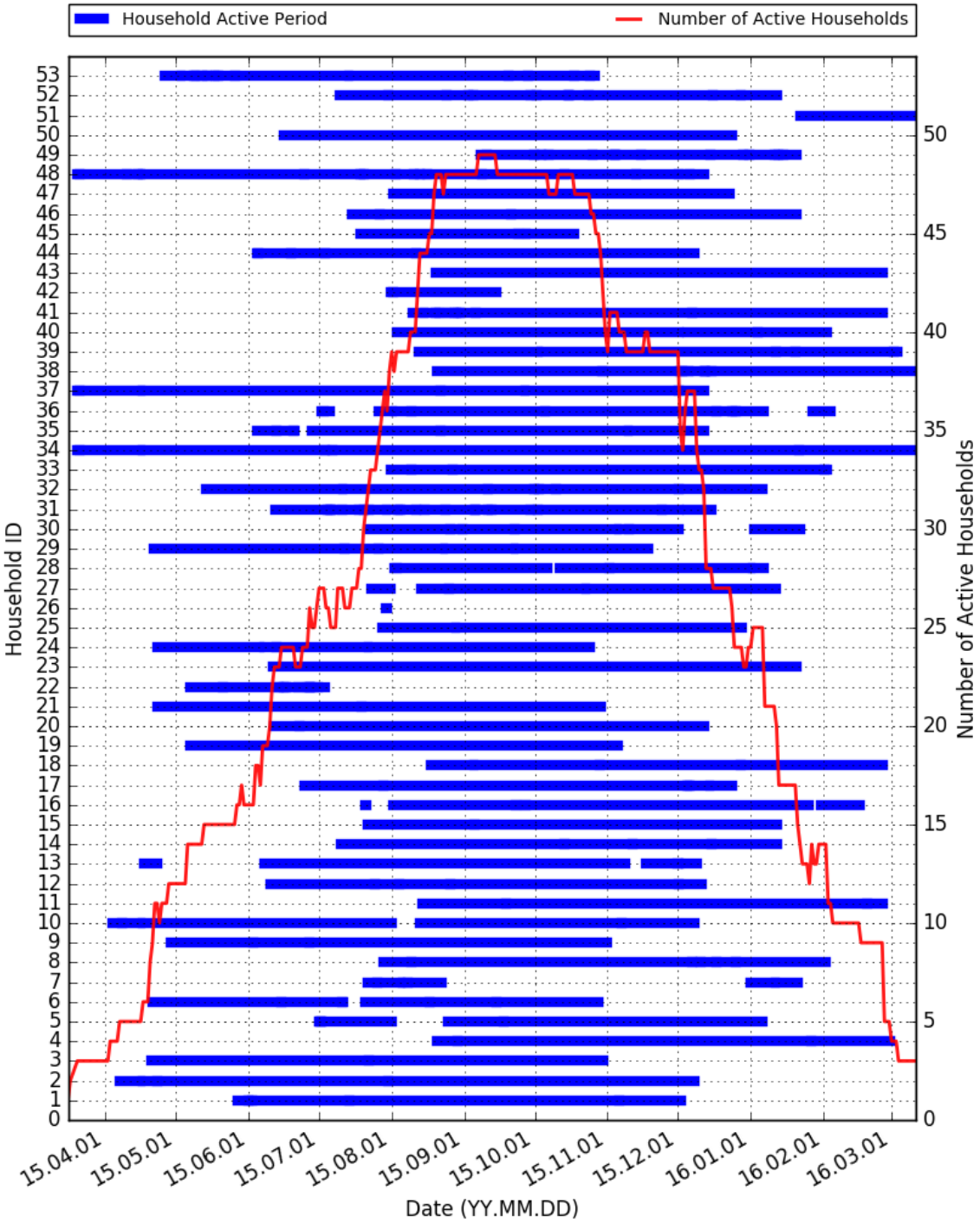


Figure 6.1: The period of data contribution from all participating households

of November 2015, at least 40 routers were active each day, making this the most extensive *published, passive* measurement study of home networks.

Table 6.1 summarizes some statistics of our dataset. "Min", "Max", etc. are taken over the set of households. Thus, data on a total of over two million five-minute intervals was collected; about 40,000 intervals were collected from the median household, which translates to between five and six months (though not all intervals were contiguous). This is an order of magnitude longer than the study reported in 2014 by Sundaresan et al [6]. About half the households observed a billion or more packets during the study, and the number of packets observed varied by three decimal orders of magnitude across homes. The table also contains number of outside hosts that the household home network has exchanged data with. The number of outside hosts ranges from several thousand to several million with a median value of 44,644.

Table 6.1: Summary of the dataset

	Min.	Max.	Avg.	Med.	Ttl.
Intervals	8,842	100,616	43,247	40,845	2.2 M
Edges	0.2 M	15.7 M	2.1 M	1.5 M	111.0 M
Packets	6.5 M	4.7 B	1.3 B	1.0 B	68.9 B
Bytes	3 GB	3745 GB	1135 GB	890 GB	57.6 TB
Outside Hosts	5,745	2.8 M	0.16 M	44,644	*

- The dataset contains data from 52 households.
- M: million, B: billion
- GB: Gigabyte, TB: Terabyte
- *Identifiers of outside hosts are not comparable among households. It is not possible to calculate a total number of outside hosts across all households.

Table 6.2 summarizes protocol usage over all the edges (five-minute interval with traffic between an inside and an outside host) in the study. Unsurprisingly, TCP was the dominant protocol. The 3.5% of edges that contain both TCP and UDP is likely due to DNS. The vast majority of edges involve traffic to a single external port.

Some of the limitations of the dataset are similar to those described in [5]. The subjects of the study may not be representative. Also, not all routers uploaded data

Table 6.2: Protocols observed in the dataset

Among all edges		
Total	110,990,858	100%
w/ ICMP	2,094,624	1.89%
TCP/UDP-only	108,896,234	98.11%
Among TCP/UDP-only edges		
TCP only	76,292,623	70.01%
UDP only	28,710,164	26.36%
TCP and UDP	3,893,447	3.58%
Single port	104,459,409	95.93%
Two ports	3,678,649	3.31%
Three ports	380,274	0.38%
Four or more ports	377,902	0.35%

continuously. However, the measurement system described in this dissertation is designed to ensure that any data files created eventually are uploaded, so only router outages would prevent measurement. Moreover, the hardware platform features a hardware bridge that handles traffic between hosts on the inside network; such traffic never reaches our measurement software. The HNFL software therefore has no insight into inside-only traffic.

The anonymization approach is not prefix-preserving: flows to different outside hosts on the same subnet (e.g., in the same data center) cannot be recognized as such. Another limitation is that it is (intentionally) infeasible to determine whether/when different *households* communicate with the same external endpoint. Although it prevents analysis of the popularity of various Internet sites, the method does ensure that inside and outside endpoint identities are constant over time, so the evolution of a household's bipartite graph can be studied. Importantly, the anonymization also enables us to make the dataset available to the community, unlike some earlier passive measurement studies [6].

The HNFL software does not explicitly track individual TCP connections or DNS request-response pairs. However, in most cases, a "flow" corresponds to a single connection when the protocol is TCP. Also, in the aggregate data, the HNFL software

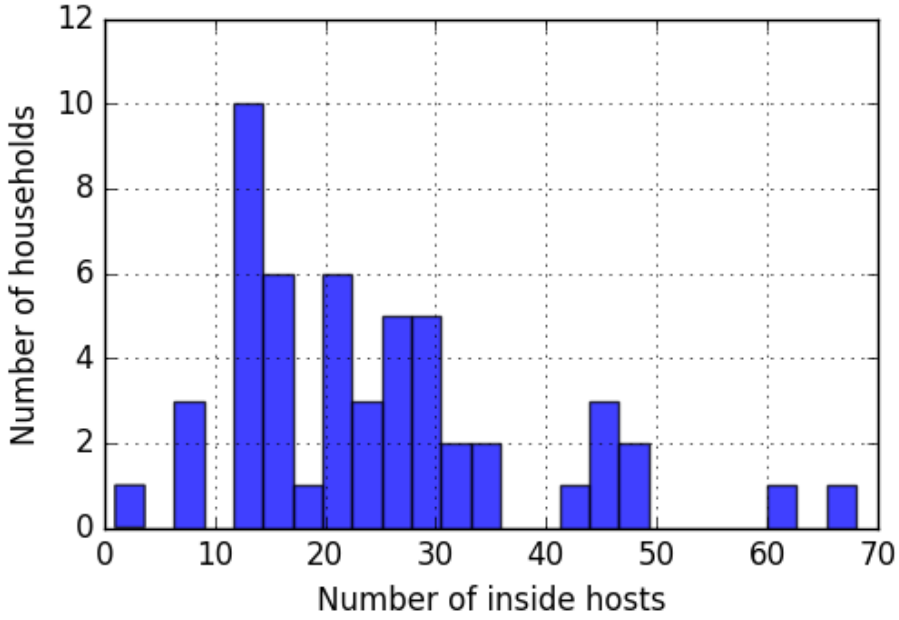


Figure 6.2: The number of networked devices in each household.

only keeps track of the three most common port numbers for a pair of hosts. If more than three port numbers are observed in a single five-minute interval, some information is lost. However, the number of edge/intervals in which this occurred was negligible (0.34%).

6.2 Home Network Devices

Individual devices are the basic participants in any network activities occurring in home networks. The questions examined in this section are what and how many devices are present in home networks, as well as how users are using them.

6.2.1 Number of Devices and Activities

Figure 6.2 shows the distribution of the number of devices present in participating households. Most households observed 12 to 35 active networked devices over the duration of the study. The number of devices is also related to the demographic composition of a household: households with the most devices tend to have more

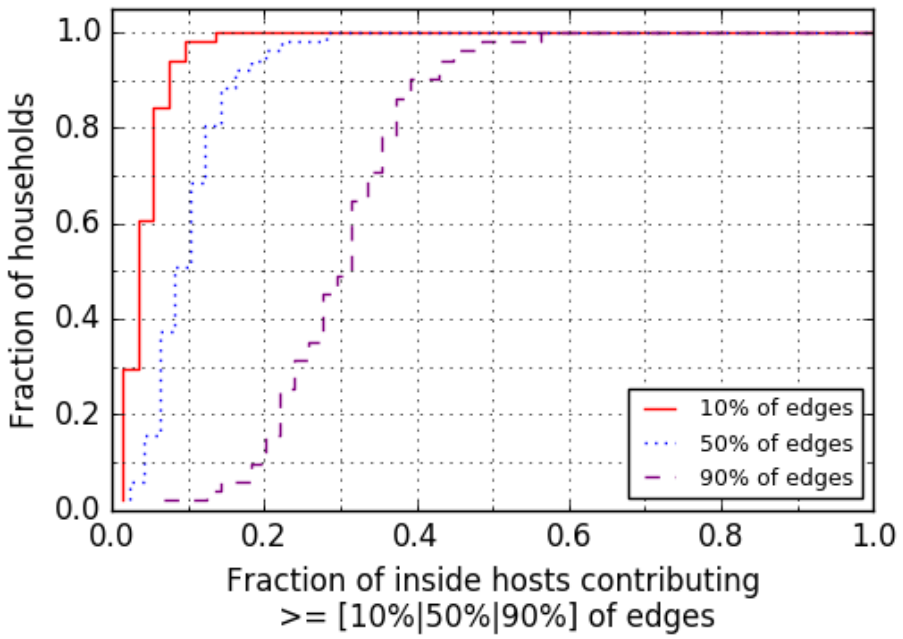


Figure 6.3: The traffic distribution among home network devices.

children (under 18 years old) or be shared by young university students (from 18 to 25 years old). This result shows more households with a significantly higher number of devices than in [5], where only half of households had more than five devices, even in developed countries. This may reflect the leading edge of the "Internet of Things", or it may be an artifact of the demographics of the sample—i.e., more children might correlate with more visiting friends using the network.

Not all devices seen in the dataset are equally active: in half of the households, 10% of the devices contribute more than 50% of the edges (Figure 6.3). Moreover, in half the homes less than 30% of devices contribute more than 90% of edges. In virtually all households, 30% of the devices contribute more than half the traffic. While computing these statistics, the single household with only one device (a Belkin router) is removed in order not to bias the results. Every remaining household had at least seven devices connected at some point in the study.

Figure 6.4 indicates that 70% of the devices are active for no more than half of the days in the study. A device is considered "active" in a day if it is involved in

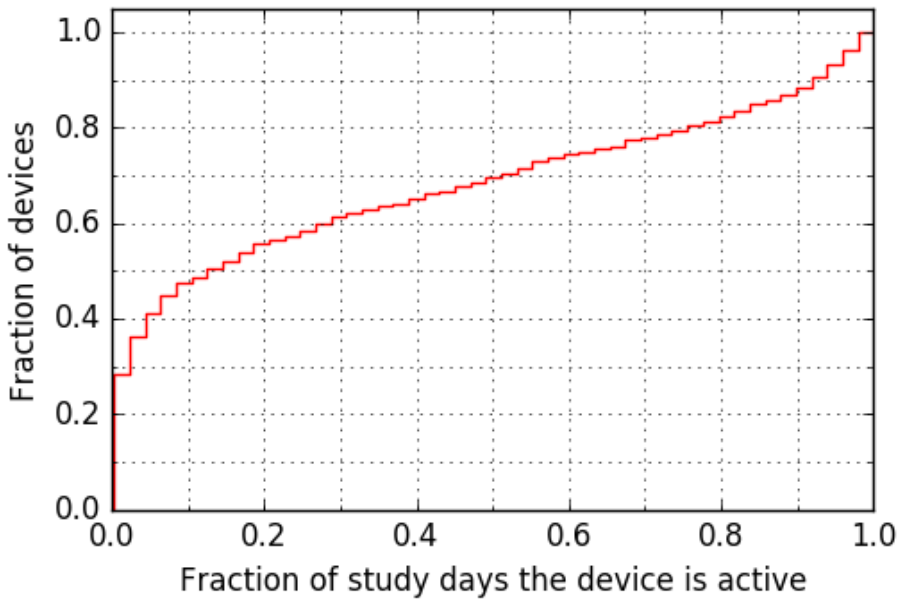


Figure 6.4: Activeness of devices.

Table 6.3: Manufacturers of daily active devices

Manufacturer	Device Type
Apple, Samsung	Smartphone/Tablet/PC
Hewlett Packard, Asus	PC
TiVo, Roku, Ecostar	Internet TV
AzureWave, Redpine Signals	Probably PC
Ecobee	WiFi thermostat

some traffic during that day. Overall, 169 devices from 48 households are "one-day" devices. These "one-day" devices could be anything from a rarely used device (e.g., an old laptop) to a guest device (e.g. visiting friend's smartphone). Only 30 devices from 23 households out of 1,286 total devices were active every day.

Table 6.3 lists the manufacturers and possible device types of these daily devices. Based on the list, Internet TVs, PCs, smart home devices, and smartphones are more likely to be used every day or always connected. Section 6.2.2 further discusses device types and manufacturers.

Figure 6.5 is a CDF of the time span of the longest-duration edge in each household, defined as the fraction of the intervals in which the edge was continuously

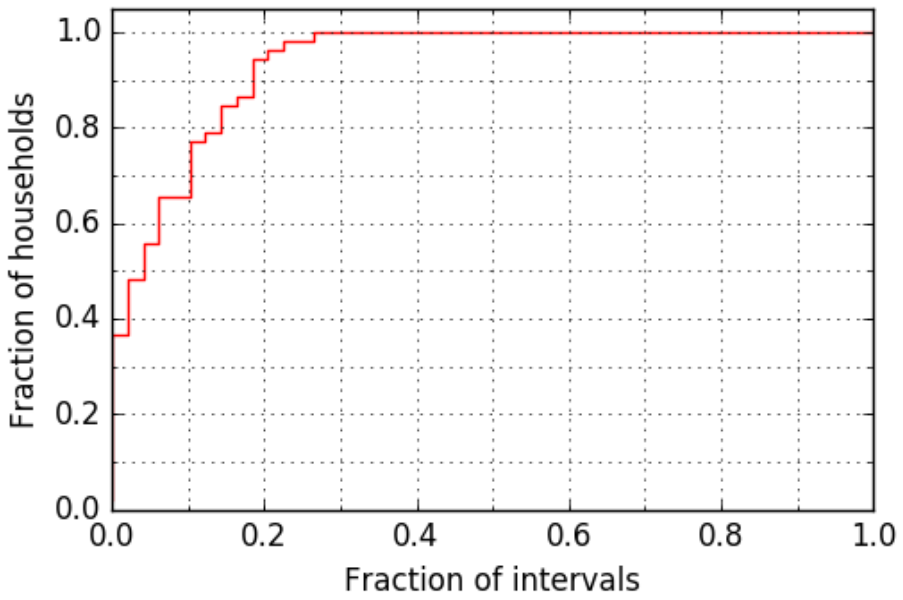


Figure 6.5: The lifespan of edges with longest duration in households.

present in the reported household data. About 28 households do not have obvious long-term edges. The longest-duration edges lived up to about 25% of total intervals from the corresponding household; no edge was continuously active throughout the study. Some devices, such as personal computers, may appear to be always active due to the background applications (e.g., email clients and operating system updater) if these computers are not turned off by users. The lack of always-active edges may be due to the measurement being interrupted for several times during the study for other reasons (e.g., intentional power off or power failure).

6.2.2 Device Vendors

Figure 6.6 shows the distribution of device types from 47 manufacturers, and the number of edges generated by each manufacturer's devices. The 47 unique manufacturers are reduced from 93 vendor entries derived from the IEEE 802 OUI information in the dataset. The long tail of the distribution is omitted by showing only devices contributing over 99% of all edges generated by all devices with OUI information attached; this removes 1/3 of all OUIs. The organization names of

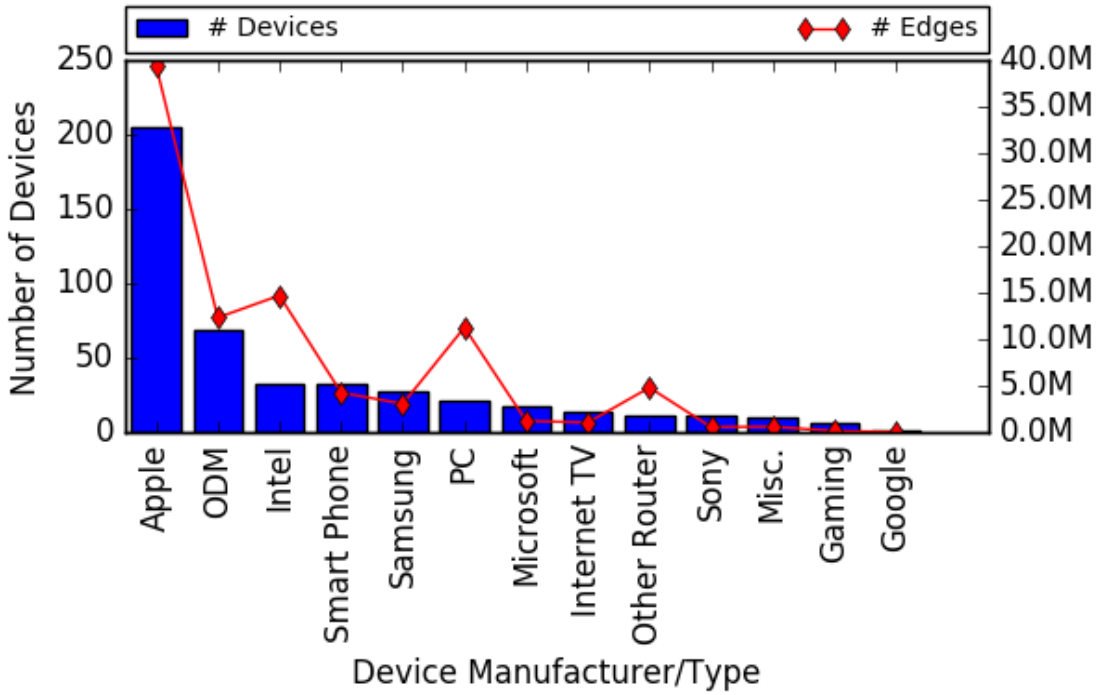


Figure 6.6: The number of devices and edges from different manufacturers or devices types.

big manufacturers are displayed separately in the figure since there is not enough information to distinguish device types solely from the manufacturer's name (e.g., an Apple device cannot be distinguished as an iPhone or a Macbook). For smaller manufacturers, devices are grouped into common device types, as shown in Table 6.4.

Apple is—by far—the most popular manufacturer in the participating households, followed by Intel. In addition, Apple devices are involved in more than 40% of the edges in the study. Although Apple has employed randomized MAC addresses since iOS 8[1] [63], this new feature does not affect the tracking of devices by Apple because iOS devices only use randomized MAC addresses when running Wi-Fi[2] scans. Once an iOS device is associated with a Wi-Fi access point, the device uses its real MAC address.

Devices related to desktops or laptops tend to produce larger network "footprints",

[1]iOS 8 is released in September 2014.
[2]Wi-Fi: a wireless network connection technology.

Table 6.4: Manufacturers of devices

Device Type	Manufacturers
Original Device Manufacturer (ODM)	Murata, Hon Hai Precision, Liteon, AzureWave, Gemtek, Castlenet, Wistron NeWeb, Foxconn, Redpine Signals, Universal Global Scientific Industrial, and Kaparel
Smartphone	Motorola, HTC, LG, NEC Casio, RIM, TCT Mobile, ZTE, and Shenzhen RF
Personal Computer (PC)	Asus, Giga-Byte, Dell, Hewlett Packard, Micro-Star Int'l, and ASRock
Internet TV	TiVo, Roku, Vizio, and Echostar
Other Router	Belkin, TP-Link, Cisco-Linksys, NetGear, and Arris
Misc.	Ecobee (WiFi thermostat), Chicony (network camera), Sercomm (network camera), Barnes&Noble (ebook), and LiFi Labs (smart lightbulb)
Gaming	Nintendo and Mitsumi (manufacturer of WiFi sub-PCB for Nintendo DS and controllers for Wii, PlayStation, and Xbox)

i.e., more bytes. The router vendors that show up in Figure 6.6 indicate that several households placed a separate router behind the measurement router. Overall, this distribution of device types and manufacturers is very similar to that presented in [5], indicating that users' preferences in network devices have not changed much in the intervening years.

6.3 Internet Destinations

Measurement results agree with those of [5] in showing that traffic distribution among Internet destinations is long-tailed. Figure 6.7 shows that for 100% of all households, less than 20% of outside hosts contribute more than 80% of edges and less than 30%

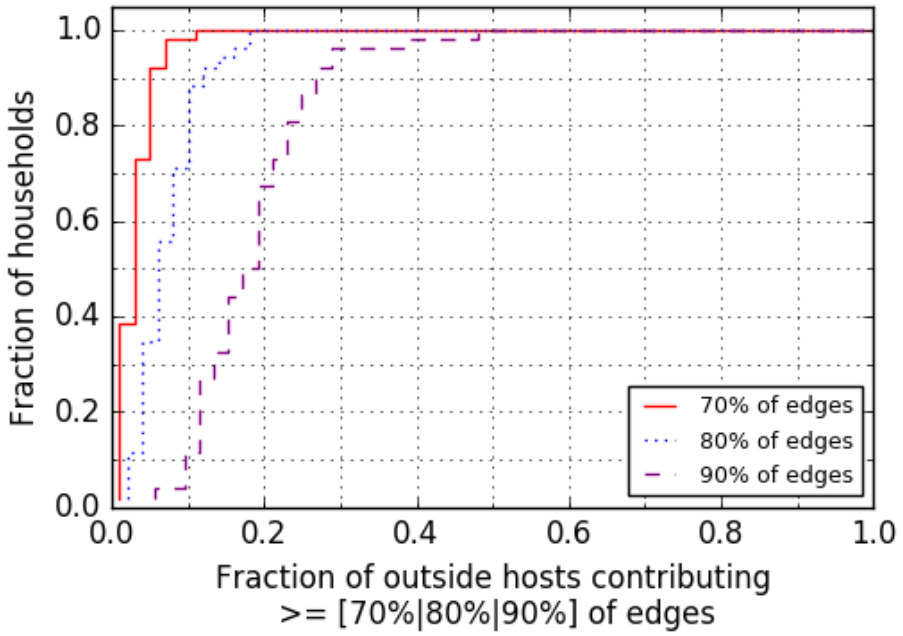


Figure 6.7: The traffic distribution in edges among Internet destinations.

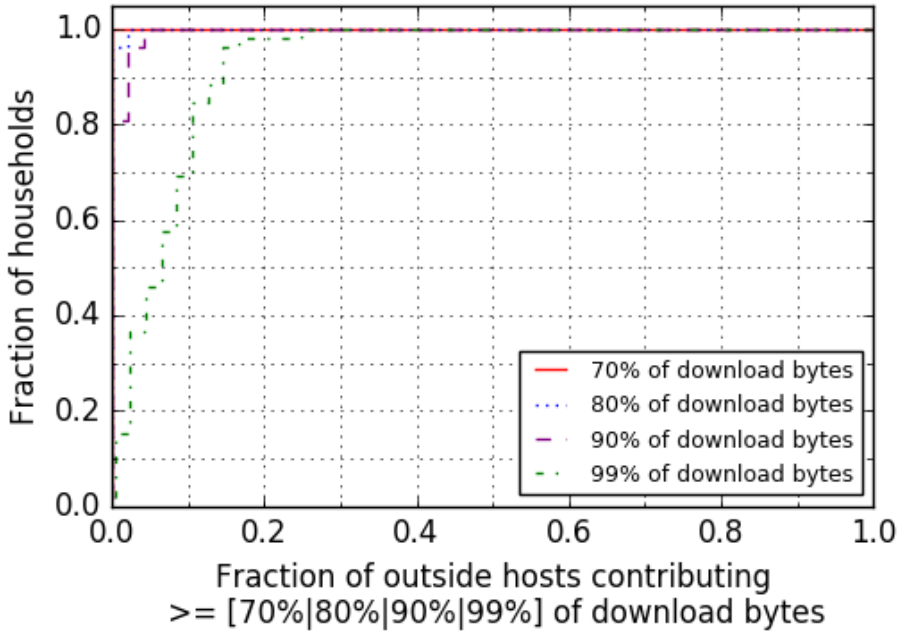


Figure 6.8: The traffic distribution in download bytes among Internet destinations.

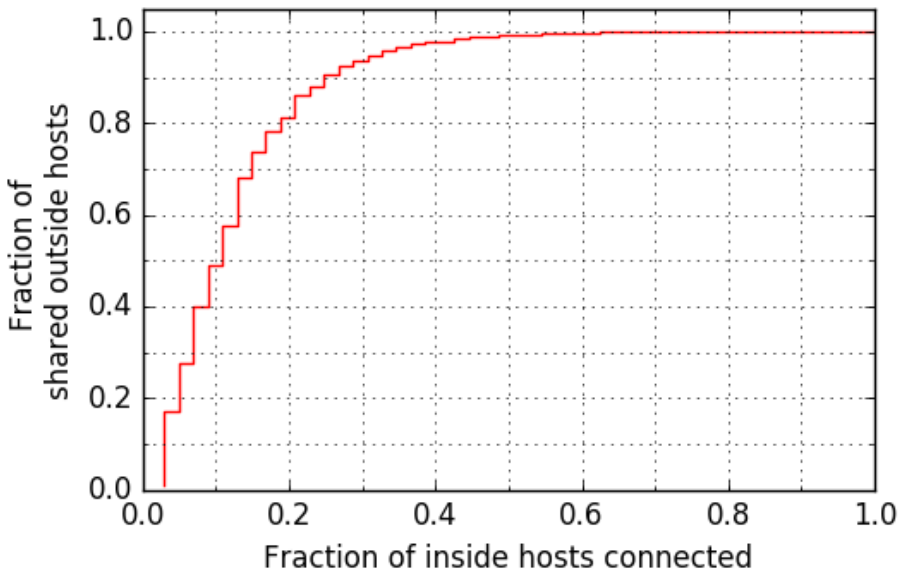


Figure 6.9: How Internet destinations are shared among devices in a household.

of all outside hosts contribute more than 90% of edges in most households (about 90%). On the other side, Figure 6.8 shows that bytes transmitted in the downlink are more concentrated to a small number of outside hosts. Only 15% of outside hosts are responsible for over 99% of all download bytes for 95% of the participating households.

For the 51 households with at least seven devices, 12.6% out of over eight million outside hosts have connections from more than one device in a household. Figure 6.9 presents the popularity of these shared Internet destinations in households, which ranges from about 3% to 65% of devices.

Then, are outside hosts with more traffic the ones also shared most by inside hosts? The data support this conclusion. Figure 6.10 plots the size of the common subset of top 20% of outside hosts with most edges and top 20% of outside hosts with most connected inside hosts. For over 85% of all households, more than half of the top edge generating outside hosts have connections with two or more inside hosts. But at the same time, 87.4% of all outside hosts have connections to only one inside hosts. In other words, users tend to access the popular Internet applications

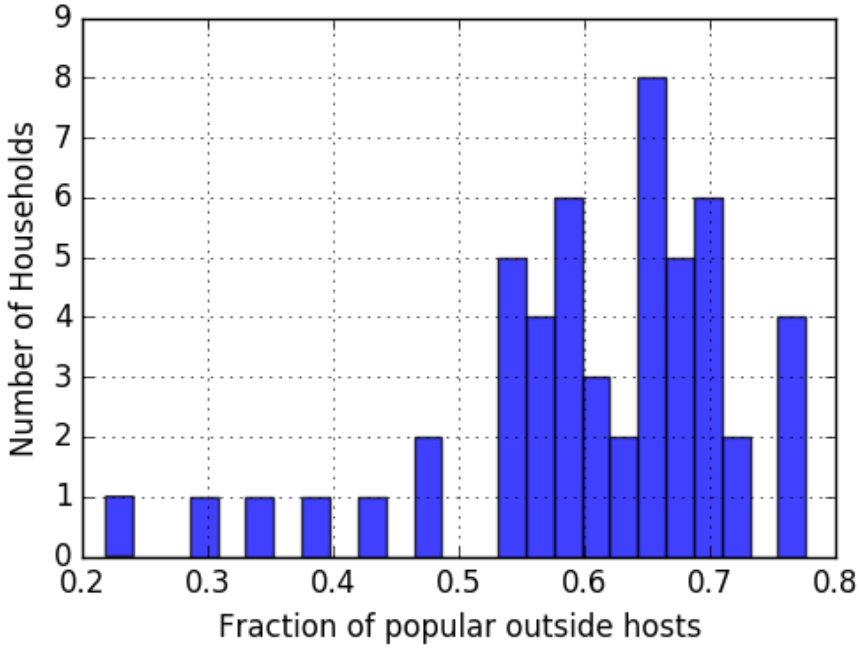


Figure 6.10: Popular Internet destinations.

from different devices. For example, a user may watch a movie from Netflix in the afternoon and continue the movie while having the dinner using a tablet device. Also, some users prefer to set up their email accounts on all their devices including laptops and smartphones. All these popular applications are more likely to trigger network traffic from different devices to the same outside host.

6.4 Households

The continuous passive measurement at the vantage point of the gateway also enables us to better understand activities of home networks as a whole.

6.4.1 Achievable Transmission Rate

One of the most-studied attributes of home networks is access channel speed (upload and download bandwidth) [19, 4, 5]. Due to the nature of passive measurement, the transmission rate cannot be directly tested by sending and receiving data on the host machine directly. However, the approximate bandwidth can be estimated from

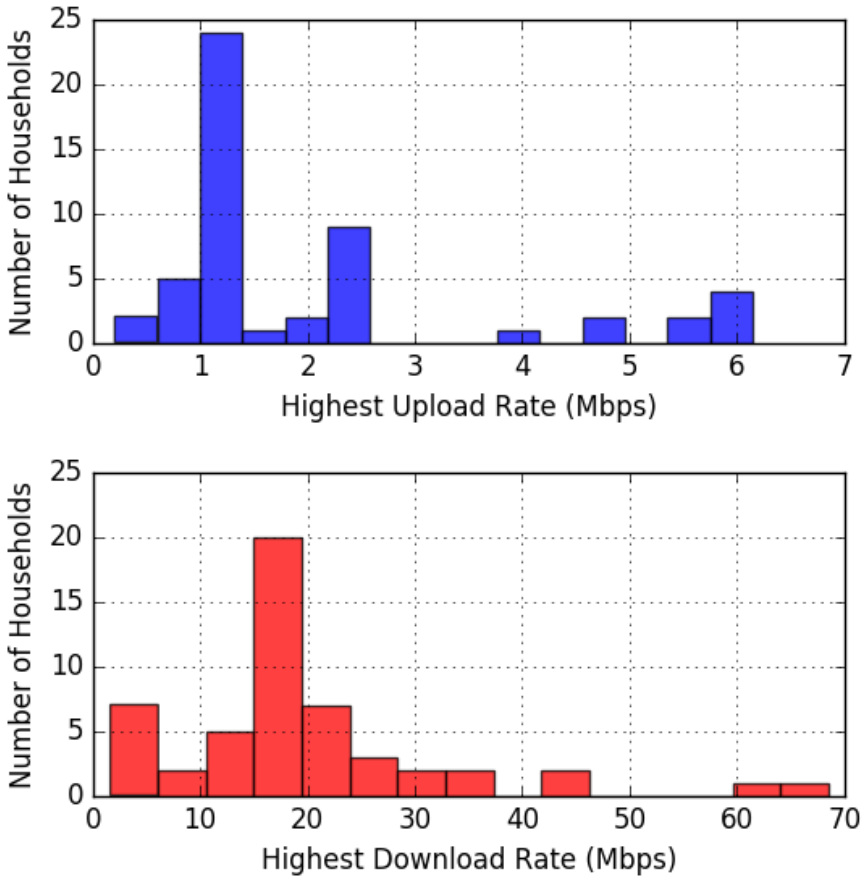


Figure 6.11: Highest upload/download rate observed from 52 households.

the highest achieved upload and download rate by calculating bytes transmitted in each five-minute interval. Figure 6.11 presents the distribution of highest upload and download rate observed from all 52 participating households. In general, the highest upload rates are less than one-tenth of the highest download rates, which is similar to the results in [19].

Knowing the amount of data transmitted in each five-minute interval for all households, Section 6.4.2 explores the usage patterns of home networks.

6.4.2 Diurnal Usage Patterns

According to the analysis, although households differ, there are clear diurnal patterns. The traffic data are divided into smaller chunks according to clock hours as shown in

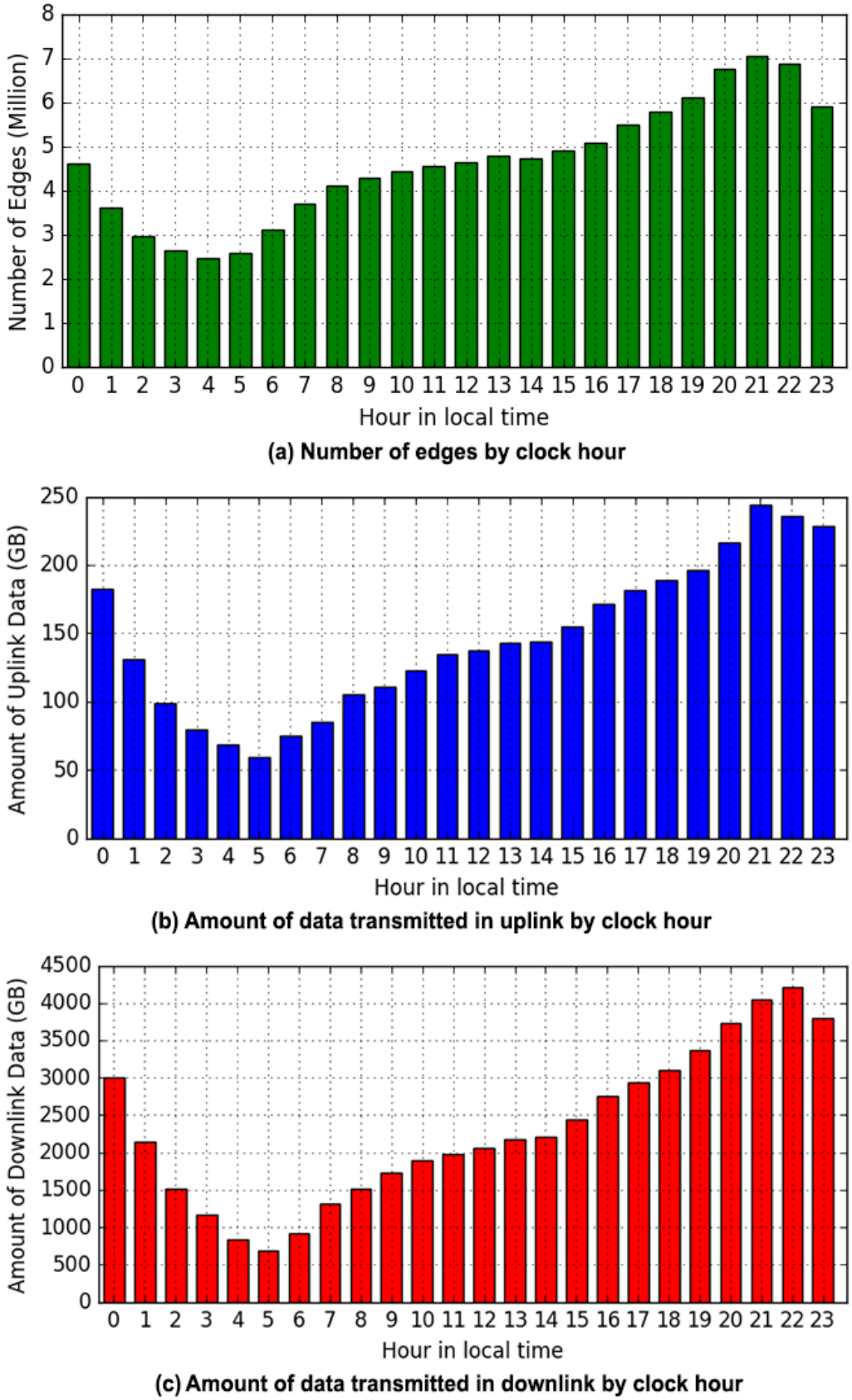


Figure 6.12: Diurnal network activities across all households.

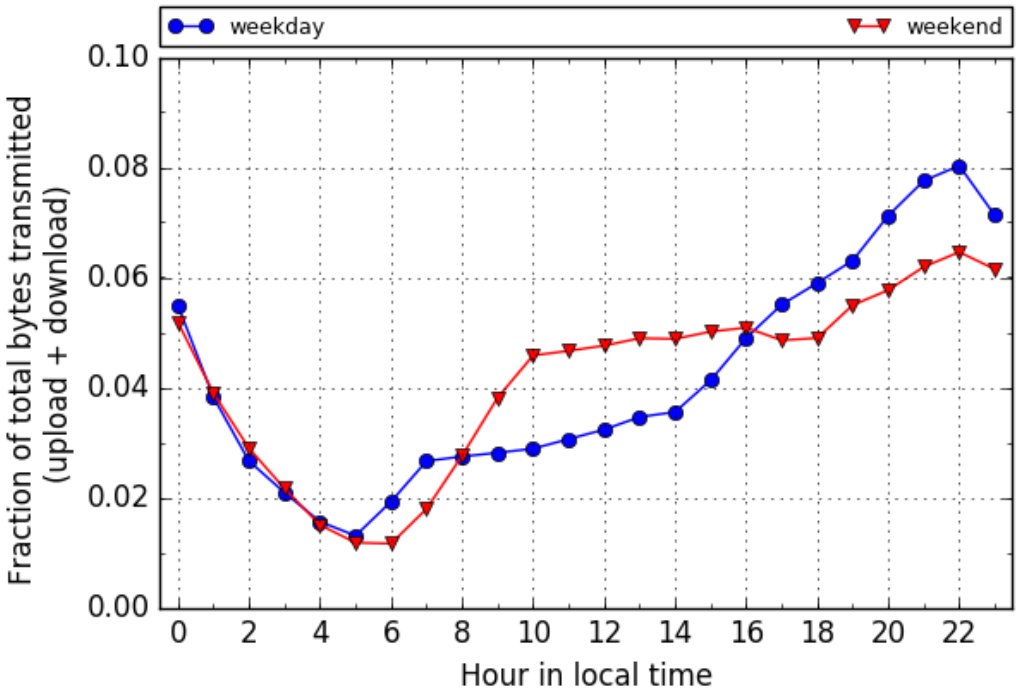


Figure 6.13: Data transmitted (combined upload and download) on weekdays and weekends across all households.

Figure 6.12. The plots using edges, uplink bytes, and downlink bytes all show similar shapes, which peak during the night and bottom before the sunrise. Figure 6.12-b and Figure 6.12-c also confirm that the data transmitted in uplink is only a fraction of the data transmitted in downlink, which is about 6%.

According to the analysis, the network usage in a household can be different on weekends than weekdays. Figure 6.13 plots the weekday and weekend diurnal patterns computed by summing the data from all households for each weekday interval and each weekend interval. As expected, network usage goes up when people wake up; usage in daytime is moderate since some family members need to go to work or school. Evenings are the peak times for network usage. After that, network usage goes down sharply as people go to sleep. The shape of weekend's usage pattern is different as more active daytime network usage is observed.

While most households have similar weekend network usage patterns, weekday

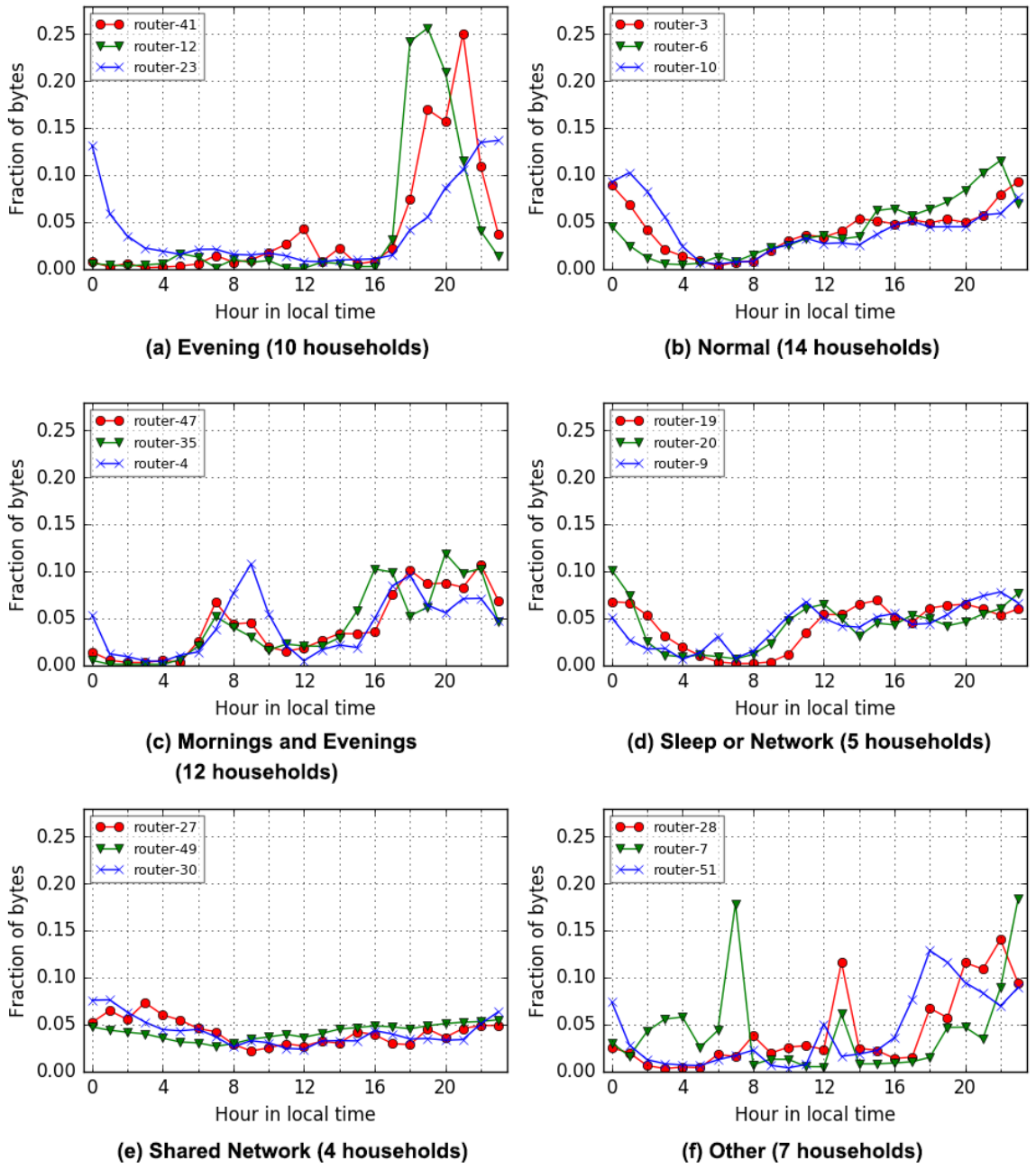


Figure 6.14: Representative traces for six types of diurnal network activity on weekdays.

usage patterns from different households show the obvious distinction. The 52 households are clustered into six groups as shown in Figure 6.14:

- **Evenings.** The households have only one major usage spike during the period of late afternoon and evening.
- **Normal.** Patterns in this group are close to the weekday pattern shown in Figure 6.13. Network usage gradually grows until the peak hour in the evening. Also, this group is the largest group.
- **Mornings and evenings.** The households have a network usage peak during the morning. Then the network activity keeps low until late afternoon.
- **Sleep or network.** The patterns in this group are more weekend-like. An even level of network usage throughout the day is observed except for hours of sleep.
- **Shared network.** The network is kept busy throughout the 24 hours. This kind of usage patterns is observed mainly from households shared by multiple young people.
- **Other.** Any usage patterns that do not fit in the previous five groups.

To obtain these clusters, the Ward's hierarchical agglomerative clustering method [64] is applied on the weekday usage patterns from each household. The distance between two households' patterns is their shift-minimum standard deviation. Shift-minimum means we shift the pattern curve of one of the household forward or back 0 to 4 hours, among which the minimum standard deviation value calculated is used as the distance between two patterns. We adopt the shift-minimum method due to the observation that the shapes of usage patterns from different households could match closely except for a simple linear shift. For example, the patterns of router-3 and router-6 in Figure 6.14-b are very similar if we shift router-6 to the right for two hours.

The results from the hierarchical clustering algorithm are confirmed and refined (from three to six clusters) by eyeballing the graphs. The weekday and weekend pattern graphs of all households besides household #26 are available in Appendix D.

Not every household follows the diurnal usage pattern shown in Figure 6.14. Actually, the clustering results prove that home networks are heterogeneous since households differ with each other in terms of demographic composition and collection of network devices.

6.5 Internet Applications

Internet applications are identified by the TCP/UDP port number used on the Internet side. In order to get more accurate application data, edges with multiple port numbers are removed before calculating the results in this section. The results are probably biased by the removal of multi-port edges from the data. Fortunately, the removed multi-port edges only contribute 4.39% of all edges and 4.23% of all bytes in the dataset.

6.5.1 HTTP vs HTTPS

During the study, there is a trend of increasing HTTPS traffic over HTTP traffic. With the purpose of verifying the universality of this trend on the broader Internet, the anonymized Internet Traces from CAIDA [65] are also analyzed.

There are three traces taken from CAIDA's equinix-chicago monitor carrying data within the period of the study (20150521, 20150917, and 20151217). The number of bytes transmitted as HTTP (port 80) and HTTPS (port 443) traffic is aggregated according to calendar days in both CAIDA's traces and the study's dataset. Figure 6.15 illustrates the trend of HTTPS's fraction among the two applications. Both the home router dataset and CAIDA's Internet traces show a clear trend of increasing HTTPS footprints over HTTP.

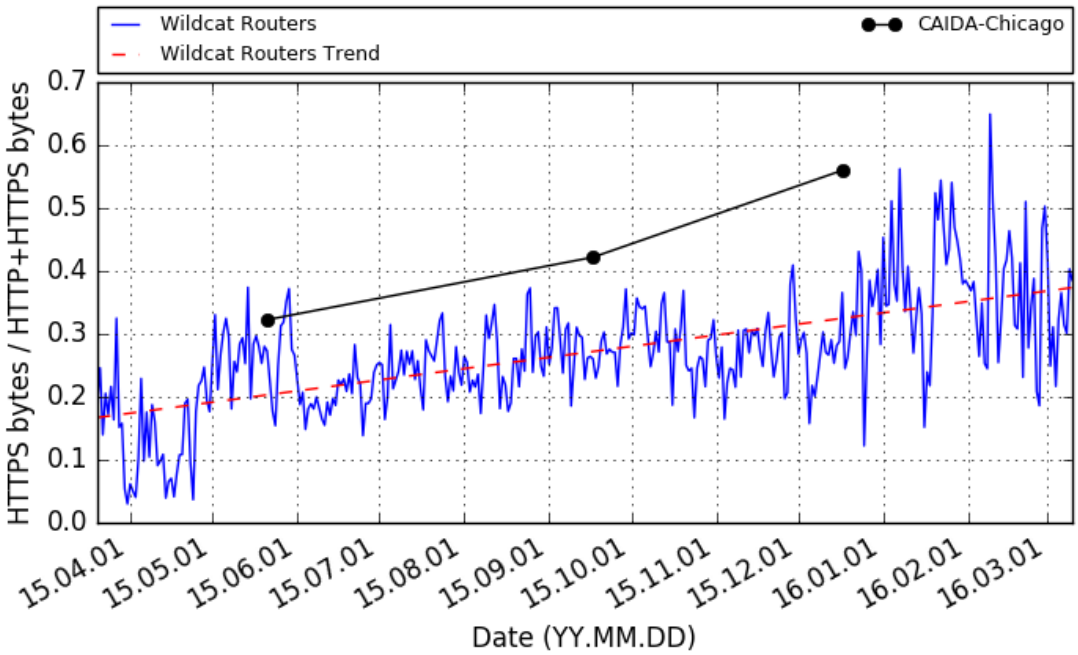


Figure 6.15: The trend in increasing HTTPS traffic.

However, the fraction of bytes transmitted as HTTPS in the dataset is apparently smaller than the one from CAIDA's traces. Thus, insights from other studies are necessary before certain conclusions can be made. It's also worth mentioning that the plotting shows a higher degree of variance at the periods around the start and end of the study. The cause for this variance is that the study has much less active participating households at those two periods.

6.5.2 Download vs Upload

Although HTTP and HTTPS contribute most of the data transmitted in the dataset, other applications are also interesting, such as mail services, BitTorrent, and smartphone-related applications. Table 6.5 summarizes the top applications observed. The table lists each application's contribution to the data transmitted and the relative amounts of upload and download traffic, which reveals the attributes of certain applications. For example, the download side dominates HTTP traffic. Various online streaming services are probably the major contributors. But more

Table 6.5: The data transmitted by applications.

Application	Percent of Bytes	Download : Upload
HTTP	70.39%	44.45
HTTPS	25.60%	6.79
Mail[1]	0.15%	10.66
BitTorrent[2]	0.04%	1.86
Apple[3]	0.008%	1.25
Google[4]	0.002%	1.86
Other	3.81%	3.46
Total	100.00%	16.48

[1] Mail: IMAP (port 143 and 993), POP (port 110 and 995), SMTP (port 25 and 465).
[2] BitTorrent: official BitTorrent port range from 6881 to 6889.
[3] Apple: Apple push notifications (port 5223). This port is also used by TiVo devices and some PlayStation games.
[4] Google: Google Play Store or Chrome sync (port 5228).

download traffic from HTTPS is expected in the near future because major data-heavy services, such as Netflix, are rolling out HTTPS for delivering their content.

It is also somewhat surprising that BitTorrent traffic has such a limited footprint—only 0.04% data transmitted. It's partially due to the application identifying method. Only edges carrying official BitTorrent port numbers are counted as BitTorrent application traffic. However, in [19], BitTorrent is the largest contributor to transmitting data, even on top of HTTP.

Other lightweight services, such as Apple push notification, and Google Chrome settings sync are more balanced between upload and download transmission.

6.5.3 Edges vs Packets vs Bytes

Figure 6.16 illustrates traffic distribution among HTTP, HTTPS, and other applications using number of edges, number of packets, and number of bytes transmitted for all households. There are more HTTPS edges rather than another applications from most households as shown in Figure 6.16-a. However, considering number of packets

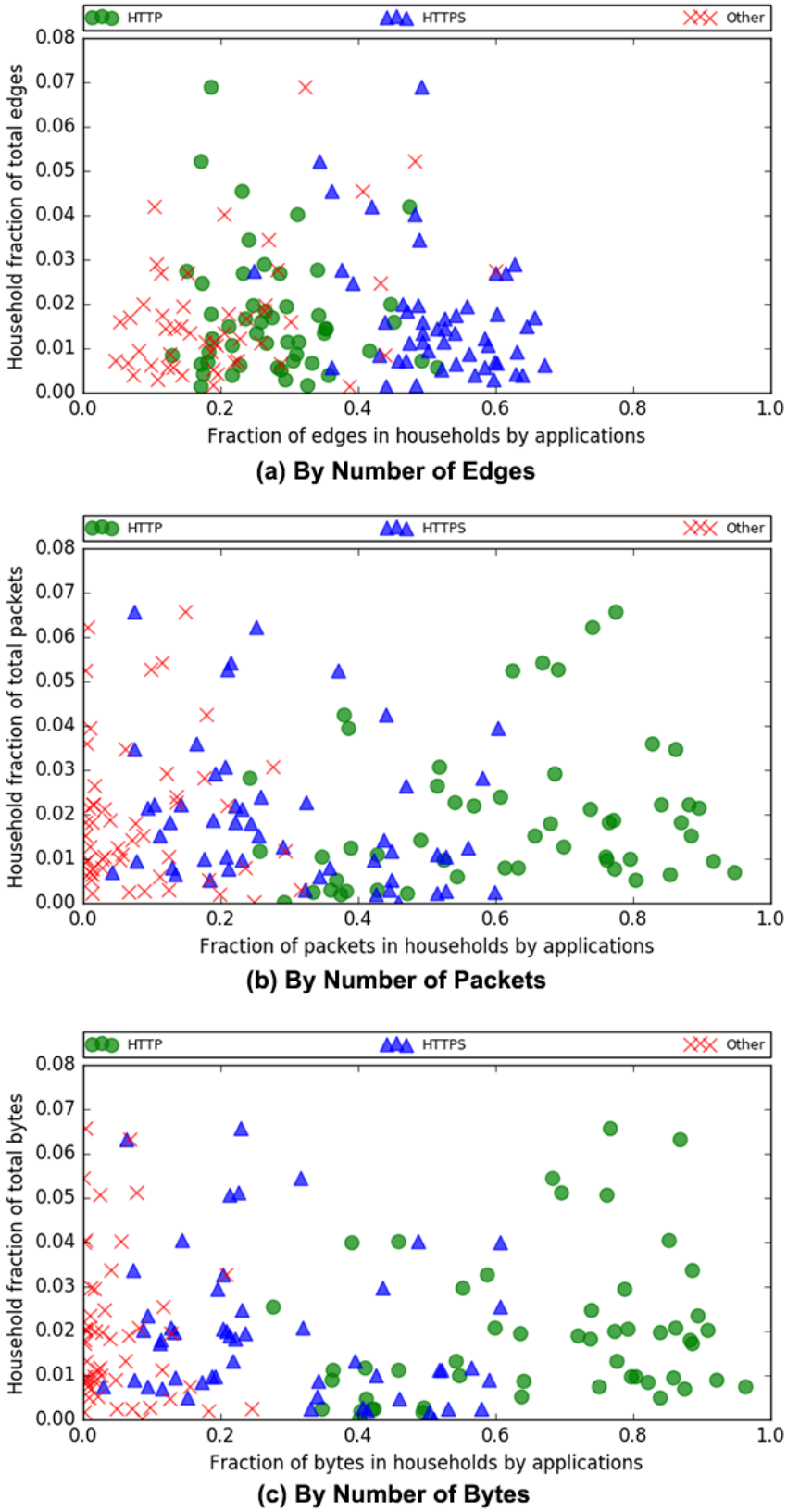


Figure 6.16: The traffic distribution among Internet applications for all households.

Table 6.6: Device/application affinity

Application	Threshold	Total # devices	# Households w/ such devices
HTTP	$\geq 50\%$	82	38
HTTPS	$\geq 50\%$	566	52
Mail	$\geq 30\%$	9	8
BitTorrent	$\geq 10\%$	11	8
Apple[1]	$\geq 30\%$	13	12
Google[2]	$\geq 30\%$	3	2

[1] Apple: Apple push notifications (port 5223). This port is also used by TiVo devices and some PlayStation games.
[2] Google: Google Play Store or Chrome sync (port 5228).

or number of bytes transmitted (Figure 6.16-b and Figure 6.16-c), HTTP replaces HTTPS as the most popular application.

One possible explanation for transposition between HTTP and HTTPS observed with different measurements is that HTTPS is mostly used for lightweight traffic, such as website authentication, online banking transactions, and social media services, while HTTP is used heavily in applications with the higher load like video streaming and file downloading. In general, HTTP edges carry more data in terms of packets or bytes than HTTPS.

Considering applications other than HTTP and HTTPS, they usually only make up less than 40% of edges or less than 30% of packets and bytes in most households. Section 6.5.4 discusses other applications in more details.

According to the observation presented in Figure 6.16, edges reveal different aspects of network applications in comparison with packets and bytes. Edges quantify the connection between inside and outside hosts, while packets and bytes measure the actual amount of data transmission of the application. Future studies on the dataset should consider edges along with packets and bytes in the analysis.

6.5.4 Devices and Applications

At last, the affinity between devices and applications is examined. Unsurprisingly, a lot of devices have affinity to HTTP/HTTPS applications (Table 6.6). However, there are a significant number of devices associated with other "smaller" applications. For the 9 devices attached to mail applications, unfortunately, only two of them are confirmed as Apple devices, while other devices do not have related MAC OUI information in the dataset. Considering the 11 devices with more than 10% BitTorrent edges, their manufacturers are Intel, Dell, and several other ODMs. Thus, these devices are probably PC products.

As noted in Table 6.5, port 5223 is also used by TiVo and Sony PlayStation games besides being used by Apple push notification services. It turns out that the manufacturers of the 13 devices are Apple, TiVo, Sony, and Wistrom NeWeb (one of the iPhone manufacturers). Otherwise, the 3 devices using Google play store or Chrome sync service are manufactured by HTC, Samsung, and an ODM.

6.6 Use of Home Network Traffic Dashboard

The home network traffic dashboard introduced in Section 4.1 is an attempt to help users understand their network usage. However, as shown in Figure 6.17, most users in the study rarely used the dashboard. About 60% of all households did not use the dashboard at all during the study. Only one household used the dashboard more than once a day. Most households used the dashboard around the start and the end of the study period. A reasonable assumption is that most users do not have the motivation to track their network usage. Unlike the households using uCap [42, 43], the participants of the passive measurement study do not have a monthly usage cap on their bandwidth subscriptions. The dashboard use around the end of the study is probably due to dashboard related questions in the post-study survey.

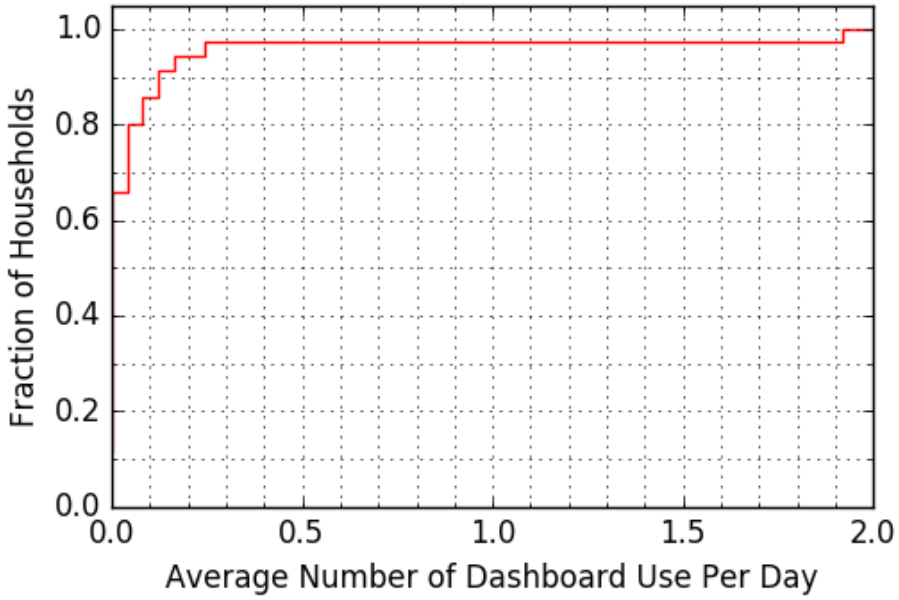


Figure 6.17: The usage of Home Network Traffic Dashboard.

Chapter 7

Conclusions and Future Work

This dissertation describes and evaluates a lightweight privacy-preserving passive measurement tool, Home Network Flow Logger (HNFL), for home networks. The HNFL tool is lightweight that a commodity router running the tool can capture 100% of network flow data while still achieving a transmission rate over 360 Mbps, which is about 87% of the maximum capacity of the hardware without running any extra software. In contrast, BISmark's passive measurement software degrades the transmission rate to only 16 Mbps on the same hardware, which is lower than the transmission speed provided by most of the broadband plans today. Also, a Home Network Dashboard and an iOS Management application were developed to help home network users to understand and manage their network.

A passive measurement study of home networks was conducted using the measurement system based on the HNFL tool. The passive measurement study involves more homes and longer duration than any other previously published studies. The study collects un-NATed data from 52 homes over an eleven-month period on the structure of the bipartite graph representing communications between inside hosts and outside host. To prevent inference attacks based on comparing traffic to similar destinations across households, external address information was anonymized in a manner specific to each household. Although this method is not prefix-preserving, it still allows exploration of interesting statistics. For example, the study shows that in

most households, a few inside devices dominate in communication with the greater Internet. And similarly, in virtually all households, 90% or more of the traffic (number of edges as measured in active 5-minute intervals) involves half of the external hosts.

The results on the types and numbers of devices on the home network agree with those of Grover et al [5]: Apple is the most popular device manufacturer in participating homes by at least a factor of five, constituting about 40% of all devices in the study whose manufacturer could be determined. Moreover, at least 40% of the observed network activity involved an Apple device; the actual percentage is almost certainly higher, since about 20% of households did not report OUIs of devices.

Like Grover et al., a pronounced diurnal activity pattern across most homes is observed, though the difference between the weekday and weekend patterns seems to be less pronounced in the study [5]. Using clustering, daily activity patterns are categorized into six groups. Three of these groups—which differ significantly from each other—account for 70% of the homes in the study.

Over the eleven months in which the study collected data, there is an observation of a general increase in the amount of web data transmitted securely. This trend generally matches that observed in three backbone traces from the CAIDA equinix-chicago monitor [65], though it is somewhat less pronounced than in the backbone traffic.

The study of home networks has many limitations in common with prior work. Participants were self-selected and therefore not necessarily representative. Privacy and resource constraints force a tradeoff between granularity of data on one hand, and privacy-preservation and degradation of the user experience on the other. Nevertheless, the study represents a contribution to the field, in showing that it is possible to collect useful traffic data on commodity platforms without interfering with user-observed performance, and that such data can produce interesting results even when it is fully anonymized.

The Home Network Flow Logger (HNFL) software is available on Github[1] along with documentation about system requirement, installation instructions, and usage guide. The anonymized dataset of Wildcat Home Routers project is available to researchers on request, through the project website[2].

7.1 Implications of Data Analysis

According to the results from analyzing data collected from residential households, there are some facts and reasonable guesses that can be meaningful to many parties including ISPs, network device manufacturers, and Internet application developers:

- Over all participating households of the Wildcat Home Routers study, there are more than 16 bytes transmitted in the downlink for every one byte transmitted in the uplink (Table 6.5). This observation justifies providing asymmetric capacity in the last mile. It can also be seen as supporting the case for future Internet architectures that focus on content retrieval.

- The study shows more network devices in each household than prior studies which might be consistent with the emergence of "Internet of Things" devices. It is possible that a household might have several hundreds of active network devices in the near future. However, the majority of commodity routers on the market only support up to 253 devices because these routers usually (by default) use the last eight bits of IPv4 address space to assign IP addresses to local devices. The home network router vendors may need to consider adding support for more concurrent devices.

- Developers of popular Internet applications may consider the possibility of content sharing within the home network domain to reduce redundant Internet

[1]HNFL Github repository: https://github.com/UKY-netlab/privpresMon
[2]Wildcat Home Routers: http://www.netlab.uky.edu/wildcat-home-router/

traffic. For example, if a user switches from a laptop to a tablet while streaming a video, the tablet can get the partial content of the video from the cache on the laptop directly instead of downloading the whole video from a remote server again.

7.2 Future Work

The Home Network Flow Logger (HNFL) shows promising accuracy and performance as a passive measurement tool within the home network environment. However, the following aspects are in consideration to make a more feature-complete HNFL:

Wider platform adaption

The HNFL software is only evaluated on several home gateway routers running OpenWrt Attitude Adjustment release. Giving the large device base supported by OpenWrt [51] and the new OpenWrt releases, the measurement software needs a modification to adapt newer systems and more hardware platforms in order to scale out adoption of HNFL in home networks.

IPv6 Support

Currently, the HNFL software only measures network traffic using Internet Protocol version 4 (IPv4). However, IPv4 uses a 32-bit field, which can only represent 4,294,967,296 unique addresses. Since the exhaustion of IPv4 addresses is anticipated in the near future, the whole industry is moving towards Internet Protocol version 6 (IPv6). According to the statistics from Google, the percentage of users that access Google over IPv6 has almost doubled in the past twelve month (from 6% to 12%) [66]. To passively measure IPv6 traffic, the new software can make use of the IPv6 related NetFilter hooks, which are available since Linux Kernel version 2.4. The support for IPv6 will not alter the format

of resulting HNFL measurement data since the outside host anonymization algorithm, AES-128, is still applicable to IPv6 addresses.

Adjustable Interval

The measurement data of HNFL is not ideal for some uses (e.g. traffic engineering and real-time monitoring) due to the current fixed five-minute intervals. It is desirable that the updated *hnflc* can adjust the length of intervals according to available compute and storage resources and current system load. Thus, time sensitive applications can benefit from the finer granularity on more powerful hardware platforms.

Application Tagging

Port numbers are the only identifiers used to identify network applications in the current version of HNFL tool. However, the DNS resource records available locally can be used to tag individual edges with the main application types, which are related to the domain name of the edge's outside node. With this kind of information, researchers can determine the application of the edge more accurately.

Open API to Other Applications

Many applications can benefit from the passive measurement data captured by the HNFL tool. An Application Programming Interface (API) can enable other researchers and developers to contribute new applications for home networks, such as advanced network diagnosis and troubleshooting, dynamic traffic engineering, and advanced network security.

Meanwhile, better applications can make the measurement system more appealing to users.

Improved Dashboard

The existing dashboard only visualizes network usage data of the most recent interval to users. More information is available to users by using historical data, which is already saved locally on the router. Otherwise, the integration of traffic control functions into the dashboard is also desirable so that users may adjust traffic priority among devices while some users in the household are experiencing degraded performance.

Fully-Functional Mobile App

Unlike users of off-the-shelf routers from large manufacturers, users of OpenWrt routers do not have the complementary mobile app ready for them. A fully-functional mobile app designed specifically for OpenWrt routers running the HNFL software will be a great incentive for home network users. The mobile app should offer direct access not only to the visualized dashboard but also to configuration functions.

Appendices

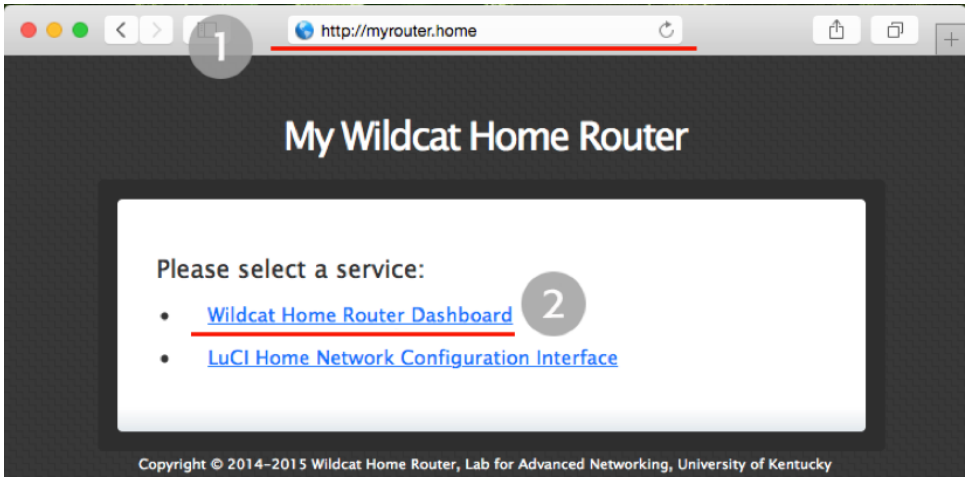
Appendix A

Understanding Network Usage Via Dashboard: the Instructions Given to Wildcat Home Router Users

Wildcat Home Router takes snapshots of the network usage every five minutes. You can learn about your home network status with metrics like online devices, bandwidth usage, network traffic distribution, and so forth.

A.1 Open Dashboard

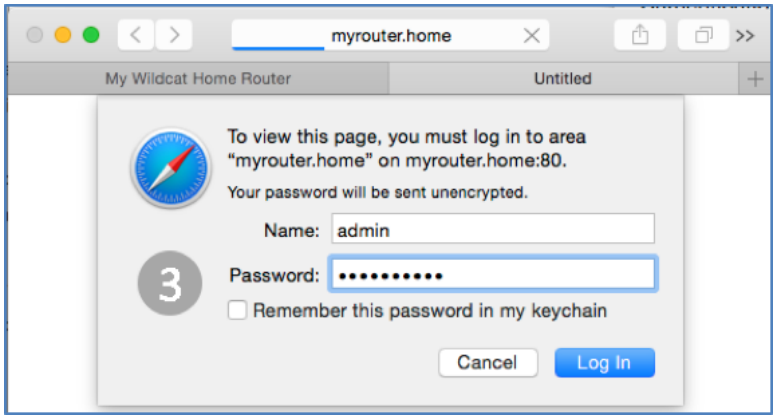
Follow the step below to open the dashboard:



Open the dashboard

1. Open a browser on your computer and type in http://myrouter.home in the address bar to go to the dashboard. (Note: You must type the "http://" in order to get to the dashboard page. If you omit it, you will get a page of search results.)

2. Click on "Wildcat Home Router Dashboard"



Dashboard login

3. Enter the username and password provided (there is a sticker on the back of the router with this information), then click "Log In".

A.2 Understanding the Dashboard

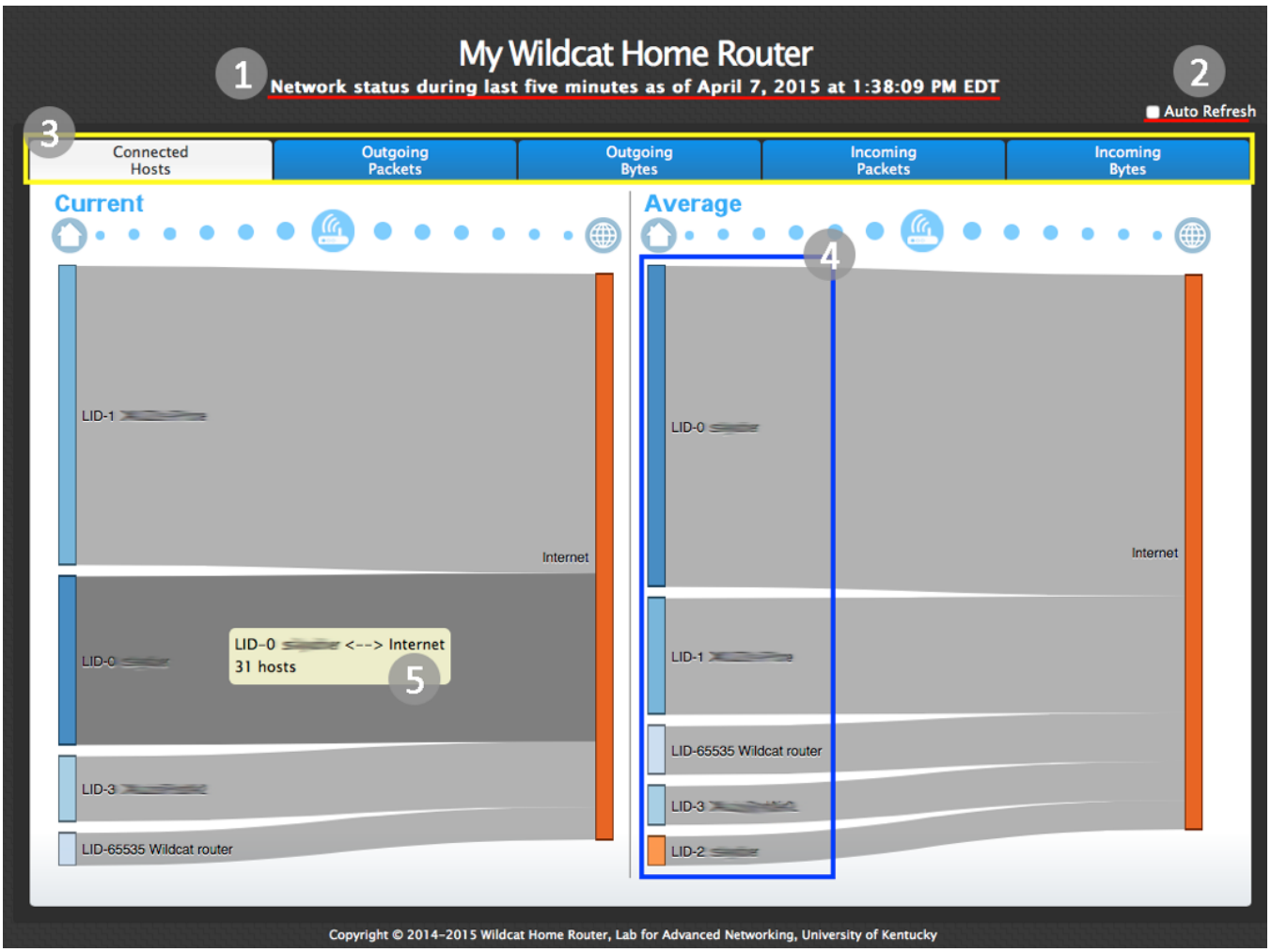
You will see three different graphs from the dashboard:

- **Current:** Show the overall usage of devices during the five-minute period according to the selected metric.

- **Average:** Show the approximate average usage of devices during the past hour according to the selected metric.

- **Top 5 Links (not in "Connected Hosts" tab):** Show top 5 most popular Internet destinations of each local device according to the selected metric.

The dashboard is explained as following:



Dashboard interface - a

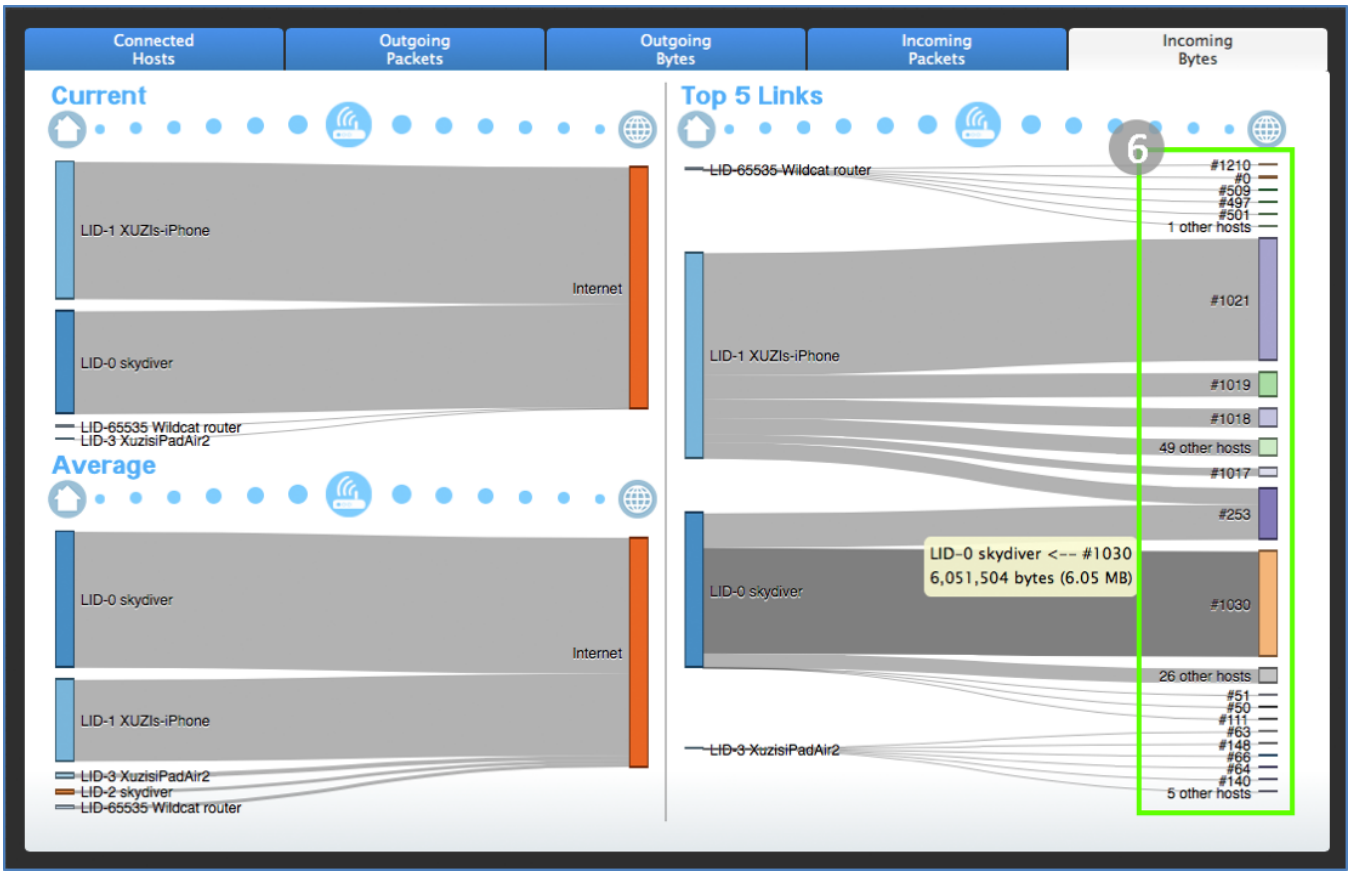
1. On the top of the page, there is a time stamp showing when the measurements were made.

2. The dashboard page does not automatically refresh itself by default. But you could check the "Auto Refresh" box to enable the feature.

3. The dashboard page has five tabs indicating five different metrics the wildcat router is tracking for you:

   - *Connected Hosts:* show how many devices are using your home network
   - *Outgoing Packets:* network packets sent from local devices to Internet
   - *Outgoing Bytes:* size of data sent from local devices to Internet
   - *Incoming Packets:* network packets received by local devices from Internet
   - *Incoming Bytes:* size of data received by local devices from Internet

4. On the left side of each graph, you can see a list of local devices in your home network. The width of each node indicates the proportion of its network usage.



Dashboard interface - b

5. You can move the mouse over the graphic elements (nodes or paths) to check detailed metrics.

6. For tabs related to packets and bytes, you can see more detailed network traffic distribution in "Top 5 Links" graphs. On the right side of a "Top 5 Links" graph, you will see network usage related to specific Internet destinations. These outside hosts are anonymized with numbers for privacy considerations.

Appendix B

MySQL Database Tables

The data collected from Wildcat Routers project are managed in the following four MySQL tables:

1. Devices

MySQL table: devices

Field	Type	Primary Key
router_name	char(20)	√
lid	bigint(20) unsigned	√
oui	char(6)	
org	char(90)	

- lid: foreign(outside) host identifier
- oui: Organizationally Unique Identifier (OUI)
- org: organization/manufacturer name

2. Summary of data from routers

MySQL table: router_info

Field	Type	Primary Key
router_name	char(20)	√
timezone	char(32)	
start_time	int(10) unsigned	
end_time	int(10) unsigned	
num_uploads	bigint(20) unsigned	
oui_enabled	tinyint(3) unsigned	

3. Network edges from router's traffic

MySQL table: router_edges

Field	Type	Primary Key
timestamp	int(10) unsigned	√
router_name	char(20)	√
fid	char(22)	√
flows	bigint(20) unsigned	
flows_in	bigint(20) unsigned	
pkts_out	bigint(20) unsigned	
pkts_in	bigint(20) unsigned	
bytes_out	bigint(20) unsigned	
bytes_in	bigint(20) unsigned	
flows_out_itvl	bigint(20) unsigned	
flows_in_itvl	bigint(20) unsigned	
pkts_out_itvl	bigint(20) unsigned	
pkts_in_itvl	bigint(20) unsigned	
bytes_out_itvl	bigint(20) unsigned	
bytes_in_itvl	bigint(20) unsigned	
prot	tinyint(3) unsigned	
port_1	smallint(5) unsigned	
port_1_flows	bigint(20) unsigned	
port_2	smallint(5) unsigned	
port_2_flows	bigint(20) unsigned	
port_3	smallint(5) unsigned	
port_3_flows	bigint(20) unsigned	
dur[1]	bigint(20) unsigned	

- timestamp: the Unix time value in seconds
- fid: foreign(outside) host identifier
- *_out: field related to outgoing traffic
- *_in: field related to incoming traffic
- prot: transport layer protocols

4. Network edges from user's traffic

MySQL table: user_edges

Field	Type	Primary Key
timestamp	int(10) unsigned	√
router_name	char(20)	√
lid	bigint(20) unsigned	√
fid	char(22)	√
flows_out	bigint(20) unsigned	
flows_in	bigint(20) unsigned	
pkts_out	bigint(20) unsigned	
pkts_in	bigint(20) unsigned	
bytes_out	bigint(20) unsigned	
bytes_in	bigint(20) unsigned	
flows_out_itvl	bigint(20) unsigned	
flows_in_itvl	bigint(20) unsigned	
pkts_out_itvl	bigint(20) unsigned	
pkts_in_itvl	bigint(20) unsigned	
bytes_out_itvl	bigint(20) unsigned	
bytes_in_itvl	bigint(20) unsigned	
prot	tinyint(3) unsigned	
port_1	smallint(5) unsigned	
port_1_flows	bigint(20) unsigned	
port_2	smallint(5) unsigned	
port_2_flows	bigint(20) unsigned	
port_3	smallint(5) unsigned	
port_3_flows	bigint(20) unsigned	
dur	bigint(20) unsigned	

Appendix C

MySQL Queries for Data Analysis

This appendix presents a list of SQL queries used in data analysis. Details of database tables are available in Appendix B. The selected list of queries are:

1. Count total number of active calendar days for a specific router:

```
SELECT t1.router_name, count(*)
FROM (
  SELECT DISTINCT
    router_name, DATE(FROM_UNIXTIME(timestamp)) AS date
  FROM user_edges
  WHERE router_name = [ROUTER_ID]
) AS t1
GROUP BY t1.router_name
```

2. Count number of intervals for a specific router:

```
SELECT t1.router_name, COUNT(*)
FROM (
  SELECT router_name, timestamp
  FROM user_edges
  WHERE router_name = [ROUTER_ID]
  GROUP BY timestamp
) AS t1
```

3. Count number of active routers for each calendar day:

```
SELECT t1.date, COUNT(*)
FROM (
  SELECT
    DATE(FROM_UNIXTIME(timestamp)) as date, router_name
  FROM user_edges
  GROUP BY CONCAT(date, router_name)
) AS t1
GROUP BY t1.date
```

4. Count the number of TCP/UDP only edges (protocol: 1 for TCP and 2 for UDP):

```
SELECT COUNT(*)
FROM user_edges
WHERE port_2 = 0 and prot = 1
```

5. Get the number of intervals of the most long-lived edge for each router:

```
SELECT router_name, MAX(dur)
FROM user_edges
GROUP BY router_name
```

6. Count the number of outside hosts connected to each router during the study:

```
SELECT t1.router_name, COUNT(t1.fid)
FROM (
    SELECT DISTINCT router_name, fid
    FROM user_edges
) AS t1
GROUP BY t1.router_name
```

7. Get number of inside hosts that have contact with a specific outside host for each calendar day:

```
SELECT t1.fdate, count(t1.lid)
From (
  SELECT DISTINCT
    DATE(timestamp, `unixepoch', `localtime') AS fdate, lid
  FROM user_edges
  WHERE router_name = [ROUTER_ID]
  AND fid=[SOME FID]
) AS t1
GROUP BY t1.fdate
```

8. List the top 500 outside hosts for a specific router order by the number of edges related to the outside host:

```
SELECT router_name, fid, count(fid) AS fid_cnt
FROM user_edges
WHERE router_name = [ROUTER_ID]
GROUP BY fid
ORDER BY fid_cnt DESC LIMIT 500
```

9. List the top 100 applications (according to port numbers) for specific router:

```
SELECT port_1, COUNT(port_1) AS cnt
FROM user_edges
WHERE router_name = [ROUTER_ID]
GROUP BY port_1
ORDER BY cnt DESC LIMIT 100
```

10. List all pairs of inside host and outside host related to a specific application according to port numbers. For example, query for email applications (IMAP, POP, SMTP):

```
SELECT router_name, lid, fid, count(*) as fcount
FROM user_edges
WHERE port_1 in (110, 995, 143, 993, 25, 2525, 465)
  OR port_2 in (110, 995, 143, 993, 25, 2525, 465)
  OR port_3 in (110, 995, 143, 993, 25, 2525, 465)
GROUP BY CONCAT(router_name, lid, fid);
```

11. Count number of devices appeared for all routers:

```
SELECT router_name, COUNT(*)
FROM devices
GROUP BY router_name
```

12. Find the number of active days of all devices from a specific router and also display the manufacturer information of the devices if available:

```
SELECT
  t1.router_name, t1.lid, count(t1.lid), t2.oui, t2.org
FROM (
  SELECT DISTINCT
    router_name, lid, DATE(FROM_UNIXTIME(timestamp)) as date
  FROM user_edges
  WHERE router_name = [ROUTER_ID]
) AS t1
INNER JOIN devices as t2
ON t1.router_name=t2.router_name AND t1.lid = t2.lid
GROUP BY t1.lid
```

13. Get the highest downlink transmission rate observed:

```
SELECT PRINTF(`\%.2f', bytes_in_itvl / 39321600.0)
FROM user_edges
WHERE router_name = [ROUTER_ID]
ORDER BY bytes_in_itvl DESC LIMIT 1
```

14. Get the highest uplink transmission rate observed:

```
SELECT PRINTF(`\%.2f', bytes_out_itvl / 39321600.0)
FROM user_edges
WHERE router_name= [ROUTER_ID]
ORDER BY bytes_out_itvl DESC LIMIT 1
```

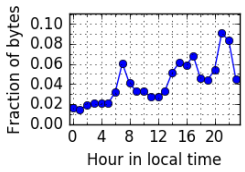
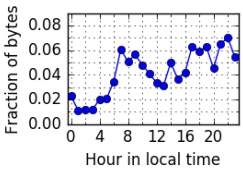
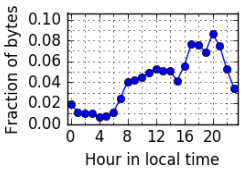
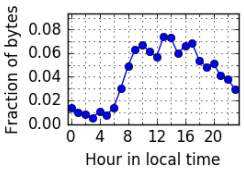
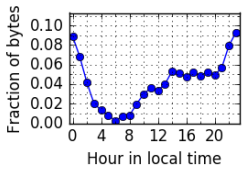
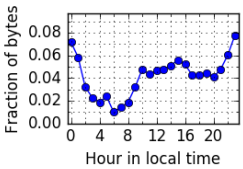
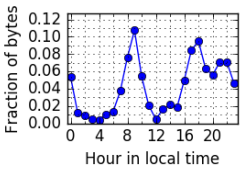
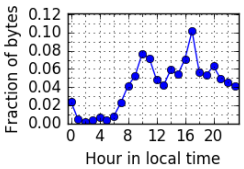
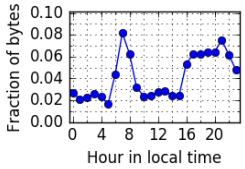
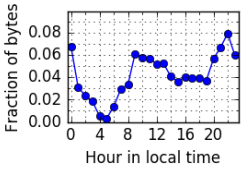
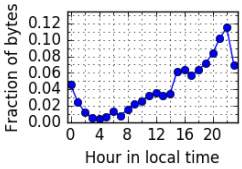
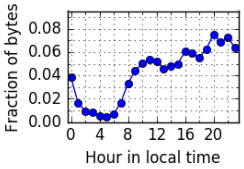
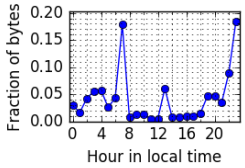
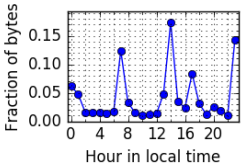
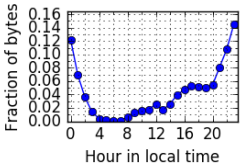
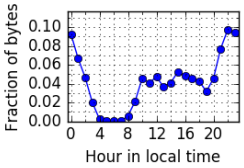
15. Calculate the sum for each traffic parameter (number of outgoing/incoming packets, number of outgoing/incoming bytes) within each clock hour for a specific router:

```
SELECT
  HOUR(FROM_UNIXTIME(timestamp)) AS hour,
  SUM(pkts_out_itvl) AS pkts_out,
  SUM(pkts_in_itvl) AS pkts_in,
  SUM(bytes_out_itvl) AS bytes_out,
  SUM(bytes_in_itvl) AS bytes_in
FROM user_edges
WHERE router_name = [ROUTER_ID]
GROUP BY hour
```

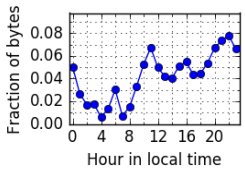
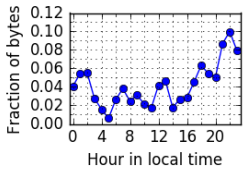
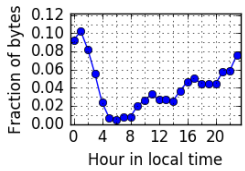
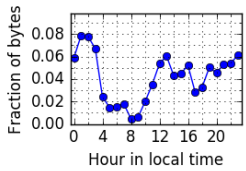
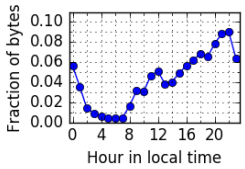
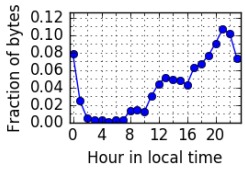
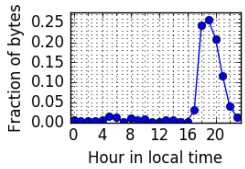
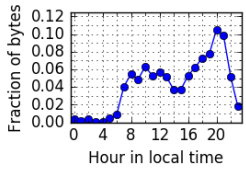
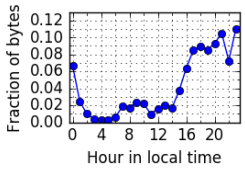
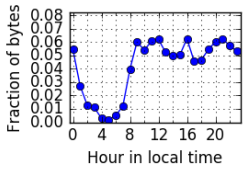
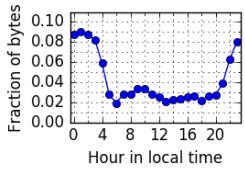
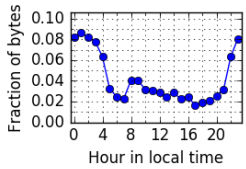
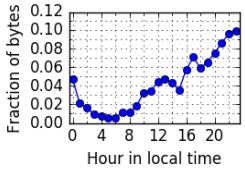
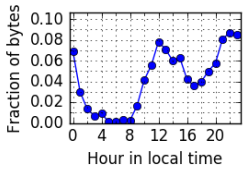
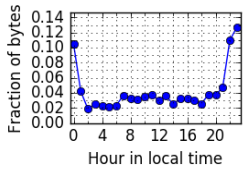
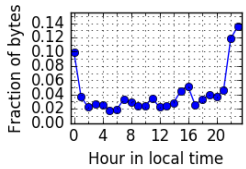
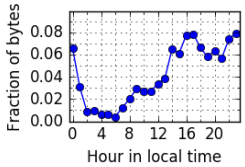
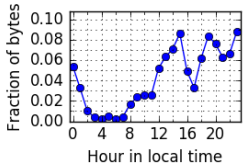
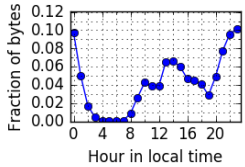
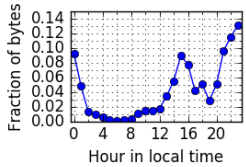
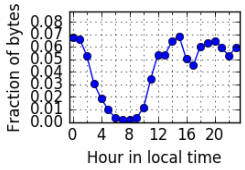
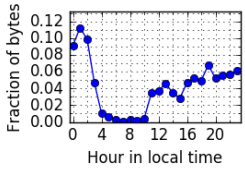
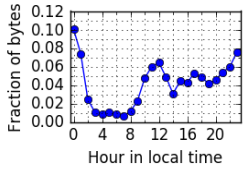
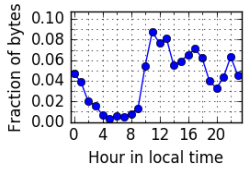
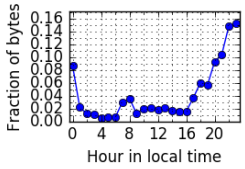
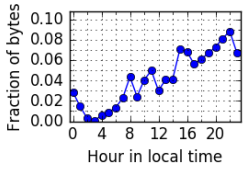
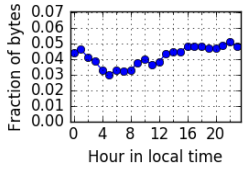
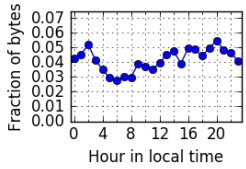
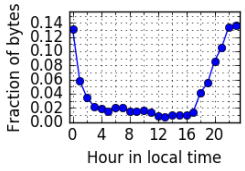
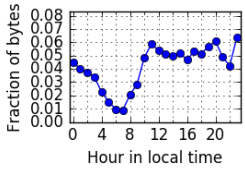
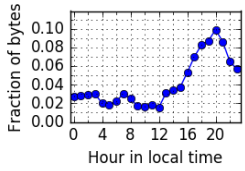
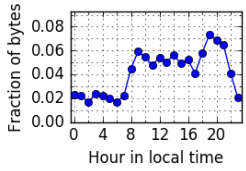
Appendix D

Diurnal Network Usage Patterns of All Households

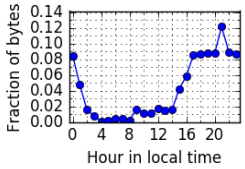
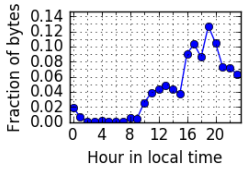
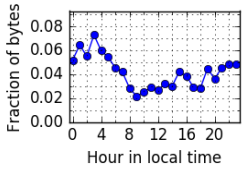
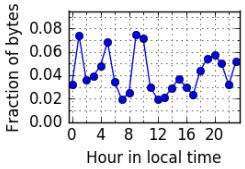
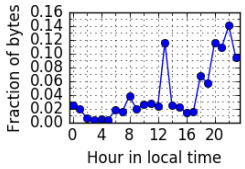
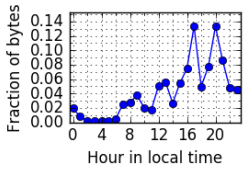
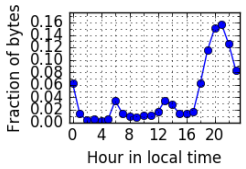
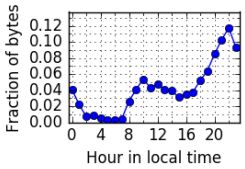
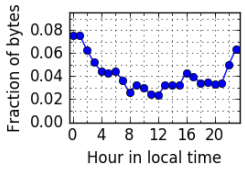
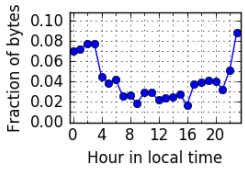
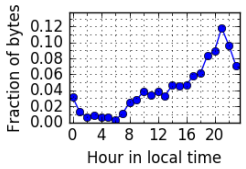
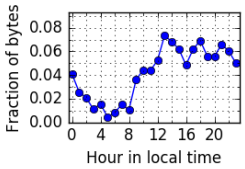
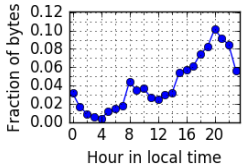
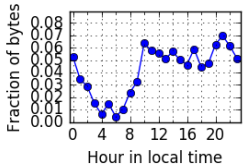
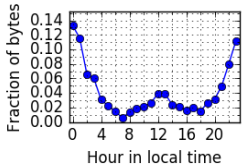
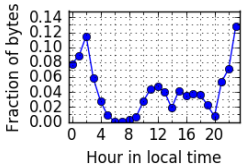
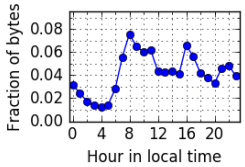
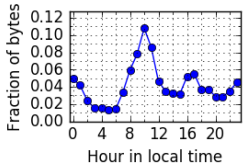
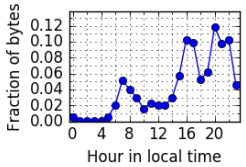
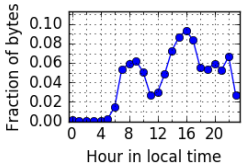
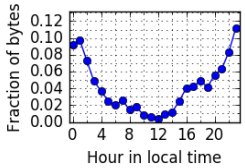
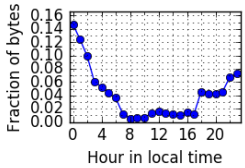
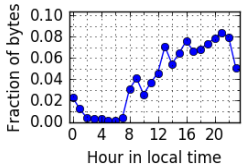
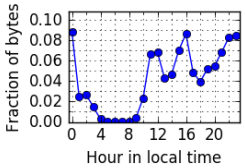
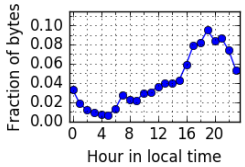
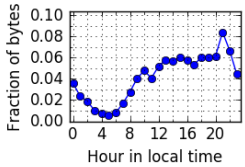
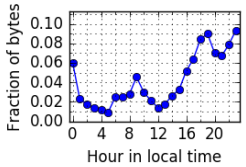
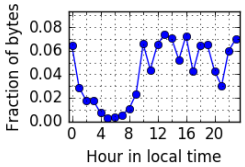
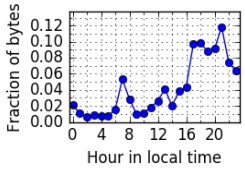
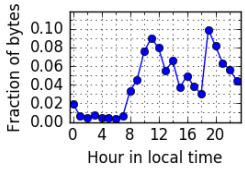
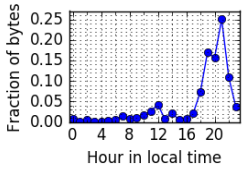
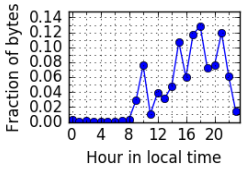
The weekday and weekend network usage patterns of all participating households are listed in tables below sorted by the household ID in ascending order:

ID	Weekday Pattern	Weekend Pattern	ID	Weekday Pattern	Weekend Pattern
01			02		
03			04		
05			06		
07			08		

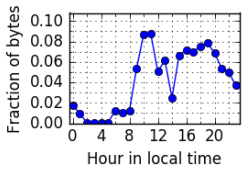
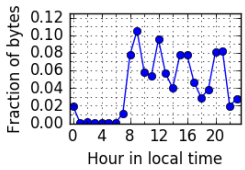
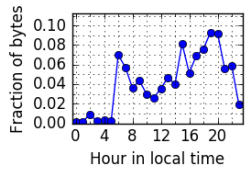
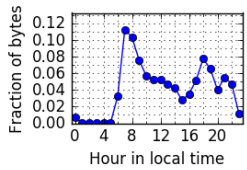
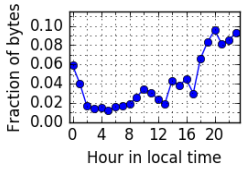
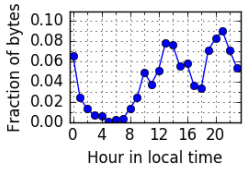
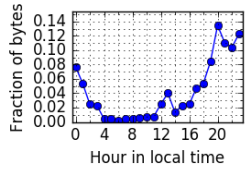
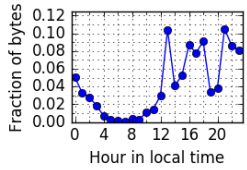
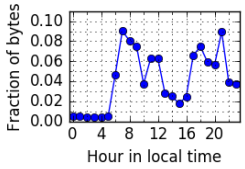
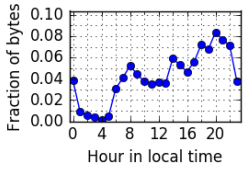
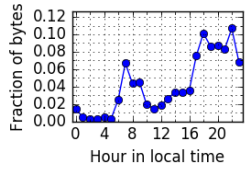
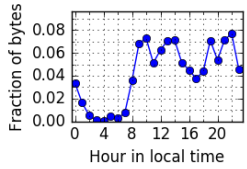
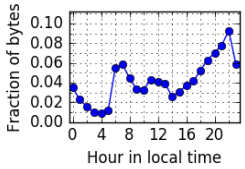
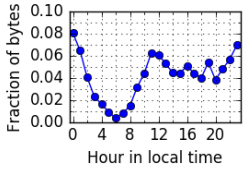
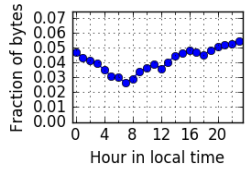
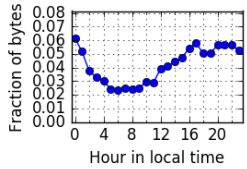
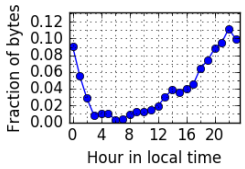
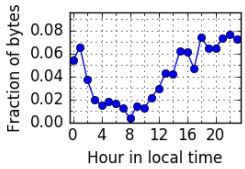
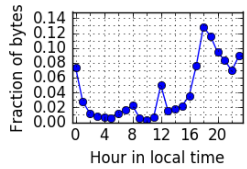
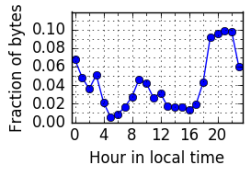
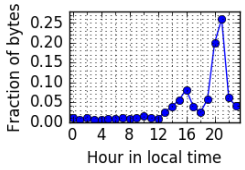
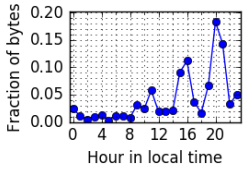
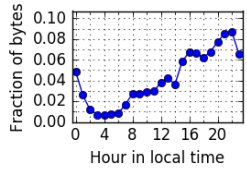
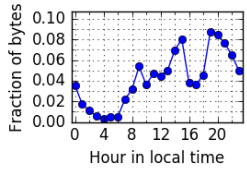
Diurnal usage patterns (household 01 - 08)

ID	Weekday Pattern	Weekend Pattern	ID	Weekday Pattern	Weekend Pattern
09			10		
11			12		
13			14		
15			16		
17			18		
19			20		
21			22		
23			24		

Diurnal usage patterns (household 09 - 24)

ID	Weekday Pattern	Weekend Pattern	ID	Weekday Pattern	Weekend Pattern
25			27		
28			29		
30			31		
32			33		
34			35		
36			37		
38			39		
40			41		

Diurnal usage patterns (household 25, 27 - 41)

ID	Weekday Pattern	Weekend Pattern	ID	Weekday Pattern	Weekend Pattern
42			43		
44			45		
46			47		
48			49		
50			51		
52			53		

Diurnal usage patterns (household 42 - 53)

Bibliography

[1] ITU and UNESCO. The state of broadband 2015. http://www.broadbandcommission.org/documents/reports/bb-annualreport2015.pdf, September 2015. Last retrieved 2016-07-01.

[2] M. Linsner, P. Eardley, Burbridge T., and Sorensen F. Large-Scale Broadband Measurement Use Cases. https://tools.ietf.org/html/rfc7536, May 2015. Last retrieved 2016-07-01.

[3] Gregor Maier, Fabian Schneider, and Anja Feldmann. Nat usage in residential broadband networks. In *Proceedings of the 12th International Conference on Passive and Active Measurement*, PAM'11, pages 32–41, Berlin, Heidelberg, 2011. Springer-Verlag.

[4] SamKnows. How we measure fixed-line broadband. https://www.samknows.com/meet-the-whitebox. Last retrieved 2016-07-01.

[5] Sarthak Grover, Mi Seon Park, Srikanth Sundaresan, Sam Burnett, Hyojoon Kim, Bharath Ravi, and Nick Feamster. Peeking behind the nat: An empirical study of home networks. In *Proceedings of the 2013 Conference on Internet Measurement Conference*, IMC '13, pages 377–390, New York, NY, USA, 2013. ACM.

[6] Srikanth Sundaresan, Sam Burnett, Nick Feamster, and Walter de Donato. Bismark: A testbed for deploying measurements and applications in broadband access networks. In *Proc. of. USENIX*, 2014.

[7] K. Claffy, Y. Hyun, K. Keys, M. Fomenkov, and D. Krioukov. Internet Mapping: from Art to Science. In *IEEE DHS Cybersecurity Applications and Technologies Conference for Homeland Security (CATCH)*, pages 205–211, Watham, MA, Mar 2009.

[8] Richard Mortier, Tom Rodden, Peter Tolmie, Tom Lodge, Robert Spencer, Andy crabtree, Joe Sventek, and Alexandros Koliousis. Homework: Putting interaction into the infrastructure. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology(UIST)*, pages 197–206, 2012.

[9] Xuzi Zhou and Ken L. Calvert. Lightweight privacy-preserving passive measurement for home networks. In *IEEE International Conference on Communications (ICC)*, 2015.

[10] Ping: send icmp echo_request to network hosts. http://linux.die.net/man/8/ping. Last retrieved 2016-07-01.

[11] Traceroute: print the route packets trace to network host. http://linux.die.net/man/8/traceroute. Last retrieved 2016-07-01.

[12] iperf: the network bandwidth measurement tool. https://iperf.fr/. Last retrieved 2016-07-01.

[13] Tcpdump: a powerful command-line packet analyzer. http://www.tcpdump.org/. Last retrieved 2016-07-01.

[14] Wireshark: Go Deep. https://www.wireshark.org/. Last retrieved 2016-07-01.

[15] Marcel Dischinger, Andreas Haeberlen, Krishna P. Gummadi, and Stefan Saroiu. Characterizing residential broadband networks. In *Proceedings of the ACM/USENIX Internet Measurement Conference (IMC'07)*, Oct 2007.

[16] Aaron Schulman and Neil Spring. Pingin' in the rain. In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, IMC '11, pages 19–28, New York, NY, USA, 2011. ACM.

[17] Brent Chun, David Culler, Timothy Roscoe, Andy Bavier, Larry Peterson, Mike Wawrzoniak, and Mic Bowman. Planetlab: An overlay testbed for broad-coverage services. *SIGCOMM Comput. Commun. Rev.*, 33(3):3–12, July 2003.

[18] Gregor Maier, Anja Feldmann, Vern Paxson, and Mark Allman. On dominant characteristics of residential broadband internet traffic. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference*, IMC '09, pages 90–102, New York, NY, USA, 2009. ACM.

[19] Matthew Sargent and Mark Allman. Performance within a fiber-to-the-home network. *ACM SIGCOMM Computer Communication Review*, 44(3):22–30, July 2014.

[20] Christian Kreibich, Nicholas Weaver, Boris Nechaev, and Vern Paxson. Netalyzr: Illuminating the edge network. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, IMC '10, pages 246–259, New York, NY, USA, 2010. ACM.

[21] Mohan Dhawan, Justin Samuel, Renata Teixeira, Christian Kreibich, Mark Allman, Nicholas Weaver, and Vern Paxson. Fathom: A browser-based network measurement platform. In *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*, IMC '12, pages 73–86, New York, NY, USA, 2012. ACM.

[22] Mario A Sánchez, John S Otto, Zachary S Bischof, David R Choffnes, Fabián E Bustamante, Balachander Krishnamurthy, and Walter Willinger. Dasu: Pushing experiments to the internet's edge. In *Proc. of USENIX NSDI*, 2013.

[23] Vuze Bittorrent Client. http://www.vuze.com.

[24] Lucas DiCioccio, Renata Teixeira, Martin May, and Christian Kreibich. Probe and pray: Using upnp for home network measurements. In *Proceedings of the 13th International Conference on Passive and Active Measurement*, PAM'12, pages 96–105, Berlin, Heidelberg, 2012. Springer-Verlag.

[25] UPnP Forums. UPnP Specifications. http://www.upnp.org. Last retrieved 2016-07-01.

[26] Lucas DiCioccio, Renata Teixeira, and Catherine Rosenberg. Measuring home networks with homenet profiler. In *Proceedings of the 14th International Conference on Passive and Active Measurement*, PAM'13, pages 176–186, Berlin, Heidelberg, 2013. Springer-Verlag.

[27] OpenWrt. https://openwrt.org/. Last retrieved 2016-07-01.

[28] SamKnows Test Methodology: Methodology and Technical Information Relating to The SamKnows Testing Platform. https://www.samknows.com/broadband/uploads/methodology/SQ301-005-EN-Test-Suite-Whitepaper-4.pdf. Last retrieved 2016-07-01.

[29] SamKnows Performance Monitoring. https://reporting.samknows.com. Last retrieved 2016-07-01.

[30] Zachary S. Bischof, Fabian E. Bustamante, and Rade Stanojevic. Need, want, can afford: Broadband markets and the behavior of users. In *Proceedings of the 2014 Conference on Internet Measurement Conference*, IMC '14, pages 73–86, New York, NY, USA, 2014. ACM.

[31] Federal Communications Commission (FCC). Measuring Broadband America. http://www.fcc.gov/measuring-broadband-america. Last retrieved 2016-07-01.

[32] Policy by the Numbers. International Broadband Pricing Study: Updated dataset. http://policybythenumbers.blogspot.com/2013/05/international-broadband-pricing-study.html. Last retrieved 2016-07-01.

[33] Srikanth Sundaresan, Walter De Donato, Nick Feamster, Renata Teixeira, Sam Crawford, and Antonio Pescapè. Measuring home broadband performance. *Communications of the ACM*, 55(11):100–109, 2012.

[34] Partha Kanuparthy and Constantine Dovrolis. Shaperprobe: End-to-end detection of isp traffic shaping using active methods. In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, IMC '11, pages 473–482, New York, NY, USA, 2011. ACM.

[35] S. Avallone, S. Guadagno, D. Emma, A. Pescape, and G. Ventre. D-itg distributed internet traffic generator. In *Proceedings of the The Quantitative Evaluation of Systems, First International Conference*, QEST '04, pages 316–317, Washington, DC, USA, 2004. IEEE Computer Society.

[36] Brice Augustin, Xavier Cuvellier, Benjamin Orgogozo, Fabien Viger, Timur Friedman, Matthieu Latapy, Clémence Magnien, and Renata Teixeira. Avoiding traceroute anomalies with paris traceroute. In *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement*, IMC '06, pages 153–158, New York, NY, USA, 2006. ACM.

[37] Kenneth L Calvert, W Keith Edwards, Nick Feamster, Rebecca E Grinter, Ye Deng, and Xuzi Zhou. Instrumenting home networks. *ACM SIGCOMM Computer Communication Review*, 41(1):84–89, 2011.

[38] Natasha Gude, Teemu Koponen, Justin Pettit, Ben Pfaff, Martín Casado, Nick McKeown, and Scott Shenker. Nox: Towards an operating system for networks. *SIGCOMM Comput. Commun. Rev.*, 38(3):105–110, July 2008.

[39] STREAM. Stanford stream data manager. http://infolab.stanford.edu/stream/.

[40] Marshini Chetty, Richard Banks, Richard Harper, Tim Regan, Abigail Sellen, Christos Gkantsidis, Thomas Karagiannis, and Peter Key. Who's hogging the bandwidth: The consequences of revealing the invisible in the home. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 659–668, New York, NY, USA, 2010. ACM.

[41] Marshini Chetty, David Haslem, Andrew Baird, Ugochi Ofoha, Bethany Sumner, and Rebecca Grinter. Why is my internet slow?: Making network speeds visible. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1889–1898, New York, NY, USA, 2011. ACM.

[42] Marshini Chetty, Richard Banks, A.J. Brush, Jonathan Donner, and Rebecca Grinter. You're capped: Understanding the effects of bandwidth caps on broadband use in the home. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 3021–3030, New York, NY, USA, 2012. ACM.

[43] Marshini Chetty, Hyojoon Kim, Srikanth Sundaresan, Sam Burnett, Nick Feamster, and W. Keith Edwards. ucap: An internet data management tool for the home. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 3093–3102, New York, NY, USA, 2015. ACM.

[44] dd-wrt.com. http://www.dd-wrt.com/site/index. Last retrieved 2016-07-01.

[45] A.I. Rana and B. Jennings. Semantic Uplift of Monitoring Data to Select Policies to Manage Home Area Networks. In *Advanced Information Networking and Applications (AINA), 2012 IEEE 26th International Conference on*, pages 368–375, 2012.

[46] TShark. http://www.wireshark.org/docs/man-pages/tshark.html. Last retrieved 2016-07-01.

[47] Nick Feamster. Outsourcing home network security. In *Proceedings of the 2010 ACM SIGCOMM Workshop on Home Networks*, HomeNets '10, pages 37–42, New York, NY, USA, 2010. ACM.

[48] Mukarram Bin Tariq, Murtaza Motiwala, and Nick Feamster. Nano: Network access neutrality observatory. 2008.

[49] Mukarram Bin Tariq, Murtaza Motiwala, Nick Feamster, and Mostafa Ammar. Detecting network neutrality violations with causal inference. In *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, pages 289–300. ACM, 2009.

[50] Bhavish Agarwal, Ranjita Bhagwan, Tathagata Das, Siddharth Eswaran, Venkata N Padmanabhan, and Geoffrey M Voelker. NetPrints: Diagnosing Home Network Misconfigurations Using Shared Knowledge. In *NSDI*, volume 9, pages 349–364, 2009.

[51] OpenWrt-Table of Hardware. http://wiki.openwrt.org/toh/start. Last retrieved 2016-07-01.

[52] The netfilter.org project. http://www.netfilter.org/. Last retrieved 2016-07-01.

[53] National Institute of Standards and Technology. Announcing the Advanced Encryption Standard (AES). http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf, November 2001. Last retrieved 2016-07-01.

[54] Nping: Network packet generation tool. http://nmap.org/nping/. Last retrieved 2016-07-01.

[55] Dig - DNS lookup utility. http://linux.die.net/man/1/dig. Last retrieved 2016-07-01.

[56] LuCI by OpenWrt. https://github.com/openwrt/luci/wiki. Last retrieved 2016-07-01.

[57] Data-Driven Documents JavaScript Libaray. http://standards-oui.ieee.org/oui.txt. Last retrieved 2016-07-01.

[58] uHTTPd by OpenWrt. https://wiki.openwrt.org/doc/howto/http.uhttpd. Last retrieved 2016-07-01.

[59] NETGEAR Genie App. https://www.netgear.com/home/discover/apps/genie.aspx. Last retrieved 2016-07-01.

[60] TP-Link Tether App. http://www.tp-link.com/en/pages/common/promos/app_tether_v2.html. Last retrieved 2016-07-01.

[61] The UCI System. https://wiki.openwrt.org/doc/uci. Last retrieved 2016-07-01.

[62] IEEE Standards OUI. http://standards-oui.ieee.org/oui.txt. Last retrieved 2016-07-01.

[63] Recommended settings for Wi-Fi routers and access points. https://support.apple.com/en-us/HT202068. Last retrieved 2016-07-01.

[64] Joe H. Ward. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963.

[65] CAIDA. The CAIDA UCSD anonymized internet traces 2015. http://www.caida.org/data/passive/passive_2015_dataset.xml.

[66] Google IPv6 Statistics. https://www.google.com/intl/en/ipv6/statistics.html. Last retrieved 2016-07-01.

Vita

- Education
  - Shandong University, Jinan, Shandong, China, *B.Eng. in Electronic Commerce and B.Sc. in Finance*, Sep. 2005 – July 2009
  - National Cheng Kung University, Tainan, Taiwan, *Exchange Student in Computer Science*, Sep. 2007 – Jan. 2008
- Employment History
  - EMC DataDomain, Santa Clara, CA, Software Engineer Intern, Jun. 2015 – Sep. 2015
  - University of Kentucky, Lexington, KY, Research Assistant, Jan. 2010 - May. 2015 & Sep. 2015 – Aug. 2016
  - University of Kentucky, Lexington, KY, Teaching Assistant, Aug. 2009 – Jan. 2010
- Publications
  - Xuzi Zhou and Kenneth L. Calvert, "Living on the edge: A passive measurement study of home network traffic", *Under review in INFOCOM 2017 as of Aug. 25th 2016*
  - Xuzi Zhou and Kenneth L. Calvert, "Lightweight privacy-preserving passive measurement for home networks", *In Proceedings of International Conference on Communications (ICC), IEEE, 2015*
  - Kenneth. L. Calvert, W. K. Edwards, N. Feamster, R. E. Grinter, Y. Deng, and X. Zhou, "Instrumenting home networks", *ACM SIGCOMM Computer Communication Review, vol. 41, no. 1, pp. 84-89, 2011*
  - Xiwei Wang, Erik Osten, Xuzi Zhou, and Hui Lin, "A case study of recommendation algorithms", *in Proceedings of International Conferences on Computational and Information Sciences (ICCIS), IEEE, 2011*
- Patent
  - Xuzi Zhou, A New Structure for Multi-functioning Triangular Scale, China Patent CN2885600
- Awards
  - Best Presentation Award, Xuzi Zhou, Scott workman, Mohammad Islam, Nathan Jacobs, and James Griffioen, "Cyber Infrastructure for the VOEIS Project", 26th Annual Eastern Kentucky University Symposium in the Mathematical, Statistical, and Computer Science, 2013
  - Innovation Award Credit, Shandong University, 2007
  - Outstanding Student Scholarship, Shandong University, 2007

- Software
  - **privpresMon**: an OpenWrt implementation of the HNFL software.
    - https://github.com/UKY-netlab/privpresMon
  - **SuperBoard**: an iOS app simulating the LED board with three different display and editing modes.
    - https://itunes.apple.com/us/app/id951480740
- Certification
  - Certified Associate in Software Testing, QAI Global Institute, #50530