Packet Capture on Home Gateways: Is it feasible?

Ahlem Reggani, Fabian Schneider

Packet Capture on Home Gateways: Is it feasible?

Ahlem Reggani
UPMC Sorbonnes Universités and CNRS
ahlem.reggani@lip6.fr

Fabian Schneider
UPMC Sorbonnes Universités and CNRS
fabian@ieee.org

ABSTRACT

Residential Internet access and home networks recently receive a lot of attention from the research community, the regulatory agencies, and ISPs. Home gateways provide Internet connectivity for increasing numbers of devices, serving several purposes such as telephony, media-streaming, data, or gaming. Thus, troubleshooting and monitoring home networks is 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 & US) rely on active measurements from the home gateway. In this paper, we want to explore the next step: Passive measurements on home gateways. We experimentally analyze resource consumption of running `tcpdump` capturing traffic forwarded by different home gateways. We find that it is feasible to do so for throughputs in the order of typical DSL access speeds. Yet, capturing fully loaded Fast Ethernet or faster links is more challenging.

1. INTRODUCTION

Recently, home networks experience more and more attention. 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 divers requirements and challenges that home networks pose.

This trend is also reflected in research (Section 6). Researchers want to improve troubleshooting at home and understand the characteristics of home network usage. Projects like Netalyzr [6], HostView [5], and RIPE Atlas [10] aim at understanding the network performance at home. Yet, these tools suffer from the unobservability of activities of other devices happening inside the home that can bias the results [2]. Therefore, Calvert et al. [1] propose to measure from the home gateway, recording events on the home network. They report that preliminary test of capturing typical home traffic with `tcpdump` misses up to 10 %. SamKnows deploys home gateways in order to repeatedly measure the access link performance [11]. The EU projects Nanodatacenters [8] and Figaro [3] design the gateway for next-generation Internet services and also include monitoring on the home gateway.

Home gateways connect the home network to the Internet. They allow for being always connected and are low cost, which translates into offering only limited resources compared to laptops or workstations.

Given the trend to move the measurement from the end-host to the home gateway, the question arises if home gateways are capable of executing those measurements. While active measurements usually do not consume a lot of resources, passive measurements are costly in terms of CPU, memory, and disk resources. Passive measurement applications include recording packets to disk, exporting flow records (e. g., NetFlow), or DPI solutions for application detection, characterization, or security.

Currently, passive measurements in the home are used to troubleshoot network problems (e. g., [5]) and characterize network usage [1]. Yet, passive monitoring on home gateways enables network traffic inspection and block malicious traffic. This can be beneficial for several reasons, *(i)* the user's privacy is preserved, *(ii)* the user receives immediate feedback on corrupted devices and *(iii)* no host outside the home can be infected. Moreover, passive measurements on the gateway enable new billing models for home customers such as per traffic class/flow pricing.

We, in this paper, aim at systematically evaluating the performance of a key component of all passive monitoring tools: Packet capture and delivery to the analyzing application in user-space. In our testbed (Section 2) we experimentally compare the performance of five different gateways, while capturing packets with `tcpdump`. As gateways we select the well-known Linksys WRT54GL, the off-the-shelf D-Link DIR-615, and customized embedded systems with AMD Geode, Marvel Kirkwood (ARM), and Intel Atom processors (sorted by increasing performance). Using tools (Section 3) to generate different levels of network load, which correspond to popular broadband access speeds and to monitor the system utilization, we evaluate and explain the results. Our experimentation method (Section 4) enables automatic execution and repetition of all experiments.

Table 1: Evaluated Hardware (Abbreviations: FE – 100 Mbps Ethernet, GE – Gigabit Ethernet)

Name	Manufacturer & Model	Processor @Speed	InSet	RAM	NICs	Storage	OS or FW/Kernel
MIPS1	Linksys WRT54GL	Broadcom 5352 @200 MHz	MIPS	16 MB	5*x*FE	4 MB Flash	Tomato/2.4.20
MIPS2	D-Link DIR-615	RaLink 3052F @384 MHz	MIPS	32 MB	5*x*FE	4 MB Flash	DD-WRT/2.6.23
Geode	Soekris net5501	AMD Geode LX @500 MHz	i586	512 MB	3*x*FE	80 GB SATA	Debian/2.6.26
ARM	OpenRD Ultimate	Marvell Kirkwood @1.2 GHz	ARM	512 MB	2*x*GE	1 GB USB	Debian/2.6.32
ATOM	TranquilPC T2WHSA2	Intel Atom 330 2@1.6 GHz	i686	2 GB	3*x*GE	500 GB SATA	Debian/2.6.26

Our results indicate (Section 5) that all of our gateways except the elder WRT54GL are able to forward and capture the generated traffic up to a network load of 20 Mbps. The top three gateways can also sustain 100 Mbps, while still leaving CPU resources for traffic analysis and packet processing.

2. EXPERIMENT SETUP

In this section, we first describe the testbed used for our measurements, then we explain the selection of home gateways under test.

Figure 1 represents our testbed setup. We use two edge machines: *Server* and *Client*[1]. We selected five representatives for home gateways, named MIPS1, MIPS2, 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.

Home gateways exist in different models with different features. For our experiments, we select five models, whose hardware and software details are shown in Table 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 and MIPS2) 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 firmwares 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. [4]. This system represents what is currently sold as home gateway in stores and is a close sibling of the gateway (Netgear WNR3500L, 480 MHz MIPS 74K processor) used by SamKnows [9].

- The Soekris net5501 (Geode) is, according to AMD, the ideal family for set top boxes, residential gateways, and embedded systems. A similar system is used for the BISMark measurement deployment [1, 15] 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 [14]. 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.

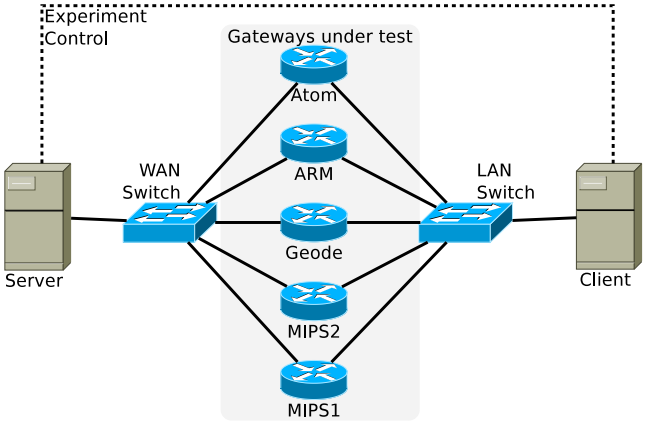

Figure 1: Experiment Setup

3. 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

[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.

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`[2] 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 to disk nor analyze the data[3]. 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 not offering storage at all (MIPS1, MIPS2).

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. Thus, the take away lesson is not to rely on the assumption that standard software behaves identical on different platforms.

4. 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−idle) value. Furthermore, for scenarios with `tcpdump`, we also measure the fraction of packets captured on the gateway.

[2]On systems running Debian Linux we use the tool `sar` and on the MIPS architectures we use an `awk` script, which can be downloaded from: `http://cmon.lip6.fr/~fabian/cpusage.awk`

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

Table 2: Maximum throughput with 0 % E2E-Loss. (No CPU monitoring, no `tcpdump`)

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

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 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
   (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

5. RESULTS

This section shows our results for all scenarios. We start with the baseline scenario, then move on to the `tcpdump` scenarios, and finish with a discussion of the insights.

5.1 Baseline Scenario

Figure 2 illustrates the results of the no-tcpdump scenario. The plot on top shows the E2E-Loss and the bottom plot shows the CPU utilization for each gateway (different colors/line-types/symbols) and different throughputs on the logarithmic x-axis. In this figure, we can see that the ATOM gateway (circles) does not lose any packets and its CPU remains idle even at 832 Mbps traffic rate. Apparently, the maximum sustainable rate (recall Table 2) is not determined by the CPU utilization. Likewise the Geode (pluses) and the MIPS2 (crosses) are also not limited by their CPU, reaching 67 % and 80 % CPU utilization at maximum achievable

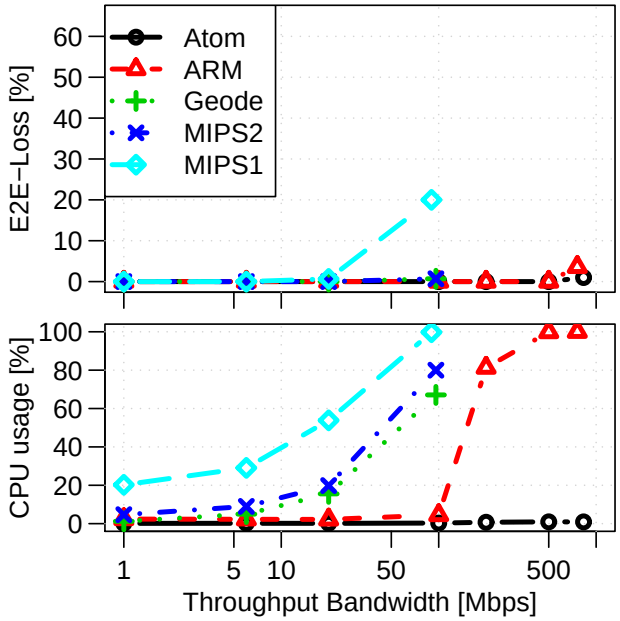

Figure 2: Evaluation Results from the no-tcpdump scenario. E2E-Loss (top) and CPU usage (bottom) vs. traffic bandwidth (x-axis in logscale).

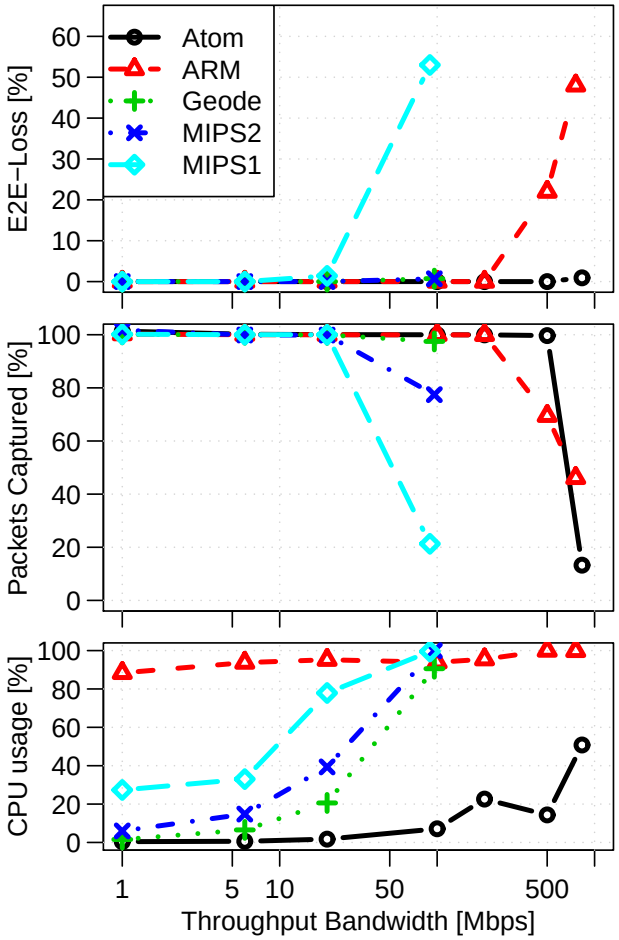

Figure 3: Evaluation Results from the tcpdump-68 scenario. End-to-End loss (top), Packets captured (middle), and CPU utilization (bottom) vs. traffic bandwidth (x-axis in logscale).

throughput. A likely explanation is a hardware limitation of the gateways interfaces.

Contrary both the ARM gateway (triangles) and the MIPS1 (diamonds) both consume all their CPU resources at maximum rate, indicating that the CPU is the limiting factor for the throughput here. Furthermore, these gateways also suffer from E2E-Loss at maximum sustainable rate: 3.6 % for the ARM and significant 20 % for the MIPS1. The cause of these losses is likely due to the additional overhead of CPU utilization monitoring. Yet, we assume the CPU monitoring is not by itself responsible for the losses. It rather pushes CPU utilization over its limit, causing the problem of receive live-lock [7] where incoming interrupts preempt the kernel and user-space application, which in turn causes more context switches.

The CPU utilization at 20 Mbps allows to compare the performance of all the gateways and rank them. The MIPS1 has to invest by far the most resources and is thus has the worst performance.

5.2 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 3, for this scenario we expectedly find overall increased CPU utilization. The results for capturing full packets (1500 bytes) are shown in Figure 4. 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 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 `se-`

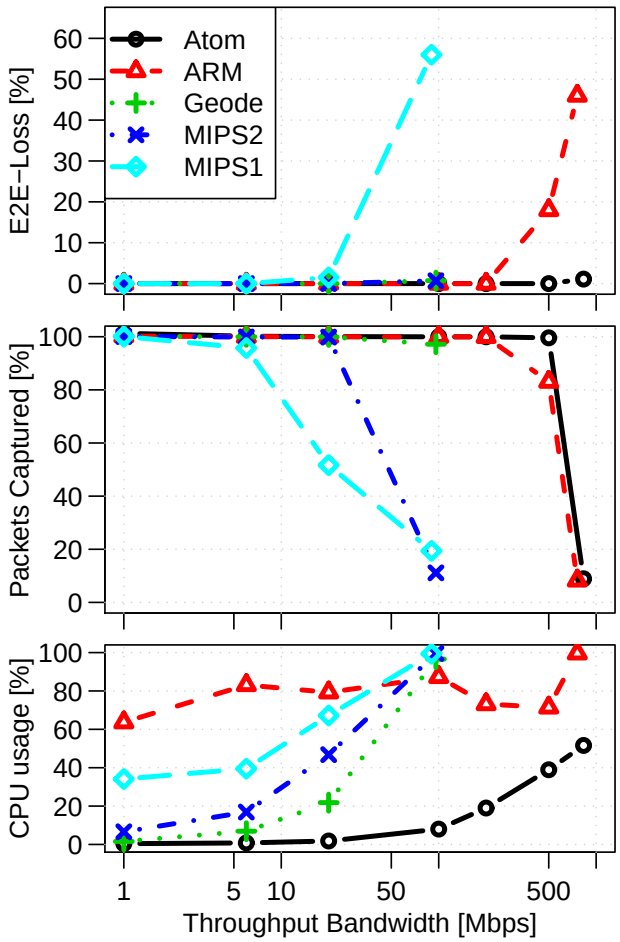

Figure 4: Evaluation Results from the tcpdump-1500 scenario. End-to-End loss (top), Packets captured (middle), and CPU utilization (bottom) vs. traffic bandwidth (x-axis in logscale).

`lect()` or `usleep()`[4]. 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.

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.

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

Contrary, the MIPS2 does not manage to capture all the packets at maximum bandwidth, 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.

5.3 Discussion of Results

To sum up 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 throughputs 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) or if the home network itself (up to 1 Gbps) should be monitored, we can still achieve reasonable performance with the ARM or the ATOM. Even a Geode-like box with Gigabit-Ethernet support could 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 throughputs, one of the ATOM's cores is still idle. Or, when looking at 20 Mbps (typical DSL speed) even the MIPS2 only uses half of its CPU.

To interpret these results we utilize insights from our previous work on capturing performance of server architectures [12, 13]: 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, snort[5] or Bro[6] can be executed on our gateways.

6. RELATED WORK

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. [1] underline the need for a "Home Network Data Recorder" to allow a more detailed understanding and troubleshooting of the home networks. Their work is the first to propose passive measurements on a home gateway. They base their concept on the NOXbox, a system very similar to our Geode. 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. Yet, they neither

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

report on system utilization nor systematically vary the workload. This inspired us to perform this study.

Several other studies measure network performance from the end hosts [5, 6, 10], partly using passive measurements. Yang et al. [16] found that users prefer tools already installed on the end-host or the gateway for the purpose of troubleshooting their home network. As typically `tcpdump` is not pre-installed on gateways, maybe the findings of this paper foster the integration of `tcpdump` on gateways.

Hätönen et al. [4] compared 34 off-the-shelf home gateways in terms of maximum throughput, binding timeouts, and number of concurrent bindings. Because the MIPS2 performed well in their tests we added it to our testbed.

7. CONCLUSION

In this paper, we evaluate the performance of home gateways while capturing packets using `tcpdump`. We discard the captured packets and do not analyze them any further. We compare five different gateways representing four different architectures (MIPS, AMD Geode, ARM, and Intel Atom). We find that all our candidates except the 6-year-old Linksys WRT54GL are capable of forwarding and capturing traffic up to 20 Mbps without losses and drops. Moreover, the non-MIPS architectures can deal with up to 100 Mbps, leaving comfortable resources for packet analysis and processing.

For future work we plan to better understand the limitations (e. g., by profiling the gateway in more detail). We also plan to evaluate several passive monitoring tools such as writing packets to disk, running a NetFlow exporter, or running a network intrusion detection system (NIDS). This will enable us to devise and explore possibilities to increase the capturing performance and lower the resource consumption of passive monitoring tools for gateways.

8. REFERENCES

[1] Calvert, K. L., Edwards, W. K., Feamster, N., Grinter, R. E., Deng, Y., and Zhou, X. Instrumenting home networks. *SIGCOMM Comput. Commun. Rev. 41*, 84–89.

[2] DiCioccio, L., Teixeira, R., and Rosenberg, C. Impact of home networks on end-to-end performance: controlled experiments. In *Proceedings of the 2010 ACM SIGCOMM workshop on Home networks* (2010), HomeNets '10, pp. 7–12.

[3] Figaro. `http://www.ict-figaro.eu`.

[4] Hätönen, S., Nyrhinen, A., Eggert, L., Strowes, S., Sarolahti, P., and Kojo, M. An experimental study of home gateway characteristics. In *Proceedings of the 10th annual conference on Internet measurement* (2010), IMC '10, pp. 260–266.

[5] Joumblatt, D., Teixeira, R., Chandrashekar, J., and Taft, N. Hostview: annotating end-host performance measurements with user feedback. *SIGMETRICS Perform. Eval. Rev. 38* (January 2011), 43–48.

[6] Kreibich, C., Weaver, N., Nechaev, B., and Paxson, V. Netalyzr: illuminating the edge network. In *Proceedings of the 10th annual conference on Internet measurement* (2010), IMC '10, pp. 246–259.

[7] Mogul, J. C., and Ramakrishnan, K. K. Eliminating receive livelock in an interrupt-driven kernel. *ACM Trans. Comput. Syst. 15* (August 1997), 217–252.

[8] Nanodatacenters. `http://www.nanodatacenters.eu`.

[9] 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.

[10] RIPE atlas. `http://atlas.ripe.net`.

[11] Samknows. `http://www.samknows.com`.

[12] Schneider, F. Performance evaluation of packet capturing systems for high-speed networks. Diplomarbeit, Technische Universität München, Munich, Germany, Nov. 2005.

[13] Schneider, F., Wallerich, J., and Feldmann, A. Packet capture in 10-gigabit ethernet environments using contemporary commodity hardware. In *Proceedings of the 8th international conference on Passive and active network measurement* (2007), PAM '07, pp. 207–217.

[14] Sheeva plug. `http://www.plugcomputer.org`.

[15] Sundaresan, S., de Donato, W., Feamster, N., Pescapè, A., and Teixeira, R. Benchmarking broadband internet with bismark. `http://www.cc.gatech.edu/~ssundar3/docs/bismark-internet2-102010.pdf`, Nov. 2010. Presentation at Internet2 Fall Members Meeting.

[16] Yang, J., and Edwards, W. K. A study on network management tools of householders. In *Proceedings of the 2010 ACM SIGCOMM workshop on Home networks* (2010), HomeNets '10, pp. 1–6.