Domestic and mobile networks : measurements, analyses and patterns

Ahlem Reggani

Thèse de doctorat

UNIVERSITÉ PIERRE ET MARIE CURIE
UPMC SORBONNE UNIVERSITÉS

École doctorale

EDITE DE PARIS
INFORMATIQUE, TÉLÉCOMMUNICATIONS ET ÉLECTRONIQUE

présentée par

Ahlem Reggani

pour obtenir le grade de

Docteur de l'Université Pierre et Marie Curie

Réseaux domestiques et mobiles : Mesures, analyses, et modèles

soutenue le 07 Janvier 2014 devant le jury composé de :

Yacine Ghamri-Doudane	Rapporteur	Professeur Université de La Rochelle
Nathalie Mitton	Rapporteur	Chargée de Recherche INRIA
Farid Benbadis	Examinateur	Ingénieur de Recherche Thalès
Nadjib Achir	Examinateur	Maitre de Conférence Université Paris-Nord
Serge Fdida	Examinateur	Professeur UPMC Sorbonne Universités
Marcelo Dias de Amorim	Directeur	Directeur de Recherche CNRS

PhD Thesis

UNIVERSITY PIERRE AND MARIE CURIE
UPMC SORBONNE UNIVERSITÉS

Doctoral school

EDITE DE PARIS
COMPUTER SCIENCE, TELECOMMUNICATIONS, AND ELECTRONICS

presented by

Ahlem Reggani

submitted in partial fulfillment of the requirements for the degree of

Doctor of Science of the University Pierre and Marie Curie

Domestic and mobile networks: Measurements, analyses, and patterns

Commitee in charge:

Yacine Ghamri-Doudane	Reviewer	Professor Université de La Rochelle
Nathalie Mitton	Reviewer	INRIA Research Fellow
Farid Benbadis	Examiner	Research Engineer Thalès
Nadjib Achir	Examiner	Associate Professor Université Paris-Nord
Serge Fdida	Examiner	Professor UPMC Sorbonne Universités
Marcelo Dias de Amorim	Advisor	CNRS Research Director

Résumé

Cette thèse est structurée autour de contributions dans les domaines des réseaux domestiques et mobiles. Dans le contexte des réseaux domestiques, nous nous occupons à la fois de la caractérisation du trafic et de la dégradation des performances des applications. Dans le cas des réseaux mobiles, nous sommes intéressés par comprendre la relation entre la technologie sans fil et les opportunités de contact entre les nœuds mobiles. Nous résumons les principales contributions de cette thèse dans ce qui suit.

Partie I (*Optimisation des performances des applications dans les réseaux domestiques*). L'augmentation du taux d'accès à Internet à la maison conduit à plus de populations avec des réseaux domestiques. Un réseau domestique connecte plusieurs appareils à l'Internet permettant aux différents membres d'un ménage de partager l'accès à Internet et aux ressources du réseau local. Par conséquent, les applications fonctionnant en parallèle peuvent interférer les unes avec les autres. Par exemple, les enfants peuvent jouer à des jeux en ligne ralentissant la navigation sur le web de leurs parents. Le premier objectif de cette thèse est de contrôler l'utilisation des ressources du réseau domestique afin d'optimiser la performance des applications concurrentes. La passerelle domestique est responsable de la connexion du réseau domestique au reste de l'Internet. Parce que la passerelle a une vue d'ensemble de tout le trafic en provenance et vers le réseau domestique, elle est le point de départ idéal pour l'optimisation des applications. Dans cette thèse, nous proposons un système qui fonctionne sur la passerelle domestique pour *détecter* des dégradations de performances et optimiser l'*allocation* des ressources pour obtenir les meilleures performances des applications.

En même temps, les passerelles résidentielles classiques ne comportent aucun mécanisme pour garantir une performance optimale aux applications. Une autre contribution de cette thèse est donc de proposer une approche d'*optimisation des performances des applications* pour les réseaux domestiques. En particulier, nous étudions la faisabilité du suivi des performances des applications sur les passerelles résidentielles. Nous montrons que, bien que la passerelle domestique a des ressources limitées, elle a encore la capacité de faire plus que simplement la transmission des paquets. Elle peut recueillir et exporter toutes les informations nécessaires pour effectuer notre méthode d'optimisation des performances.

Partie II (*Reproduction de traces de mobilité*). La meilleure façon d'analyser ou de valider un protocole ou même le choix de conception dans les réseaux tolérants aux perturbations est à travers un déploiement réel. Néanmoins, en raison des difficultés de mise en œuvre et même de coûts financiers,

seulement quelques expérimentations ont été rapportées dans la littérature. En conséquence, plusieurs travaux s'appuient toujours sur des modèles de mobilité synthétiques. Alors que les modèles de mobilité synthétiques sont utiles pour isoler les paramètres spécifiques d'une solution ou aider à enquêter sur l'évolutivité d'un système, ils ne peuvent pas toujours refléter les conditions réelles. D'autre part, les traces de contact sont connues pour mieux représenter la mobilité de la vie réelle, mais aussi d'être difficile à obtenir. Et si une trace réelle était suffisante pour obtenir plusieurs autres, comme si nous avions effectué plusieurs expérimentations ? À cette fin, nous nous appuyons sur la mobilité plausible, un algorithme capable d'inférer un mouvement spatial à partir de traces de contact et nous proposons un système de *reproduction* de traces de mobilité qui, à partir d'une unique trace de contact réelle, offre de multiples traces de contact inspirées de la trace originale. Nous vérifions la conformité de notre proposition en comparant les résultats de notre système avec la trace de contact originale d'un réseau mobile généré synthétiquement et montrons que le résultat de notre système reste cohérent avec la trace d'origine.

Mots-clefs

Réseaux domestiques, passerelle domestique, optimisation de performance, réseaux mobiles à connectivité intermittente, caractérisation des contacts, dynamique des réseaux, analyse basée sur graphes.

Abstract

This thesis is structured around contributions in the areas of domestic and mobile networks. In the context of home networks, we deal with both home traffic characterization and application performance degradation. In the case of mobile networks, we are interested in understanding the relationship between wireless technology and contact opportunities among nodes on the move. We summarize the main contributions of this thesis in the following.

Part I (*Application performance optimization in home networks*). The increasing penetration ratio of residential Internet access leads to more people with home networks. The home network connects many devices to the Internet allowing different members of a household to share internet access and local network resources. Thus, applications running in parallel can interfere with one another. For instance, children playing online games slow down their parents browsing over the web. The first focus of this thesis is to control the utilization of home network resources to optimize the performance of competing networked applications. The home gateway is in charge of connecting the home network to the rest of the Internet. Because it has an overall view of all the traffic coming from and going to the home network, it is the ideal point for application optimization. In this thesis, we propose a system that runs on the home gateway to *detect* performance degradation and optimize resource *allocation* to obtain the best application performance.

At the same time, typical home gateways do not include any mechanism to guarantee optimal *application performance*. Another contribution of our work is an application performance optimization approach for home networks. In particular, we study the feasibility of application performance tracking on home gateways. We show that, although the home gateway has limited resources, it still has the capacity to do more than just forwarding packets. It can collect and export all the information needed to perform our application performance optimization method.

Part II (*Mobility trace breeding*). The best way to analyze or validate any protocol or design choice in disruption-tolerant networks is through a real deployment. Nevertheless, because of implementation challenges and even financial costs, only a few experimentations have been reported in the literature. As a consequence, several works still rely on synthetic mobility models. While synthetic mobility models are useful to isolate specific parameters of a solution or help investigate the scalability of a system, they cannot always reflect real life conditions. On the other hand, contact traces are known to better represent real-life mobility but also to be hard to get. What if one real trace were sufficient to get multiple others, just as if we performed multiple experimentations? To this end, we rely on

plausible mobility, an algorithm capable of inferring spatial movement from contact traces and we propose a mobility trace *breeding* system that, from a single real-life contact trace, derives multiple contact traces inspired from the original trace. We check the conformity of our proposal by comparing the results of our breeding system with the original contact trace of a mobile network generated synthetically and show that the outcome of our system does correspond to the original trace.

Keywords

Domestic network, home gateway, performance optimization, intermittently-connected mobile networks, contact characterization, network dynamics, graph-based analysis.

Contents

Chapter 1

Introduction

UNDERSTANDING network dynamics is fundamental for the deployment of efficient communication protocols, strategies, or even hardware. For example, Facebook is being widely studied given the high dynamics of this social network. An interesting case study analyzes the effect of Facebook users on the evolution of the underlying network[1]. In other domains such as disruption-tolerant networks (DTN), protocols need to be tested in a dynamic network and thus require a fine understanding of communication opportunities that emerge between users. Another example is a residential area network (domestic networks), whose dynamics has a direct impact on how service providers should behave to provide the best possible service to their customers.

The last decade has seen an outstanding increase in connected devices[37]. People are practically always connected, inside their homes, at work, or even in the streets. Many devices offer multiple services to users, from fixed PC to laptops and smartphones. Often, each user has more than one of these devices. A consequence is the increase in the complexity of the interactions within the network, especially because they change over time.

This thesis addresses such problems in two main areas. Domestic and mobile networks. In home networks, we study application dynamics that helps the design, the monitoring, and the service level guarantee for a network. In mobile networks, we focus on node mobility and more specifically on the dynamics of mobility traces that are important to support protocol design and testing.

1.1. Problem space

1.1.1. From analysis to diagnosis in domestic networks

Different technologies (e.g., Wi-Fi and power-line communication) allow users in a household to connect a number of devices inside the home network and access the Internet using a single access link (see Figure A.1). In this context, all devices connected to the home network share the capacity of the access link and the local network. Hence, applications and services running on these devices can interfere with one another. For instance, a user can play an online game while another is starting a big file download. Given that both applications require bandwidth, there may be a negative effect

Figure 1.1: Home network example.

on the performance that users perceive. In such a case, it is hard for the user to determine where the problem is. Solving performance degradation is difficult for expert users and vague for the others[50]. Simple home network users apply simple solution strategies to fix their connectivity issues, such as unplugging and replugging their devices or rebooting the machine. But sometimes simple solutions are not enough. For instance, in case of poor wireless performance because of interferences between mobile devices, rebooting the access point does not fix the problem. If none of the users has a minimum technical skill, the problem will remain unsolved until an external hand is called. Many studies have been made around the access link performance[19;106] but so far only few are toward home networks performance.

To improve user experience on the Internet, home networks should have *automatic* solutions that do not require user intervention. Such solutions should automatically diagnose the home network and optimize its performance in order to avoid poor user experience. But, to this end, it is important to first measure and characterize the network dynamics.

Many techniques exist to measure Internet topology and performance but little is known about home networks. Even if home networks receive more and more interest across the years, still little knowledge is available about home network characteristics[9;15]. For instance, what are typical devices at home? What do users like to do within the home network? What are the most popular services accessed from home? Previous work used active probing from end-hosts or servers in the Internet[30;55;59;91] to measure and characterize the access link. The lack of home network studies is due to different challenges such as the one related to data collection or the incomplete view of home network traffic using current probing methods.

Partial traffic view from end-hosts. Users involved in a home network traffic study are usually asked to install a tool on their end devices or plug and configure a third party device to their home network in order to collect traffic. The problem with end-hosts is that they are often personal machines and moved out of the home network for some time. Such behavior cuts the measurement and alter the data. Moreover, end-hosts have only a partial view of the traffic, which limits the studies with the collected dataset. To have a full knowledge of the traffic, one solution is to install the tool on all the

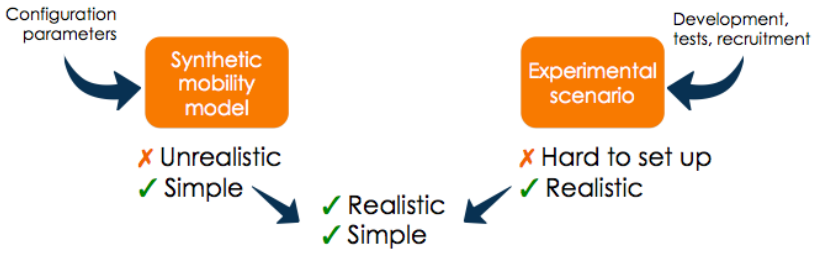


Figure 1.2: Mobility trace breeding benefits.

end-hosts within the same home network; thus, cooperation of all the household members is required. Unfortunately, this solution is weak and unrealistic because a single disconnected node can distort all the dataset.

In this thesis, we measure traffic inside the home network from the *home gateway* in order to have the whole picture of the dynamics.

1.1.2. From single to multiple mobility experiments

Reliable communications in intermittently-connected mobile networks (ICMN), also known as disruption-tolerant networks (DTN), require a deep understanding of the network dynamics. We are surrounded by communicating devices: phones, access points, vehicles, and so on. Because mobile nodes are subject to unpredictable interruptions, protocols that have been designed for wired networks such as TCP/IP which requires end-to-end connectivity does not stand anymore. That is why understanding the dynamic of mobile networks is key for new protocols design or testing.

Building and running reliable experiments to measure contact opportunities in ICMN is a long and challenging process. The final goal from such experiments is to get a contact trace which we can analyze to help understand the network dynamics. As only few real contact traces are publicly available, it would be interesting to think at *reusing* the existing traces to get multiple others. Figure A.2 resumes the benefit of the method we propose, which combines advantages of both synthetic data and real experiments. We focus on the ability of reproducing contact traces while avoiding the difficulties of setting up a multiple real experiments.

1.2. Contributions

As a summary, this thesis makes the following contributions.

Contributions in the area of home networking:

1. We present a comparison of local traffic in different network environments using end-hosts. We run our study with traces from 47 users who ran the collection tool (Hostview) for at least one week; 32 of them ran it for more than a month. End-hosts were connected from 185 unique networks spread over 18 different countries where 34 networks were residential and 38 were work environments (universities and enterprises). Our results show a large diversity in local

traffic but still wide-area traffic dominates. In networks like airports and coffee shops, the local traffic is rare (except for DNS traffic).

2. We analyze home and work traffics separately and compare their dynamics in terms of applications. In both networks, we do observe a non-negligible local traffic where most connections are short. But sometimes local connections transfer a large number of bytes. Besides DNS, we find that typical local applications are network file system and backup. Still, the composition of the local traffic depends on the user and the network.

3. We design a performance optimization system that operates from the home gateway to avoid performance degradation of all active applications in a home network. We propose a two step strategy: (i) track the performance of applications and (ii) identify applications in real time.

4. With our modified version of the home gateway, we show that even with an additional overhead generation, the results are promising on the possibility of using a traffic collection process along with an application identification strategy. Also, we propose light application identification techniques. Following our guidelines, the results show that it is possible to run an application performance tracking technique from the home gateway.

Contributions in the area of mobile networking:

1. We propose a *breeding* system to derive possible contact traces from a single real experiment. Our synthetic evaluations indicate that the original and bred traces follow the same characteristics. Also, using a variety of real datasets, we show that our system produces accurate contact traces strongly inspired from the original traces.

2. We use a particular real dataset to show the valuable network observations that our breeding method provides (which would only have been possible with a new experimental campaign). We also explain the guidelines to extract such information.

1.3. Overview of the document

This dissertation is organized in two parts as follows. The first part includes: Chapter 2 defines the problem of application performance degradation and presents related research in the area. Chapter 3 shows the application dynamics in home networks. Chapter 4 investigates the feasibility of tracking application network performance in home gateways. Then in the second part: Chapter 5 introduces the problem of contact traces reproducibility and talk about prior work. Chapter 6 studies contact traces breeding. We conclude in Chapter 7.

Part I

Domestic networks performance optimization

Chapter 2

Toward a better user experience in domestic networks

INTERNET access and home networks have been receiving a lot of attention lately from the research community, regulatory agencies, and ISPs. Home gateways provide Internet connectivity serving several purposes such as telephony, media-streaming, data, or gaming. In this part of the thesis, we analyze home network traffic and show how to track performance in such networks.

Regulatory agencies become interested in comparing the access link speed offered by ISPs with what they deliver. ISPs face the ever increasing bandwidth and nowadays also delay demands of users. New applications and devices contribute to the several requirements and challenges that home networks pose. This trend makes troubleshooting and monitoring home networks fundamental to understand their problems and challenges. Since monitoring the home from an end-device is restricted in terms of what can be monitored, projects such as SamKnows (UK and US) rely on active measurements from the home gateway. Even if measuring at the gateway is valuable, it is also resource consuming and can interfere with users' resource needs. For this reason, we explore the feasibility of *passive* measurements at home gateways.

Previous studies highlight the difficulties that users face when setting up, maintaining, and troubleshooting a home network[29]. Sometimes, users cannot even correctly articulate what the problem is[50]. After running two different sets of tests (interviews, surveys) in both UK and US, the experiment shows that a great potential exists for developing applications that help householders. Furthermore, it appears that home maintenance is challenging even for advanced users. Interviews also show that users expect that, whenever new technologies or features are brought to the home, they should be included into the existing infrastructure.

Our approach brings an *automated* help using an already existing equipment in any household (home gateway). As home networks are more complex nowadays, housekeeping is a new users' concern, especially for digital media management. A study has been made on digital housekeeping[28]; it points out the importance of managing media services, disk space, and also maintaining order when

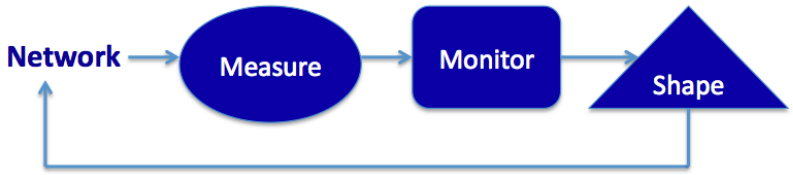


Figure 2.1: Network traffic optimization basics.

scaling home networks. Domestic routines need to be respected as well, considering the fact that they change from one home to another and overtime within any home.

We can find various tools for bandwidth measurement over the Internet such as BWMeter, Netmeter for single computers, or SpeedMeter Pro for multi-computer visualization. Unfortunately, none of them offers a central display interface for managing the whole network. For this purpose, other tools were developed to help home users manage and troubleshoot their network[12;14]. The tools come in a separate public appliance that allows anyone to manage any machine in the network.

In order to guarantee users a good experience within their home network, it is crucial to optimize resource consumption among all devices in the network. We consider three basic steps for optimizing network utilization (see Fig. 2.1). In the following, we talk about research efforts in the area of measuring and shaping home networks. Then, we focus on the monitoring step to explain in detail where our work comes at play.

2.1. Home traffic measurements

Wide-area traffic measured from inside the network has been analyzed from different angles over the past decades[8;56;59;97]. These measurements, however, cannot capture local traffic at the edge. In this thesis, we analyze local traffic and how it compares with wide-area traffic using data collected directly at end-hosts using HostView[42]. Other studies have collected and analyzed similar end-host data in the past[26;84]. In particular, Giroire et al. have compared network traffic from end-hosts across three network environments (inside the company, VPN to company, and outside the company)[26]. Different from ours, their study has not characterized local network traffic in depth and although it measured laptops of a larger number of users than HostView measured, they are all employees of a single enterprise.

More similar to our work are the studies of one enterprise network[68;72] and of three home networks[48]. These prior studies instrument the local network to collect packet traces and can hence differentiate between local and wide-area traffic. Existing experiment results focus mainly on network performance, not on traffic characterization[48]. Their few traffic characterization results show that wide-area traffic dominates local traffic in the three homes, but there are some, rare spikes of local traffic. In this thesis, we study measures from at least one end-host (or at most a couple) in each network and hence cannot have such a complete view of each of the studied networks, but it

can sample a larger number of networks. Later, we contrast the analysis in the enterprise study[72] with our analysis on traffic in enterprise environments. Given that Internet traffic can vary significantly among sites and over time[74], our study contributes to show the diversity of traffic patterns in different network environments.

Projects like Netalyzr[55], HostView[42], and RIPE Atlas[82] aim at understanding network performance at home. Yet, these end-host based tools suffer from the unobservability of activities of other devices existing inside the home that can bias the results[17]. To overcome such issues, Calvert et al.[9] propose to measure from the home gateway. They report through preliminary tests of capturing typical home traffic that occasionally, their methodology leads to 10% loss rate under heavy load. SamKnows deploys home gateways in order to repeatedly measure the access link performance[86]. To the best of our knowledge, there has been no work that systematically explored the feasibility of passive monitoring on different home gateways.

Calvert et al. underline the need for a Home Network Data Recorder (HNDR) to allow a more detailed understanding and troubleshooting of home networks[9]. Their work is the first to propose passive measurements at a home gateway. They base their concept on the NOXbox. In the next chapter, we test the performance of such a box. For a heavy load-case (two P2P downloads, one Hulu streaming, and two Youtube downloads) of unspecified throughput, they report `tcpdump` drops up to 10 % of the packets while recording to disk. Unfortunately, they neither report on system utilization nor systematically vary the workload. In the next chapter, we play with different workloads and report system utilization.

2.2. Network traffic shaping

Traffic shaping includes different tools like *Trickle*[22] that rate limits the TCP connections of a process or group of processes. Another tool is *WonderShaper*, a traffic shaping software that provides low latency for interactive traffic, allows web surfing at reasonable speeds while uploading/downloading, and ensures that uploads/downloads do not hurt each other[102].

When we consider Internet traffic, we can separate it in two different traffic areas. Most research studies focus on traffic at border routers or companies for the measurement simplicity. We call it *wide-area* traffic. But there is only few interest in traffic that remains inside the network -*local traffic*- because of measurement challenges. We detail the efforts in this area in the following.

2.3. Network monitoring

A recurring problem concerns bandwidth management. For this purpose, tools were developed to help home users manage and troubleshoot their networks. We classify them according to their location in the network.

End-device based. The Home Watcher, for instance, is a domestic tool for bandwidth management. It shows home user bandwidth consumption along with an option for limiting bandwidth usage per person (up to 20 % of the total capacity to avoid severe limitations)[12]. After its trial in 6 homes (24 people) for 8 weeks, users answered questions about the experience of everybody in the home. The tool was not only appreciated for its ability to control bandwidth utilization but also for allowing parents to know their children's activities at a given time. For example, at homework or bed time, if bandwidth usage is not low parents know that the kids are playing with their computers instead. Yet, it brought the issue of who decides at home and how severe are the limitations to apply. Results differ depending on the social make up in a household. Therefore, an automatic limitation system is preferable.

In contrast with the work cited above, more automatic solutions are proposed. HomeMaestro is a host-based system that monitors the performance of local and global applications and automatically detects contention for network resources[46]. It uses per flow and per process statistics (throughput, RTT, loss rate) to detect competing flows for the same resource. In some cases, the system is able to detect up to 85 % of the problems, where the vast majority is caused by applications competing across hosts. This technique suffers from the way the traffic is captured; it uses 2 wireless monitors to capture wireless traffic which creates interference and thus leads to missing traffic. For this reason, we believe that the home gateway is a better candidate to perform these types of tests.

Router based. Another tool inspired from the Home Watcher is Kermit. It has the same goal but collects data from a flashed router with DD-WRT[14]. It comes with an interface that includes *who's online* and *who's hogging the bandwidth* and a picture of all connected devices attached to a cloud (Internet). In particular, users like the fact that they can personalize it with pictures and real names to identify each other. It provides an additional option that can prioritize users within the home network. This study comes with interesting insights into how to design a management tool for the home. As such tools need to gather a large amount of data, a smart solution is required to filter and store the valuable data or decide whenever it is no longer useful. Besides, users always have privacy concerns that ask for more unobtrusive mechanisms.

In the same perspective of helping users with their Internet experience at home, another project called Homenet was conducted with 93 Pittsburgh families[50]. This work highlights the lack of technical support for homes in opposite to workplace, where people can easily find someone with the appropriate skills to ask for help. Relying on telephone logs, mail reports from users (237 members) to dedicated staff (persons to call in case of problems) and automated probes to calculate time where users are active, they noticed that 70 % of householders ask for set-up help and 95 % for technical support. This shows that help at home is under-estimated even for houses with skilled consumers. Before asking technical support staff for help, consumers try to find someone in the home who can help, this person is usually a teenager. This child will play a new role of technical-advisor that affects her behavior (authority, independence). Even if other solutions exists like Austin project, which aim

at forming teenagers in taking advantage of technology to improve involvement and helping others, automatic systems without human intervention are still needed.

Existing automatic solutions use software-defined networking (SDN) approaches. An example involves a prototype router on top of NOX and Open Vswitch that brings per-flow traffic control[64]. This method introduces a set of switching and protocol modifications (address allocation, medium access control, Internet access control, flow-level traffic control) and its heterogeneity is corroborated with a large range of IP-enabled devices. The perspective is to translate users hierarchy at home into traffic components.

Another solution toward self-tuning home networks exists. The case study in this work is a poor wireless performance experienced in a home network because of interference (stations operating on the same frequency), which adds delays in the network when backing-off or increasing loss rate when facing collisions. The policy for this case is changing the channel which seems simple but has to be dynamic and based on distributed information from multiple nodes across the network. In order to automate that, a proposal is to use State Machine based Policies (SMP), which consists in various components that execute *actions* under *conditions*[77].

The closest project to our topic is the BISMark project[96]. The idea is to deploy gateways that allow running active and passive measurements remotely to investigate home networks. In our work, we use *BISMark-passive*. We choose this software that passively monitors network traffic because it sends differential updates only periodically to a central server. We detail the trace collection process later in this thesis.

2.4. Open issues and contributions

As detailed above few is known about network dynamics and how to automatically improve user experience in households. In the first part of this thesis, we answer two main questions. First, what is the dynamics of domestic networks and how does it compare to other networks? We use traffic from several home networks and analyze their diversity and how they compare to enterprise traffic in Chapter 3 . Second, is it feasible to track performance from the home gateway ? In Chapter 4, we show a methodology to perform such a tracking from the home gateway without exhausting resources.

Chapter 3

Application Dynamics in Home Networks

THIS chapter compares local and wide-area traffics from end-hosts connected to different home and enterprise networks. We base our analysis on network and application traces collected from 47 end-hosts for at least one week. We compare traffic patterns in terms of number of connections, bytes, duration, and applications.

3.1. Problem definition

The past couple of decades has seen many studies that characterize Internet traffic [8;56;59;97]. These studies are based on packet traces collected in ISP networks, at border routers of university campuses or enterprise networks. As such, most prior studies focus on wide-area traffic. Little is known about the traffic that stays inside a network, which we call *local traffic*. The main exception is the study of traffic from one enterprise [68;72], which shows that local traffic is different from wide-area traffic with a significant amount of name service, network file system, and backup traffic. As the authors point out their study is "an example of what modern enterprise traffic looks like" [72]. It is crucial to reappraise such analysis in other enterprises and more important in other types of edge networks. For instance, the spread of broadband Internet has caused an increase in the number of households that have a home network. Yet, there has only been limited analysis of local traffic volumes in three home networks [48], but no in-depth characterization of in-home traffic patterns. The challenge of studying local traffic across multiple edge networks is to obtain measurements from *inside* multiple networks.

This chapter characterizes local network traffic of multiple networks from the perspective of an end-host that connects inside an edge network. This approach is in contrast with previous work [48;72], which instruments routers in the local network. Although instrumenting routers could capture all traffic traversing the local network, it is hard to have access to routers in more than a few networks. By monitoring traffic directly at end-hosts, we can sample a larger number of networks, but we can only see the traffic from the host we used for measuring the network. For smaller networks (such as

home networks) a single host's traffic captures a significant fraction of total traffic, whereas for larger networks (as enterprises) this fraction is less significant.

We rely on data collected at end-hosts using the HostView monitoring tool[42]. HostView records packet header traces and information about applications and user environment. The data we study was collected from 47 users who ran HostView for more than a week each. Given that users move between different networks, this dataset contains end-host traffic from a total of 185 different networks spread over 18 different countries. Section 3.2 gives an overview of the HostView data. The analysis of local and wide-area traffics from HostView is challenging though, because HostView has no information of which traffic flows are local. Worse, HostView scrapes the end-host IP address from the traces to protect user's privacy, which makes the identification of local traffic even more challenging. Therefore, we propose a heuristic to separate local from wide-area traffic. Section 4.6 describes this heuristic together with our method to categorize environments and applications in the HostView data.

Our analysis (presented in Section 3.4) raises some high-level questions: How does the volume of an end-host's local traffic compare to wide-area traffic? Do local and wide-area applications differ? How does traffic vary between home and work? The results show that for most users wide-area traffic dominates local traffic, but that some users have over 80 % of local traffic. Local connections are mostly shorter and smaller than wide-area connections, but sometimes they transfer a larger amount of traffic than large wide-area connections. We find that typical local applications are DNS, ssh, and network file systems (confirming previous findings[72]). Moreover, common applications at work include backup, printing, and web. Yet, these applications are rarely used at home.

3.2. Summary of HostView Data

We use three of the datasets collected by the HostView tool[42]: network packet traces, application labels, and the end-host's network environment. HostView logs all the data directly at the end-host into a trace file, which is periodically uploaded to a server. A new trace is created every four hours or when a change in the network interface or the IP address is detected.

Network traces and application context HostView logs the first 100 bytes of each packet sent and received by the end-host with *libpcap*. For DNS packets, it records the whole packet to enable offline hostname to IP address mappings. In this work, we use connection summaries generated by HostView[43]. Each connection summary record describes both directions of a TCP or UDP connections and includes (among other fields): *(i)* source and destination IP addresses (replacing the host IP address with "0.0.0.0" to comply with French privacy laws), source and destination port numbers, and network protocol; *(ii)* number of bytes, number of packets, and duration of the connection; and *(iii)* the name of the process executable that generated the connection.

Network environment HostView labels each trace file with information describing the network environment the end-host is connected to, including the network interface, a hash of the wireless network SSID and of the BSSID of the access point for wireless networks or a hash of the MAC

address of the gateway for wired networks. It also records the ISP, the city, and the country for each trace using the MaxMind GeoIP commercial database from March 2011. When the end-host connects to a new wireless network, HostView asks the user to specify the network type from a pre-defined list: Home, Work, Airport, Hotel, Conference meeting, Friend's home, Public place, Coffee shop or Other (with the possibility to specify). This user tag is used to classify the network the user connects to according to an environment type. Unfortunately, this tag is not available for wired connections and users sometimes skip the questionnaire. Originally, only 40 % of HostView traces had a user tag, but after applying some heuristics (which exploit the fact that users connect to the same network with both wireless and wired, for instance) previous work was able to label 78 % of the traces[43]. Still, the data includes at least one unlabeled trace per user. The next section describes our method to label most of the remaining traces with an environment type.

Dataset characteristics and biases HostView was announced in networking conferences and researcher mailing lists. Volunteer users downloaded HostView (which is available only for Mac OS and Linux) and ran it during different time intervals between November 2010 and August 2011. In this study, we use traces from 47 users who ran HostView for at least one week; 32 of these users ran HostView for more than a month.

Because of the way HostView was advertised and its limited operating-system support, the user population is biased towards networking researchers. We acknowledge that networking researchers probably use different applications than the average user and may also work from home. It is still interesting to study examples of the differences between local and wide-area traffic. We do observe a diverse set of applications among different users and our users do use some popular applications like YouTube, Facebook and BitTorrent. Furthermore, this bias influences the types of networks we study. Importantly, "work" is often a university. Overall, we study end-hosts connected to 185 unique networks spread over 18 different countries (Italy: 25, France: 22, Germany: 21, Rest of Europe: 31, Asia: 19, US: 63, Australia: 3, and Brazil: 1); 34 distinct home networks and 38 distinct work environments (29 are universities and 9 enterprises).

Another bias comes from using data collected for a limited time period on only one single end-host in the network. It is well known that traffic characteristics can vary considerably between different networks and over time[74]. HostView can only see a small fraction of the network's traffic and there are some types of traffic that it can never observe. For example, some homes may have a media server that serves content to the TV; this type of traffic traverses the home network, but it is not originated or consumed by an end-host. Despite these shortcomings, we believe that this end-host perspective on local versus wide-area traffic offers the unique opportunity to sample traffic in a relative large number of networks. Whenever appropriate, we also contrast our findings with previous work.

3.3. Methodology

In this study, we compare local and wide-area traffic in networks of different types. In addition, we are interested in the traffic application mix. We follow three steps to label HostView traces before

our analysis: *(i)* Differentiation of local and wide-area traffic, *(ii)* Extension of the incomplete network type labeling, and *(iii)* Categorization of connection records into application groups.

Local vs. wide-area HostView does not collect the host IP address, so we cannot identify the local subnet based on the host IP prefix. We develop a number of heuristics to classify traffic as local or wide-area. We define *local* traffic as all the traffic exchanged between an end-user machine and a private IP address, i. e., `192.168/16`, `172.16/12`, `10/8`. We expect this classification to correctly match most local traffic at homes, as those typically connect through a NAT gateway sharing one public IP on the outside. To avoid misclassification when the ISP employs carrier-grade NAT, we develop a second heuristic that analyzes the remote IP addresses of all traffic flows classified as local. When we observe that the remote IP addresses fall in more than five different subnets, we compute the number of connections and bytes for each remote /24 to identify whether there is a "preferred subnet", i.e., a remote subnet that carries most of the traffic (>99.9 %). If there is a preferred subnet, then we leave all traffic classified as local. Otherwise, we flag the network for manual inspection. The HostView data had a total of five home networks which contacted more than five different remote subnets, four of these had a preferred subnet. We manually inspected the remaining home network and found that a large fraction of P2P traffic going to IPs in `10.*` networks. In fact, this user's home ISP is known deploy carrier-grade NAT, so we label this `10.*` traffic as wide-area and we leave the `192.*` traffic as local. For work networks, we might misclassify local traffic as wide-area when hosts connected to the local network have public IP addresses. We address this issue with a third heuristic that labels all traffic to a destination IP address that has the exact same organization name as that of the source network as local. Finally, we classify all broadcast traffic as local. We label all the remaining traffic as *wide-area*.

Extension of network environment labels As discussed in Section 3.2, some of the HostView traces have no network type tag (e. g., Home or Work). We manually inspect the ISP, the network interface, and the geo-location of each unlabeled trace and assign a label. For example, we label a trace annotated with *ISP: "University of California"; City: "Santa Cruz, California"; Country: "United States"* as *Work*. Another example containing *ISP: "Free"; City: "Paris"; Country: "France"* is labeled *Home*. This manual classification reduced the fraction of unlabeled traces to 2 %. Some traces have no information that indicates the type of network.

Application Categorization For our analysis of popular applications we rely on a two-staged categorization process. First, we assign one of eleven application categories or "unclassified" to each connection based on the process executable name. Second, we label any connection that remains unclassified based on the application protocol as derived from the port number using the IANA mapping. We assign categories to those process names and application protocols that account for the most connections and the most volume. Table 3.1 lists the eleven categories and gives example process names and application protocols for each of them.

Table 3.1: Examples of process names and network services to category mappings. This list is not complete and only intended to give an idea.

Category	Process name (Examples)	Application protocols
Backup	retroclient	amanda
Chat	Skype, iChat, Adium, Pidgin	ircd, SIP, msnp, snpp, xmpp
DistantControl	ssh, sshd, VNC, screen sharing	ssh(22), webmin
Email	Mail, Outlook, Thunderbird	IMAP(S), POP3(S), (S)SMTP
Personal	Media players, games, productivity	rtsp
FileTransfer	ftp, dropbox, svn, git, SW updates	ftp, rsync, svn, cvspserver
Management	traceroute, iperf, nmap, ntpd, uPNP	BOOTP, MySQL, VPN, SNMP, whois
Miscellaneous	perl, python, VirtualBox, openvpn	—
NameService	dns, nmblookup, named, nmbd, nscd	domain(53), mdns, netbios-ns
NetworkFS	smbclient, smbd, AppleFileServer	AFP, AFS, LDAP, netbios, nfs
P2P	amule, uTorrent, transmission	amule, Kazaa, BitTorrent
Printing	cupsd, lpd, HP, Lexmark	ipp, printer
Web	Firefox, Chrome, Safari, Opera, httpd, plugin-container, WebKitPluginHost	HTTP(S)

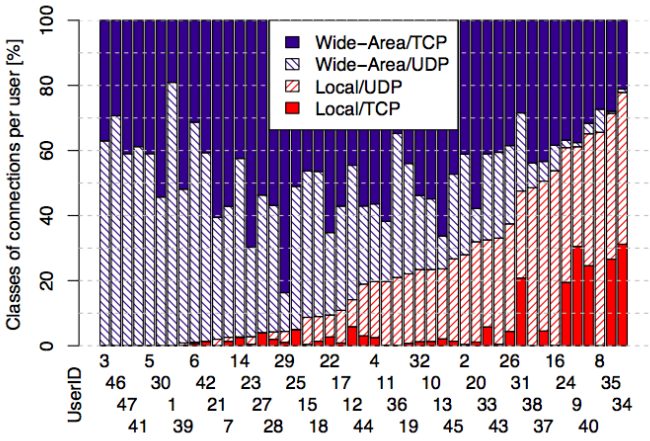
3.4. Results

This section first compares local and wide-area traffic in general. Then, it studies the split of local and wide-area traffic at home and at work.

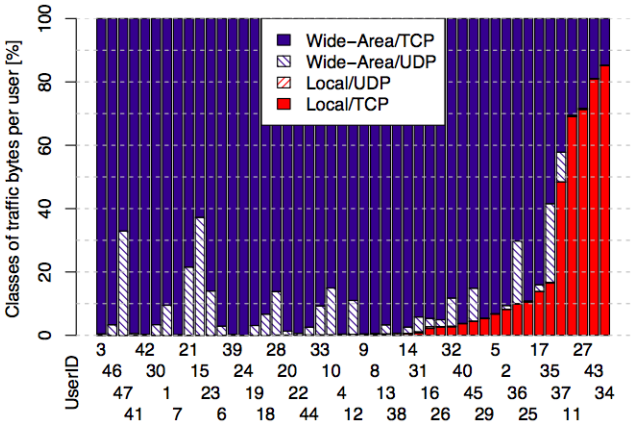
Local vs. Internet: Connection and Bytes Figures 3.1(a) and 3.1(b) show the fraction of local (two bottom bars) and wide-area (two top bars) traffic for each user (UserIDs are the same across figures for comparison). For each user, we separate UDP (shaded bars) from TCP (solid bars) traffic. We consider the composition of traffic by number of connections (Figure 3.1(a)) and bytes (Figure 3.1(b)).

Take the example of the rightmost user in Figure 3.1(a), UserID *34*, 77 % (46 % UDP and 31 % TCP) of this user's connections are local. The remaining traffic is directed to the Internet (0 % UDP and 23 % TCP). In general, we observe that Internet traffic dominates both in number of connections and bytes, although this dominance is much more pronounced for bytes. In total, we classify 780 GB as local and 3 TB as wide-area traffic. Furthermore, we see that UDP dominates local connections for almost 80 % of the users. The absence of shaded bars in Figure 3.1(b) clearly shows that almost all bytes are transferred in TCP connections (>89 %).

We observe that the four rightmost users in Figure 3.1(b) transfer more bytes locally than in the wide-area. As we discuss in the next section, most of this traffic corresponds to network file system, so these users could be playing music or watching videos from a local network storage. In Figure 3.1(b), more than half of the users exchange almost all traffic with hosts in the wide-area (corroborating previous findings[48]). In the rare cases these users do exchange traffic with hosts in the local network, they mainly perform file transfers.



(a) Local vs. wide-area connections per user (Total number of connections per user varies between 2.5 K and 3 M.).



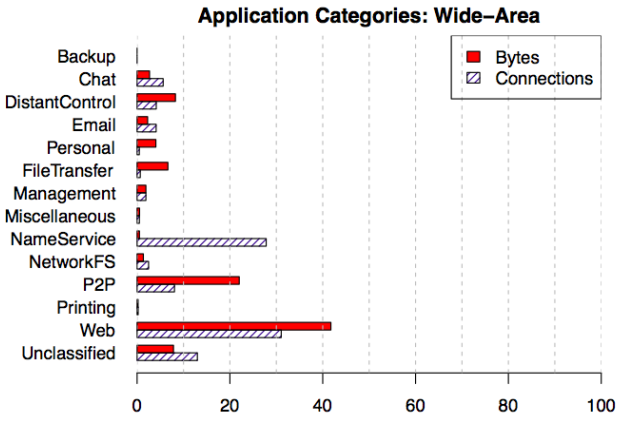
(b) Bytes transferred on local vs. wide-area connections per user (Total amount of traffic per user varies between 800 MB and 770 GB).

Figure 3.1: Local vs. wide-area traffic.

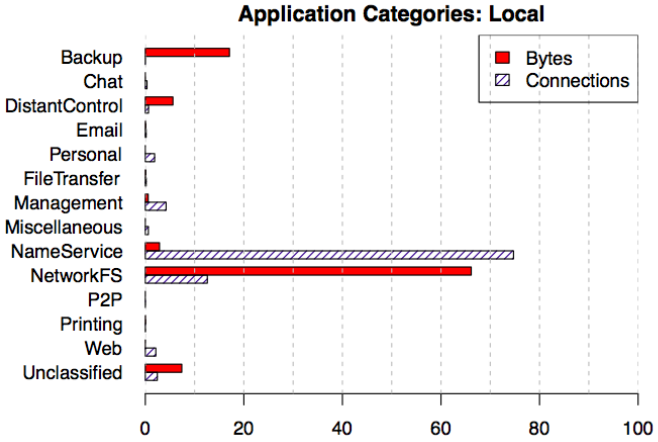
Local vs. Internet: Application Mix We now study how local and wide-area applications differ. Figure 3.2 shows the application mix in terms of connections (shaded bars) and data bytes (solid bars). These figures use the application categorization method described in Section 4.6, which leaves no more than 12 % of connections and 7 % of bytes *unclassified*.

Figure 3.2(a) shows the application mix for wide-area traffic. We see that the proportion of bytes per application class agrees with results from previous studies[56?]. Web traffic and P2P are the top applications. In addition, we see some file transfers and distant control traffic (ssh and VNC). When we classify in terms of number of connections, the mix changes and name services take the second place behind Web. Chat and Email are also more prevalent in terms of connections than bytes.

Figure 3.2(b) shows that name services (e. g., DNS) dominates local traffic in terms of connections, whereas backup and network file systems (e. g., AFP and SMB) in terms of bytes. A previous



(a) Application mix for wide-area traffic.



(b) Application mix for local traffic.

Figure 3.2: Local vs. wide-area application mix.

study of enterprise traffic[72] also found that network file system and name service dominate local traffic, but their study found considerably more local email and web traffic than what we find. A significant part of our data is of home traffic, which may explain this difference. We now split the traffic into home and work.

Traffic at Home and Work Our analysis so far has mixed traffic from multiple network environments, including home, work, airports, coffee shops, or hotels. Based on our extended environment labels (see Section 4.6), we investigate the differences not only between local and wide-area traffic, but also across different types of network environments. Figure 3.3 shows the distribution of traffic and users over the different environments. Note that a single user can visit multiple environments. After applying our heuristics the 'Other' category, which includes instances when users labeled the environment as other and when our heuristic could not label the environment, only accounts for 12 % of the bytes and 18 % of the connections. We see that users (light shaded bars) are primarily at home

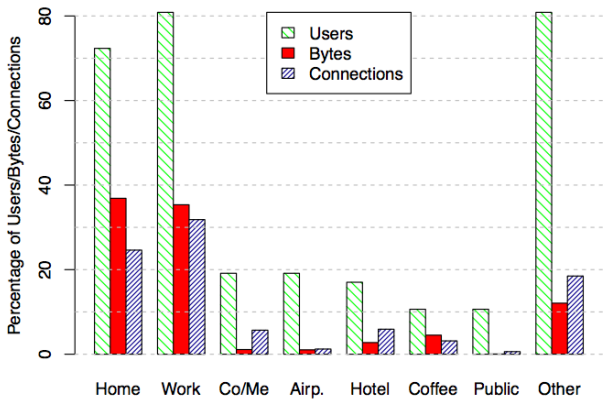


Figure 3.3: Percentage of Users, volume, and connections by environment.

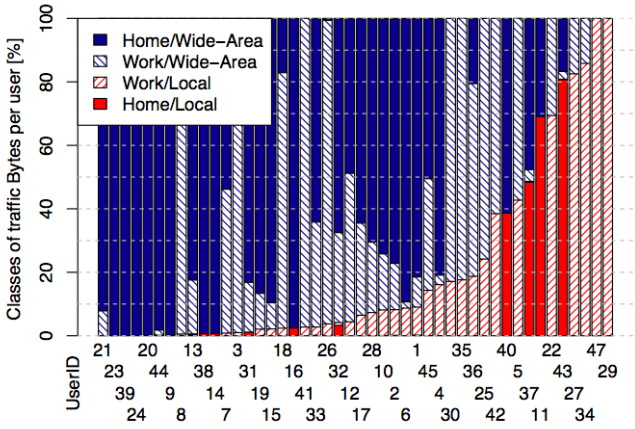


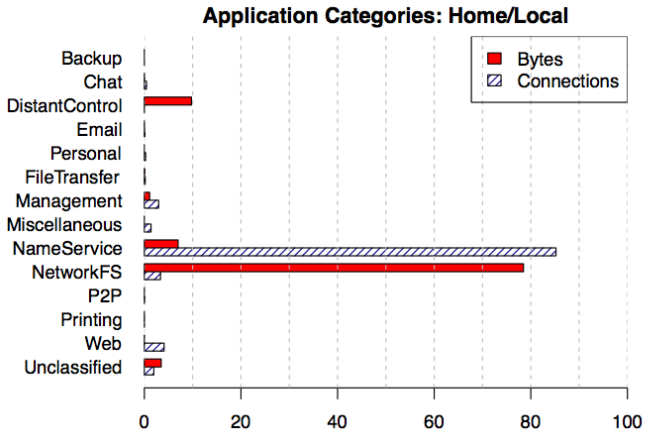
Figure 3.4: Bytes transferred at home vs. work and traffic target per user.

or work, thus we select these two environments for further study. These environments include 56 % of the connections (heavy shaded bars) and 72 % of the bytes (solid bars). Moreover, our analysis of local traffic in different environments (not shown) shows that the fraction of local traffic in all environments but home and work is marginal (<1.25 %).

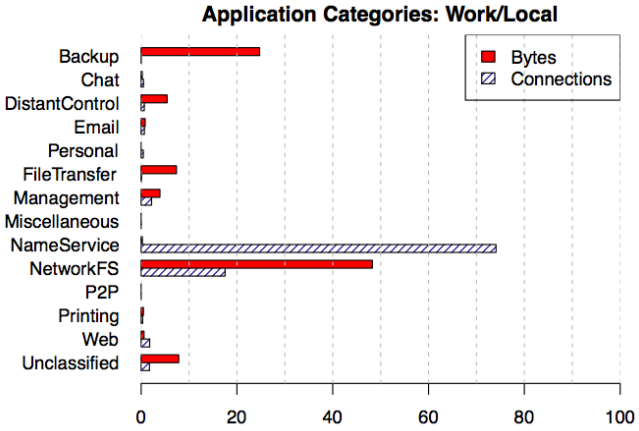
Figure 3.4 shows the number of bytes sent and received per user for all four combinations: home/wide-area, work/wide-area, work/local, and home/local. As expected, we see a similar split between local (bottom) and wide-area (top) traffic The differences between Figure 3.4 and Figure 3.1(b) happen because here we only include traffic from home and work. The majority of users has more local traffic at work. Only four users have a significant fraction of local traffic at home.

Application Mix at Home and Work Now that we established a basic understanding of how traffic differs between home and work as well as local and wide-area, we investigate the application mix in each of these cases. The analysis of wide-area traffic at work (omitted for conciseness) shows almost no P2P traffic, but a considerable fraction of file transfers and distant control traffic. These results are consistent with previous findings by Pang et al.[72].

We study the application mix of local traffic at home in Figure 3.5(a) and at work in Figure 3.5(b). Local traffic at work includes file transfers and backup traffic, which are not present in home traffic.



(a) Application mix for Home Local traffic.

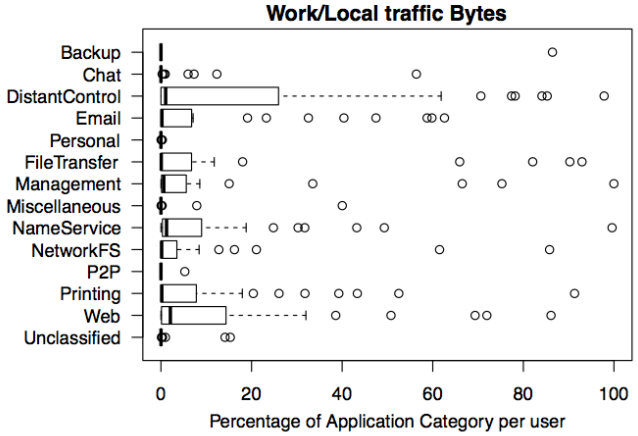


(b) Application mix for Work Local traffic.

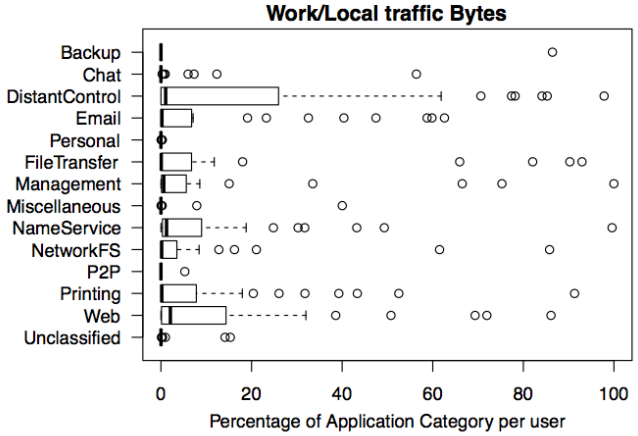
Figure 3.5: Local Home/Work application mix.

Different from Pang et al.[72], we see little local email or web traffic at work. Indeed, it turns out that email traffic of most HostView users is wide-area. A possible explanation is that they are typically mobile and hence rely less on local infrastructure.

Another difference is the lack of backup traffic at home, which may reflect users' preference to backup directly at external disks when at home, instead of over the network. The backup traffic at work is mainly from a single user, who is responsible for almost all the bytes of backup traffic in Figure 3.5(b). We do also observe some file transfer traffic locally at work. Most of that is transmit (file transfer client for Mac OS) and FTP, but some is Dropbox (a cloud storage/synchronization service). Given it is a cloud service (cloud = wide-area) we did not expect to find Dropbox locally. It turns out that Dropbox is using a direct connection for synchronization across devices in the same LAN. Dropbox constitutes half of the file transfers in our local home traces.



(a) Boxplot of application mix per user for Home/Local traffic.

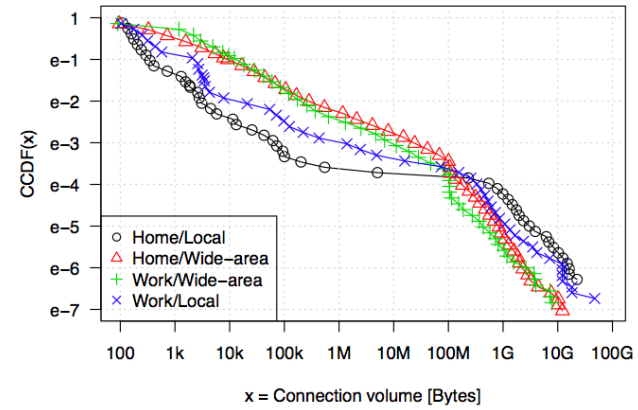


(b) Boxplot of application mix per user for Work/Local traffic.

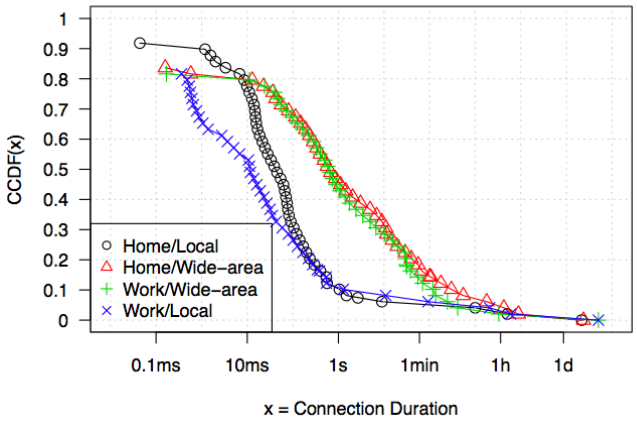
Figure 3.6: Local Home/Work application mix per user.

As single users can have a distorting impact on the overall traffic composition, we now calculate the application mix per user. Figure 3.6 shows boxplots[1] of the application mix per user in terms of bytes. Each row shows the distribution of the individual contribution of the corresponding application category across all users. We find that although network file system traffic dominates local traffic, most users have less than 10 % of traffic in this category both at home and at work. Reversely, although name service represents a small percent of the total number of bytes in Figure 3.6(a), the median across all users is over 50 %. We find similar effects for file transfers at home. At work, contrary to Figure 3.6(b), we do see web, email, and printing usage.

[1]The box (line inside the box) shows the quartiles (median); whiskers show nearest values not beyond a standard span from the quartiles; points beyond (outliers) are drawn individually.



(a) CCDF of connection volumes.



(b) CCDF of conn. durations (log-linear)

Figure 3.7: CCDF of connections and conn. durations.

Connection size and duration We end our analysis with a study of the characteristics of local and wide-area connections both at home and work. We show the complimentary cumulative distribution of the number of bytes per connection in Figure 3.7(a) and connection durations in Figure 3.7(b). For example, the 'work/local' point at x = 10kB in Figure 3.7(a) indicates that only 1 % (y-axis) of all the connections are larger that 10kB.

In terms of bytes, we observe in general larger (further to the right) connections for wide-area traffic. Local connections are typically small, but the largest local connections exceed the size and duration of wide-area connections. This observation confirms one previous study showing that home traffic sometimes have short spikes[48]. Although the connection durations in Figure **??** are limited by the 4 hour trace file cutoff, most connections are shorter than this limit. We also see the local connections (circles and crosses) are up to two orders of magnitude shorter than wide-area connections.

3.5. Summary

This chapter presented a comparison of local traffic in different network environments from the perspective of end-hosts. The advantage of using end-hosts as vantage points is that we study traffic collected from over one hundred different edge networks. Our results showed that there is a large diversity in importance of local traffic relative to wide-area traffic, but that in general wide-area traffic dominates. In some networks (like airports and coffee-shops), we rarely see any local traffic, the only local traffic is DNS. At home and work, we do observe a non-negligible fraction of local traffic. Most local traffic is composed by short connections, but sometimes local connections transfer an extremely large number of bytes. Besides DNS, the most typical local applications are network file system and backup, but the composition of local traffic depends on the user and the network. The drawback of measuring local traffic from end-hosts is that we can only see a small fraction of each network's traffic.

Chapter 4

Tracking Application Network Performance in Home Gateways

HOME gateways offer Internet connectivity for all devices in the home, allowing services such as telephony or gaming. However, typical home gateways do not include any mechanism to guarantee optimal performance when applications are competing for the same resources. In this chapter, we outline an application performance optimization approach for home networks. In particular we study the feasibility of application performance tracking on home gateways, which involves both identification of active applications and monitoring their performance.

4.1. Problem definition

With the spread of broadband Internet access[37], more and more people have Internet at home. Various services allow users in a household to perform professional and personal tasks. But running different network services simultaneously can lead to performance degradation. Home users face many performance problems due to various reasons[28;79]. For instance, a kid downloading a big file can disturb the quality of his parents conference call over Skype. Although, the download only marginally suffers from the minimal bandwidth requirements of the Voice over IP call, the latter requires low latency, which is negatively affected by the download if both share a single queue. The problem is twofold: First, most home users only have limited technical skills and thus have no understanding of performance degradation reasons. Second, even if these skills are present contemporary home network devices offer almost no options to prioritize traffic and identify and resolve resource conflicts.

As for the previous example, when network performance degrades users can only wonder why their conference call quality is bad? Are there competing applications? Is it the router? Or the ISP? Maybe the VoIP server is overloaded ... In this chapter, we want to propose a new approach to help users in such situations by tracking home network application performance. Our solution leverages the home gateway as the tracking point. Not only is all home network traffic passing through the

home gateway. It is also the ideal point to tell apart the traffic from different user devices and network services as well as to distinguish between internal and external problems (home/ISP).

Home performance tracking consists of two main parts, identifying active applications and monitoring their performance. The first step is application identification, where we analyze traffic to identify the set of active applications. The second, performance monitoring, is about extracting performance metrics from the flows belonging to each individual application. Applications have different requirements on the network. If we consider the previous example, using the performance tracking metrics we can evaluate the actual bandwidth used for download and the impact on the latency required for the Skype conference call. Thus, we can optimize these two parameters to ensure that the quality of the Skype call will not degrade by enforcing a maximum delay for its packets. Based on the application type and the performance metrics an optimal network resource utilization can be determined and enforced. The resource requirements of applications can either be manually configured or learned from past measurements (e.g. when an application was the sole active application at a time).

Solutions for application performance optimization exists but only based on end-hosts[13;47]. Performance optimization in home networks however requires a complete view of all home network traffic. This can either be achieved by placing the monitor on the home gateway or by putting a monitor on each end-host. With the increasing number (tablets, smartphones) of more and more different types (laptops, gaming consoles, set-top-boxes, smart-home, eHealth, etc.) of network devices at home, the task of developing a solution that runs on every platform appears non-viable. Our home gateway based solution only needs to support a single platform. It can easily replace the existing home gateway, given that home gateways are small and cheap and mainly passive devices without user's data stored on them. On the other hand this also means it has limited resources which makes it challenging to run computationally expensive performance optimization algorithms on them. In this chapter we:

1. Present our overall approach on how to optimize the utilization of network resources in home networks (in Section 4.2) consisting of several components including monitoring, application detection, metric computation, resource optimization and traffic shaping;

2. Discuss which metrics are required and review existing measurement tools that provide these metrics (in Section 4.3);

3. Study the CPU and I/O capacity of typical home gateways beyond pure forwarding of traffic (Section 4.4) and assess the resource consumption of the traffic monitoring component of our solution (Section 4.7);

4. Examine the overhead involved in exporting the monitored data to an external metric computation and optimization element (in Section 4.8).

Moreover, we discuss how to best perform application identification (in Section 4.9), present our current state of implementation, (in Section 4.10) and conclude (in Section 3.5).

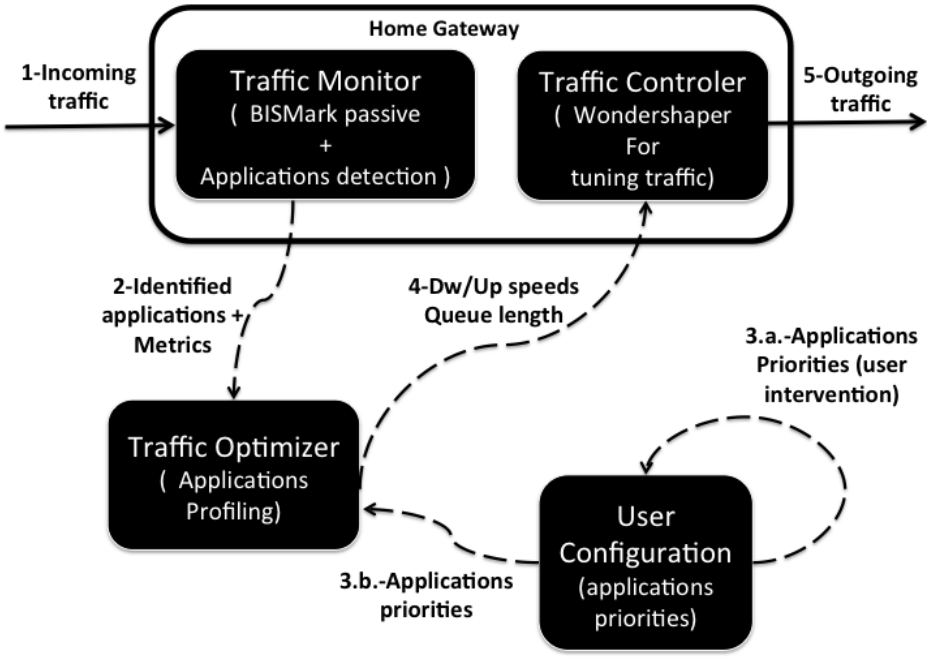


Figure 4.1: Performance Optimization System.

4.2. The Overall Approach

We envision a system that will control traffic in the home network according to application requirements and household priorities. The network resource requirements vary from one application to another. Some will need low delay while others require high throughput. Household priorities also change depending on activity and over time. A user can give more importance to his mails while working, his file download while installing software or his video streaming when relaxing. Both application requirements and household priorities help to decide how to allocate resources. For this optimization system to exercise the required level of control, it must be installed mainly in the home gateway. Figure 4.1 shows the complete approach of our performance optimization system at home.

When traffic traverses the home gateway (step 1), the *traffic monitoring* module records flow and packet information (step 2) for the purpose of determining current active applications along with their performance metrics (*application identification and performance metric computation*). This information is send to the *traffic optimizer* (step 3). The optimizer processes this information and assigns the corresponding application traffic profiles. Further it identifies the resource requirements for a given profile and according to the *user configuration* (application/household priorities) sends control parameters (download/upload speed, queue length) to the *traffic controller* (step 4) which will tune the traffic to prevent performance degradation.

In the following we explain each module of this system in more detail:

- **Traffic Monitor**, this module is composed of a modified version of the BISMark-passive[96] function to perform passive traffic measurement. BISMark-passive was originally developed by researcher from Georgia Tech for the purpose of passively monitoring network traffic and peri-

odically sending small anonymized updates to a central server for analysis to help understand home network usage. The recorded packet and flow information is send to the ...

- **Application identification & performance metric computation**, a process to detect applications and their performance metrics. For any incoming traffic, this process will identify active applications and their corresponding metrics (bandwidth, packet loss, latency, etc.) and send them to the ...
- **Traffic Optimizer**, this module takes as input the active applications with their performance metrics. In addition, it needs existing knowledge (e. g., learned before) about application performance profiles (ranges of acceptable performance for different metrics). This module will combine these information and the user priority to give as output the optimal shaping parameters for the ...
- **Traffic Controller**, this module is designed to apply traffic shaping and traffic prioritization in order to forward the traffic in the best form that avoids performance degradation. Some tools as tc[35] and/or netem[24] could be used for traffic shaping.
- **User configuration**, a learning module that records the priority that the user assigns to each application (class of applications). Its output will be included in the *Optimizer* decision process.

While it is clear that the traffic monitor and the traffic controller need to run inside the home gateway, there is some design space on where to run the application identification and performance metrics calculation, the traffic optimizer, and the user configuration.

The proposed modules and their functions need a considerable processing power to be performed. In the remainder of this chapter we want to address the question how much processing can be done in the home gateway and discuss the overhead of exporting information from the gateway. In this chapter, we focus primarily on the Traffic Monitor and Application identification and metric computation modules. In the following section we will discuss which performance metrics we need to monitor.

4.3. Collection of Application Performance Metrics

Understanding network application performance is a prerequisite to allocating network resource in a way that users are satisfied. In networking application performance is represented with performance metrics such as *throughput*, *delay* and *jitter*. Other metrics such as the number of *retransmissions* in a connection as well as the number of *concurrent connections* will help us better diagnose the current situation. In addition we need to be able to identify the (type of) *application* which causes a certain piece of traffic. Thus we aim to collect all these metrics on a gateway.

Inferring and monitoring network performance metrics has been well studied. Different tools have been developed to help users and researchers measure simple network metrics. In the following we explore the suitability of existing tools for our purposes.

Dedicated metric measurements: For instance, Pathload measures bandwidth[38], T-rat evaluates the rates at which flows transmit data[105], or King estimates delay by measuring the delay between

the closest DNS servers[94]. These tools are actively monitoring a dedicated metric by issuing probes. Despite being fairly accurate the required measurement overhead is a concern to our approach which is supposed to operate permanently.

Network performance diagnose: There are also passive tools that extract network performance metrics. For example, *tcptrace* uses traces collected on end-hosts to compute a set of metrics for each observed connection such as amount of bytes sent and received, number of retransmissions, throughput and others[71]. Similarly, HostView is a monitoring tool[42] that records packet header traces and information about applications and user environment. HostView relies on `gt`[27], for application identification.

As we can see, many tools already exists to measure various metrics. But, these tools have been developed to perform measurements on end-hosts. Moreover, they are not optimized for low resource consumption but rather for high accuracy happily investing more resources. For our performance tracking technique however, we need gateway-based monitoring tools. We can not directly apply the existing solutions for end-hosts in gateways. First, because of the limited resources in home gateways. Second, because on the gateway we do not have access to the same information as we have on end-hosts (e. g., process executable or network stack details).

In-network traffic monitoring An approach closer to our needs is collecting network flow measurements. Protocols like NetFlow collect IP traffic information, for instance source and destination IP's, ports, Timestamps for the flow start and finish time, number of bytes and packets observed in the flow and so on. NetFlow is powerful for collecting IP traffic statistics on all interfaces where it is enabled. Some of our required metrics can be computed from such information. However, it does not collect all the information our tracking needs (e.g. domain names) and is hard to extend.

This is why we base our work on the BISMark firmware[96]. While monitoring a similar set of information compared to NetFlow it is tailored for the use on home gateways (low resource consumption), and easily extensible. The firmware includes the BISMark-passive function which passively collects traces including flow and packet records (timestamps, size, ports, IP addresses, transport protocol and IP to domain name mappings from DNS traffic). These (anonymized) traces are periodically send to a central server for analysis.

Looking at the list of metrics we are interested in at the beginning of this section, we can already calculate many but not all of our desired metrics. In fact, the collection process does not include sequence numbers, acknowledgment numbers and TCP flags which are required to compute round-trip-time, jitter, and retransmissions. For that purpose, we extend the BISMark-passive software to collect the missing metrics by directly changing the implementation to fit our needs.

BISMark achieves low resource utilization through three major mechanisms. First, no computation of metrics is performed on the gateway. Second, it only sends incremental updates to the server, e. g., , the flow information (IPs, ports, transport protocol) is send only once and referred to with a connection identifier afterwards or most timestamps are only relative timestamps and thus require less

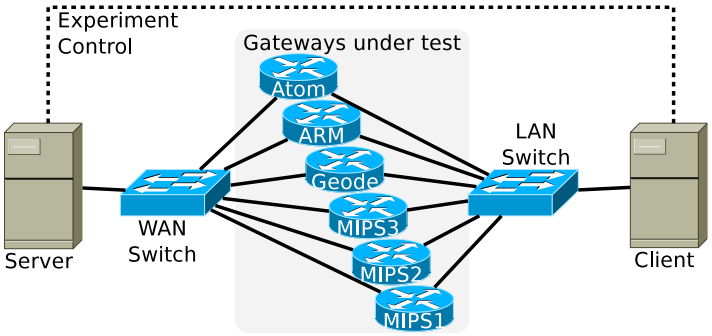


Figure 4.2: Experiment Setup

Table 4.1: Evaluated Hardware (Abbreviations: FE – 100 Mbps Ethernet, GE – Gigabit Ethernet)

Name	Manufacturer & Model	Processor @Speed	InSet	RAM	NICs	Storage
MIPS1	Linksys WRT54GL	Broadcom 5352 @200 MHz	MIPS	16 MB	5*x*FE	4 MB Flash
MIPS2	D-Link DIR-615	RaLink 3052F @384 MHz	MIPS	32 MB	5*x*FE	4 MB Flash
MIPS3	Netgear 3700 v2	MIPS 32bit @680 MHz	MIPS	64 MB	5*x*FE	16 MB Flash
Geode	Soekris net5501	AMD Geode LX @500 MHz	i586	512 MB	3*x*FE	80 GB SATA
ARM	OpenRD Ultimate	Marvell Kirkwood @1.2 GHz	ARM	512 MB	2*x*GE	1 GB USB
ATOM	TranquilPC T2WHSA2	Intel Atom 330 2@1.6 GHz	i686	2 GB	3*x*GE	500 GB SATA

bits to encode. Third, Bismark limits it memory usage for trace data. It records traces in chunks of 30 seconds and each chunk will not store more than a preset number of packets and flows (default: 2^{16}). Then, each 10 minutes all chunks are compressed, sent to a server and deleted on the gateway.

4.4. Home Gateway Constrains

Home gateways typically come as small and cheap boxes build out of embedded hardware with low resources. Their limitations include low processing power, small memory size and most times no storage. Different types and brands of gateways are available.

Given the low resources in home gateways, our first question is: Can we do more than just forward packets through the gateway?

Figure 4.2 represents our testbed setup. We use two edge machines: *Server* and *Client*[1]. We selected six representatives for home gateways, named MIPS1, MIPS2, MIPS3, Geode, ARM, and ATOM in our study. On these gateways, we use Ethernet for both WAN and LAN interfaces and disable the wireless where present. Note, we do not use PPPoE or DSL on the uplink. The WAN (LAN) uplink of each gateway is connected to the Server (Client) via the *WAN (LAN) Switch*. For management and monitoring, the Server and the Client are connected via an additional *Experiment Control* network.

[1]Both run a Linux 2.6.32 kernel. The server has a dual-core Intel Core2 E8400 3 GHz CPU and the client has an octa-core Intel Core i7 860 2.8 GHz CPU.

Home gateways exist in different models with different features. For our experiments, we select six models, whose hardware and software details are shown in Table 4.1. We focus on breadth of different architectures, which also represent different resource capacities. Our selection ranges from home gateways commonly used by customers (MIPS1, MIPS2, and MIPS3) over low power embedded machines (Geode and ARM) to a medium performance system as used in net-books (ATOM):

- The Linksys WRT54 (MIPS1) is the most popular platform for open-source Linux-based firmware since its release in 2002. This system allows us to understand what is commonly available in most households connected via broadband Internet.
- The D-Link DIR-615 (MIPS2) is a recent home gateway that has been found to perform well by Hätönen et al.[33]. This system represents what is currently sold as home gateway in stores.
- The Netgear WNDR3700 (MIPS3) is a popular Wireless-N Gigabit router. It is currently used in the BISMark measurement project[96] and a close sibling to the one used by SamKnows[69].
- The Soekris net5501 (Geode) is, according to AMD, the ideal family for set top boxes, residential gateways, and embedded systems. A similar system was used for the initial BISMark deployment[9;95] in Atlanta, GA.
- The OpenRD Ultimate (ARM) is based on an ARM architecture using the Marvell Kirkwood platform. The OpenRD is the development branch of the well-known Sheeva plugs, a mini computer the size of a power supply unit which directly plugs into a power outlet[90]. These devices are fairly popular as home servers.
- The TranquilPC (ATOM) being equipped with an Atom 330 dual-core CPU with Hyperthreading, is a full-fledged PC. This system allows to explore how increased budget and thus increased resources perform when monitoring.

4.5. Software Tools

Our measurements require software for three tasks: traffic generation, observation of resource consumption, and a passive monitoring tool.

In terms of traffic generation, we chose `iperf` in UDP mode (1500 Bytes packet size), since `iperf` in TCP mode does not allow to determine the bandwidth of the generated traffic. It reports the number of generated packets, the loss rate, and the achieved throughput.

Next, because all gateways run Linux, we can rely on information from `/proc/stat` for the purpose of resource monitoring. This allows us to capture how much CPU time was spend in different CPU modes, such as user, system, idle, or interrupt handling. For the ATOM we multiply the obtained results by 2 since on that system Hyper-Threading creates two virtual CPUs per core which share the same resources. For `/proc/stat`'s point of view this creates fake resources, which we remove by the multiplication.

Finally, we want to understand the impact of passive monitoring on each gateway's load. We select `tcpdump`, the most basic tool for passive monitoring. We configure it to neither write a trace

Table 4.2: Maximum throughput with 0% E2E-Loss. (No CPU monitoring, no `tcpdump`)

MIPS1	MIPS2	MIPS3	Geode	ARM	ATOM
90 Mbps	96 Mbps	96 Mbps	96 Mbps	759 Mbps	832 Mbps

to disk nor analyze the data[2]. This allows us to evaluate the task of capturing packets and delivering them into user-space. Thereby, we can also identify resource requirements of the key component of passive monitoring tools. We decided against writing packets to disk, as the gateways employ very different storage technologies, some simply offering insufficient storage for our task (MIPS1, MIPS2, and MIPS3).

During our preliminary experimentation we noticed that `tcpdump` on the ARM did not close properly. After sending a `SIGKILL`, the process stopped but did not terminate. Continuously increasing E2E-Loss made us aware of the problem and we solved it by using `SIGHUP` instead. A take away lesson is not to rely on the assumption that standard software behaves identical on different platforms.

4.6. Measurement Method

In this section we explain the steps involved in our measurements. We distinguish three scenarios, each consisting of several experiments. For each experiment several metrics are captured. Moreover, each experiment has two parameters, the bandwidth and which gateway to test.

The scenarios define if and how `tcpdump` is used on the gateway. The no-tcpdump scenario serves as a baseline and determines the resource consumption for the forwarding and NATing of packets. In the tcpdump-68 scenario we additionally run `tcpdump` with snap-length 68 bytes (default) on the gateway. In the tcpdump-1500 scenario we use a snap-length of 1500 bytes, corresponding to full packet capture.

In terms of metrics we extract the end-to-end loss (*E2E-Loss*) from the `iperf` server log. We, as well, monitor the CPU utilization on the gateway and report the averaged $(1 - \text{idle})$ value. Furthermore, for scenarios with `tcpdump`, we also measure the fraction of packets captured on the gateway.

All the tests are done with UDP using a throughput in the order of typical DSL access speeds: 1 Mbps, 6 Mbps, 20 Mbps. For the ARM and the ATOM, which have Gigabit interfaces, we also test 100 Mbps, 200 Mbps, and 500 Mbps. Moreover we include the maximum sustainable throughput that varies depending on the gateway, see Table 4.2.

We perform experiments for all possible combinations of bandwidths and gateways. Each combination is repeated three times and the average of all runs is reported. Note, that we did not experience significant deviations. Thus to increase readability we omit to show minimum and maximum values in the results. Each experiment consists of the following steps:

1. Prepare

[2]By defining `-w /dev/null` on the command line.

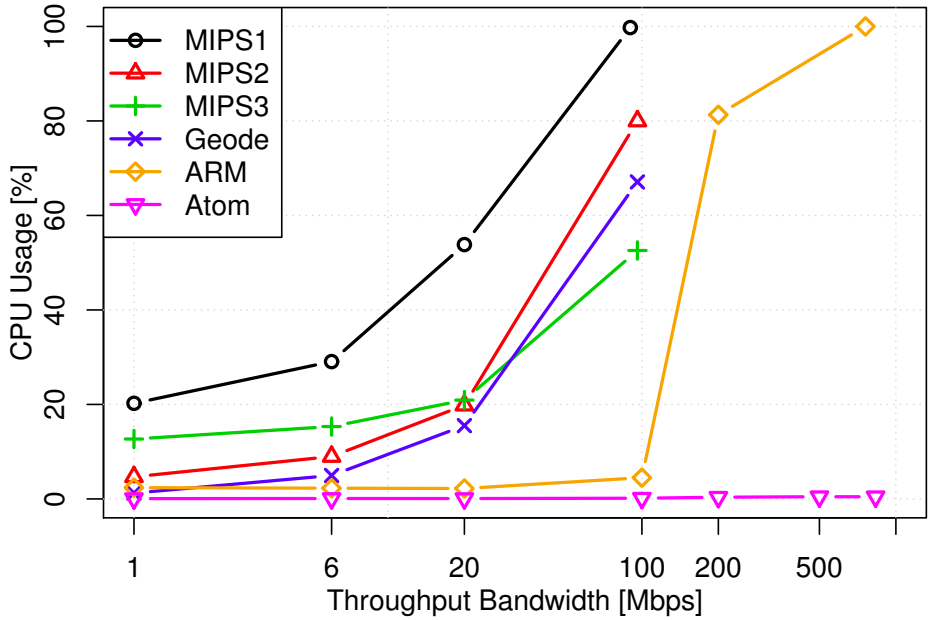


Figure 4.3: CPU consumption for six home gateways (MIPS1: Linksys WRT54GL, MIPS2: Dlink DIR-615, MIPS3: Netgear WND3700, AMD Geode LX, OpenRD Kirkwood ARM, Intel Atom 330)

   a) Set client's default route to the selected GW
   b) Start CPU monitoring
   c) Start `tcpdump` (depends on scenario)
2. Generate traffic with `iperf`
3. Clean-up
   a) Stop `tcpdump` (depends on scenario)
   b) Stop CPU monitoring
4. Collect reports

4.6.1. Baseline Scenario

Figure 4.3 shows the CPU consumption for the 6 different gateways in the market, while forwarding packets at different bandwidth (x-axis). Each gateway has a maximum forwarding bandwidth, beyond which no measurements are reported in this plot.

We observe that even at 100 Mbps most gateways have CPU resources left to capture and process packets. Only the aged WRT54GL (MIPS1) already reaches its CPU limit. While the Atom and ARM boxes easily achieve several hundreds of Mbps, with today's Internet access link speeds, 50 Mbps should suffice for almost all users. The Netgear WNDR3700 (MIPS3) is a good compromise between cost (80 USD) and performance (60 % remaining CPU @ 50 Mbps). We conclude that more than half of the CPU cycles are left for, e. g., packet capture, network metrics computation, or application detection. Thus we select it for further study (see Section BISMark Evaluation).

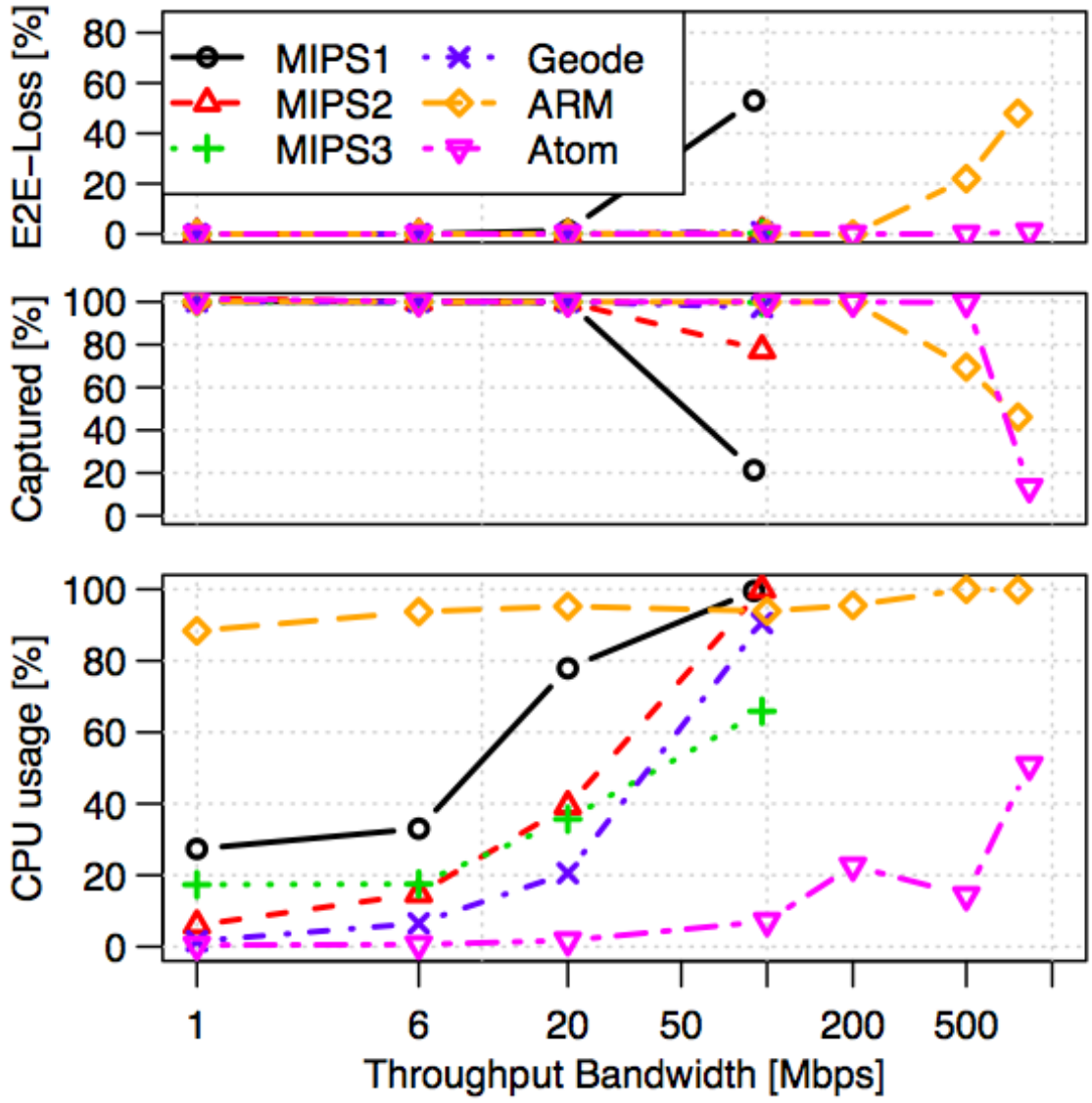


Figure 4.4: Evaluation Results from the tcpdump-68 scenario. End-to-End loss (top), Packets captured (middle), and CPU utilization (bottom) vs. traffic bandwidth (x-axis in logscale).

4.6.2. Scenarios Including Packet Capture

The tcpdump-68 scenario extends the no-tcpdump by additionally executing `tcpdump` on the gateway. It captures all packets on the external interface with the default snap-length of 68 bytes. Looking at the load levels, in Figure 4.4, for this scenario we expectedly find overall increased CPU utilization. The results for capturing full packets (1500 bytes) are shown in Figure 4.5. Both figures have an additional third plot in the middle showing the percentage of captured packets.

The ATOM does not expose a big impact on the through traffic when `tcpdump` is running. Only at maximum bandwidth it looses around 1 %. It is also able to capture all packets except at maximum bandwidth where only few packets are captured (down to 13.2 % for tcpdump-68 and 8.8 % for tcpdump-1500). The reason why the CPU is not fully utilized is the dual-core nature of the system. Here, a CPU utilization of 50 % translates into one core being fully utilized. Given that the

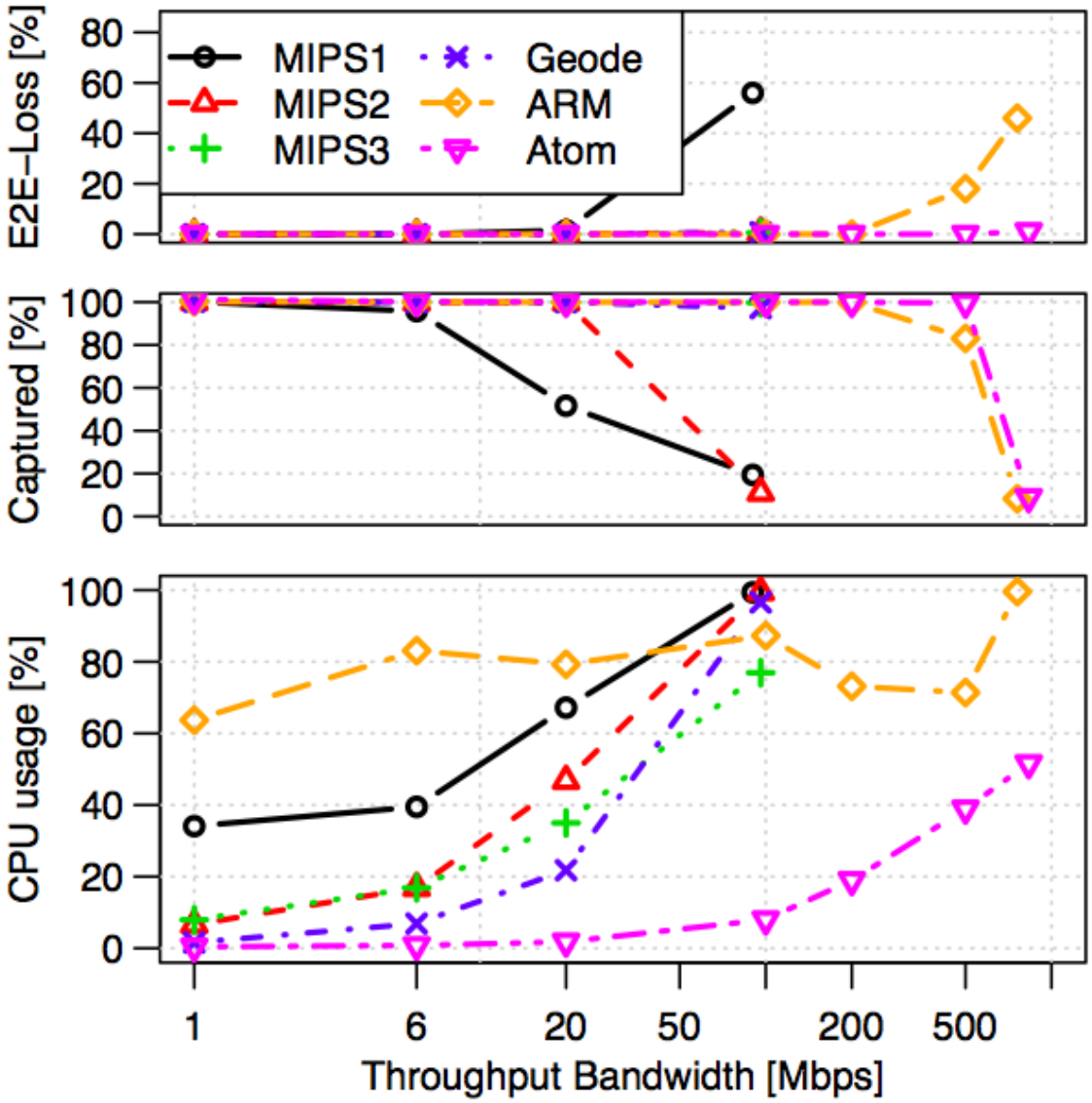


Figure 4.5: Evaluation Results from the tcpdump-1500 scenario. End-to-End loss (top), Packets captured (middle), and CPU utilization (bottom) vs. traffic bandwidth (x-axis in logscale).

kernel, including interrupt handling and packet capturing, is done by only one core, the low number of captured packets is due to CPU capacity limitations.

Interestingly, we find that the ARM is under high CPU load even for throughput as low as 1 Mbps. A possible explanation might be a different implementation of the capturing stack on the ARM based architecture, such as using busy waiting instead of using `select()` or `usleep()`[3]. If this explanation is correct, there are additional cycles available even though an almost fully utilized CPU is reported. Furthermore, the E2E-Loss increases significantly over the baseline up to 20 % and 50 % at 500 Mbps and maximum bandwidth, respectively. As for the ATOM, the percentage of captured packets drops for maximum bandwidth, but also for 500 Mbps. It seems like the ATOM is prioritizing the forwarding path and therefore cannot capture as many packets. However the ARM loses roughly as much as it cannot capture.

[3]We have not yet verified this in the source code of the kernel, libpcap, or `tcpdump`.

Yet, the ARM and the ATOM do not cause losses or drop packets when operated under a network load less than or equal to 200 Mbps and 500 Mbps, respectively. The Geode is comparable to these former ones, with the exception that its interfaces limits it to 96 Mbps for which it does neither lose packets nor drop packets while capturing.

Contrary, the MIPS2 and MIPS1 do not manage to capture all the packets at maximum bandwidth. MIPS2 is dropping 22 % in the tcpdump-68 scenario and 88 % in the tcpdump-1500 scenario. While the MIPS2 does not experience E2E-Loss at maximum bandwidth, the MIPS1 does. Not only it loses roughly twice the amount (60 %) as compared to the baseline, it also misses to capture around 80 % of the packets. Here despite running on the same architecture the MIPS3 performs much better. It is able to sustain its maximum rate while capturing all the packets (less than 1 % loss) for both `tcpdump` scenarios and leaving 25 % of CPU capacity idle.

Although CPU capacity is the prime concern, home gateways also have only a small amount of flash memory and typically no disk storage space. Thus memory management is an important concern for our approach. The WNDR3700v2 ships with 64 MB of RAM and 16 MB of flash memory. After this general performance evaluation we now continue evaluating BISMark-passive's performance on the selected home gateway.

4.6.3. Discussion

We find that except the MIPS1 all our gateways operate without any losses on the end-to-end path and manage to capture all packets up to a bandwidth of 20 Mbps. Given that most DSL access links are not offering higher throughput our findings are encouraging to implement passive monitoring of traffic on the access link on home gateways. In case of fiber-to-the-home access links (approx. 100 Mbps) our results show that the top four MIPS3, Geode, ARM and ATOM are viable option. However if the home network itself (up to 1 Gbps) should be monitored only the ARM and the ATOM provide reasonable performance. A Geode-like box with Gigabit-Ethernet support could also work.

The three example monitoring applications (writing packets to disk, collecting NetFlow, and running a DPI tool) will of course consume additional resources. Yet, when e. g., considering Gigabit throughput, one of the ATOM's cores is still idle. Or, when looking at 20 Mbps (typical DSL speed) even the MIPS2 and the MIPS3 only uses half of their CPU.

To interpret these results we utilize insights from our previous work on capturing performance of server architectures[87;88]: Filtering packets and writing to disk does not consume a lot of additional resources (less than 10 %). On the other hand processing packets (simulated via `memcpy()`s and `gzip`ing) roughly doubles the CPU consumption.

With that in mind it seems likely that passive monitoring tools like NetFlow, bismark-passive[4], snort[4] or Bro[5] can be executed on our gateways. As an example of a real application, we test bismark-passive's resource consumption in the next chapter. For those tests we select the MIPS3. It is the cheapest gateway that allows capturing up to 100 Mbps while leaving considerable resources to run

[4] `www.snort.org`

[5] `www.bro-ids.org`

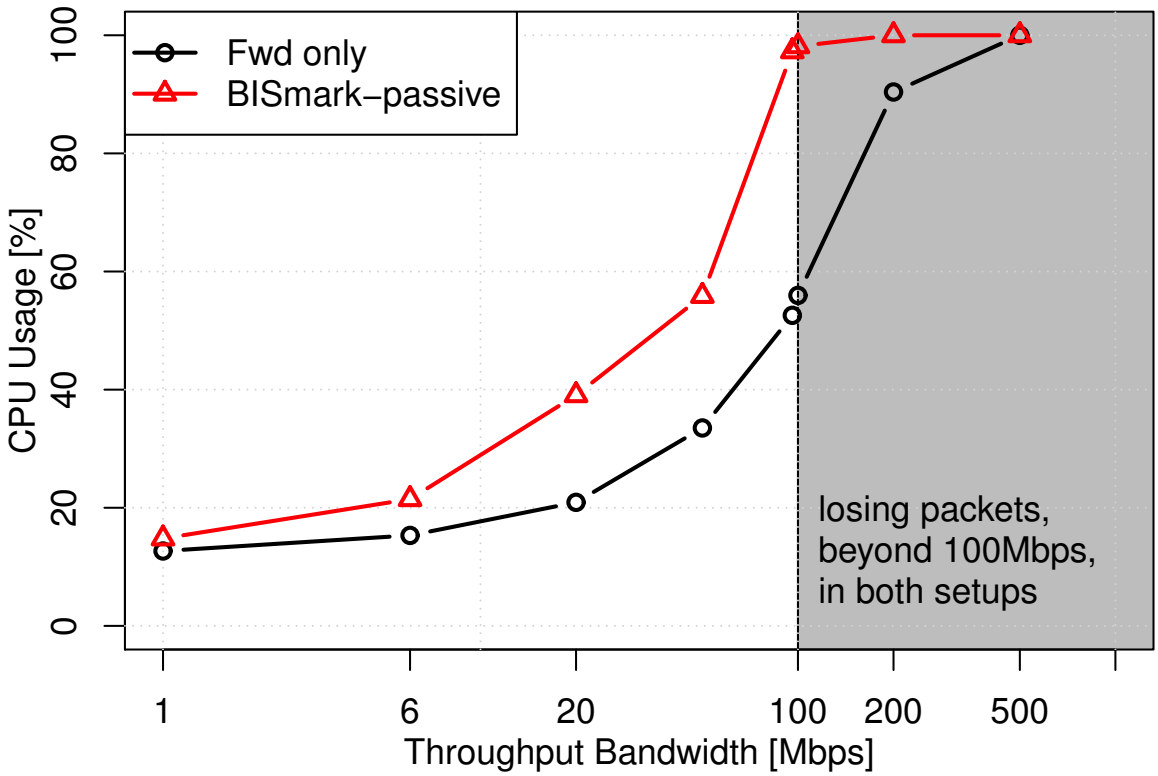


Figure 4.6: CPU consumption for forwarding only and running Bismark-passive. Note that the gateway starts losing packet in both setups beyond 100 Mbps.

a passive measurement application. Extrapolating the results from this section, if bismark-passive works properly in the MIPS3, it will also likely run in the Geode, ARM and ATOM.

In order to evaluate the performance of our home gateways in a realistic monitoring scenario we repeat our experiments for the MIPS3 with bismark-passive. Like `tcpdump`, bismark-passive relies on libpcap to capture all traffic passing through the gateway. In addition bismark-passive analyzes the recorded packets and extracts information such as packet sizes, flow information (timestamp, flow id, ports, transport protocol, IP's, size, mac id, domain names...)

4.7. BISMark Passive Evaluation

As explained in Section 4.3, we rely on BISMark-passive software to build an extended technique for application performance tracking. Bismark-passive captures all traffic passing through the gateway using the de facto standard libpcap library. It then analyzes the recorded packets and extracts the information described in Section 4.3. Periodically differential updates of the extracted information[4] are gzipped and exported to a server. In Figure 4.6, we evaluate the BISMark-passive software by comparing its resource consumption against the case of just forwarding packets from last section, when running in the MIPS3 gateway.

As expected the additional task of capturing packets and analyzing them causes noticeably higher CPU consumption at bandwidths beyond 5 Mbps. At typical home network speed (20Mbps), BISMark-passive requires twice as much resources (40% in total) than just forwarding the packets. However the

additional CPU requirements of BISMark-passive remain almost constant when looking at 50 Mbps still leaving 40 % of the CPU for other tasks. Yet, at maximum forwarding rate (96 Mbps) BISMark-passive uses the entire CPU capacity.

This evaluation tells that BISMark-passive already consumes a decent amount of CPU. Thus, if in addition we compute all our required metrics, run the performance optimization and the resulting traffic shaping in the gateway, it will generate a high overhead in terms of CPU/memory usage and is likely to run out of resources at high rates.

To avoid a possible lack of resources, our idea is to split the application performance tracking between the home gateway, which is mainly monitoring and extracting packet and flow information, and an additional computation element, which will analyze the compute the performance metrics and determine an optimized resource utilization. The result of this optimization is a set of policies (prioritization rules, QoS parameters) that the home gateway will need to apply to the traffic passing through it. The computation element, could be a laptop or media server at home or "cloudified" resources of a dedicated performance management service.

However with this approach, we need to export information from the gateway in order to be processed outside (computation element). This incurs additional bandwidth requirements, which we will study in the next section. But with this approach we can trade-off home gateway CPU utilization for increased upload bandwidth.

4.8. Overhead of performance tracking

With the insights from the last section on scarce CPU resources and our approach to ship of the collected information from BISMark to a computation element, we need to ensure the offloading overhead is manageable. Recall that we extended BISmark-passive to collect more traffic information in order to enable our application performance tracking. This causes additional overhead which has to be taken into account.

The difficulty is now that our deployment of the extend BISMark software has not left the lab experimentation stage, and we have no data from a real world deployment. Yet, from analyzing the code of BISMark we know that the overhead which is equivalent to the size of the collected traces only depends on the number of packets and the number of flows that it contains. We also know how much data is produced per packet and per flow seen by the home gateway. Thus, we can combine these constants with real world data collected with the unmodified BISMark firmware, to estimate the required overhead of our extended version. This data was made available to us by the researcher from the BISMark project, which collected these real world traces from 24 different home users using BISMark gateways.

Figure 4.7 shows the distribution of the estimated overhead (MB) from real world traces as a function of packet rate (K packets/s) in a two-dimensional histogram, where darker regions indicate high numbers of traces matching the region. In addition we show theoretical boundaries (worst and best case). At a given packet rate these boundaries represent the two extreme cases: *(i)* Either every

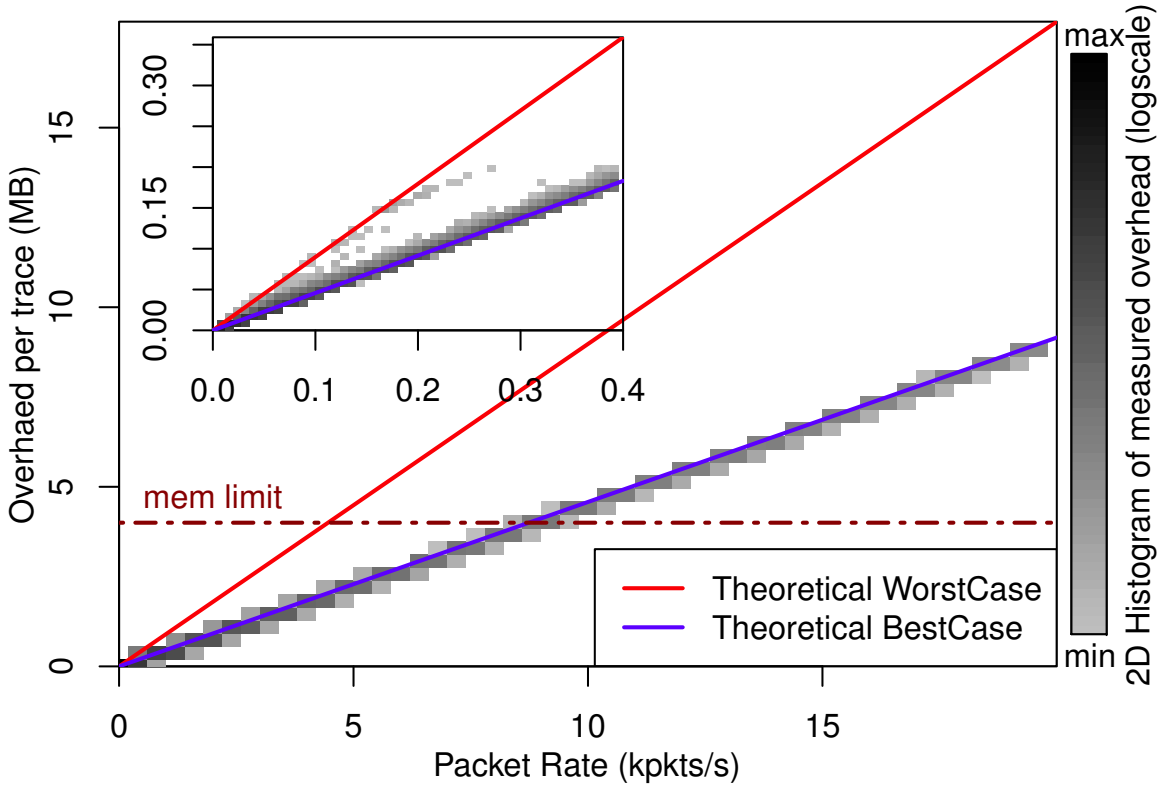


Figure 4.7: Distribution of the real collected traces overhead as a function of packet rate

packet in the trace belongs to a different (and new) traffic flow. This represents the worst case as we need both a packet and a flow entry for each packet. Or *(ii)* where every packet in the trace belongs to the very same flow. In which case we still need to store a packet entry for each packet but only one flow entry. Finally we assume that we use no more than 4MB of the gateway memory (RAM only) which is shown as a discontinuous helper line.

As a first takeaway from Figure 4.7, we find that the real world data matches closely with the theoretical best case. This means that we observe many more packets than different flows, which matches our expectation. If we look at the zoomed inset (small picture, <400 packet/s, 97 % of all the traces), we see a higher variation in the overhead, with some traces reaching the worst case boundary. Yet, at low packet rates also the absolute overhead is small.

To better understand the packet rate, we transform (K packets/s) to (Mbps) assuming an average packet size of 660 bytes from[5]. Thus, based on average packet sizes 5K, 10K, and 15K packets/s would translate into 26, 53, and 79 Mbps. The figure shows that the memory limitation is respected up to 8K packets/s, as the best case is prevailing at these packet rates. Although this would translate to 42 Mbps, using average packet sizes, we already know that in the best case we see many more packet than flows. Thus we have less small TCP maintenance packets (i. e., SYN, ACK, FIN) and likely many full sized packets. Using full sized (1500 bytes) packets 8K packets/s translate into 90 Mbps.

Looking at our overall approach the findings from this overhead estimation show that it is no problem to collect and export the collected data. Although a worst case of 4 MB memory limit every 30 sec would translate into an uplink bandwidth requirement of 1 Mbps, this case is highly unlikely.

On the one hand the traces are compressed once they are collected which reduces their size to 25 %–50 %. On the other hand from the distribution of packet rates we observe in the data, high packet rate occur only in very few cases (<0.1 %).

4.9. Application identification

Many projects already map network flows to application names. What makes application identification in our case more challenging is that we need to develop techniques that offer fast application identification and consume small amount of resources if we want to run it in the gateway. On the other hand for our purpose it might suffice to only detect a certain class of traffic as defined by its network performance profile, i. e., the classes requirements on the network.

In general like for the network performance measurement tools there are two major classes of tools for labeling network traffic with their corresponding application. There are end-host based tools such as ETW[36] (Windows only), *lsof* or *GT*[27] (for UNIX based systems) which associate flows with the appropriate socket entry from the kernel socket tables. Despite their high accuracy, these tools are not possible to use in our gateway-based approach. Thus we need to resort to non end-host based application identification techniques such as port based classification or deep packet inspection (DPI). As an additional requirement for our application performance tracking, we need to identify active applications in real-time.

The BISMark project already tested a DPI solution, namely the TIE tool[16] which proved to consume too much resources for permanent operation. Given that other DPI solutions such as *snort*[92] or *bro*[75;76] have a extended functionality over TIE, those will likely also consume too much resources. Thus, a gateway-based application identification requires a simpler and less heavy solution.

Considering our data collection process which does not include full packet headers, real-time application classification[57] can be a suitable solution. Although, this technique has a training phase that might be resource consuming, we believe it is a good start because it only needs the timings and sizes of the first packets of a TCP connection[3]. This solution could also be implemented on a computational element outside the home gateway, and the information collected with our extended BISMark software suffices as input. Our idea is to combine this application classification tool with port-based classification and the analysis of domain names collected from BISMark-passive. For HTTP traffic, destination names are especially useful to identify related services. Then to distinguish between different HTTP services, we can also use content-type. If any traffic carries content-type, we can identify whether it is an image, a video, or simple text.

4.10. Implementation

Given the implementation challenges that home gateways bring, the evaluation in this chapter is crucial for any implementation decision we choose. In order to track application performance, we modify BISMark-passive implementation. For our collection process we add sequence numbers,

acknowledgment numbers and TCP flags in the collected traces. They will be valuable to compute round-trip-time, jitter, and retransmissions.

We are working on a first small deployment of BISMark boxes to collect traffic using our modified BISMark-passive function. The dataset will allow us to deepen our understanding of home traffic and help avoid performance degradation issues. We aim at performing the collection period for a long time in different homes. We already started this step by changing the BISMark-passive code, setting up the gateways and preparing the agreement for the users who will be hosting the gateways at their homes.

We also have already investigated implementation solutions of our final optimization solution. We will start by testing available tools such as WonderShaper[102]. We choose this tool because it already provides low latency for interactive traffic, allows web surfing at reasonable speeds while uploading/downloading, and ensures uploads/downloads do not hurt one another.

4.11. Summary

Home networks and application performance are two challenging areas. In our work, we aim at avoiding performance degradation of active applications in a home networks by tracking their performance from home gateways. We discussed the gateway resource limitations along with the trade-off between different implementation strategies (end-host vs. home gateway). We introduced a modified version of BISMark-passive that collects valuable information for application identification. We showed that even if it generates an additional overhead, the results are promising for an application identification technique along with a traffic collection process. We explained possible application identification techniques and discussed our solution implementation. Our overall evaluation showed that it is possible to perform an application performance degradation technique from the home gateway following our guidelines.

Part II

Getting the most of mobility experiments

Chapter 5

Toward maximum exploitation of mobility traces

ACHIEVING efficient communication in disruption-tolerant networks (DTN) depends on a deep understanding on the dynamic laws governing the network. In particular, it is important to investigate the frequency and duration at which nodes meet each other using the notions of *contact* and *intercontact* patterns[45;73].

The best way to analyze or validate any protocol or design choice in DTN is through real deployments. Nevertheless, because of implementation challenges and even financial costs, only a few experimentations have been reported in the literature[89;104]. As a consequence, several works still rely on synthetic mobility models.

While synthetic mobility models are useful to isolate specific parameters of a solution or help investigate the scalability of a system, they cannot always reflect real life conditions. Synthetic mobility models are always used as a perfect model for evaluation because of their full knowledge of the mobility and the mimic of real human movement. Researchers made an additional effort to develop models based on real mobility traces in order to guarantee the highest possible realism.

To construct a trace-based model, multiple human movement scenarios are performed. From multiple experimentations, several real movement traces are collected to extract mobility characteristics. Based on the observed characteristics, a trace-based model is built. The goal is to have a model that offers synthetic traces that are close to the original human movement traces. Many trace-based mobility models were developed[2;39;40;51;65;70]. There are simple mobility models[62] and more complex models, such as those based on real GPS traces[58;81].

Because of the challenges in the collection process, a limited number of contact traces is available for the research community[53]. A representative contact trace involves recruiting a large number of users, setting up the measurement devices (e.g., frequency at which nodes will send beacons) and clean the data. A representative contact trace involves recruiting a large number of users at first. Then, we must set up all the measurement devices using the right parameters (e.g., determining the

Figure 5.1: Using *Bred* mobility traces allows for more realism than pure synthetic mobility models.

frequency at which nodes will send beacons to potential neighbors). Finally, it is important to clean the data and get it ready to use.

In the next chapter, we propose a *mobility trace breeding* system that, from a single real-life contact trace, derives plausible contact traces *inspired* from the original trace. The objective is to get as close as possible to a realistic scenario while avoiding setting up a new experiment (see Fig. 5.1). The motivation for this work came up during an experiment where we had to choose between Bluetooth and Wi-Fi for the ad hoc channel. In fact, what would be the impact of such a decision on the resulting contact trace? The right answer to this question would require setting up from scratch as many experiments as the number of different configurations we are interested in.

Our system only needs a single contact trace from a real experiment to *breed* several traces, just as if we perform several experiments with different configurations. It follows two basic steps:

1. **Spatial inference.** We derive a plausible mobility trace from the simple contact trace by using an inference system that takes as input a contact trace and generates a "plausible" spatial mobility.

2. **Trace breeding.** From the spatial mobility trace, we set the values of the parameters (communication range, measurement interval) and rewrite the original scenario with the resulting "bred" contact traces. In the current version of the system, although this is not a requirement, we consider a disk propagation model.

Most of available contact traces gather human contact opportunities and involve, in general, a limited number of nodes. Some examples are described in Table 5.1. In the Reality Mining experiment (MIT), a group of 100 participants ran an application in their mobile phones to capture closeness information during a whole year[21]. Details about the rest of the datasets are given in Section 6.4. The Haggle project collected contact information between attendees of the IEEE Infocom conference Intel iMotes[10]. Rollernet, is another experiment that considers iMotes to capture contacts between rollerbladers. The difference between these projects is the measurement interval. While both Reality Mining and Haggle traces relied on measurement intervals of 600 and 120 seconds, respectively, Rollernet used 15-second measurement intervals. In the same range of measurement intervals (20s) but with many more participants (789), Stanford high is an interesting experiment held in a US high school to record face-to-face contact opportunities[85]. Gaito et al. conducted an even finer measurements using a dedicated device known as the pocket mobility trace recorder (PMTR), which measured contact opportunities every second[25].

Those traces are interesting but unfortunately are not all publicly available and the results we can extract from their analysis are limited to specific situations. But still, we can find some public traces

Table 5.1: Real contact traces examples

Name	Duration(days)	Devices(#)	Frequency(s)
PMTR	15	44	1
Rollernet	0.125	62	15
Stanford	0.375	78	20
Infocom	3	70	120
MIT	365	100	600

collected from experiments using more than one technology (bluetooth and Wi-Fi)[60;66;80]. The problem is that they do not contain any contact information required for our methodology. That is why several works have started investigating how to improve the experimental methodology, especially in terms of reproducibility. Many calls have been made for network research experiment and results reproducibility[7;20;31;99]. Recent work from Handigol et al. proposes an approach to enable runnable network systems experiment using Container-Based Emulation (CBE)[31]. They envision an environment with virtual hosts, switches, and links running on a multicore server, using real application and kernel code with software-emulated network elements. However, for the best of our knowledge no work has been proposed neither on the reproducibility of contact traces nor on their creation. It is exactly on this point that our work comes at play.

It is important to underline that our methodology is not intended to give exact results, but to generate possible contact traces *strongly inspired* from real experiments.

Assume one real contact trace was sufficient to have multiple reliable contact traces. If that was the case, several research efforts could have benefited from that approach. For example, there have been many studies using dartmouth college dataset[11;34;52;54] or UCSD[61].

Traces in these datasets only consider a contact opportunity when users are within the same Wi-Fi access point. Thus, missing contacts when users are within each one communication range. If the experiment was done using bluetooth at first in order to obtain contact opportunities defined by the communication range, our system would have bred bluetooth traces to obtain Wi-Fi traces. Thus, enlarging the dataset and collecting all the contact opportunities.

In this chapter, we make the following contributions:

- We discuss the problem of real-life measurements collection and real experiments challenges.
- We propose the *breeding* system as a combination of two different parts. To the best of our knowledge, this is the first time a such realistic solution is proposed.
- We check the conformity of our proposal by comparing the results of our breeding system with the original contact trace of a mobile network generated synthetically. Then, we apply the system to a real-world dataset and we show that indeed, changing the communication technology has a considerable impact on the contact trace.

Chapter 6

Breeding contact traces

THE second part of the thesis is about communications in disruption-tolerant networks (DTNs). K.Fall presents DTNs as a very robust network to encompass challenges[49]. The proposed tradeoff is indeed interesting; accept longer delivery delays to get higher reliability in return. This reliability is achieved by using the *store-and-forward* strategy. The principle is simple, the whole message is sent over each link of the network before being transferred to the next hop. Naturally, it requires more storage on routers in the path but in exchange it takes data gradually to the destination. Understanding network dynamics is key for reliable communications in DTNs which brings an actual need for real contact traces.

Collecting real contact traces in disruption-tolerant networks is a complex procedure. From the experiment setup to the data cleaning, a lot of efforts is required. We propose a *breeding* system to derive possible contact traces from a single real experiment. We check the consistency of our system using synthetic and trace-based mobility traces. Then, using real dataset extracted from real experiments.

We present the breeding system and explain step by step how it proceeds with a contact trace.

6.1. Methodology

Let us consider a general scenario of a real mobility experiment – a fixed number of nodes moving in a defined space. Each contact between any two nodes is recorded in a contact trace. Each record is a quadruplet $[i, j, t_1, t_2]$ indicating that nodes i and j were in contact between times t_1 and t_2.

Traces are bred following the steps depicted in Fig. 6.1. As stated before, we use a single original contact trace during the breeding process. This process involves two main steps, described in the following.

Step 1: Inferring spatial mobility from a contact trace. Predicting mobility is a concern for various services guarantee such as location-aware applications[83] or network optimization[93]. To understand user mobility, many mobility models are available but in our case, we are interested in mobility that can be inferred only using contact traces. Thus, we rely on the *plausible mobility*,

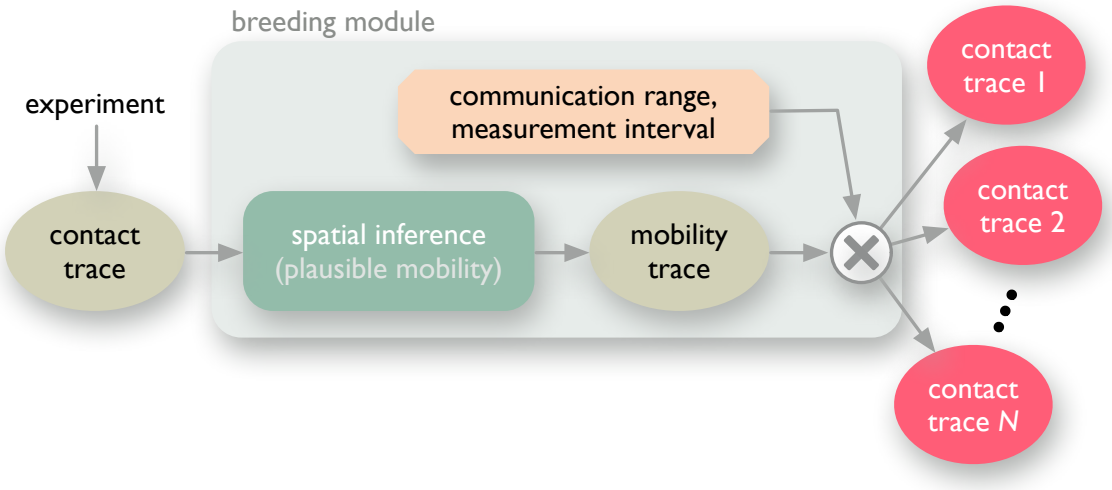


Figure 6.1: Breeding System.

a fast heuristic algorithm to infer a plausible spatial mobility from contact traces[100]. Plausible mobility relies on dynamic force-based graph drawing, in which attractive and repulsive forces are applied to nodes according to the connectivity graph. As in a real physical system, the nodes then converge to a minimum stress (or energy) position. Force-based algorithms are particularly well suited to the plausible mobility problem because each pair of nodes that are in contact will tend to be geographically close to each other.

Plausible mobility has two main benefits. First, it allows realistic translation of the contact trace to a mobility one. Second, it enables to see and understand the network dynamics. But, the quality of the inferred mobility trace is linked to many parameters such as the precision of the measured contact trace. It has been shown that the inferred mobility is more precise for high measurement frequencies. Moreover, the presence of reference nodes (anchors) in the contact trace helps determine the position of the remaining nodes[101]. Plausible mobility, at this stage, involves careful configuration.

Step 2: Breeding contact traces from spatial mobility. The mobility trace obtained with plausible mobility is then used to breed traces. Our method includes two parameters that users can configure to produce the measurement scenario they want:

- **Communication range (ρ).** In an experimental setup, the technology used for measuring the contacts between nodes has a direct impact on the observations (each technology has, for example, a maximum transmission range). For example, if the real experiment is performed using Bluetooth (with a range of about 10 meters), the underlying connectivity graph would be completely different than the one obtained with Wi-Fi (much longer transmission range).

- **Measurement interval (ϕ).** Our method gives the ability to change the frequency at which nodes send probes to potential neighbors. Of course, the higher the frequency, the more representative the contact traces. If the real experiment did not allow using high frequency measurements for some practical reason (e.g., to save energy at the measurement nodes), our system can help make a projection of the network in such a case.

We use the mobility trace from plausible mobility to identify couple of contacts using the specified measurement interval ϕ and range ρ. From the mobility trace, we consider the positions of all nodes every ϕ units of time (in our case, seconds). We define then that two nodes are in contact if they are within the transmission range ρ of each other.

6.2. Constrains

In our approach, The contact trace to breed can be either synthetic or real.

Mobility models allow to produce contact traces by simulating a proximity-based model for example, where two nodes are in contact if they are in the transmission range of each other. We consider the resulting contact traces as *perfect*, since we record all the contact opportunities and control the whole parameters. We use this approach for our conformity check in Section 6.3.

In contrast, real-life contact traces are considered *noisy*. Real traces are measured according to a measurement interval which fails in sometime to record all contact opportunities. Using a communication technology as ZigBee or bluetooth for instance, takes few seconds but still may miss contacts. The longer the measurement interval, the higher the probability to miss contacts. Indeed, long measurement intervals make it harder to catch short contacts. Worse, consecutive short contacts can be considered as a single long contact thereby distorting the results. A solution named Pocket Mobility Trace Recorders [25] has been developed to overcome such issues. Yet, it can not avoid typical wireless limitations such as interferences.

In order to translate contact traces to spatial mobility traces, we use *plausible mobility*. It is so far the only algorithm that allows to propose a spatial mobility of nodes only based on their contact times. The algorithm does not provide the exact mobility but ensures two main properties. First, the nodes speed remains realistic and limited. Second, the original trace can possibly be reproduced as the concept of nodes in contact (within the transmission range of each other) is always respected. The more constrains, the closer the mobility trace will be to the original contact trace. For example, additional information as fixed nodes positions help the inferring algorithm. But they are not always available, worse the network can be large and/or sparse which makes the inference harder. Even though plausible mobility offers a reliable solution, it also introduces a layer of possible errors.

From spatial mobility to the bred contact trace, we consider that the resulting trace is complete. In other words, we do not apply any interference or loss model on top of the results. An interesting perspective could be to reproduce the original environment constrains (measurement losses, interferences) and apply them to the bred traces in order to mimic the initial environment challenges of the experiment .

6.3. Conformity checking: synthetic traces

Before evaluating our system with real-world contact traces, we first check its conformity by using synthetic traces generated using the random walk mobility model. Relying on a synthetic model is

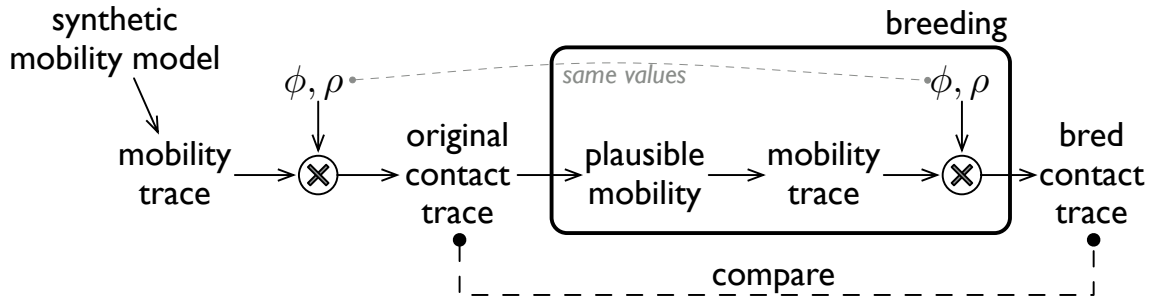


Figure 6.2: Checking the conformity of our breeding system using synthetic contact traces.

Table 6.1: Synthetic mobility traces used to check the consistency of the breeding system.

Trace	**Duration(s)**	**#devices**	ϕ **(sec)**	ρ **(meters)**
RT1	300	200	1	100
RT2	1350	50	120	100

the only means we dispose to check whether the mechanism leads to results that are consistent with the original trace, as we explain below.

Because real traces are subject to physical phenomena (e.g., interferences and data losses), we use a synthetic mobility model to check whether our system conforms to the original trace (complete as we explain below). We perform the conformity check as illustrated in Fig. 6.2.

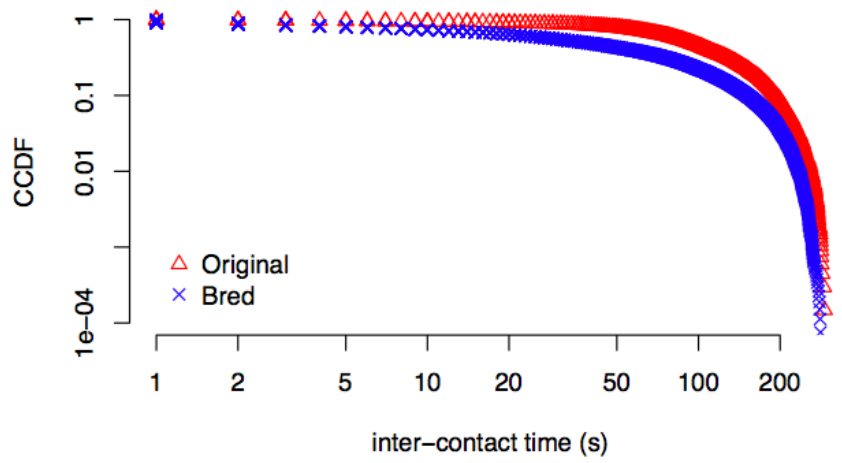
We check our system with two scenarios with different values of ϕ and ρ. We call these scenarios RT1 and RT2, as they rely on a synthetic contact trace following the Random Trip (RT) mobility model[6]. Nodes move on a 1,000m $\times$ 1,000m area at speeds chosen uniformly between 1m/s and 10m/s, with no pause time. Their transmission range is 100m (leading to an average node degree of 0.5π). Note that the node degree is quite low, which makes the inference harder for the plausible mobility algorithm. We run the network for a duration between 300s and 1,350s.

As explained in Section 6.2, the smaller the measurement interval ϕ, the better the results of plausible mobility. We show the results for two different combinations of values. The first is to stress the system, so we run it with a larger number of nodes (200) and small measurement intervals ($\phi =$ 1s). The second test is intended to comply with typical real contact traces (which have, in general, fewer nodes and longer measurement intervals) – we consider 50 nodes and $\phi = 120$s. Table 6.1 summarizes the parameters of the mobility models we use.

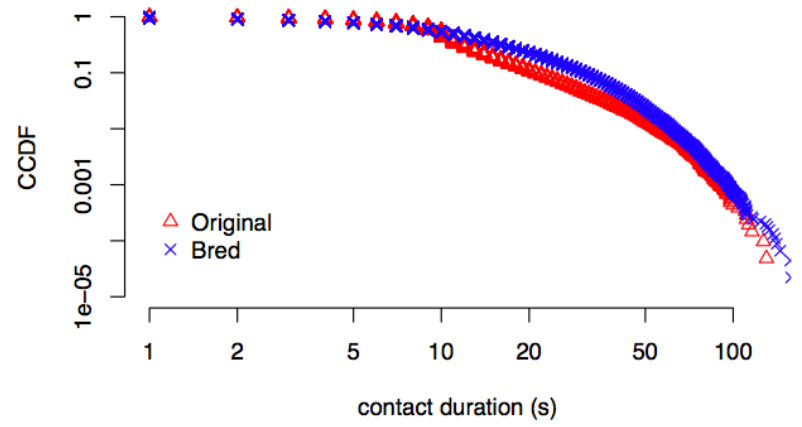
As depicted in Fig. 6.2, to evaluate the correctness of our system, we run the breeding system using the same (ϕ, ρ) of the original trace. In this way, we are able to check if the breeding system introduces any bias or not.

Fig. 6.3(a) and 6.4(a) show the complementary cumulative distribution function (CCDF) of the aggregated intercontact times for both the original and bred traces. As expected, both distributions follow a truncated power law[10]. Moreover, the results show that our breeding system produces a contact trace that is really close to the original one. Fig. 6.4(a) shows that, even for large measurement intervals, our breeding system produces good results.

In addition, we show in Fig. 6.3(b) the CCDF of the contact duration over all contacts. The distributions of both traces are close with a slight increase in contact durations for the bred trace. For



(a) Intercontact.

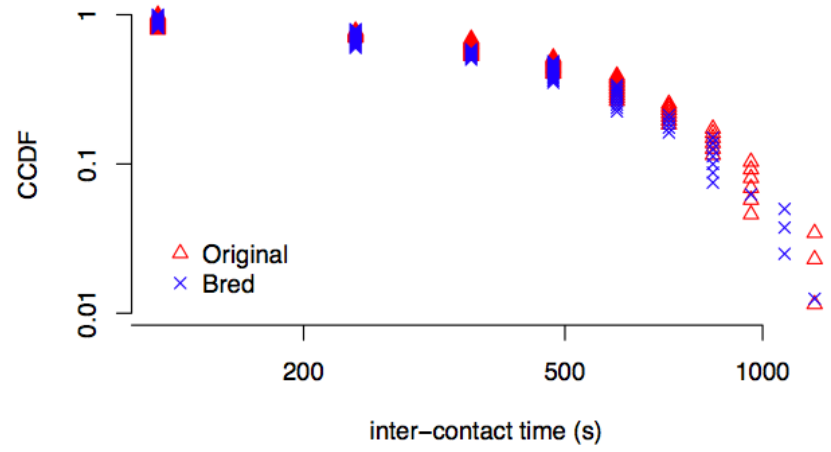


(b) Contact.

Figure 6.3: Intercontact times and contact durations of the original and bred traces from the RT1 mobility model.

the RT2 model in Fig. 6.4(b), we basically report the same observations as in the RT1 scenario. In more detail, Fig. 6.4(b) shows that both distributions are close and the longest contact lasts twice a measurement interval time (2X120). Moreover, the proportion of long contacts is greater in the bred trace than in the original one and conversely, the proportion of short contacts is higher in the original trace compared to the bred one.

Other Mobility Models. The evaluation below focused on the Random Trip model using a perfect sampling algorithm. But naturally, our breeding system works with other mobility models. We run an additional evaluation using the recently claimed best human mobility model, *SMOOTH*[65]. It is a realistic mobility model built on the basis of seven known features of real human movement. The model has been validated to mimic real human movement by comparing its synthetic traces against several real mobility traces For the validation process, various real traces were used : two campuses,



(a) Intercontact.

(b) Contact.

Figure 6.4: Intercontact times and contact durations of the original and bred traces from the RT2 mobility model.

a metro city, two outdoor sites, and two conference scenarios. Not only, the model was also compared to three other human mobility models SWIM, SLAW, and TLW[58;62;81].

The novelty SMOOTH brings is the guarantee of the seven statistical features found in real human walk patterns. The first, second, and third features specify that the distributions of mobile location distances (*"flight"*), inter-contact times, and pause-times follow a truncated power-law (TPL). Fourth, nodes are distributed non-uniformly in the network. Fifth, node movements are not random. Indeed, movement patterns can be predicted to some extent due to the some movement regularity. Sixth, the probability to visit a new location (*Pexplore*) is inversely proportional to the total number of locations the node visited so far. Last feature, the probability of visiting an old location is *1-Pexplore*.

The social human behavior is reflected in the simulation area as mobile nodes visiting few locations more often than others. Thus, creating communities that are represented as clusters. SMOOTH includes this feature as an input parameter. For our simulation, we use 4 clusters.

Table 6.2: SMOOTH trace configuration.

Model	SMOOTH	**Duration (days)**	3
Area (m^2)	500X500	**Clusters**	4
Range (meters)	25	(α, $\mathbf{f}_{min}$, $\mathbf{f}_{max}$)	(2.1, 5, 100)
Nodes	78	(β, $\mathbf{p}_{min}$, $\mathbf{p}_{max}$)	(2.1, 10s, 12h)

In SMOOTH, clusters do not have boundaries but instead each cluster has a single *landmark*. A landmark is an *(x,y)* coordinate position for a cluster. Over the simulation, every cluster landmark is placed uniformly satisfying the condition that no two landmarks are within transmission range R of each other. Initially when a node chooses a cluster in the network, it is placed within 0.5R of the chosen landmark. In our scenario, R value is set to 25.

It also considers two additional human mobility features. The distributions of mobile location distances (*"flight"*) and node pause-times follow a truncated power-law. They are translated in simulation in six parameters (α, f_{min} , f_{max}) for the flights distribution. and (β, p_{min} , p_{max}) for the pause-time distribution. We sum up all our simulation configuration in Table 6.2.

Nodes move on a 500m $\times$ 500m area at speeds proportional to the distance length (see[65] for more details). We simulate the network for a duration of 3 days. When we breed the original trace, we use a measurement interval of 60 sec in order to challenge our system.

As depicted in Fig. 6.2, to evaluate the correctness of our system, we run the breeding system using the same (ϕ, ρ) = (60, 25) of the original trace we chose.

Fig. 6.5(a) and 6.5(b) also show the CCDFs of intercontact times and contact durations for both the original and bred trace. For this more realistic mobility model, results distinctly show an even better (Random Trip) accordance between the original and the bred trace for both intercontact and duration times.

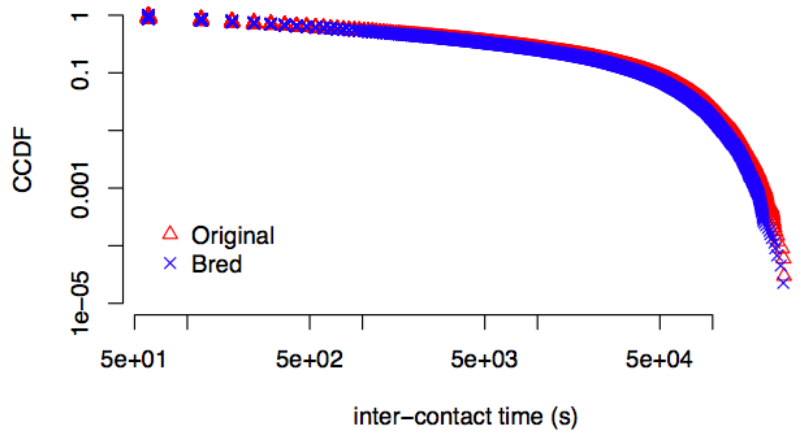
This section shows clearly the conformity of our proposed system for different measurement intervals even the more challenging (highest) intervals.

6.4. Real-life traces evaluation

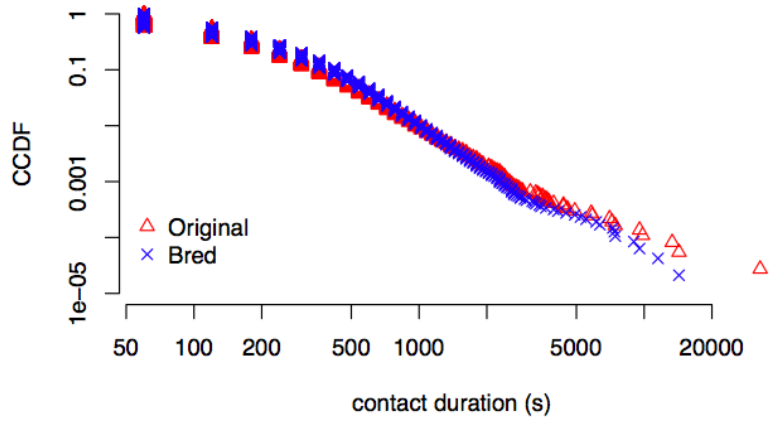
The perfect evaluation using real traces should be performed with contact traces extracted from real experiments using at least two different technologies to evaluate our parameters tuning step as well. For instance, it would have been ideal to have two contact traces. The first, collected using bluetooth and the other one with WiFi. Then, breed the bluetooth trace using the WiFi transmission range. The bred trace should be close to the WiFi contact trace (experimental).

As explained before such traces are not available. Still, we can compare the original contact trace against the bred trace using the same parameters to evaluate the difference with the original trace.

In this section, we focus on real-life contact traces. We choose contact traces derived from four real experiments Rollernet, Infocom, PMTR, and Stanford high (see Table 1). We picked these traces because of the difference in the measurement interval (frequency) and experiment duration.



(a) Intercontact.



(b) Contact.

Figure 6.5: Intercontact times and contact durations of the original and bred traces from the SMOOTH mobility model.

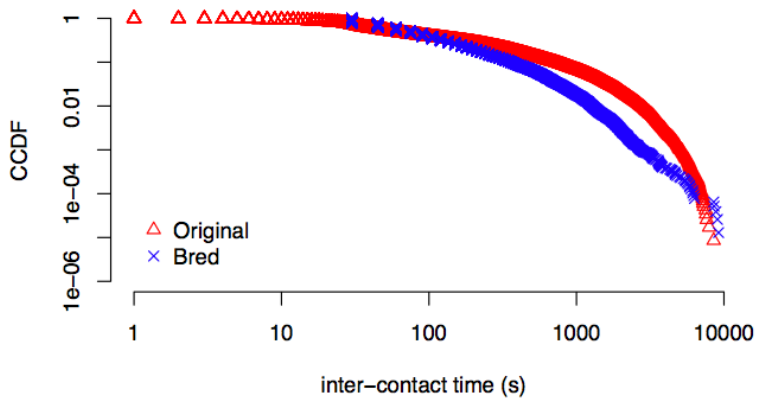
As we already checked the conformity of the bred traces, in this section we use all datasets to show that even with noisy traces, the bred traces still have the same characteristics as the originals.

6.4.1. Rollernet

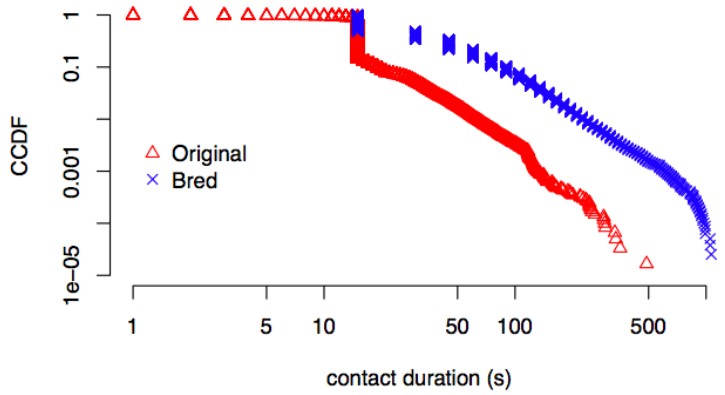
We decided to use the Rollernet dataset because it uses a reasonable measurement interval of $\phi = 15$s. The trace was collected from 62 rollerbladers carrying bluetooth iMotes for 3 hours around Paris.

As in the synthetic evaluation, we do not use any reference nodes to help better infer mobility. Still, the experiment involves a known head and tail nodes. The rest of the rollerbladers moved between the head and the tail during all the experiment. Nodes are highly mobile and present an average contact duration of 26s.

We configure the breeding system trying to mimic the original trace. As the Rollernet experience used Bluetooth technology for probing the neighborhood, we set $\rho = 10$m (approximately the Blue-



(a) Intercontact.



(b) Contact.

Figure 6.6: Intercontact times of the original and bred traces in Rollernet with $\phi = 15$s and $\rho = 10$m.

tooth transmission range). For the measurement interval, we directly apply $\phi = 15$s. The goal is to evaluate the bred trace behavior compared to the original. Fig. 6.6 shows the CCDF of the aggregated intercontact and contact durations.

In Fig. 6.6(a), both distributions have the same shape. We can see however that the original trace has shorter intercontact times (between 1s and 15s); this is due to the asynchronous nature of the scanning process in the experimentation, while the breeding system is synchronous (a requirement of plausible mobility). This forces our contact durations to be at least equal to one measurement interval, i.e., 15s.

Both traces have the same distribution between 15s and 700s. Then, the original trace shows longer intercontact periods than the bred trace. This happens because the inferring mobility algorithm detects more contacts than in the original one, thus reducing intercontact durations in average. This observation also appears in Fig. 6.6(b), where the contact durations of the bred trace are larger than the original one (>500s). This result can be explained by the inherent properties of a real environment,

which suffers from physical phenomena such as fading and signal interference. Although the bred trace tends to compute longer contacts than the original one, the results remain consistent.

For the rest of the datasets, we continue to use the same methodology as below to show the similarity between the original and the bred trace.

6.4.2. Infocom

The Infocom contact trace[10] was collected using Bluetooth Intel iMotes during three to four days (April 23-27, 2006) during the INFOCOM 2006 conference. The iMotes were distributed to 70 users moving between the three different levels of the conference area. Among the iMotes, 20 were static motes deployed within the same area.

We use this dataset because it has a large measurement interval of 2 minutes ($\phi = 120$s) which makes the task even harder for our breeding system. We configure the transmission range to $\rho = 100$m as according the experiment. Fig. 6.7 shows the CCDF of the aggregated intercontact and contact durations.

In Fig. 6.7(a), the original and bred trace follow the same distribution. Again, the original trace has shorter inter contact times (between 1s and 120s) due to the asynchronous nature of the the experiment scanning process. Thus, the shorter intercontact is at least equal to 120s. (one measurement interval).

But for this dataset, both contact traces have the same shape even in the extreme right of the figure. In other words, we do not notice a deviation such in the rollernet dataset. The same behavior also appears in Fig. 6.7(b) where the contact durations of the original and bred trace are practically similar from the left to the right of the figure. Except the starting point where the minimum contact duration in the bred trace is equal to the measurement interval as always (120s) for the reasons cited below.

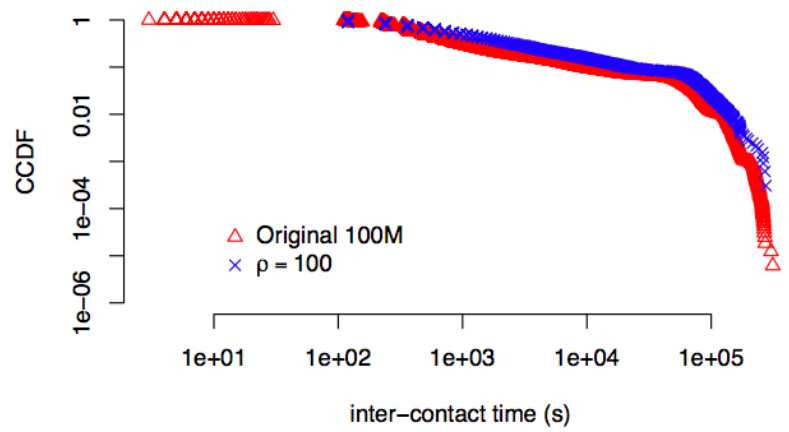
These results demonstrate that even for large measurement intervals that challenge the breeding system, the results stay consistent.

6.4.3. PMTR

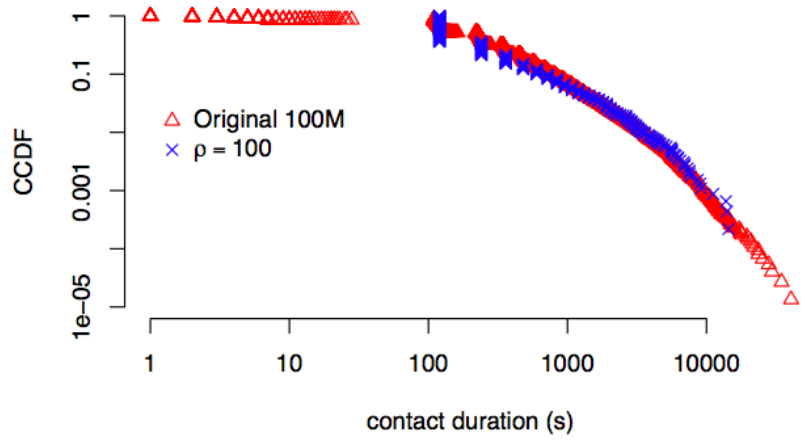
The need of finer contacts measurements motivated the design of a new device for recording short contacts that appears in dense mobility areas and are missed when using technologies such bluetooth. The Pocket Mobility Trace Recorder[63] is designed to run with measurement intervals starting from 1 second.

The dataset we use in this section is extracted from an experiment that have been run for 15 days in Novermber 2008 at University of Milano. It has the shortest measurement interval 1second. The 44 PMTRs were distributed to PhD students, faculty members, and technical stuff. The population is dispersed in laboratories and offices in a three-floor building of 200X100 m large and sometimes visits a nearby cafeteria for lunch or coffee breaks.

The dataset we use in this section is extracted from an experiment that have been run for 15 days in Novermber 2008 at University of Milano. It has the shortest measurement interval 1second. The



(a) Intercontact.



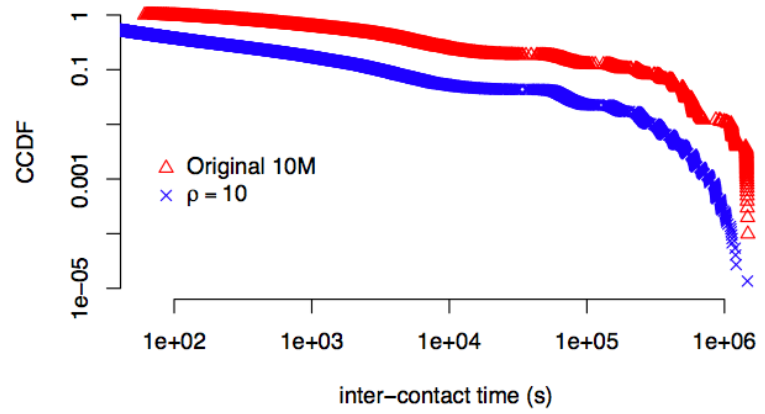
(b) Contact.

Figure 6.7: Intercontact times of the original and bred traces in Infocom with $\phi = 120$s and $\rho = 100$m.

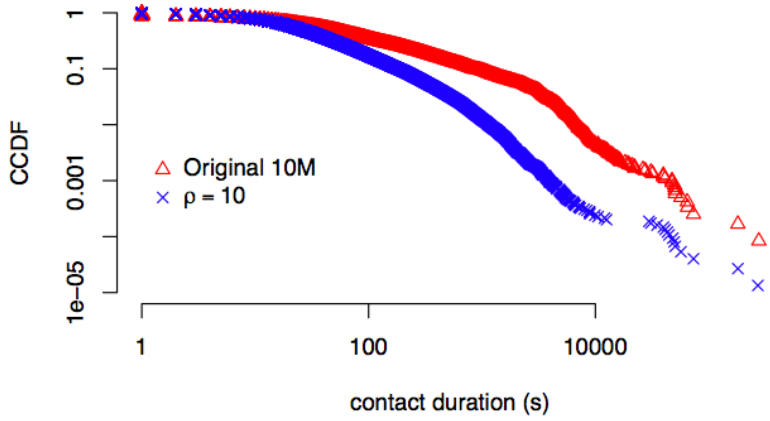
44 PMTRs were distributed to PhD students, faculty members, and technical stuff. The population is dispersed in laboratories and offices in a three-floor building of 200X100 m large and sometimes visits a nearby cafeteria for lunch or coffee breaks.

As for the previous datasets, we evaluate the deviation of the breeding system by comparing the original trace against the bred trace. We configure the measurement interval $\phi = 1$s and the transmission range to $\rho = 10$m to stay similar to the device. Fig. 6.8 shows the CCDF of the aggregated intercontact and contact durations.

In Fig. 6.8(a) both distributions have the same shape, but the difference remains in contact durations. In Fig. 6.8(b), we see that the bred trace has less long contacts than in the original one. Our explanation is that due the fine grain measurement of the PMTR dataset, the breeding system detects very easily short contacts and in opposite to the original trace which might consider multiple short contacts as a long one, our system make a clear difference between multiple short contacts and a single long one.



(a) Intercontact.



(b) Contact.

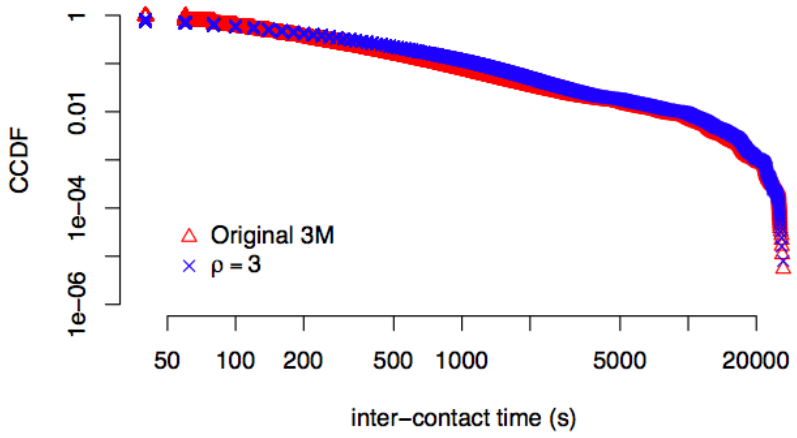
Figure 6.8: Intercontact times of the original and bred traces in PMTR with $\phi = 1$s and $\rho = 10$m.

An interesting observation from PMTR and Infocom results rises. When comparing the four figures, we note that the use of fixed nodes (anchors in Infocom) even with a high measurement frequency gives better results than low frequencies without anchors (PMTR).

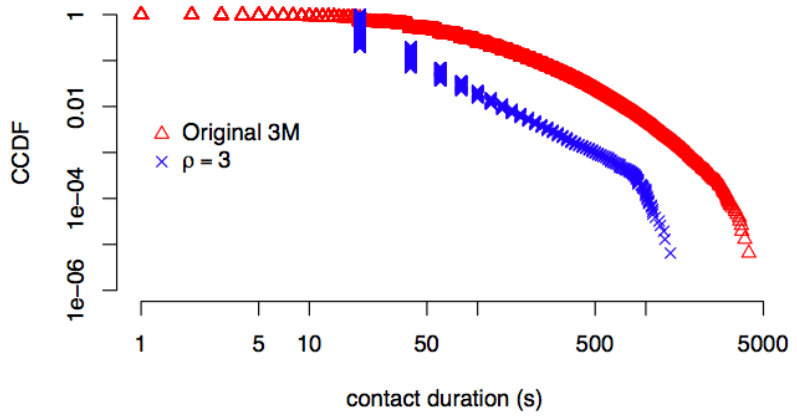
6.4.4. Stanford

A nine hours experiment (between 7AM and 4PM) was conducted in a US high school including 789 participants, among them were students, teachers, staff, and other volunteers. They measured contact opportunities each 20 second within a range of 3 meters in order to represent face to face contacts. We keep the same parameters for the evaluation.

Fig. 6.9 shows the CCDF of the aggregated intercontact and contact durations. Even if this experiment uses a small communication range (3M), we see an almost prefect match in the intercontact times (Fig. 6.9(a)) while keeping the same shape for the contact durations (Fig. 6.9(b)). Record that the bred trace starts at 20sec which translates into minimum contact duration equals at a least



(a) Intercontact.



(b) Contact.

Figure 6.9: Intercontact times of the original and bred traces in Stanford high with $\phi = 20$s and $\rho = 3$m.

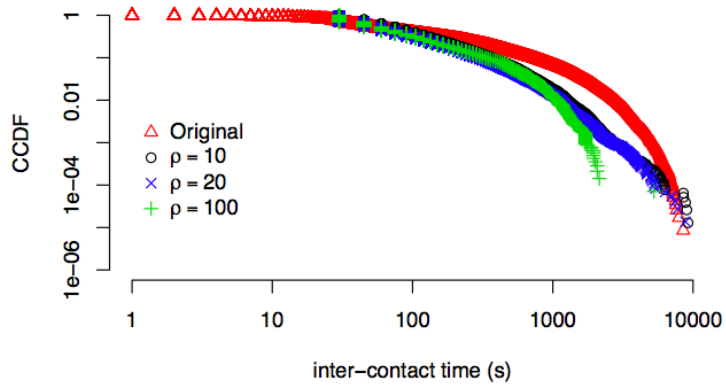
one measurement interval. We also record the same observation as in the Rollernet dataset for short contacts in the bred trace.

In this section, we show that even with the most challenging contact traces (high measurement interval, small communication range, no fixed nodes) and the imperfection of the real contact traces (losses, interference), the breeding system still produces contact traces strongly inspired from reality.

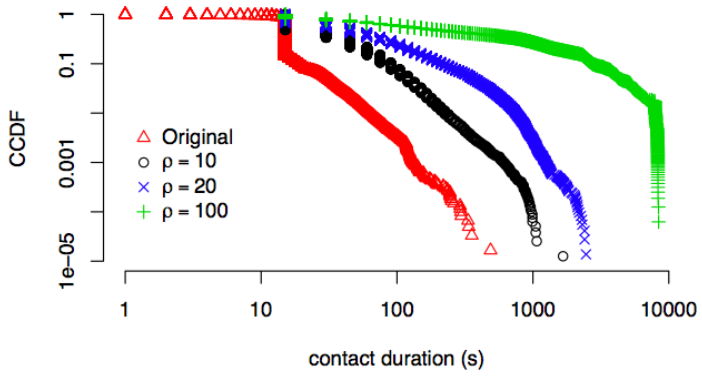
6.5. Use cases

In this section, we use Rollernet dataset as an example to show the valuable information that breeding contact traces can add to a single experimental trace.

Varying the transmission ranges. We compare now the original Rollernet trace against variants with different transmission ranges ρ. Fig. 6.10 shows the CCDF of the aggregated intercontact and contact durations over all nodes for the same ϕ and different values of ρ (10m, 20m, and 100m).



(a) Intercontact.



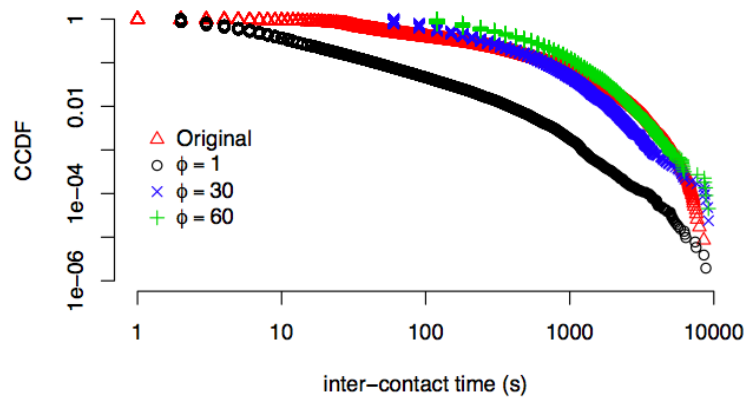
(b) Contact.

Figure 6.10: Intercontact times of the original and bred traces in Rollernet by keeping the same ϕ and varying ρ.

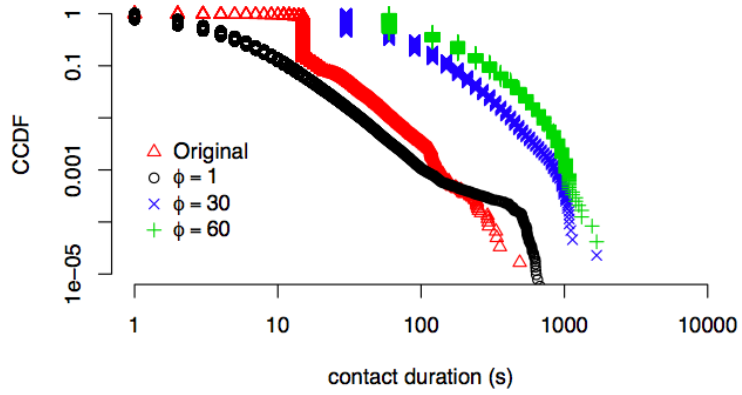
Fig. 6.10(a) clearly shows that, independently from the transmission range, all distributions follow a truncated power law. As expected, the longer the transmission range, the shorter the intercontact times. What is interesting to note here is how intercontact time relates to the transmission range. For instance, if we use a technology that covers an area within 100m, we reduce the maximum intercontact time by almost 2 hours if compared to the case where the transmission range is 20m.

Another interesting takeaway with this dataset is shown in Fig. 6.10(b). We see that, for $\rho =$ 100m, the maximum contact duration reaches the full duration of the experiment. In other words, some pair of nodes would have likely been in contact all the time if the experiment designers had chosen a technology with this transmission range. But if we go back to Fig. 6.10(a), we see that it makes only a little difference in terms of intercontact times. In conclusion, there would be a limited gain in investing in a technology that gives larger coverage.

Varying the measurement intervals. Finally, we compare the original Rollernet trace against variants with different measurement frequencies ϕ. The goal is to investigate if the measurement interval has any impact on the contacts. Fig. 6.11 shows the CCDF of the aggregated intercontact and contact durations over all nodes for the same ρ and different values of ϕ (1s, 30s, and 60s). As explained



(a) Intercontact.



(b) Contact.

Figure 6.11: Intercontact times of the original and bred traces in Rollernet by keeping the same ρ and varying ϕ.

previously, smaller values of ϕ give the most accurate results. Even though, our results show that, even for high values of ϕ, we can always generate reliable bred traces.

6.6. Summary

We proposed a *breeding* system able to infer as many traces as we wish from one single real experiment. The main advantage of our system is to avoid all the challenges of setting up a real experiment. Our approach offers the ability to tune measurement parameters (transmission range, measurement interval), allowing obtaining new insights into the network that would have been possible only with a new experimental campaign.

Our results show that, despite the constrains of the inferring mobility algorithm, our system can give traces that are compliant with the original dataset. When using synthetic and trace-based contact traces, the system offers very similar resulting traces. In particular, even with noisy and imperfect real traces, the bred traces follow the characteristics of the original traces.

Chapter 7

Conclusion and Perspectives

HIGH speed internet access, diversity of personal user devices, and high quality contents are factors in constant development leading to more and more exigency from customers. To guarantee a large bandwidth, low delays, and continuos connectivity to users, it is crucial to have a good understanding of the network. From a restricted area such as home networks to larger and more dynamic networks like wireless networks, the challenges vary but user exigency does not. Whatever the resources required for the desired content, the user will always go towards the best service guarantee at home or in the street. Therefore, it is not only important to understand the dynamic of the network but also to be able to diagnose and optimize the performance of the network in an automatic way to satisfy users.

7.1. Summary of contributions

This thesis develops a methodology to automatically diagnose and optimize network application using the minimum gateway resources on one hand and helps understanding network dynamics using the minimum available real datasets on the other hand.

Investigating local traffic dynamic. We compare the local traffic in different network environments instrumenting end-hosts. Relying on the end-host perspective allowed to gather traces from one hundred different networks. We use the dataset to separate local from wide-area traffic and compare the composition of local traffic through different networks. Our dataset involves 47 users who ran *Hostview* for at least a week. Our results show that, in general, wide-area traffic is more dominant than local traffic, even if in some networks we do not observe any local traffic (as in airports).

In home and work environments, we observe a non-negligible amount of local traffic. Moreover, the composition of local traffic at work is split between network file systems and backup, which are not present in home traffic. The lack of backup traffic at home may reflect users' preference to backup directly at external disks when at home. Still, the composition depends on the user and the network.

Application performance optimization system design. Applications in a home network access the Internet by sharing a single access link, thus competing for resources. In this thesis, we propose an application performance optimization system composed of different modules. Each module is responsible for a different and complementary network resource optimization task. The system is designed to run from the heart of the home network, the home gateway.

We discuss both resource limitations and the implementation design choices. Then, we detail the functioning of our modified version of the home gateway running BISMark firmware. Our results show that, even with extra overhead, there is still resources left for running a traffic collection process along with an application identification strategy. Using real collected data, we show that it is feasible to run an application performance degradation technique from the home gateway.

Mobility trace breeding. The best methodology to validate any research approach is to prove its reliability in a real environment. As only few real traces are publicly available for researchers, we had the idea to develop a mobility trace *breeder*. In this approach, we only need a single real trace to derive multiple others strongly inspired from the real trace. Our results indicate that for both synthetic and real datasets, the bred traces do follow the characteristics of the original trace. We use Rollernet to show some use cases. We play with the system parameters in order to change the used technology and the measurement frequency. Then, we extract valuable information from the range of bred traces we derived. For instance, if we use a technology that covers an area within 100m, we reduce the maximum intercontact time by almost 2 hours if compared to the case where the transmission range is 20m. Such information is important for content dissemination for example.

7.2. Future research directions

In this thesis, we investigate underlying phenomena in both domestic and mobile networks. The following explains future research directions from each contribution and also from the global scope of the thesis.

Measurement efforts. The plan is to collect data directly from home gateways to measure all traffic from a single home over a longer period of time. We work with the developers of Bismark (http://projectbismark.net/) to collect passive traffic measurements as well. The final goal would be to improve user perception. We believe that our system combined with end-host measurements would be an efficient solution to achieve this goal. In fact, working on the correlation between user perception and application is complementary to our system and could be used to improve user experience.

Application performance optimization system. The next step would be to build the complete system that controls and avoids performance degradation in home networks. Here, we need to perform another resource consumption study for the traffic control box on the home gateway. An additional

idea for the future is to integrate our home performance optimization approach with a prediction technique for user (dis-)satisfaction

Mobility traces breeding. We have identified several directions for further research. First, we need to dig into the role of plausible mobility on our system and improve it and reproduce more faithfully the original network, especially in the case of low measurement frequency. Also, the results we had are specific to each dataset and are by no means a general rule. In a future work, we intend to provide results for a much larger set of real-world datasets, and then try to derive more general conclusions. At last but not least,it would be interesting to introduce interference and loss models in our systems, for instance the Gilbert-Elliott Model for packet loss[32] to get even closer to the real environment.

Combination of results. The combination of the work from the home to the mobile networks would rise interesting research questions. Given the little (public) knowledge about the application mix in mobile areas in general, it is interesting to use our application characterization method on mobile devices from home networks to characterize applications in the context of mobile networks. The results would indicate the top used applications. Then, in order to ensure the best performance for each application, we would employ our *breeding* system to represent the different technologies to rely on and assess their individual performance. The final outcome would be the best technology to use on a per-application basis.

Appendices

Annexe A

Résumé de la these en français

COMPRENDRE les phénomènes sous-jacents dans les réseaux est fondamental pour le déploiement de protocoles réseaux efficaces, stratégies ou même matériel. La connaissance de la dynamique des réseaux est utile pour tester des solutions ou prendre des décisions sur l'évolution du réseau. Par exemple, Facebook est largement étudié étant donné la forte dynamicité de ce réseau social. Une étude de cas intéressante analyse l'effet des utilisateurs de Facebook sur la dynamique de réseau [1]. Dans des domaines tels que les réseaux tolérants aux perturbations (DTNs), les protocoles doivent être testées dans un réseau dynamique et donc, exigent la compréhension de la dynamique des utilisateurs. Un autre exemple est les réseaux résidentiels (réseaux domestiques). Comprendre la dynamique des applications dans les réseaux domestiques est essentiel pour les fournisseurs de services afin d'offrir le meilleur service possible à leurs clients.

The last decade has seen an outstanding increase in connected devices[37]. People are practically always connected, inside their homes, at work, or even in the streets. Many devices offer multiple services to users, from fixed PC's to laptops and smartphones. Often, each user has at least one personal laptop, smartphone, or computer. Additional network devices also share the Internet such as home gateways, routers, and WiFi access points. Each device comes in variety of platforms and brands. These connected devices constitute a more and more complex network. Devices within the same area (home, work, street) can share a single access to the Internet. If users face poor Internet connectivity, it is hard to determine which part of the network is responsible for the performance degradation. Worse, it is even not always possible to find out the culprit. Therefore, it is crucial to understand the dynamic of the network at first and have the ability to diagnose poor network performance thereafter or retroactively, make design choices.

This thesis addresses such problems in two main areas. Domestic and mobile networks. In home networks, we study application dynamics that helps the design, the monitoring, and the service level guarantee for a network. In mobile networks, we concentrate on nodes mobility and more specifically mobility traces dynamics to support protocol design and testing.

La dernière décennie a vu une augmentation remarquable dans les dispositifs connectés[37]. Les gens sont pratiquement toujours connectés, à l'intérieur de leur maison, au travail, ou même dans

FIGURE A.1 – Exemple de réseau domestique.

les rues. De nombreux dispositifs offrent de multiples services aux utilisateurs, à partir du PC fixe, des ordinateurs portables ou des smartphones. Souvent, chaque utilisateur dispose d'au moins un ordinateur portable personnel, smartphone ou un ordinateur. Les autres appareils réseaux partagent également l'Internet tels que les passerelles à domicile, les routeurs et les points d'accès WiFi. Chaque appareil est livré dans une variété de plates-formes et de marques. Ces appareils connectés constituent un réseau de plus en plus complexe. Des dispositifs dans la même zone (domicile, travail, rue) peuvent se partager un seul accès à l'Internet. Si les utilisateurs font face à une mauvaise connexion Internet, il est difficile de déterminer quelle partie du réseau est responsable de la dégradation des performances. Pire, il n'est même pas toujours possible de trouver le coupable. Par conséquent, il est crucial de comprendre la dynamique du réseau dans un premier temps et avoir la capacité de diagnostiquer une mauvaise performance du réseau par la suite ou rétroactivement, à faire des choix de conception.

Cette thèse aborde ces problèmes dans deux domaines principaux. Réseaux domestiques et mobiles. Dans les réseaux domestiques, nous étudions la dynamique des applications qui contribue à aider à la conception, le suivi et la garantie d'un bon niveau de service pour un réseau. Dans les réseaux mobile , nous nous concentrons sur la mobilité des nœuds et plus particulièrement la dynamique des traces de mobilité pour soutenir la conception de protocol et leurs tests.

A.1. Espace du problème

A.1.1. De l'analyse au diagnostic des réseaux domestiques

Différentes technologies (par exemple WiFi) permettent aux utilisateurs dans un ménage de connecter un certain nombre de dispositifs à l'intérieur du réseau domestique et d'accéder à Internet en utilisant un lien d'accès unique. Figure A.1 montre un exemple de réseau domestique. Dans ce contexte, tous les périphériques connectés à au réseau domestique se partagent la capacité de la liaison accès

à Internet et du réseau local. Par conséquent, les applications et les services fonctionnant sur ces appareils peuvent interférer les uns avec les autre. Par exemple, un utilisateur peut jouer à un jeu en ligne, tandis qu'un autre commence un gros téléchargement de fichier. Étant donné que les deux applications nécessitent une bande passante, il peut y avoir un effet négatif sur la performance que les utilisateurs perçoivent. Dans un tel cas, il est difficile pour l'utilisateur de déterminer où est le problème. Résoudre une dégradation de performances est difficile pour les utilisateurs même experts et vague pour le reste des autres[50]. Les simple utilisateurs du réseau domestique appliquent des stratégies de solutions simples pour résoudre leurs problèmes de connectivité , comme débrancher et rebrancher leurs appareils ou redémarrer la machine. Mais les solutions parfois simples ne suffisent pas. Par exemple, en cas de mauvaises performances sans fil en raison d'interférences entre appareils mobiles, le redémarrage ou la mise OFF/ON ne réglera pas le problème. Si aucun des utilisateurs n'a un minimum de connaissances de la technologie, le problème restera sans solution jusqu'à ce qu'une main externe soit appelée. De nombreuses études ont été faites autour de la performance de la liaison d'accès[19;106] mais jusqu'ici seuls quelques-uns sont orientés vers la performance des réseaux domestiques.

Pour améliorer l'expérience des utilisateurs sur Internet, les réseaux domestiques devraient avoir des solutions automatiques qui ne nécessitent pas l'intervention de l'utilisateur. Ces solutions devraient automatiquement diagnostiquer le réseau domestique et optimiser sa performance afin d'éviter une mauvaise expérience de l'utilisateur. Mais avant, il est important de mesurer et de caractériser la dynamique du réseau.

De nombreuses techniques existent pour mesurer la topologie de l'Internet et la performance, mais on en sait peu sur les réseaux domestiques. Même si les réseaux domestiques reçoivent de plus en plus d'intérêt à travers les années, peu de connaissance est disponible sur les caractéristiques du réseau domestique[9;15]. Par exemple, quels sont les dispositifs typiques à la maison ? Qu'est-ce que les utilisateurs aiment faire dans le réseau domestique ? Quels sont les services les plus populaires accessibles à partir de la maison ? Des travaux antérieurs utilisant des sondes actives à partir d'hôtes terminaux ou serveurs sur Internet[30;59;91**?**] pour mesurer et caractériser le lien d'accès. L'absence d'études de réseaux domestiques est due à différents défis tels que les défis liés à la collecte de données ou à la vue incomplète du trafic du réseau domestique en utilisant des méthodes de sondage actuels.

Vue partielle du trafic par les hôtes terminaux. Les utilisateurs impliqués dans une étude de trafic sur le réseau domestique sont généralement invités à installer un outil sur leurs équipements terminaux ou de configurer un tierce périphérique de leur réseau domestique afin de collecter le trafic. Le problème avec les hôtes d'extrémité, c'est que ce sont souvent des machines personnelles et peuvent être déconnectés du réseau domestiques pendant un certain temps. Un tel comportement réduit la mesure et modifie les données. En outre, les hôtes d'extrémité n'ont qu'une vue partielle de la circulation du trafic, ce qui limite les études avec l'ensemble des données recueillies. Pour avoir une connaissance complète du trafic, une solution consiste à installer l'outil sur tous les hôtes d'extrémité au sein du même réseau domestique, ce qui signifie que la coopération de tous les membres du ménage

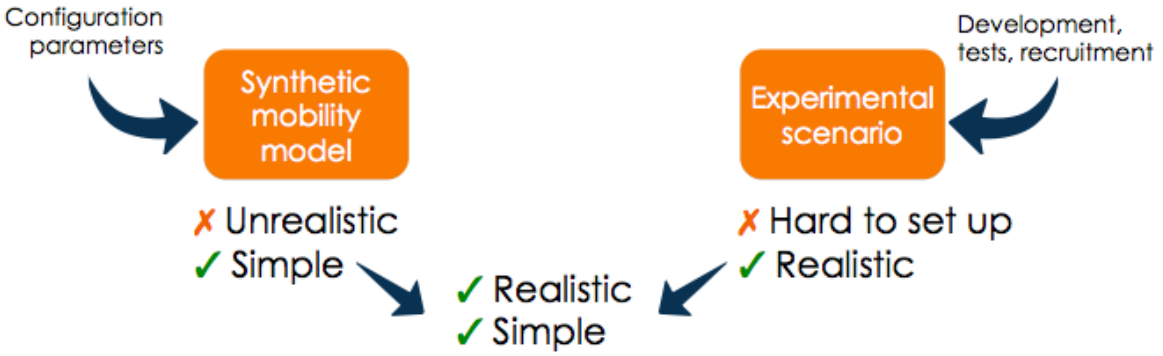


FIGURE A.2 – Avantages de reproduction de trace de mobilités

est nécessaire. Cette solution est faible et peu réaliste, car un seul nœud déconnectée peut fausser tout le jeu de données.

Dans cette thèse , nous mesurons le trafic à l'intérieur du réseau domestique à partir de la passerelle domestique afin d'avoir l'image complète.

A.1.2. D'unique à de multiple expériences de mobilité

Les communications fiables dans les DTNs nécessitent une compréhension approfondie de la dynamique des réseaux. Nous sommes entourés par des dispositifs de communication : téléphone, les points d'accès, des véhicules et ainsi de suite. Le défi supplémentaire que ces dispositifs ont apporté est une certaine mobilité. Parce que les nœuds mobiles sont soumis à des interruptions imprévisibles, les protocoles qui ont été conçus pour les réseaux câblés tels que TCP/IP qui nécessite une connectivité de bout en bout ne résiste plus. C'est pourquoi la compréhension de la dynamique des réseaux mobiles est essentielle lors une nouvelle conception ou de test de protocoles .

Les réseaux mobiles peuvent être très dynamique et souffrir de connexions intermittentes. La propriété instable des nœuds dans ces réseaux conteste la convergence des protocoles. Un exemple est l'expérience Rollernet[98]. Des capteurs Bluetooth ont été distribués à un groupe de patineurs lors d'une tournée autour de Paris. Même si les noeuds sont proches en raison de la technologie utilisée, des interruptions se sont produites entraînant de nombreux nœuds connectés par intermittence. Mais rollernet est seulement l'une des rares expériences réelles disponibles pour la communauté. En effet, il existe un besoin réel de traces réelles à utiliser pour les tests de protocol ou de validation.

Dans cette thèse, nous nous intéressons aux possibilités de contact résultant de réseaux mobiles de manière intermittente (ICMN). La construction et l'exploitation des expériences fiables pour mesurer les possibilités de contact dans les ICMNs est un processus long et difficile. L'objectif final de ces expériences est d'obtenir une trace de contact que nous pouvons analyser pour mieux comprendre la dynamique du réseau. Seulement quelques traces de contact réelles sont accessibles au public, il serait intéressant de réfléchir à la réutilisation des traces existantes pour obtenir plusieurs autres. La figure A.2 reprend l'avantage de la méthode que nous proposons, qui combine les avantages de l'expérimentation réelles à la fois et la méthode synthétiques. Nous nous concentrons en particulier sur la capacité de reproduire ces expériences tout en évitant tous les défis d'expériences réelles.

A.2. Contributions

Cette thèse apporte les contributions suivantes aux mesures de réseaux domestiques et mobiles, leur caractérisation et modèles. Pour la première partie :

1. Nous présentons une comparaison du trafic local dans différents environnements réseaux à l'aide d'hôtes d'extrémité. Nous faisons notre étude avec des traces de 47 utilisateurs qui ont installé l'outil de collecte (Hostview) pendant au moins une semaine, 32 d'entre eux l'ont eu pendant plus d'un mois. Les hôtes ont été connectés à partir de 185 réseaux uniques répartis sur 18 pays où 34 réseaux ont été résidentiels en plus d'environnements de travail 38 en tout (universités et entreprises). Nos résultats montrent une grande diversité dans le trafic local, néanmoins le trafic externe domine. Dans les réseaux tels que les aéroports et les cafés, le trafic local est rare. On observe quelque trafic DNS.

2. Nous analysons le trafic dans les résidentiel et professionnels séparément et comparons leur dynamique en termes d'applications. Dans les deux réseaux, nous constatons un trafic local non négligeable avec la plupart des connexions étant courtes. Mais parfois, les connexions locales transférent un grand nombre d'octets. Outre DNS, nous constatons que les applications typiques sont les systèmes de fichiers réseau et de sauvegarde. Cependant, la composition du trafic local dépend de l'utilisateur et le réseau.

3. Nous concevons un système d'optimisation de la performance qui opère à partir de la passerelle domestique, afin d'éviter la dégradation des performances de toutes les applications actives dans un réseau domestique. Nous proposons une stratégie en deux étapes. Tout d'abord, suivi de la performance des applications. Ensuite, l'identification des applications en temps réel.

4. Avec notre version modifiée de la passerelle domestique, nous montrons que, même avec une génération d'une charge supplémentaire, les résultats sont prometteurs pour l'utilisation d'un processus de collecte de trafic avec une stratégie d'identification de la demande. Aussi, nous vous proposons une techniques *légère* d'identification d'application. Suivant nos directives, les résultats montrent qu'il est possible d'exécuter une technique de suivi des performances de l'application à partir de la passerelle domestique.

Dans la seconde partie de cette thèse :

1. Nous proposons un système de *multiplication* de traces de contact possibles à partir d'une seule expérience réelle. Nos évaluations synthétiques indiquent que les traces d'origine et multipliés suivent les mêmes caractéristiques. Aussi, en utilisant une variété de données réels, nous montrons que notre système produit des traces de contact fortement inspirées des traces originales.

2. Nous utilisons un exemple de données réelles en particulier, pour montrer des observations de valeur sur le réseau que notre méthode de reproduction déduit et qui aurait été possible seulement avec une nouvelle campagne expérimentale. Nous expliquons les orientations nécessaires pour extraire ces informations.

A.3. Plan de thèse

Cette thèse est organisée en deux parties comme suit. La première partie comprend : Chapitre 2 définit le problème de la dégradation de performance des applications et présente les recherches connexes dans le domaine. Chapitre 3 montre la dynamique des applications dans les réseaux domestiques. Chapitre 4 étudie la faisabilité de suivi de la performance des applications du réseau à partir des passerelles domestiques. Puis, dans la deuxième partie : Chapitre 5 introduit le problème de reproductibilité de traces de contact et de parle du travail préalable. Chapitre 6 étudie la reproductibilité des traces de contact. Nous concluons au Chapitre 7. Dans ce qui suit. Nous allons donner un apreçu des principaux chapitres.

A.4. Dynamique des applications dans les réseaux domestiques

Les deux dernières décennies ont vu de nombreuses études qui caractérisent le trafic Internet[8;56;59;97]. Ces études sont basées sur des traces de paquets collectés dans les réseaux de FAI, routeurs de bordure du campus de l'université ou des réseaux d'entreprises. En tant que tel, la plupart des études antérieures se concentrent sur le trafic *wide-area*. Peu est connu sur le trafic qui reste à l'intérieur d'un réseau, que nous appelons le trafic *local*. La principale exception est l'étude du trafic d'une entreprise[68;72], ce qui montre que le trafic local est différent de celui dans le wide-area avec une quantité importante de service de noms, de système de fichiers réseau, et de trafic de sauvegarde. Comme les auteurs le soulignentleur étude est "un exemple de ce à quoi peut ressembler le trafic de l'entreprise"[72]. Il est essentiel de réévaluer cette analyse dans d'autres entreprises et plus important dans d'autres types de réseaux de bord. Par exemple, la propagation de l'Internet haut débit a provoqué une augmentation du nombre de ménages qui ont un réseau domestique. Pourtant, il n'y a eu qu'une analyse limitée des volumes de trafic locales dans trois réseaux domestiques[48], mais aucune caractérisation approfondie des habitudes des utilisaturs dans la maison n'a été faite. Le défi de l'étude sur le traffic local à travers de multiples réseaux est d'obtenir des mesures à *l'intérieur* de ces réseaux.

Cette partie caractérise le trafic du réseau local de plusieurs réseaux de la perspective d'utilisateurs finaux qui se connectent à l'intérieur d'un réseau de bord. Cette approche est en contraste avec des travaux antérieurs[48;72], qui instrumentens les routeurs dans le réseau local. Bien que l'instrumentation des routeurs pourrait capturer tout le trafic traversant le réseau local, il est difficile d'avoir accès aux routeurs de plus que quelques réseaux. En surveillant le trafic directement aux hôtes finaux, nous pouvons analyser un plus grand nombre de réseaux, mais nous pouvons voir le trafic qu'avec la vision dede l'hôte que nous avons utilisé pour mesurer le réseau. Pour les réseaux plus petits (comme les réseaux domestiques) le trafic d'un hôte unique capte une part importante du trafic total, alors que pour les grands réseaux (comme les entreprises), cette fraction est moins significative.

Les réseaux d'accès à Internet et à domicile ont reçu un grand nombre d'attention ces derniers temps de la communauté de recherche, les organismes de réglementation et les FAI. les passerelles

fournissent une connectivité Internet sert à plusieurs fins telles que la téléphonie, médias en streaming, les données, ou les jeux. Dans cette partie de la thèse, nous analysons le trafic réseau à domicile et montrons comment suivre la performance dans ces réseaux.

Les organismes de réglementation intéressés à comparer la vitesse de liaison d'accès offert par les FAI avec ce qu'ils offrent. FAI face à la bande passante de plus en plus de nos jours et retardent également des demandes des utilisateurs. De nouvelles applications et de dispositifs contribuent à plusieurs exigences et les défis que posent les réseaux domestiques. Cette tendance rend le dépannage et la surveillance des réseaux domestiques fondamentaux à comprendre leurs problèmes et de défis. Puisque le contrôle de la maison à partir d'un dispositif d'extrémité est limité en termes de ce qui peut être surveillé, des projets tels que SamKnows (Royaume-Uni et États-Unis) comptent sur mesures actifs de la passerelle domestique. Même si la mesure à la passerelle est précieux, il est également consommatrice de ressources et peut interférer avec les besoins en ressources des utilisateurs. Pour cette raison, nous explorons la faisabilité de mesures passives des passerelles résidentielles.

Notre approche apporte une aide automatisée à l'aide d'un matériel déjà existant dans tous les ménages (de passerelle domestique). Comme les réseaux domestiques sont plus complexes de nos jours, la performance est une préoccupation de nouveaux utilisateurs, en particulier pour la gestion des médias numériques. Une étude a été faite sur le ménage numérique, il souligne l'importance de la gestion de services de médias, de l'espace disque et aussi pour maintenir l'ordre lors du redimensionnement des réseaux domestiques. les routines internes doivent être bien respectés, compte tenu du fait qu'ils changent d'une maison à l'autre et au sein de la même maison .

Nous pouvons trouver différents outils de mesure de la bande passante sur Internet tels que BWMeter, Netmeter pour les ordinateurs individuels, ou Speedmeter Pro pour la visualisation multi-ordinateurs. Malheureusement, aucun d'entre eux offre une interface d'affichage central pour la gestion de l'ensemble du réseau. A cet effet, d'autres outils ont été élaborés pour aider les utilisateurs à domicile de gérer et de dépanner leur réseau . Les outils sont dans un appareil public distinct qui permet à quiconque de gérer n'importe quelle machine du réseau.

Afin de garantir aux utilisateurs une bonne expérience au sein de leur réseau domestique, il est essentiel d'optimiser la consommation des ressources entre tous les périphériques du réseau. Nous considérons que les trois étapes de base pour l'optimisation de l'utilisation du réseau. Nous parlons des efforts de recherche dans le domaine de la mesure et de mise en forme des réseaux domestiques. Ensuite, nous nous concentrons sur l'étape de surveillance d'expliquer en détail où notre travail entre en jeu.

Le trafic wide-area mesuré à partir de l'intérieur du réseau a été analysé sous différents angles au cours des dernières décennies. Ces mesures cependant, ne peuvent pas capturer le trafic local au bord. Dans cette thèse, nous analysons le trafic local et comment il se compare avec un trafic étendu en utilisant des données recueillies directement à la fin hôtes utilisant HostView. D'autres études ont collecté et analysé des données de fin hôtes similaires dans le passé. En particulier, Giroire et al . ont comparé le trafic réseau des hôtes d'extrémité à travers trois environnements de réseau (à l'intérieur de l'entreprise, VPN à la société, et à l'extérieur de l'entreprise). Différente de la nôtre, leur étude n'a

pas caractérisé le trafic de réseau local en profondeur et bien qu'il mesure les ordinateurs portables d'un grand nombre d'utilisateurs que HostView mesurées, ils sont tous les employés d'une même entreprise .

Plus proche de notre travail sont les études d'un réseau d'entreprise et de trois réseaux domestiques . Ces instruments d'études avant le réseau local pour recueillir des traces de paquets et peut donc la différence entre la zone piétonne large locale et . Résultats expérimentaux existants se concentrent principalement sur les performances réseau , pas sur la caractérisation de la circulation . Leurs quelques résultats de la caractérisation de la circulation montrent que le trafic étendu domine le trafic local dans les trois maisons , mais il ya des pointes , rares trafic local . Dans cette thèse , nous étudions des mesures d'au moins une extrémité de l'hôte (ou tout au plus un couple) dans chaque réseau et par conséquent ne peut pas avoir une telle vue d'ensemble de chacun des réseaux étudiés.

Plus tard, nous comparons l'analyse dans l'étude de l'entreprise avec notre analyse sur le trafic dans les environnements d'entreprise. Étant donné que le trafic Internet peut varier de manière significative entre les sites et au fil du temp, notre étude contribue à montrer la diversité des modèles de trafic dans différents environnements réseau .

Des projets comme NetAlyzer, HostView et Atlas RIPE visent à comprendre réseau performance à la maison. Pourtant, ces outils basés fin d'accueil souffrent de la non-observabilité des activités d'autres dispositifs existants à l'intérieur de la maison qui peut biaiser les résultats. Pour surmonter ces problèmes, Calvert et al . proposer de mesurer à partir de la passerelle domestique. Ils signalent par des essais préliminaires de trafic de maison typiquement capture ce que de temps en temps, leur méthodologie conduit à taux de perte de 10% sous une charge lourde. SamKnows déploie passerelles résidentielles afin de mesurer plusieurs fois sur la performance de la liaison d'accès. Au meilleur de notre connaissance, il n'y a pas de travail qui explore systématiquement la faisabilité de la surveillance passive sur les différentes passerelles résidentielles.

Calvert et al. souligner la nécessité d' un enregistreur de données de réseau domestique (HNDR) pour permettre une compréhension plus détaillée et le dépannage des réseaux domestiques. Leur travail est le premier à proposer des mesures passives à une passerelle domestique. Ils fondent leur concept sur la NOXbox. Dansnotre travail, nous testons la performance d'une telle boîte. Pour une charge lourde cas (deux téléchargements P2P , le streaming Hulu , un et deux téléchargements Youtube) de débit non spécifié, ils signalent tcpdump baisse jusqu'à 10% des paquets lors de l'enregistrement sur le disque. Malheureusement, ils ne rendent compte de l'utilisation du système, ni varient systématiquement la charge de travail. Dans le chapitre suivant, nous jouons avec différentes charges de travail et l'utilisation du système de rapport.

Le lissage du trafic comprend différents outils tels que Trickle taux limite les connexions TCP d'un processus ou groupe de processus. Un autre outil est WonderShaper, un logiciel de mise en forme de trafic qui fournit une faible latence pour le trafic interactif, permet la navigation sur Internet à des vitesses raisonnables pendant le chargement/déchargement, et veille à ce que uploads / téléchargements ne blessent pas l'autre.

Lorsque l'on considère le trafic Internet, nous pouvons séparer en deux zones de circulation différentes. La plupart des recherches portent sur le trafic au niveau des routeurs de frontière ou entreprises pour la simplicité de mesure. Nous appelons cela le trafic dans une vaste zone. Mais il n'y a que peu d'intérêt dans le trafic qui reste à l'intérieur du réseau local du trafic en raison de problèmes de mesure. Nous détaillons les efforts dans ce domaine dans la suite.

Un problème récurrent concerne la gestion de la bande passante. Pour ce faire, des outils ont été élaborés pour aider les utilisateurs à domicile de gérer et de dépanner leurs réseaux. Nous les classons en fonction de leur localisation dans le réseau .

Basé sur dispositifs finaux. Home Watcher, par exemple , est un outil interne de gestion de la bande passante . Il montre utilisateur à domicile consommation de bande passante avec une option permettant de limiter l'utilisation de la bande passante et par personne (jusqu'à 20% de la capacité totale pour éviter des limitations sévères) . Après son procès dans six maisons (24 personnes) pendant 8 semaines , les utilisateurs répondent aux questions sur l'expérience de tout le monde dans la maison . L'outil a été non seulement apprécié pour sa capacité à contrôler l'utilisation de la bande passante , mais aussi pour permettre aux parents de connaître les activités de leurs enfants à un moment donné . Par exemple , au travail ou l'heure du coucher , si l'utilisation de la bande passante n'est pas peu de parents savent que les enfants jouent avec leurs ordinateurs à la place. Pourtant, il a soulevé la question de savoir qui décide à la maison et la gravité sont les limites à appliquer. Les résultats varient en fonction de la composition sociale dans un foyer . Par conséquent, un système de limitation automatique est préférable. En contraste avec l'ouvrage cité ci-dessus, des solutions plus automatiques sont proposées.

HomeMaestro est un système basé sur l'hôte qui surveille les performances des applications locales et globales et détecte lice pour ressource réseau automatiquement. Il utilise par flux et par les statistiques de processus (débit, RTT, taux de perte) pour détecter les flux en compétition pour la même ressource. Dans certains cas , le système est capable de détecter jusqu'à 85% des problèmes, où la grande majorité est causée par des demandes concurrentes sur des hôtes . Cette technique souffre de la façon dont le trafic est capturé , il utilise deux moniteurs sans fil pour capturer le trafic sans fil qui crée des interférences et entraîne donc le trafic manquante. Pour cette raison, nous croyons que la passerelle domestique est un meilleur candidat pour effectuer ces types de tests .

Basé sur des routeurs. Un autre outil inspiré de la maison Watcher est Kermit. Il a le même objectif mais recueille des données à partir d'un routeur flashé avec DD-WRT. Il est livré avec une interface qui comprend qui est en ligne et qui est monopolisant la bande passante et une image de tous les appareils connectés attachés à un nuage (Internet). En particulier, les utilisateurs apprécient le fait qu'ils peuvent personnaliser avec des photos et des noms réels pour identifier l'autre. Il fournit une option supplémentaire qui peut hiérarchiser les utilisateurs au sein du réseau domestique. Cette étude vient avec des idées intéressantes sur la façon de concevoir un outil de gestion pour la maison. Comme ces outils doivent recueillir une grande quantité de données , une solution intelligente est nécessaire pour filtrer et stocker les données de valeur ou décider quand il n'est plus utile. En outre,

les utilisateurs ont toujours des problèmes de confidentialité qui demandent des mécanismes plus discrets .

Dans la même perspective d'aider les utilisateurs avec leur expérience de l'Internet à la maison, un autre projet appelé Homenet a été réalisée avec 93 familles de Pittsburgh. Ce travail met en évidence le manque de soutien technique pour les maisons en face de lieu de travail, où les gens peuvent facilement trouver quelqu'un ayant les compétences appropriées à demander de l'aide. S'appuyant sur des rondins de téléphone, des rapports de messagerie des utilisateurs (237 membres) à un personnel dévoué (personnes à appeler en cas de problèmes) et des sondes automatiques pour calculer le temps où les utilisateurs sont actifs, ils ont remarqué que 70% des ménages demander set-up de l'aide et 95% pour le support technique. Cela montre que l'aide à domicile est sous- estimé, même pour les maisons avec les consommateurs qualifiés. Avant de demander au personnel de support technique pour de l'aide, les consommateurs tentent de trouver quelqu'un à la maison qui peut aider, cette personne est généralement un adolescent. Cet enfant va jouer un nouveau rôle de conseiller technique qui affecte son comportement (autorité, l'indépendance). Même si d'autres solutions existent comme projet Austin, qui visent à former les adolescents à prendre avantage de la technologie pour améliorer la participation et aider les autres, les systèmes automatiques sans intervention humaine sont encore nécessaires.

Solutions automatiques existants utilisent définie par logiciel en réseau (SDN) approches. Un exemple concerne un routeur de prototype au-dessus de NOX et Open Vswitch qui apporte contrôle du trafic par flux . Cette méthode présente un ensemble de commutation et de protocole modifications (attribution d'adresse , de contrôle d'accès au support , contrôle d'accès à Internet , le contrôle du trafic au niveau de l' écoulement) et son hétérogénéité est corroborée avec une large gamme de dispositifs compatibles IP . La perspective est de traduire hiérarchie des utilisateurs à la maison dans les composants de la circulation .

Une autre solution vers les réseaux domestiques auto-tuning existe. L'étude de cas dans ce travail est une performance sans fil pauvres connu dans un réseau domestique en raison des interférences (stations fonctionnant sur la même fréquence), qui ajoute des retards dans le réseau lors de la sauvegarde à pied ou en augmentant le taux de perte face à des collisions. La politique de cette affaire est en train de changer le canal qui semble simple mais doit être dynamique et fondée sur l'information diffusée à partir de plusieurs nœuds dans le réseau. Afin d'automatiser cela, une proposition est d'utiliser Politiques État machine sur la base (SMP), qui consiste à divers composants qui exécutent des actions sous conditions .

Le projet le plus proche de notre sujet est le projet Bismark. L'idée est de déployer des passerelles qui permettent l'exécution des mesures actives et passives à distance pour enquêter sur les réseaux domestiques. Dans notre travail, nous utilisons Bismark-passif . Nous avons choisi ce logiciel qui surveille passivement le trafic réseau , car il envoie des mises à jour différentielles que périodiquement à un serveur central. Nous détaillons le processus de collecte de trace plus tard dans cette thèse .

Des études antérieures mettent en évidence les difficultés que rencontrent les utilisateurs lors de l'installation , la maintenance et le dépannage d'un réseau domestique . Parfois , les utilisateurs ne

peuvent même pas articuler correctement quel est le problème . Après l'exécution de deux ensembles différents de tests (interviews, enquêtes) au Royaume Uni et des États-Unis , l' expé-rience montre qu'il existe un grand potentiel pour le développement d'applications qui aident les ménages . De plus , il apparaît que le maintien à domicile est difficile , même pour les utilisateurs avancés . Entretiens montrent également que les utilisateurs s'attendent à ce que , chaque fois que de nouvelles technologies ou fonctionnalités sont portées à la maison , ils doivent être inclus dans l'infrastructure existante .

Les organismes de réglementation intéressés à comparer la vitesse de liaison d'accès offert par les FAI avec ce qu'ils offrent . FAI face à la bande passante de plus en plus et aujourd'hui aussi retarder demandes des utilisateurs. De nouvelles applications et de dispositifs contribuer aux différentes exigences et les défis que les réseaux domestiques poser . Cette tendance rend les réseaux domestiques dépannage et de surveillance fondamental de comprendre leurs problèmes et de défis. depuis suivi la maison à partir d'un dispositif d'extrémité est limité en termes de ce qui peut être suivi , des projets tels que SamKnows (Royaume-Uni et États-Unis) reposent sur des mesures actives à partir de la passerelle domestique . Même si la mesure à la passerelle est précieux , il est également consommatrice de ressources et peut interférer avec les besoins en ressources des utilisateurs . Pour cette raison , nous explorons la faisabilité de textit passifs mesures au niveau des passerelles de la maison .

Nous nous appuyons sur des données recueillies à des hôtes finaux à l'aide de l'outil de surveillance de HostView[42]. HostView enregistre des paquets et des informations sur les applications et l'environnement de l'utilisateur. L'étude a été faiet à partir de données recueillies à partir de 47 utilisateurs qui ont fait tourner HostView pour plus d'une semaine chacun. Étant donné que les utilisateurs se déplacent entre différents réseaux, cette base de données contient du trafic d'hôte d'extrémité sur un total de 185 réseaux différents, répartis sur 18 pays différents. L'analyse du trafic local et wide-area à partir de HostView est difficile cependant, car HostView n'a aucune information sur la nature des flux de trafic. Pire, HostView gratte l'adresse IP de l'hôte des traces pour protéger la vie privée de l'utilisateur, ce qui rend l'identification du trafic local encore plus difficile. Par conséquent, nous proposons une heuristique pour séparer le traffic local du wide-area.

Dans cetet partie on présente une comparaison du trafic local dans les différents environnements de réseau du point de vue d'hôtes d'extrémité. L'avantage d'utiliser despoints de vue d'hôtes d'extrémité est que nous étudions le trafic recueillies auprès de plus d'une centaine de réseaux de points différentes. Nos résultats montrent qu'il existe une grande diversité de l'importance du trafic local par rapport au wide-area, mais en général le trafic wide-area domine. Dans certains réseaux (comme les aéroports et les cafés), on voit rarement le trafic local, le seul trafic local est DNS. À la maison et au travail, nous n'observons une fraction non négligeable du trafic local. La plupart du trafic local est composé par des liaisons courtes, mais parfois les connexions locales transférées ont très grand nombre d'octets (volume). Outre DNS, les applications locales les plus typiques sont les systèmes de fichiers en réseau et de sauvegarde, mais la composition du trafic local dépend de l'utilisateur et du réseau. L'inconvénient de mesurer le trafic local à partir d'hôtes d'extrémité est que nous ne pouvons voir qu'une petite fraction du trafic de chaque réseau.

A.5. Suivi de la performance des applications dans les passerelles domestiques

Les passerelles résidentielles offrent une connectivité Internet pour tous les appareils de la maison, permettant aux services tels que la téléphonie ou les jeux. Cependant, les passerelles domestiques typiques ne comprennent pas de mécanisme pour garantir des performances optimales lorsque les applications sont en compétition pour les mêmes ressources. Dans ce chapitre, nous présentons une approche d'optimisation des performances des applications pour les réseaux domestiques. En particulier, nous étudions la faisabilité du suivi des performances des applications sur les passerelles domestiques, qui implique à la fois l'identification des applications actives et le suivi de leur performance.

Avec la généralisation de l'accès Internet à haut débit[37], de plus en plus de gens ont Internet à la maison. Divers services permettent aux utilisateurs dans un ménage à effectuer des tâches professionnelles et personnelles . Services de réseau différentes mais fonction-ner simultanément peut conduire à une dégradation des performances . Les utilisateurs à domicile face à de nombreux problèmes de performance pour diverses raisons[28;79]. Par exemple, un enfant lance le téléchargement d'un gros fichier qui peut perturber la qualité de la conférence des parents qui effectuent un appel sur Skype. Bien que, le téléchargement ne souffre que de besoins minimum en bande passante alors que la Voix sur IP pour l'appel nécessite une faible latence, qui est affectée par le téléchargement si les deux partagent une seule file d'attente. Le problème est double : d'abord, la plupart des utilisateurs à domicile seulement ont limité les compétences techniques et n'ont donc pas la compréhension des raisons de la dégradation de performance. Deuxièmement, même si ces compétences sont présente, les dispositifs de réseau domestique contemporains n'offrent presque pas d'options pour hiérarchiser le trafic et identifier et résoudre les conflits de ressources.

Comme pour l'exemple précédent, lorsque les performances du réseau se dégradent, les utilisateurs peuvent uniquement se demandent pourquoi leur qualité de conférence téléphonique est mauvaise ? Y a t-il des applications concurrentes ? Est-ce le routeur ? Ou le fournisseur d'accès Internet ? Peut-être que le serveur de VoIP est surchargé . . . Dans cette partie, nous voulons proposer une nouvelle approche pour aider les utilisateurs dans de telles situations par le suivi de la performance des applications de réseau domestique. Notre solution s'appuie sur la passerelle domestique comme point de cheminement. Non seulement tout le trafic réseau de la maison passe par la passerelle domestique, il est également l'endroit idéal pour distinguer le trafic de différents dispositifs de l'utilisateur et les services de réseau, ainsi que de faire la distinction entre les problèmes internes et externes (home / ISP).

Le suivi de la performance des réseaux domestiques se compose de deux parties principales, l'identification des applications actives et la surveillance de leur performance. La première étape est l'identification de l'application, où nous analysons le trafic pour identifier l'ensemble des applications actives. La deuxième, suivi de la performance, consiste à extraire les paramètres de performance des flux appartenant à chaque application. Les applications ont des exigences différentes sur le réseau. Si

l'on considère l'exemple précédent, en utilisant les mesures de suivi de la performance, nous pouvons évaluer la largeur de bande utilisée pour le téléchargement et l'impact sur la latence nécessaire pour la conférence téléphonique Skype. Ainsi, nous pouvons optimiser ces deux paramètres pour s'assurer que la qualité de l'appel Skype ne se dégrade pas en appliquant un délai maximum de ses paquets. Selon le type d'application et les paramètres de performance optimale une utilisation des ressources du réseau peut être déterminée et appliquée. Les besoins en ressources des applications peuvent soit être configurés ou tirées de mesures passées (par exemple, lorsqu'une application était la seule application active à la fois) manuellement.

Des solutions pour l'optimisation des performances de l'application existe, mais seulement sur la base d'hôtes d'extrémité[13;47]. L'optimisation de performance dans les réseaux domestiques nécessite cependant une vue complète de l'ensemble du trafic du réseau domestique. Cela peut être réalisé en plaçant le moniteur sur la passerelle domestique ou en mettant un moniteur vidéo à chaque extrémité de l'hôte. Avec le nombre croissant (tablettes, smartphones) de plus en plus de types différents (ordinateurs portables, consoles de jeux, set-top-box, smart-home , eHealth , etc) de dispositifs de réseau sont utilisés à la maison. La tâche de développer une solution qui fonctionne sur chaque plate-forme semble non viable. Notre solution basée sur la passerelle de la maison doit prendre en charge une plate-forme unique. L'utilisateur peut facilement remplacer la passerelle domestique existante, étant donné que les passerelles domestiques sont petites et bon marché et surtout des appareils passifs sans les données utilisateur stockées sur eux. D'autre part, cela signifie aussi qu'on a des ressources limitées qui fait qu'il est difficile d'exécuter des calculs coûteux et des algorithmes d'optimisation de performances sur la passerelle directement. Nous envisageons un système qui permettra de contrôler le trafic dans le réseau de la maison en fonction de l'application exigences et les priorités des ménages. Les besoins en ressources du réseau varient d'une application à une autre. Certains auront besoin de peu de retard tandis que d'autres nécessitent un débit élevé. Priorités des ménages changent aussi en fonction de l'activité et au fil du temps. Un utilisateur peut donner plus d'importance à ses mails tout en travaillant, son téléchargement de fichiers lors de l'installation du logiciel ou son streaming vidéo en détendant. Les conditions d'application et les priorités des ménages aident à décider comment allouer les ressources. Pour que ce système d'optimisation d'exercer le niveau de contrôle requis, il doit être installé principalement dans la passerelle domestique.

Lorsque le trafic traverse la passerelle domestique (étape 1), les dossiers de modules de surveillance de la circulation de flux et de l'information de paquets (étape 2) dans le but de déterminer les applications actives actuelles avec leurs indicateurs de performance (identification de l'application et de la performance calcul métrique). Cette information est envoyée à l'optimiseur de trafic (étape 3). L'optimiseur traite cette information et lui attribue les profils de trafic d'application correspondantes. En outre, il identifie les besoins en ressources pour un profil donné et en fonction de la configuration de l'utilisateur (priorités application/ ménage) envoie des paramètres de contrôle (vitesse download/upload, la longueur de la file d'attente) pour le contrôleur de la circulation (étape 4), qui sera capable de capter le trafic pour éviter les dégradation de performances.

Dans ce qui suit, nous expliquons chaque module de ce système, de façon plus détaillée :

Traffic Monitor, ce module est composé d'une version modifiée de la fonction Bismark-passif pour effectuer une mesure passive du trafic . Bismark - passif a été initialement développé par le chercheur de Georgia Tech à des fins de surveillance passive du trafic réseau et l'envoi de petites mises à jour périodiquement anonymes à un serveur central pour analyse afin de mieux comprendre l'utilisation du réseau de la maison . Le paquet enregistrée et le flux d'information est envoyé à l' . . .

Identification des applications et des performances de calcul métrique, un processus pour détecter les applications et leurs indicateurs de performance . Pour tout le trafic entrant , ce processus permettra d'identifier les applications actives et leurs paramètres (bande passante , de perte de paquets , la latence , etc) correspondantes et les envoyer à la ...

Trafic Optimizer, ce module prend en entrée les applications actives avec leurs indicateurs de performance . En outre, il a besoin de connaissances existantes (par exemple , appris avant) sur les profils de performance des applications (plages de performance acceptable pour différentes métriques) . Ce module combine ces informations et de la priorité de l'utilisateur de donner en sortie les paramètres de mise en forme optimale pour le ...

Contrôleur de la circulation, ce module est conçu pour appliquer le lissage du trafic et la priorisation du trafic afin de transmettre le trafic dans la meilleure forme qui évite la dégradation des performances . Certains outils que tc et / ou netem pourraient être utilisés pour le lissage du trafic.

Configuration de l'utilisateur, un module d'apprentissage qui enregistre la priorité que l'utilisateur attribue à chaque application (classe d'applications). Sa sortie sera incluse dans le processus de décision de l'optimiseur.

S'il est clair que le moniteur de trafic et le contrôleur de la circulation doivent fonctionner à l'intérieur de la passerelle de la maison, il est un espace de conception sur l'emplacement d'exécution de l'identification de l'application et de calcul des indicateurs de performance, l'optimiseur de trafic, et la configuration de l'utilisateur.

Les modules proposés et leurs fonctions ont besoin d'une puissance de traitement considérable doit être effectuée. Dans le reste de ce chapitre, nous voulons répondre à la question combien le traitement peut être fait dans la passerelle de la maison et de discuter de la tête de l'exportation d'informations de la passerelle. Nous nous concentrons principalement sur l'identification et l'application Traffic Monitor et modules de calcul métriques. Dans la section suivante, nous allons discuter de la performance qui métriques que nous devons suivre.

Comprendre les performances des applications de réseau est une condition préalable à l'attribution des ressources du réseau de manière que les utilisateurs sont satisfaits. Dans la mise en réseau des performances de l'application est représentée avec des métriques de performance telles que le débit, le retard et la gigue. D'autres indicateurs tels que le nombre de retransmission-sions dans une connexion, ainsi que le nombre de connexions simultanées nous aideront à mieux diagnostiquer la situation actuelle. En outre, nous devons être en mesure d'identifier le (type de) l'application qui provoque un certain morceau de trafic. Ainsi, nous visons à rassembler toutes ces mesures sur une passerelle.

La déduction et le suivi des indicateurs de performance du réseau a été bien étudié. Différents outils ont été développés pour aider les utilisateurs et les chercheurs mesurent les paramètres de réseau simples. Dans ce qui suit, nous explorons la pertinence des outils existants pour nos fins.

Les réseaux domestiques et les performances des applications sont deux domaines difficiles. Dans notre travail, nous cherchons à éviter la dégradation des performances des applications actives dans un réseau domestique par le suivi de leur performance à partir des passerelles résidentielles. Nous discutons des limites des ressources de la passerelle avec le compromis entre les stratégies de mise en œuvre différentes (hôtes finaux vs. passerelle domestique). Nous avons introduit une version modifiée de *Bismark-passive* qui recueille des renseignements précieux pour l'identification de l'application. Nous avons montré que, même s'il génère une charge supplémentaire, les résultats sont prometteurs pour une technique d'identification d'application avec un processus de collecte de trafic. Nous avons expliqué les techniques possibles d'identification des applications et discuté de notre mise en œuvre de la solution. Notre évaluation globale a montré qu'il est possible d'effectuer une technique de dégradation des performances des applications à partir de la maison en suivant nos lignes directrices.

A.6. Réécriture de tracs de contact

Réaliser une communication efficace dans les réseaux tolérant aux perturbations (DTN) dépend d'une compréhension profonde sur les lois dynamiques régissant le réseau. En particulier, il est important d'étudier la fréquence et la durée au cours de laquelle les nœuds se rencontrent en utilisant les notions de modèles de *contact* et de *Intercontact* [45;73].

La meilleure façon d'analyser ou de valider un protocole ou choix de conception dans DTN est par de véritables déploiements. Néanmoins, en raison des difficultés de mise en œuvre et même les coûts financiers, à seulement quelques expérimentations ont été rapportés dans la littérature [89;104]. En conséquence, plusieurs œuvres comptent encore sur les modèles de mobilité synthétiques .

Alors que les modèles de mobilité synthétiques sont utiles pour isoler des paramètres spécifiques d'une solution ou aider à enquêter sur l'évolutivité d'un système, ils ne peuvent pas toujours refléter les conditions réelles. Les modèles synthétiques mobilité sont toujours utilisés comme un modèle parfait pour l'évaluation en raison de leur connaissance de la mobilité et de la mimique de mouvements humains réels. Les chercheurs ont fait un effort supplémentaire pour développer des modèles sur les traces de mobilité réelles afin de garantir la plus haute réalisme possible.

Pour construire un modèle à partir de traces, de multiples scénarios de mouvements humains sont effectuées. Des expérimentations de multiple traces de mouvements réels sont collectées pour extraire des caractéristiques de mobilité. Sur la base des caractéristiques observées, un modèle à partir de traces est construit. Le but est d'avoir un modèle qui offre des traces synthétiques qui sont à proximité des traces d'origine du mouvement humain. De nombreux modèles de mobilité à partir de traces ont été développés [2;39;40;51;65;70]. Il existe des modèles simples de mobilité [62] et des modèles plus complexes , tels que ceux basés sur de vraies traces GPS [58;81].

En raison des défis dans le processus de collecte, un nombre limité de traces de contact est disponible pour la communauté de la recherche[53]. Une trace de contact représentant implique le recrutement d'un grand nombre d'utilisateurs, la mise en place des dispositifs de mesure (par exemple, la fréquence à laquelle les nœuds seront envoyés balises) et nettoyer les données. Une trace de contact représentant implique le recrutement d'un grand nombre d'utilisateurs au premier abord. Ensuite, il faut mettre en place tous les dispositifs de mesure en utilisant les bons paramètres (par exemple, la détermination de la fréquence à laquelle les nœuds envoient des balises pour les voisins potentiels). Enfin, il est important de nettoyer les données et obtenir qu'il soit prêt à utiliser.

Dans notre travail, nous proposons un système de reproduction de la mobilité de la trace que d'un seul de la vie réelle contact trace, tire traces de contact plausibles inspirés de la trace originale. L'objectif est de se rapprocher le plus possible à un scénario réaliste, tout en évitant la mise en place d'une nouvelle expérience. La motivation de ce travail est venu lors d'une expérience où nous avons eu à choisir entre Bluetooth et Wi-Fi pour le canal ad hoc. En fait, ce que serait l'impact d'une telle décision sur le contact trace tion de résultat ? La bonne réponse à cette question, il faudrait mettre en place à partir de zéro autant d'expériences que le nombre de configurations différentes qui nious intéresse.

Notre système n'a besoin que d'une seule trace de contact à partir d'une expérience réelle de se reproduire plusieurs traces , comme si nous effectuons plusieurs expériences avec des configurations différentes . Il s'ensuit deux étapes de base :

1. **Inférence spatiale.** Nous tirons une trace de mobilité plausible de la trace simple de contact par en utilisant un système d'inférence qui prend en entrée une trace de contact et génère une mobilité spatiale «plausible».

2. **Trace élevage.** De la trace de la mobilité spatiale, nous avons fixé les valeurs des paramètres (intervalle de communication, intervalle de mesure) et réécrivons le scénario original avec les traces de contact résultantes " réécrites ". Dans la version actuelle du système, même si cela n'est pas une obligation, nous considérons un modèle de propagation de disque.

Dans notre approche, La trace de contact pour se reproduire peut être synthétique ou réel. Des modèles de mobilité permettent de produire des traces de contact en simulant un modèle basé sur la proximité par exemple, où deux noeuds sont en contact si elles sont dans la portée de transmission de chaque autre. Nous considérons le contact résultant retrace aussi parfait , puisque nous enregistrons toutes les possibilités de contact et de contrôler les paramètres entiers. Nous utilisons cette approche pour notre contrôle de conformité.

En revanche, les traces de contact de la vie réelle sont considérés bruyant. les traces réelles sont mesurées selon un intervalle de mesure qui échoue dans un certain temps pour enregistrer toutes les possibilités de contact. Grâce à une technologie de communication ZigBee ou Bluetooth par exemple, prend quelques secondes mais peut encore manquer contacts. Plus l'intervalle de mesure est élevée, plus la probabilité de manquer contacts. En effet , les intervalles mesure de longues rendent plus

difficile à attraper contacts court . Pire encore, les contacts courts consécutifs peuvent être considérés comme un seul contact prolongé faussant ainsi les résultats . Une solution nommée Pocket Mobilité Enregistreurs de trace a été conçu pour surmonter ces problèmes. Pourtant , il ne peut pas éviter les limitations sans fil typiques tels que les interférences .

Afin de traduire traces de contact à des traces de mobilité spatiale, nous utilisons la mobilité plausible. Il est à ce jour le seul algorithme qui permet de proposer une mobilité spatiale des nœuds seulement en fonction de leurs temps de contact . L'algorithme ne fournit pas la mobilité exact, mais assure deux propriétés principales . Tout d'abord, la vitesse des nœuds reste réaliste et limité . En second lieu, la trace original peut éventuellement être reproduite comme le concept de noeuds en contact (dans la plage de transmission de l'autre) est toujours respecté.

Les plus contraintes , plus la trace de la mobilité seront à la trace de contact d'origine. Par exemple, des informations supplémentaires comme des positions de nœuds fixes aider l'algorithme inférence. Mais ils ne sont pas toujours disponibles, pire le réseau peut être grande et / ou rares qui rend plus difficile la conclusion. Même si la mobilité plausible offre une solution fiable, il introduit aussi une couche d'erreurs possibles.

De la mobilité spatiale à la trace de contact reproduite, nous considérons que la trace résultante est complète. En d'autres termes, nous n'appliquons pas de toute interférence ou modèle de la perte sur le dessus des résultats. Une perspective intéressante pourrait être de reproduire les contraintes de l'environnement d'origine (pertes de mesure, interférences) et de les appliquer aux traces élevés afin d' imiter les défis de l'environnement initial de l'expérience. Nous proposons un système capable de déduire autant de traces que nous voulons d'une seule expérimentation réelle. Le principal avantage de notre système est d'éviter tous les défis de la mise en place d'une réelle expérience. Notre approche offre la possibilité de paramètrer les mesures (plage de transmission, intervalle de mesure), permettant d'obtenir de nouvelles connaissances sur le réseau qui aurait été possible uniquement grâce à une nouvelle campagne expérimentale.

Nos résultats montrent que, malgré les contraintes de l'algorithme de mobilitéd' inférence, notre système peut donner des traces qui sont conformes à l'ensemble de données d'origine. Lors de l'utilisation de synthèse et de traces de contact à base le système offre des traces résultantes très similaires. En particulier, même avec des traces réelles bruyants et imparfaits, les traces élevés suivent les caractéristiques des tracés originaux.

L'accès haut débit à Internet, la diversité de dispositifs personnels des utilisateurs et le contenu de haut qualité sont des facteurs de développement constant menant à de plus en plus d'exigence des clients. Pour garantir une large bande passante, de faibles retards et la connectivité continu aux utilisateurs, il est crucial d'avoir une bonne compréhension du réseau. D'une zone restreinte, comme les réseaux de la maison à des réseaux plus vastes et plus dynamiques comme les réseaux sans fil, les défis varient, mais l'exigence de l'utilisateur non. Quelles que soient les ressources nécessaires pour le contenu souhaité, l'utilisateur sera toujours aller vers le meilleur service de garantie à la maison ou dans la rue. Par conséquent, il est non seulement important de comprendre la dynamique

du réseau, mais également d'être capable de diagnostiquer et optimiser les performances du réseau d'une manière automatique pour satisfaire les utilisateurs.

Cette thèse développe une méthodologie pour diagnostiquer automatiquement et optimiser l'application de réseau en utilisant les ressources de la passerelle au minimum d'une part et contribue à la dynamique des réseaux compréhension en utilisant les données réelles au minimum d'une autre part.

Appendix B

List of publications

B.1. Conferences

Ahlem Reggani, Fabian Schneider, Renata Teixeira, *An end-host view on local traffic at home and work*, in *Passive and Active measurements conference* (PAM), March 2012.

Ahlem Reggani, Fabian Schneider, Renata Teixeira, *Tracking application network performance in Home Gateways*, in *Traffic analysis and classification workshop* (IWCMC), July 2013.

Ahlem Reggani, John Whitbeck, Marcelo Dias de Amorim, Mauro Fonseca, Vania Conan, and Serge Fdida, *Mobility trace breeding*', in *Wireless days* (poster), Nov. 2013.

B.2. Journals and Magazines

Ahlem Reggani, John Whitbeck, Marcelo Dias de Amorim, Mauro Fonseca, and Vania Conan, *Getting the most of your experiment : The impact of mobility trace breeding*, submitted to *Computer Communications Journal*.

Bibliography

[1] AA. A study of facebook power users and their effect on network dynamics, February 2012. http://www.statista.com.

[2] Nils Aschenbruck, Aarti Munjal, and Tracy Camp. Trace-based mobility modeling for multi-hop wireless networks. *Computer Communications*, 34(6):704–714, 2011.

[3] Laurent Bernaille, Renata Teixeira, and Kave Salamatian. Early application identification. In *CoNEXT'06*, Lisboa, Portugal, December 2006.

[4] BISMark Passive. BISMark-passive, 2011. https://github.com/projectbismark/bismark-passive.

[5] Pierre Borgnat, Guillaume Dewaele, Kensuke Fukuda, Patrice Abry, and Kenjiro Cho. Seven years and one day: Sketching the evolution of internet traffic. In *INFOCOM'09*, Rio de Janeiro, Brazil, April 2009.

[6] J.Y. Le Boudec and M. Vojnovic. The random trip model: stability, stationary regime, and perfect simulation. *IEEE/ACM Trans. Netw.*, 14(6):1153–1166, December 2006.

[7] J. Buckheit, B. Donoho, and L. David. Wavelab and reproducible research. In A. Antoniadis and G. Oppenheim, editors, *Wavelets and Statistics*. Springer New York, 1995.

[8] Ramón Cáceres, Peter B. Danzig, Sugih Jamin, and Danny J. Mitzel. Characteristics of wide-area TCP/IP conversations. In *SIGCOMM'91*, Zurich, Switzerland, September 1991.

[9] Kenneth L. Calvert, W. Keith Edwards, Nick Feamster, Rebecca E. Grinter, Ye Deng, and Xuzi Zhou. Instrumenting home networks. *SIGCOMM Comput. Commun. Rev.*, January 2011.

[10] A. Chaintreau, H. Pan, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Impact of human mobility on the design of opportunistic forwarding algorithms. *IEEE Transactions on Mobile Computing*, 6(6):606–620, June 2007.

[11] Augustin Chaintreau, Pan Hui, Jon Crowcroft, Christophe Diot, Richard Gass, and James Scott. Pocket switched networks: Real-world mobility and its consequences for opportunistic forwarding. Technical report, University of Cambridge Computer Laboratory, February 2005.

[12] 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 *CHI '10*, Atlanta, Georgia, USA, April 2010.

[13] 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 *CHI '10*, Atlanta, Georgia, USA, April 2010.

[14] Marshini Chetty, David Haslem, Andrew Baird, Ugochi Ofoha, Bethany Sumner, and Rebecca Grinter. Why is my internet slow?: making network speeds visible. In *CHI '11*, Vancouver, BC, Canada, May 2011.

[15] Marshini Chetty, David Haslem, Andrew Baird, Ugochi Ofoha, Bethany Sumner, and Rebecca Grinter. Why is my internet slow?: making network speeds visible. In *CHI '11*, Vancouver, BC, Canada, May 2011.

[16] Alberto Dainotti, Walter Donato, and Antonio Pescapé. Tie: A community-oriented traffic classification platform. In *TMA '09*, Aachen, Germany, April 2009.

[17] Lucas DiCioccio, Renata Teixeira, and Catherine Rosenberg. Impact of home networks on end-to-end performance: controlled experiments. In *SIGCOMM workshop on Home networks*, HomeNets '10, New Delhi, India, August 2010.

[18] Marcel Dischinger, Krishna P. Gummadi, Andreas Haeberlen, and Stefan Saroiu. Characterizing residential broadband networks. In *IMC'07*, San Diego, CA, USA, October 2007.

[19] Marcel Dischinger, Andreas Haeberlen, Krishna P. Gummadi, and Stefan Saroiu. Characterizing residential broadband networks. In *IMC '07*, San Diego, California, USA, October 2007.

[20] K. Donald. Literate programming. *The Computer Journal*, 27(2):97–111, January 1984.

[21] N .Eagle and A. Pentland. Reality mining: sensing complex social systems. *Personal Ubiquitous Comput.*, 10(4):255–268, March 2006.

[22] Marius A. Eriksen. Trickle: a userland bandwidth shaper for unix-like systems. In *ATEC '05*, Anaheim, CA, January 2005.

[23] Figaro. Figaro. `http://www.ict-figaro.eu`, 2013.

[24] Linux Foundation, 2009. `http://www.linuxfoundation.org/collaborate/workgroups/networking/netem`.

[25] S. Gaito, E. Pagani, and G. Rossi. Opportunistic forwarding in workplaces. In *ACM WOSN*, Barcelona, Spain, August 2009.

[26] Frédéric Giroire, Jaideep Chandrashekar, Gianluca Iannaccone, Konstantina Papagiannaki, Eve M. Schooler, and Nina Taft. The cubicle vs. the coffee shop: behavioral modes in enterprise end-users. In *PAM'08*, Cleveland, OH, USA, April 2008.

[27] F. Gringoli, Luca Salgarelli, M. Dusi, N. Cascarano, F. Risso, and k. c. claffy. Gt: picking up the truth from the ground for internet traffic. *SIGCOMM Comput. Commun. Rev.*, October 2009.

[28] Rebecca E. Grinter, W. Keith Edwards, Marshini Chetty, Erika S. Poole, Ja-Young Sung, Jeonghwa Yang, Andy Crabtree, Peter Tolmie, Tom Rodden, Chris Greenhalgh, and Steve Benford. The ins and outs of home networking: The case for useful and usable domestic networking. *ACM Trans. Comput.-Hum. Interact.*, June 2009.

[29] Rebecca E. Grinter, W. Keith Edwards, Marshini Chetty, Erika S. Poole, Ja-Young Sung, Jeonghwa Yang, Andy Crabtree, Peter Tolmie, Tom Rodden, Chris Greenhalgh, and Steve Benford. The ins and outs of home networking: The case for useful and usable domestic networking. *ACM Trans. Comput.-Hum. Interact.*, June 2009.

[30] Dongsu Han, Aditya Agarwala, David G. Andersen, Michael Kaminsky, Konstantina Papagiannaki, and Srinivasan Seshan. Mark-and-sweep: getting the inside scoop on neighborhood networks. In *IMC'08*, Vouliagmeni, Greece, October 2008.

[31] N. Handigol, B. Heller, V. Jeyakumar, B. Lantz, and N. McKeown. Reproducible network experiments using container-based emulation. In *ACM CoNEXT*, nice, France, December 2012.

[32] Gerhard Hasslinger and Oliver Hohlfeld. The gilbert-elliott model for packet loss in real time services on the internet. In *GI/ITG Conference*, Dortmund, Deutschland, April 2008.

[33] Seppo Hätönen, Aki Nyrhinen, Lars Eggert, Stephen Strowes, Pasi Sarolahti, and Markku Kojo. An experimental study of home gateway characteristics. In *IMC '10*, Melbourne, Australia, November 2010.

[34] Tristan Henderson, David Kotz, and Ilya Abyzov. The changing usage of a mature campus-wide wireless network. In *MobiCom'04*, Philadelphia, PA, USA, September 2004.

[35] Bert Hubert, 2001. `http://lartc.org/manpages/tc.txt`.

[36] Insung, Park and Ricky, Buch. Improve Debugging And Performance Tuning With ETW, 2007. `http://msdn.microsoft.com/en-us/magazine/cc163437.aspx`.

[37] Internet World Stats. Internet World Stats, 2011. `http://www.internetworldstats.com/dsl.htm`.

[38] Manish Jain and Constantinos Dovrolis. Pathload: A measurement tool for end-to-end available bandwidth. In *PAM'02*, Fort Collins, Colorado, USA, March 2002.

[39] Amit Jardosh, Elizabeth M. Belding-Royer, Kevin C. Almeroth, and Subhash Suri. Towards realistic mobility models for mobile ad hoc networks. In *MobiCom '03*, San Diego, CA, USA, September 2003.

[40] Wei jen Hsu, Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Ahmed Helmy. Modeling timevariant user mobility in wireless mobile networks. In *in Proc. IEEE INFOCOM*, Anchorage, Alaska, May 2007.

[41] D. Joumblatt, O. Goga, R. Teixeira, J. Chandrashekar, and N. Taft. Characterizing end-host application performance across multiple networking environments. In *INFOCOM'12*, March 2012.

[42] Diana Joumblatt, Renata Teixeira, Jaideep Chandrashekar, and Nina Taft. Hostview: annotating end-host performance measurements with user feedback. *SIGMETRICS Perform. Eval. Rev.*, January 2011.

[43] Diana Joumblatt, Oana Goga, Renata Teixeira, Jaideep Chandrashekar, and Nina Taft. Characterizing end-host application performance across multiple networking environments. In *INFOCOM (Mini-Conference)*, Orlando, Florida, USA, March 2012.

[44] Diana Joumblatt, Renata Teixeira, Jaideep Chandrashekar, Nina Taft, and Kveton Branislav. Predicting user dissatisfaction with internet application performance at end-hosts. In *INFOCOM'13*, Turin, Italy, march 2013.

[45] T. Karagiannis, J.Y. Le Boudec, and M. Vojnović. Power law and exponential decay of inter contact times between mobile devices. In *ACM MobiCom*, Montreal, Quebec, Canada, September 2007.

[46] Thomas Karagiannis and Key Peter. Homemaestro: Distributed monitoring and diagnosis of performance anomalies in home networks. Technical report, Microsoft Research, 2008. URL http://research.microsoft.com/pubs/63875/Technical%20Report.pdf.

[47] Thomas Karagiannis and Key Peter. Homemaestro: Distributed monitoring and diagnosis of performance anomalies in home networks. Technical report, Microsoft Research, 2008. URL http://research.microsoft.com/pubs/63875/Technical%20Report.pdf.

[48] Thomas Karagiannis, Gkantsidis Christos, and Peter Key. Homemaestro: Distributed monitoring and diagnosis of performance anomalies in home networks, October 2008. Tech. Rep. MSR.

[49] F. Kevin. A delay-tolerant network architecture for challenged internets. In *ACM SIGCOMM*, Karlsruhe, Germany, August 2003.

[50] Sara Kiesler, Bozena Zdaniuk, Vicki Lundmark, and Robert Kraut. Troubles with the internet: The dynamics of help at home. *Human-Computer Interaction*, December 2000.

[51] Minkyong Kim, David Kotz, and Songkuk Kim. Extracting a mobility model from real user traces. In *INFOCOM'06*, Barcelona, Spain, April 2006.

[52] Minkyong Kim, David Kotz, and Songkuk Kim. Extracting a mobility model from real user traces. In *INFOCOM'06*, April 2006.

[53] D. Kotz and T. Henderson. A community resource for archiving wireless data. http://crawdad.cs.dartmouth.edu, 2008.

[54] David Kotz and Kobby Essien. Analysis of a campus-wide wireless network. *Wirel. Netw.*, 11 (1-2), January 2005.

[55] Christian Kreibich, Nicholas Weaver, Boris Nechaev, and Vern Paxson. Netalyzr: illuminating the edge network. In *IMC '10*, Melbourne, Australia, November 2010.

[56] Craig Labovitz, Scott Iekel-Johnson, Danny McPherson, Jon Oberheide, and Farnam Jahanian. Internet inter-domain traffic. In *SIGCOMM'10*, New Delhi, India, August 2010.

[57] L.Bernaille. *Real-time application classificationin the Internet*. Ph. d. thesis, Université Pierre et Marie Curie, 2007.

[58] Kyunghan Lee, Seongik Hong, Seong Joon Kim, Injong Rhee, and Song Chong. Slaw: self-similar least-action human walk. *IEEE/ACM Trans. Netw.*, 20(2), April 2012.

[59] Gregor Maier, Anja Feldmann, Vern Paxson, and Mark Allman. On dominant characteristics of residential broadband internet traffic. In *IMC '09*, Chicago, Illinois, USA, November 2009.

[60] Radu-Corneliu Marin, Ciprian Dobre, and Fatos Xhafa. A methodology for assessing the predictable behaviour of mobile users in wireless networks. *Concurrency Computat.: Pract. Exper.*, June 2013.

[61] Marvin McNett and Geoffrey M. Voelker. Access and mobility of wireless pda users. *SIGMOBILE Mob. Comput. Commun. Rev.*, 9(2), April 2005.

[62] A. Mei and J. Stefa. Swim: A simple model to generate small mobile worlds. In *INFOCOM 2009, IEEE*, Rio de Janeiro, Brazil, April 2009.

[63] Paolo Meroni, Sabrina Gaito, Elena Pagani, and Gian Paolo Rossi. CRAWDAD data set unimi/pmtr (v. 2008-12-01). Downloaded from http://crawdad.cs.dartmouth.edu/unimi/pmtr, December 2008.

[64] R. Mortier, T. Rodden, T. Lodge, D. McAuley, C. Rotsos, A.W. Moore, A. Koliousis, and J. Sventek. Control and understanding: Owning your home network. In *COMSNETS 2012*, Bangalore, India, jan. 2012.

[65] Aarti Munjal, Tracy Camp, and William C. Navidi. Smooth: a simple way to model human mobility. In *MSWiM '11*, Miami, Florida, USA, October 2011.

[66] Klara Nahrstedt and Long Vu. CRAWDAD data set uiuc/uim (v. 2012-01-24). Downloaded from http://crawdad.cs.dartmouth.edu/uiuc/uim, January 2012.

[67] NanoDataCenters. Nanodatacenters. `http://www.nanodatacenters.eu`, 2009.

[68] Boris Nechaev, Mark Allman, Vern Paxson, and Andrei Gurtov. A preliminary analysis of tcp performance in an enterprise network. In *INM/WREN'10*, San Jose, CA, April 2010.

[69] Netgear. Netgear announces technology collaboration with samknows for FCCs national broadband speed test. `http://www.netgear.com/about/press-releases/2010/20100601.aspx`, June 2010. Press release.

[70] AnhDung Nguyen, Patrick Senac, Victor Ramiro, and Michel Diaz. Steps - an approach for human mobility modeling. In Jordi Domingo-Pascual, Pietro Manzoni, Sergio Palazzo, Ana Pont, and Caterina Scoglio, editors, *NETWORKING 2011*. Springer Berlin Heidelberg, 2011.

[71] S. Ostermann. Tcptrace: A tcp connection analysis tool. URL: http://www. tcptrace. org, 2000.

[72] Ruoming Pang, Mark Allman, Mike Bennett, Jason Lee, Vern Paxson, and Brian Tierney. A first look at modern enterprise traffic. In *IMC '05*, Berkeley, CA, USA, October 2005.

[73] A. Passarella and M. Conti. Characterising aggregate inter-contact times in heterogeneous opportunistic networks. In *IFIP TC6*, Valencia, Spain, May 2011.

[74] Vern Paxson. Empirically-derived analytic models of wide- area tcp connections. *IEEE/ACM Transactions on Networking*, August 1994.

[75] Vern Paxson. Bro: a system for detecting network intruders in real-time. *Computer Networks*, 1999.

[76] Vern Paxson. Bro intrusion detection system, 2013. `http://www.bro-ids.org`.

[77] D. Pediaditakis, L. Mostarda, Changyu Dong, and N. Dulay. Policies for self tuning home networks. In *POLICY 2009*, London, UK, july 2009.

[78] C.E. Perkins and E.M. Royer. Ad-hoc on-demand distance vector routing. In *IEEE WMCSA '99*, New Orleans, Louisiana, USA, February 1999.

[79] Erika Shehan Poole, Marshini Chetty, Rebecca E. Grinter, and W. Keith Edwards. More than meets the eye: Transforming the user experience of home network management, 2008.

[80] Ahmad Rahmati and Lin Zhong. CRAWDAD data set rice/context (v. 2007-05-23). Downloaded from http://crawdad.cs.dartmouth.edu/rice/context, May 2007.

[81] Injong Rhee, Minsu Shin, Seongik Hong, Kyunghan Lee, Seong Joon Kim, and Song Chong. On the levy-walk nature of human mobility. *IEEE/ACM Trans. Netw.*, 19(3), June 2011.

[82] RIPE. RIPE atlas. http://atlas.ripe.net, 2004.

[83] A. Roy, S.K. Das Bhaumik, A. Bhattacharya, K. Basu, D.J. Cook, and S.K. Das. Location aware resource management in smart homes. In *PerCom'03*, Dallas, Texas, USA, March 2003.

[84] Guha Saikat, Jaideep Chandrashekar, Nina Taft, and Konstantina Papagiannaki. How healthy are today's enterprise networks? In *IMC'08*, Vouliagmeni, Greece, October 2008.

[85] Marcel Salathe, Maria Kazandjieva, Jung Woo Lee, Philip Levis, Marcus W. Feldman, and James H. Jones. Stanford high original data. Downloaded from http://www.salathegroup.com, November 20010.

[86] SamKnows. Samknows. http://www.samknows.com, 2009.

[87] Fabian Schneider. Performance evaluation of packet capturing systems for high-speed networks. Diplomarbeit, Technische Universität München, Munich, Germany, November 2005.

[88] Fabian Schneider, Jörg Wallerich, and Anja Feldmann. Packet capture in 10-gigabit ethernet environments using contemporary commodity hardware. In *PAM '07*, Louvain-la-Neuve, Belgium, April 2007.

[89] A. Seth, D. Kroeker, M. Zaharia, S. Guo, and S. Keshav. Low-cost communication for rural internet kiosks using mechanical backhaul. In *ACM MobiCom*, Los Angeles,CA,USA, September 2006.

[90] Sheeve Plug. Sheeva plug. http://www.plugcomputer.org, 2007.

[91] M. Siekkinen, D. Collange, G. Urvoy-keller, and E. W. Biersack. Performance limitations of adsl users: A case study. In *PAM'07*, Louvain-la-neuve, Belgium, April 2007.

[92] Snort. Snort, 2013. http://www.snort.org.

[93] L. Song, D. Kotz, R. Jain, and X. He. Evaluating location predictors with extensive wi-fi mobility data. In *INFOCOM 2004*, Hong Kong, China, March 2004.

[94] Krishna Gummadi Stefan, Stefan Saroiu, and Steven D. Gribble. King: Estimating latency between arbitrary internet end hosts. In *SIGCOMM'02*, Pittsburgh,USA, August 2002.

[95] Srikanth Sundaresan, Walter de Donato, Nick Feamster, Antonio Pescapè, and Renata Teixeira. Benchmarking broadband internet with bismark. http://www.cc.gatech.edu/~ssundar3/docs/bismark-internet2-102010.pdf, November 2010. Presentation at Internet2 Fall Members Meeting.

[96] Srikanth Sundaresan, Walter de Donato, Nick Feamster, Renata Teixeira, Sam Crawford, and Antonio Pescapè. Broadband internet performance: a view from the gateway. In *SIGCOMM'11*, Toronto, Ontario, Canada, August 2011.

[97] K. Thompson, G.J. Miller, and R. Wilder. Wide-area internet traffic patterns and characteristics. *Network, IEEE*, Nov/Dec 1997.

[98] P. Tournoux, J. Leguay, F. Benbadis, V. Conan, M. Dias de Amorim, and J. Whitbeck. The accordion phenomenon: Analysis, characterization, and impact on dtn routing. In *IEEE INFOCOM*, Rio de Janeiro, Brazil, April 2009.

[99] P. Vandewalle, J. Kovacevic, and M. Vetterli. Reproducible research in signal processing - what, why, and how. *IEEE Signal Processing Magazine*, 26(3):37–47, May 2009.

[100] J. Whitbeck, M. Dias de Amorim, and V. Conan. Plausible mobility: inferring movement from contacts. In *ACM MobiOpp*, Pisa, Italy, February 2010.

[101] J. Whitbeck, M. Dias de Amorim, V. Conan, M. Ammar, and E. Zegura. Fast track article: From encounters to plausible mobility. *Pervasive Mob. Comput.*, 7(2):206–222, April 2011.

[102] Wonder Shaper. Wonder Shaper, 2002. `http://lartc.org/wondershaper/`.

[103] Jeonghwa Yang and W. Keith Edwards. A study on network management tools of householders. In *HomeNets '10*, New Delhi, India, August 2010.

[104] X. Zhang, J. Kurose, B. Levine, D. Towsley, and H. Zhang. Study of a bus-based disruption-tolerant network: mobility modeling and impact on routing. In *ACM MobiCom*, Montreal, Quebec, Canada, September 2007.

[105] Yin Zhang, Lee Breslau, Vern Paxson, and Scott Shenker. On the characteristics and origins of internet flow rates. In *SIGCOMM '02*, Pittsburgh, Pennsylvania, USA, August 2002.

[106] Rui Zhang-Shen and N. McKeown. Guaranteeing quality of service to peering traffic. In *INFOCOM 2008*, Phoenix, ZA, USA, April 2008.

List of Figures

List of Tables