Impact of Home Networks on End-to-End Performance: Controlled Experiments

Lucas Di Cioccio
Technicolor / UPMC Paris
Universitas
lucas.dicioccio@technicolor.com

Renata Teixeira
CNRS / UPMC Paris
Universitas
renata.teixeira@lip6.fr

Catherine Rosenberg
University of Waterloo
cath@ecemail.uwaterloo.ca

ABSTRACT

This paper performs controlled experiments to analyze the performance of home networks. We show that the home network has a significant impact on end-to-end performance. For example, watching TV can double the time to download a file; and deploying a wireless network significantly increases round-trip times. Despite its impact on end-to-end performance, most existing diagnosis tools ignore the home network when identifying the cause of performance problems. To make matters worse, our results show that simple techniques that directly probe the home gateway cannot reliably identify that the home network is the cause of performance degradation. We are currently designing a technique based on packet bursts to identify performance problems caused by the home network. Our results for Ethernet based home networks are encouraging.

1. INTRODUCTION

The last decade has seen a steep increase in the number of Internet broadband connections [6]. More recently, home networks are also becoming more complex. Home users can now subscribe to triple-play service, which includes voice and TV; they can also install several network-enabled devices (such as gaming consoles and media servers). In addition, there is a variety of technologies to connect the devices in the home (for example, Ethernet, WiFi, or power-line). Figure 1 illustrates a typical home network with different devices that all connect to the Internet through a single point, the *home gateway*. Most often home gateways combine a router with a DSL or cable modem. The gateway is then connected via an access network to a router in the provider network, the *provider-edge router* (PE), which routes traffic from access networks to the Internet.

Home networks are fast becoming an important component of the Internet, but their performance has not been studied in detail. Although there is an increased interest in understanding residential Internet access performance [2–5, 7, 9], previous work focuses mainly on measuring and characterizing access networks (i.e., the network connecting the gateway to the provider-edge router), but not the home network itself. However, as illustrated in Figure 1, end-to-end Internet performance may be disrupted by competing traffic in the
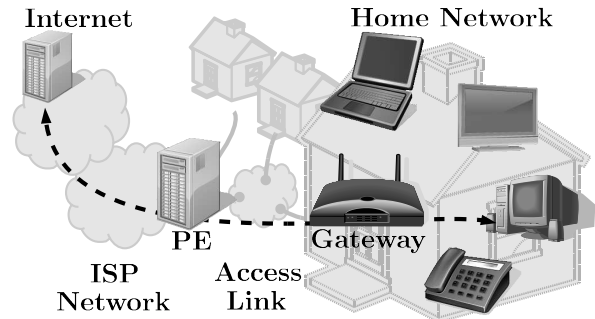

Figure 1: Example of home network Internet access.

access as well as the home network. A better understanding of home network performance and techniques that can reliably identify when the home network is the cause of performance disruptions will empower home users. For example, if a user knows that her downloads are too slow because of her TV, then she may switch off the TV during the download or upgrade her access speed. Otherwise, if the user is sure that the problem is not in her home, then she can confidently call her provider's hotline to complain.

Unfortunately, the study of home network performance is challenging. There are hundreds of millions of homes connected to the Internet with a large variety of network configurations and services. Thus, directly measuring a *representative* set of homes is practically unfeasible. In addition, even if we could place monitoring tools inside many homes (for instance, as in the grenouille project [4]), we cannot have full control of the home networks to understand their contribution to Internet performance. Alternatively, we could use an analytical or simulation model of the home gateway to understand how different devices and services in the home compete for bandwidth. However, packet scheduling in the gateway depends on vendor implementation decisions that are not publicly available.

In light of these issues, this paper uses a controlled testbed to emulate a home network connected to a DSL provider with triple-play service. This testbed (described in Section 2) allows us to study independently the effect of TV, phone, competing data downloads and uploads as well as the in-home network technology on end-to-end performance. Our main findings are:

1

1. **We quantify the significant impact that a home network can have** on end-to-end performance (Section 3). For example, when a user in the home is uploading a file, the RTTs of other users in the same home can increase by as much as one second and HTTP download rates reduce by half.
2. Identifying that the home network is the cause of performance degradations is not trivial (Section 4). Both existing diagnosis tools and simply probing the home gateway are not sufficient.
3. The technique of sending bursts of packets to the home gateway is promising (Section 5). Our results show that packet bursts can identify scenarios that may have an impact on end-to-end performance.

Section 6 ends this paper with a discussion of our plans to integrate different active and passive monitoring techniques into a tool that can reliably identify performance degradations in the home network.

2. EXPERIMENT DESCRIPTION

We study the effect of the home network on end-to-end performance using controlled experiments. The complexity of the triple-play service architecture prevents us from using a completely local testbed environment. Hence, our study uses a combination of a controlled testbed to emulate a home network, a controlled distant server, and a commercial access network, which provides triple-play services. This section presents our test cases, then it describes our testbed and the metrics we use to capture end-to-end performance as seen by an end-host.

2.1 Test cases

We independently study the most common home services: voice, TV, data upload and download. We test how five usage scenarios affect end-to-end performance as seen by a *probing computer*.

- *Idle* is when no other computer or application **in the home** uses the network. We expect to reach the best performance in this scenario. As such, it serves as reference.
- *Phone* is when a home user performs a phone call over the Internet. During the phone call, we play music to ensure that no silence detection stops the data stream.
- *TV* represents the scenario where a set-top-box streams a live video from the ISP.
- *DL* captures the case when a *competing computer* in the home downloads a file. We emulate the download with the netcat command to a remote server. We verify that changing the server location does not change the results as long as the server has good connectivity. All the computers use the same medium to connect to the gateway.
- *UL* is like DL, but the competing computer uploads a file instead.

We emulate two common home environments:

- *Ethernet environment:* we connect the computers to the gateway with full duplex 100 Mb Ethernet.
- *WiFi environment:* we connect the computers to the gateway with 56 Mbps 802.11g. Prior work [8] shows that the WiFi in the home is inherently complex. Our goal is not to study extensively WiFi performance in home networks. We focus on the case where WiFi is expected to work efficiently, i.e., when computers are close to the access point.

These scenarios and environments allow us to study ten configurations independently.

2.2 Testbed

Our testbed has an emulated home and a controlled server connected via a commercial ADSL service. The *emulated home*, located at the Technicolor lab in Paris, consists of a commercial gateway (which is both a modem and a router), a phone, a TV, and two computers. The gateway is connected to the Internet using an ADSL2+ line from France Telecom. According to its web interface, the gateway is synchronized at 7,650 kbps downstream and 620 kbps upstream at the time of the experiments. The gateway has two Ethernet 100 Mbps ports and an RJ11 port to plug a phone. A separate device, the set-top-box, is in charge of subscribing to and decoding television streams. The set-top-box is connected to one of the Ethernet ports. We perform all the measurements from a commodity business laptop connected to the Internet via the gateway. We call this laptop the *probing computer*. It has a dual core CPU at 1.80 GHz, 2 GB of RAM and runs Linux 2.6.32. For WiFi experiments, we use an external **PCMCIA** Atheros card with the ath4k driver. For UL and DL scenarios, we a use a second computer. The controlled *distant server*, located at LIP6 in Paris, is the end point of our end-to-end measurements. The path between the distant server and the emulated home is 20 hops long and stable throughout the experiments. It leaves France Telecom's network and enter Cable&Wireless in Madrid, then it returns to Paris through Renater.

2.3 End-to-End Performance Metrics

We study two metrics of end-to-end performance: the Round-Trip-Time (RTT) and the HTTP download speed. RTT captures the effect of the home on delay sensitive applications, whereas HTTP downloads represent bandwidth intensive applications.

We measure RTTs to the distant server with the ping command-line tool. We wait 500 milliseconds before each ping request. We obtain distributions from 100 pings of size 64 bytes. Our experiments with larger pings are similar. We observe no dropped packets during the whole RTT measurements.

For HTTP downloads, we use the wget command-line tool to download a file of 24 MB **with a single TCP connection**.
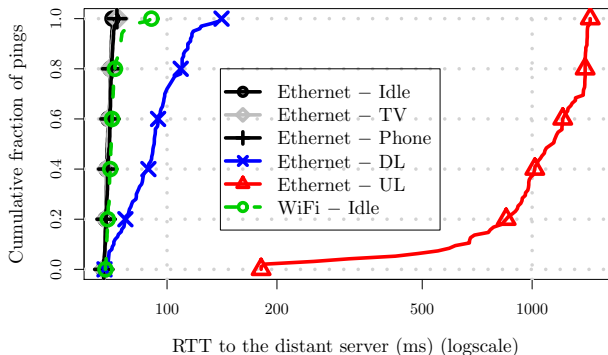
Figure 2: CDF of RTT to a distant server

Figure 3: CDF of end-to-end HTTP download rates

The file contains random bits to prevent transport compression. We verify that the distant server actually uploads the file (i.e., no cache in the ISP network serves the file). We use large files when measuring the bandwidth to minimize the effect of TCP slow start. As a result, each measurement takes time, so we only do 20 repetitions.

When performing a measurement run, we first pick a metric and do all the experiments as close as possible in time for the different scenarios. Then, we test all cases for the next metric. Performing measurements of a metric back-to-back ensures that the conditions of the ISP network are similar, and most of the differences we observe come from the emulated home. A run for RTT measurements takes around twenty minutes, whereas it takes two and a half hours for HTTP downloads.

3. END-TO-END PERFORMANCE

It is expected that the home network can impact end-to-end performance. This section quantifies this impact for RTTs and HTTP downloads under the scenarios described in Section 2.1.

3.1 Round-Trip Time

Figure 2 presents the cumulative distributions of RTTs from the probing computer to the distant server in six configurations (we omit the other WiFi scenarios because the conclusions are similar to their Ethernet counterparts). The RTTs range from 60 ms to more than 1 s, which shows that end-to-end performance of a computer in a home depends strongly on home usage. The highest impact on end-to-end performance comes from a competing computer in the home. The RTT can reach 120 ms when a competing computer does a download. This impact is even higher for uploads, RTTs are never lower than 180 ms and even larger than 1 second 60% of the cases. Uploads have a larger impact than downloads because of the high asymmetry of the ADSL line with a much lower uplink rate. The other noticeable difference is between WiFi and Ethernet. For Ethernet/Idle, RTTs are always close to 60 ms. For WiFi/Idle, RTTs have more variance and can reach larger values (up to 80 ms). **Note that to save space we have put many curves in Figure 2 and**
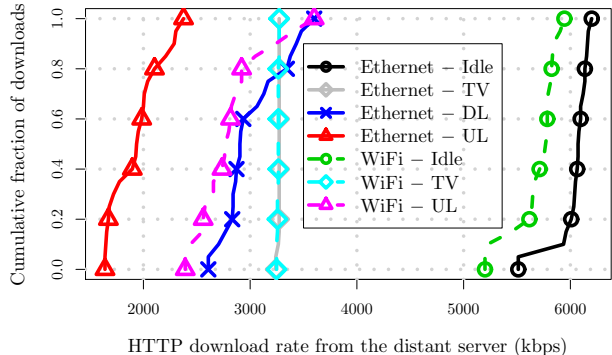
hence due to the scaling the effects of wireles cannot be easily seen on that figure. The effect of phone or TV on RTT is less significant. These services affect the variance, but not the range of RTTs.

As a result, a user with WiFi or competing computers at home may suffer long delays and jitter, which can degrade the performance of applications like online games or voice.

3.2 HTTP downloads

Figure 3 shows the distribution of HTTP download rates from the distant server for several configurations (we omit the results for the phone, because it has practically no effect on HTTP downloads). The highest achievable download rate of all scenarios is 6,200 kbps, which is 10% lower than the rate advertised by the gateway. This 10% difference is because the gateway shows a link-layer rate, which includes the encapsulation overhead. These rates also indicate that, in the Ethernet/Idle case, the bottleneck is the access link. This observation confirms the result of Dischinger et al. [3].

TV and competing downloads or uploads have a large effect on the HTTP download rates. Presence of TV limits the end-to-end bandwidth to 3,200 kbps, which doubles the file transfer time. This degradation is the same in WiFi and Ethernet (the two lines fall exactly on top of each other). Similarly to the results for RTTs, uploads from another computer in the home presents the highest impact on HTTP downloads (rates are as low as 1,500 kbps). The impact of downloads is similar to that of TV but with a higher variance. Interestingly, the upload scenario has higher HTTP upload rates in WiFi than in Ethernet. We conjecture that the contention in WiFi prevents congestion on the access link. WiFi alone can reduce the bandwidth by up to 500 kbps.

In summary, our controlled experiments show that the home network can have a significant effect on end-to-end performance. Competing computers in the home are the most important source of degradation. WiFi also affects RTTs and the TV has a large impact for HTTP downloads.

4. IDENTIFICATION

This section shows that existing diagnosis tools cannot reliably identify that the home is the cause of performance

Scenario	TTL	RTT (ms)		
Ethernet				
Idle	1 (Gateway)	0.541	0.365	0.331
	2 (PE)	40.1	40.2	39.4
UL	1 (Gateway)	0.498	0.577	0.349
	2 (PE)	398	483	527
WiFi				
Idle	1 (Gateway)	1.78	1.12	3.48
	2 (PE)	76.4	44.7	44.1

Table 1: Example of Traceroute output

Env.	Scenario	Uplink (Kbps)	Downlink (Mbps)	Latency (ms)
Ethernet	Idle	530	5.9	130
	Phone	360	4.9	130
	TV	530	3.3	130
	DL	520	4.7	160
WiFi	Idle	530	5.8	140

Table 2: Extract of Netalyzr output

degradation. Then, it shows that simply probing the gateway does not help.

4.1 Existing tools

There are many network diagnosis tools. We take two tools as example: Traceroute, which is often the first step in network diagnosis, and Netalyzr [1], which combines a number of advanced tests to diagnose network performance.

4.1.1 Traceroute

Traceroute sends packets with increasing Time-to-Live (TTL) to discover a path between a source and a destination host. Each router that forwards a probing packet decreases the TTL value. When the TTL reaches zero, the corresponding router drops the packet and sends an ICMP error message to the source. It is thus possible to estimate the RTT from the probing computer to each hop of the path to the destination.

We use Traceroute from the probing computer to the distant server with the default of three probes per hop. For each measurement, Traceroute discovers the same sequence of IP addresses for the first two hops: the *gateway* and the Provider Edge (*PE*). Although the values vary for different measurement runs of a given hop, we check that the general conclusions are consistent across runs.

Table 1 presents an example of the RTTs of the first two hops, for three configurations (we focus on the configurations that had the highest impact on RTTs in Section 3). By comparing the Ethernet/Idle case with WiFi/Idle case, we see an increased delay to the gateway. In general, we observe higher RTTs in all WiFi experiments. In the UL case, however, the RTTs increase by a factor of ten, but between the gateway and the PE, not to the gateway. Unfortunately, both cross traffic originated at the home and at the access network can explain congestion between the gateway and the PE. Therefore, **our results confirm the intuition that** Traceroute cannot tell for sure that the origin of the problem is the home network. It can only help in cases where performance degradation is due to WiFi.

4.1.2 Netalyzr

Netalyzr [1] is a web-based diagnosis tool. It performs many tests from a probing computer with the help of a distant server and generates a report for the end-user. An

execution of the tool takes several minutes to complete. Among other features, Netalyzr checks some configurations and classical security holes. It also provides many detailed hints to help the end-user interpret the tests. The reported metrics can be classified in two kinds: configuration tests (open ports, presence of HTTP proxy, etc.) and performance tests (like uplink and downlink bandwidth, or, latency). We focus here on the performance tests.

We evaluate Netalyzr in our testbed. Note that under the UL scenario over Ethernet, the applet page always times out, and hence no result can be presented for this case. Table 2 presents the results for Netalyzr's performance tests. It shows that Netalyzr measures a difference of download speed in presence of TV and upload speed in presence of phone. The latency measurements, however, present no significant difference, probably because values are rounded. There are also little differences between any given Ethernet scenario and its Wireless equivalent. Interestingly, Netalyzr measures little impact in presence of a competing download, except on latency, which is in contradiction with our findings in Section 3. Differences in results may come from the differences in methodology: Netalyzr uses UDP where we use TCP (for bandwidth) and ICMP (for latency), and the distant servers are different. This result confirms the impact of the home network on end-to-end performance, and that Netalyzr can detect it. However, Netalyzr does not explicitly attribute the performance degradation to the home network.

Similar to Netalyzr, we also tried Network Diagnostic Tool[1] as well as Network Path and Application Diagnosis[2]. Without exception, all the tools measure end-to-end performance variation between the scenarios, but they cannot identify that it comes from the home. It is important to extend these tools with techniques that explicitly identify performance problems at the home network.

4.2 Probing the gateway

The first solution to identify that the home network is the cause of performance degradation is to combine the existing troubleshooting tools with extra probing techniques. The idea is to perform extra tests to quickly decide if the home network is the source of end-to-end performance degradation. We present two candidate solutions to this purpose: pings to the gateway and using UPnP. The gateway archi-

[1]www.internet2.edu/performance/ndt/
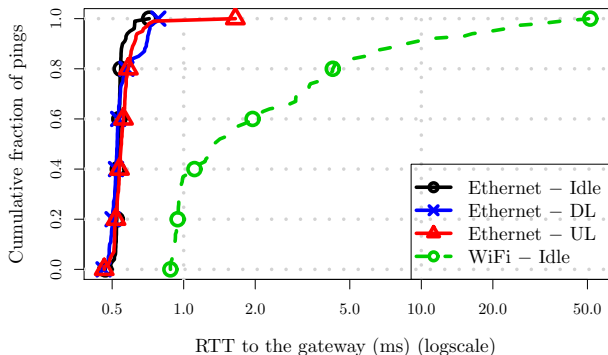[2]www.psc.edu/networking/projects/pathdiag/

Figure 4: CDF of ping to the gateway.

tecture prevents us to directly probe the set-top-box or the phone, thus, the only way to identify the presence of phone or TV is to probe the gateway.

4.2.1 Direct probing to the gateway

The simplest identification technique is to directly ping the gateway. We conduct an experiment similar to the end-to-end RTT measurements of Section 3. The only difference is that, instead of targeting the distant server, we send the pings to the gateway. Any delay that a direct ping experiences comes from the home.

Figure 4 plots the distribution of the RTTs from the probing computer to the gateway (we omit results for the other scenarios, because they show no significant deviation from their idle counterpart). For Ethernet, RTTs range from 0.5 ms to 1 ms, but almost all RTTs are close to 0.5 ms. For WiFi, the RTTs vary from 0.9 ms to 50 ms and we measure 20% of surprisingly high RTTs (i.e. larger than 10 ms). In Ethernet, the curves for the UL and DL scenarios do not deviate from the idle case. Similar experiments with different packet sizes do not change the results significantly. The fact that the RTT to the gateway does not depend strongly on the packet size indicates that most of the delay is spent scheduling and not on transmission of packets.

Consequently, direct probing of the gateway cannot identify all instances of bad performance originated at the home, unless degradation is only due to WiFi. This conclusion explains why Traceroute is not enough to identify the competing upload in Table 1, but accurately identifies WiFi.

4.2.2 UPnP probes

Protocols such as UPnP exist to let autonomous computers control the home. For our study, we use the UPnP WAN-CommonInterfaceConfig profile [10]. This profile allows an end-host to query the number of packets and bytes transferred on the WAN interface of the gateway.

In theory, with UPnP, the probing computer should be able to compare the number of bytes sent by the gateway to that sent by its applications. If the two numbers differ significantly, it means that another device is competing for the access link bandwidth. Unfortunately, in practice, UPnP is

not a tool that can be trusted 100%. Even in a small experiment with only three popular gateways in France, we find a number of issues.

1. In one gateway, UPnP is not available.
2. In another, UPnP is not enabled by default. In addition, the query always returns the same (hard coded) value for the count of bytes or packets transferred.
3. In the third gateway, the counter matches the number of data packets, but it misses packets sent by the set-top-box and the phone. This behavior is probably due to the access network architecture. Network operators may want to separate phone, TV, and data traffic on different virtual interfaces (e.g., virtual circuits in ATM). In this case, UPnP monitors only the data channel, while packets transferred on the other channels may impact the use of the access link.

This experiment indicates that UPnP is not reliable enough to be the only test of home performance but it can be useful when available.

5. CANDIDATE TECHNIQUE: BURST

Results so far are not conclusive. Home networks have considerable impact on end-to-end performance, but it is not easy to identify if the home is the source of performance degradation. We now discuss the use of packet bursts to identify if the home is the performance bottleneck.

5.1 Description

The probing computer sends a burst of ping requests. Each request requires an answer from the gateway. If the gateway receives traffic from other devices, it has to dedicate resources to it, which in turn delays the answers of some pings. Delays accumulate because packets are sent back-to-back. We then extract the largest RTT of the sequence, which accumulates the most delay in presence of cross traffic. To determine if the home network affects end-to-end performance, we send a burst to the gateway and compare its maximum RTT to that of the Idle case. If the difference of these two RTTs is significant, it means that there is some activity on the gateway. Activity on the gateway is likely to be network related, and hence may harm end-to-end performance. This burst technique needs baseline measurements, i.e., measurements when we are sure that there is no scheduling delay in the gateway.

Sending a large burst of packets might sound worrisome, but no packet reaches the ISP network. Clearly the technique might disturb legitimate applications. This concern can be addressed by starting with a small burst and increasing its size iteratively.

5.2 Evaluation

Our goal is to present a preliminary evaluation of the feasibility of the packet burst technique, not to perform a thorough parameter sensitivity analysis. We perform our evaluation with bursts of 1,000 packets. Between each burst we
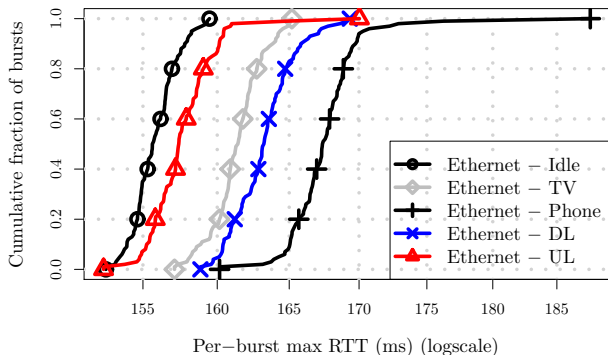
Figure 5: CDF of the per-burst maximum RTT

wait a guard time of several seconds to ensure that the buffers of the gateway and the probing computer are empty.

The time series of each burst (not shown), for the Ethernet environment and irrespective of the scenario, shows that individual RTTs increase almost linearly with the packet index. All the requests after the 700th receive no answer, which indicates that the subsequent requests were dropped by the gateway. For WiFi, the time series of the individual RTTs is more variable within a burst. We observe more packet losses, and there is no longer a clean linear increase of the RTT with the packet index.

In Figure 5, we plot the CDF of the per-burst max RTT (i.e., one point represents one burst) for the Ethernet environment. All the per-burst max RTTs are lower than 160 ms for the Idle scenario. On the other hand, for the TV and DL scenarios, only 20% and 5% respectively are under 160 ms. The Phone scenario never yields a max RTT under 160 ms. There is very little overlap between the Phone, TV, DL scenarios and the Idle case. Hence, if we can get measurements of the idle case beforehand, we should be able to distinguish when the home is overloaded. Only the RTTs in the UL scenario are harder to differentiate from that in the Idle case, although there is a consistent increase of 1.5 ms on the per-burst max RTTs.

The effect of competing download on the per-burst max RTT is larger than that of TV, which is larger than that of competing upload, because in the download and TV cases the gateway has more packets to forward. The exception is the phone which has a larger effect than any other scenario. An explanation could be that the gateway has to transcode voice packets, which consumes CPU cycles. Another explanation is that some gateways implement a QoS mechanism that privileges the handling of voice traffic. Unfortunately, we have not yet been able to validate these hypotheses.

We omit the results of packet burst on the WiFi environment, because the burst technique cannot differentiate among different scenarios. All the scenarios show RTT values between 300 ms and 700 ms. We conjecture that the burst technique is not as effective for WiFi because of the half-duplex nature of the physical layer. Both probes sent

and received compete for the medium, thus, it is hard to keep the gateway busy with incoming packets.

This work shows that bursts can help distinguish between scenarios in Ethernet but is not satisfying in WiFi. In future work, we will characterize how to tune the burst parameters, and we will analyze limitations arising from gateway hardware and software configurations. There is a tradeoff between the number of packets per burst and the number of bursts required to identify each scenario. We also plan to investigate techniques that can identify scenarios that affect end-to-end performance in WiFi environment.

6. CONCLUSIONS

The home network is a complex environment that deserves attention from the research community. We show that typical home configurations can have a significant impact on end-to-end performance. **We will study more home configurations (e.g., connecting computers with power-line and introducing gaming or peer-to-peer applications) in future works.** Consequently, end-host diagnosis tools need to explicitly take the home network into account. Directly probing the gateway cannot identify all instances of performance degradation from the home network. Even protocols designed to enable home devices to cooperate such as UPnP suffer from implementation issues. Packet bursts are promising to differentiate when the gateway is busy in some cases. Given that no single technique can identify all possible sources of performance degradation originated at the home network, we envision a hybrid approach that combines active packet bursts with passive UPnP queries.

7. REFERENCES

[1] The ICSI Netalyzr. http://netalyzr.icsi.berkeley.edu/.
[2] D. Croce, T. En-Najjary, G. Urvoy-Keller, and E. Biersack. Capacity estimation of adsl links. In *Proc. CoNEXT*, Dec 2008.
[3] M. Dischinger, A. Haeberlen, K. P. Gummadi, and S. Saroiu. Characterizing residential broadband networks. In *Proc. IMC*, Oct 2007.
[4] Grenouille. Grenouille. http://www.grenouille.com/.
[5] D. Han, A. Agarwala, D. G. Andersen, M. Kaminsky, K. Papagiannaki, and S. Seshan. Mark-and-sweep: Getting the inside scoop on neighborhood networks. In *Proc. IMC*, Oct 2008.
[6] Internet World Stats. http://www.internetworldstats.com/dsl.htm.
[7] G. Maier, A. Feldmann, V. Paxson, and M. Allman. On dominant characteristics of residential broadband internet traffic. In *Proc. IMC*, Oct 2009.
[8] K. Papagiannaki, M. Yarvis, and W. S. Conner. Experimental characterization of home wireless networks and design implications.
[9] M. Siekkinen, D. Collange, G. Urvoy-Keller, and E. Biersack. Performance limitations of adsl users: A case study. In *Proc. PAM*, Apr 2007.
[10] UPnP Forums. WANCommonInterfaceConfig:1, Nov 2001.