CESNET Technical Report 1/2015

Overview of the Local Network Monitoring Projects and Tools

Radek Krejčí, Josef Hájek

Received 10. 3. 2015

Abstract

This technical report provides insight into the area of local network monitoring. We provide summary of current standardization activities in this area as well as of currently running projects and tools focusing on both the active and passive monitoring of local networks and last mile connections. The text brings a base knowledge to our following work — development of an office network probe.

Keywords: network, monitoring, IETF, BMWG, IPPM, LMAP, M-Lab, BISmark, Atlas, SamKnows, Turris

1 Introduction

CESNET has a long-time experiences [1][2][3] with network traffic monitoring on its backbone lines. The CESNET2 network is equipped with a dedicated flow monitoring probe at every external network line and also network routers provide flow information. The peak traffic of the most heavily utilized lines easily reach 15 Gb/s. Insight into the network traffic, provided by the network monitoring tools, brings unassailable benefits for network operators, especially in the field of network security [3]

On the other hand, monitoring only the backbone lines easily overlooks problems in local networks. Although such affairs have a significant impact of the end-user experience. This text provides an overview of a tools applicable in the environment of local networks for the network traffic monitoring as well as for measuring various network characteristics. Furthermore, the overview of currently active projects related to this area is provided. These projects can serve as a source of knowledge for a future development of monitoring/measurement tools at CESNET.

1.1 Local Network Environment

Though the CESNET experiences with the network monitoring, we have to remember that, in comparison to backbone networks monitoring, local networks environment has several differences that substantially affects the network monitoring methods.

Metering device

The end-user local networks are isolated, so a metering device must be deployed in each (or in as many as possible) network. This way a) the operator is able to get a complex view on end-users but also on their transport networks from hundreds or thousands of devices and b) each end-user is allowed to get precise

information related to their network. But the need of such a huge number of devices imposes requirements on a low cost of such a device. The cost naturally impacts the performance.

Observed data amount

The decreased performance is not usually a problem due to the decreased amount of data (packets as well as bytes) to observe. On the other way, the lack of relevant data can be an issue for the data analysis tools. Therefore, it is beneficial to get information from multiple observation points. Since analysis tools receives in this case not much data from the methods typically used in backbone networks (flow monitoring), it is legitimate to think about using multiple network monitoring methods not usual in backbone environment.

These two constrains are tightly connected. They cause the measurement process is much more distributed using simpler devices and methods.

1.2 Terminology

Local Network

We focus on end-user networks (houses, offices) with a limited number (tens at maximum) of different devices (PCs, printers, entertainment device, etc.). Such a network is very often separated from the outer network using network address translation (NAT).

Active Monitoring

The measurement device creates and injects some data into the monitored network and observes the reaction of the network or/and a specific network device. This approach allows to perform required tests whenever it is needed. On the other hand, it generates extra volume of an artificial network traffic. The examples of information discoverable by this technique are round-trip time (RTT) or jitter.

Passive Monitoring

The passive approach does not inject any data into the network line. The measurement device just silently watches all the data passing by. It records all the traffic or more often only a selected characteristics of the traffic passing by the device. This approach is typical e.g., for the flow monitoring.

2 Measurement Process Standardization

Currently, there are several standardization efforts focused on the measurement of a network characteristics. The purpose of these activities is to provide common instructions to make measurement results unambiguous and comparable. The following sections shortly describes network measurement standardization activities in the Internet Engineering Task Force (IETF), Institute of Electrical and Electronics Engineers (IEEE) and the Broadband Forum (BBF).

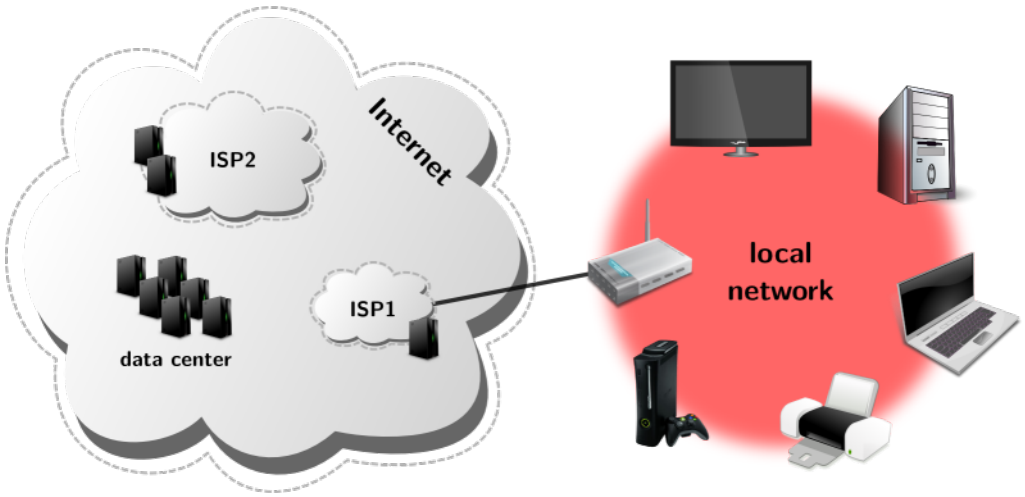

Figure 1. Local network.

2.1 IETF Benchmarking Methodology Working Group (BMWG)

Performance characteristics testing of the specific classes of network devices, systems or services is addressed by the IETF BMWG working already from 1989. The working group produces standards [4] focusing only on usage in a laboratory environment. They are not applicable for operational (local) networks.

2.2 IETF IP Performance Metrics (IPPM) Working Group

The IETF IPPM working group is over a long period (it started in 1997) focused on standardization of metrics and measurement methods applicable to the quality, performance and reliability of the IP networks and services provided by these networks. The lower network layers are out of scope of the group work. Defined metrics take into account impact of an operational environment to the measurement process and its results. The metrics defined by the working group include connectivity, one-way delay, one-way packet-loss, round-trip delay, round-trip packet loss or jitter. The complete list of the working group documents can be found at [5].

2.3 IETF Large-Scale Measurement of Broadband Performance (LMAP)

The LMAP working group is the youngest (started in 2013) IETF working group aiming to a network measurements. In general, it focuses on utilization of metrics defined by the IPPM working group for performance measurements of (wired as well as wireless) broadband access devices. The results of such a system provides accurate information for end-users as well as for their network providers.

The system being standardized includes large number of measurement devices deployed in different end-user's local networks and performing a specified set of defined tests. The data observed by the probes are consequently processed and analysed altogether in a data collector. The results provides specific and detail information for a single end-user. On the other hand, a comprehensive view to the data from the defined group of end-user networks provides a valuable information for the network providers or regulators. The standardization aims to the control

and planning the performed tests and processing the observed data. Although the passive as well as active measurement techniques are supported, definition of the specific metrics if out of scope if the IPPM working group. Working group documents, currently mostly as drafts, are available at [6].

2.4 IEEE P802.16.3

The IEEE organisation also produce activity in the network performance measurements area. The project P802.16.3, also known as *Standard for Mobile Broadband Network Performance Measurements* started in September 2012 and its goals are very similar to the goals of the IETF working groups mentioned above. However, the project is more focused on **mobile** networks (e.g., signal strength or energy efficiency measurements). Besides the active monitoring, the project contributors also consider passive monitoring as a way how to provide continuous information to the end users. More detailed information is available on the project web page [7].

2.5 BBF WT-304

Another very similar activity is provided as a working text WT-304 under the Broadband Forum. It was created to describe *Broadband Access Service Attributes and Performance Metrics* applicable for service providers, regulators and customers. The current text can be found at [8].

3 Running Projects

The following text provides an overview of the currently running projects focused on local networks monitoring. Those project make use of both the active as well as passive monitoring approaches.

3.1 Measurement Lab (M-Lab)

http://www.measurementlab.net/[1] M-Lab is an academic project providing an open, distributed server platform where the researchers can deploy their measurement tools to collect a real world data. The data are then released in the public domain. As in case of all the projects described hereafter, M-Lab goal is to provides independent and adequate information about a real parameters of the network to its end-users. However, with the publicly available data[2] it aims also to a research community.

The project continuously increase the number of a public servers collecting the data from the user measurements since early 2009. The client side is a (web, mobile, cli, browser plugin, etc.) application started on demand by a user to test their network connections.

The test suit is more a set of a publicly available open source tools (their description is included in Section 4). The main focus of the project remains on the

[1] `http://www.measurementlab.net`

[2] *https://console.developers.google.com/storage/m-lab/*[3]

server side collecting the measured data and providing them to a research community. Various libraries and tools developed/modified by the project members are available at GitHub [9].

3.2 BISmark

http://projectbismark.net/[4] BISmark is an university project (led by Georgia Tech and University of Napoli Federico II) supported byt the National Science Foundation and the Google. The project started in February 2011 and it should finish in January 2015. The main result of the project should by an OpenWRT-based platform for performing measurements of the local and ISP networks performance, which is almost the same goal as of the commercial SamKnows (see Section 3.4). However, there is a difference to the SamKnows – the BISmark allows end user to use the measurement platform also as a router (including WiFi), which is not possible in case of SamKnows Whiteboxes.

The BISmark platform base on a modified OpenWRT platform with added tools for active monitoring. Currently, there are[5] firmware images for Netgear WNDR3700v2, WNDR3800 and TP-Link WDR3600v1. The software source codes are available at GitHub [10].

Table 1. The SamKnows Whiteboxes technical parameters.

Device	WNDR3700v2	WNDR3800	TL-WDR3600
CPU	AR7161 (MIPS), 680 MHz	AR7161 (MIPS), 680 MHz	AR9344 (MIPS), 560 MHz
RAM	64 MB	128 MB	128 MB
Flash	16 MB	16 MB	8 MB
USB	1x 2.0	1x 2.0	2x 2.0
LAN	4x 1000 Mbps	4x 1000 Mbps	4x 1000 Mbps
WAN	1x 1000 Mbps	1x 1000 Mbps	1x 1000 Mbps

[4] `http://projectbismark.net/`

[5] *https://github.com/projectbismark/projectbismark/wiki/BISmark-firmware-installation*[6]

3.3 RIPE Atlas

http://atlas.ripe.net/[7] The RIPE Atlas project started in 2011 and it was one of the projects that led to the establishment of the IETF LMAP working group. The project builds the large measurement network with a big amount of small probes distributed around the world (however the project is aimed especially to Europe and Near East). Overviews of the measurements results are publicly available. More detailed information is available for the hosts of the probes. Currently, there are almost 7 000 of active measurement probes. However, the infrastructure is designed for more than 100 000 probes.

Besides the continuously performed tests controlled by RIPE, all users are able to perform a user-defined measurement. To start such a measurement, users have to pay by credits they can earn by hosting a probe, by donating (minimum price is 2048 EUR) the project or through a transfer from another user. Cost for a specific measurement depends on its type (resources required to perform it) and the number of delivered results (i.e. on a number of assigned probes multiplied by the measurement frequency). Currently available measurement tools includes ping, traceroute, DNS query, HTTP request and SSLCert, all on both IPv4 and IPv6. There is no tool for a passive monitoring. The user interface for defining the measurements is available from the project web page.

3.3.1 Technical Details

The probe is connected to the network as any other wired end device using RJ-45 connector. WiFi connection is not available (it is missing in v1 and v2 probes and turned off in v3 probes). The probes are powered via the USB cable.

Table 2. The RIPE Atlas probes technical parameters.

	probe v1/2	**probe v3**
Device	Lantronix Xport Pro	TL-MR 3020
CPU	MCF5208 (ColdFire), 167 MHz	AR7240 (MIPS), 400 MHz
RAM	8/16 MB	32 MB
Flash	16 MB	4 MB
WAN	1x 100 Mbps	1x 100 Mbps

The probes run slightly modified busybox[8]. It is extended by a schedulers

[7] `http://atlas.ripe.net/`

[8] *http://www.busybox.net/*[9]

(improved crond) eperd (that evolved from perd) and the eoogd, the daemon that runs measurements on demand. The description of other tools can be found either in available source codes [11] or in the article by Philip Homburg [12]. Another extension tools provided by the community are available at GitHub [13]. Here is the list of the most important ones:

Sagan
: Python API for accessing and processing the collected data from the Atlas probes.

Cousteau
: Python bindings to the Atlas API enabling to create user-defined measurements from Python applications.

CLI
: Command line manager for the Atlas API enabling to create user-defined measurements from the shell.

Atlas Toolbox
: Set of the Perl scripts to use the Atlas API.

3.4 SamKnows

http://www.samknows.com/[10] SamKnows Limited was founded in 2003 by Sam Crawford. The company operates internationally, though the headquarters is located in London, United Kingdom. SamKnows aims to monitoring and providing information about the end-users broadband connections. Currently, SamKnows pursues measurement within several national and international projects in EU, US, Brazil and Singapore. They declare (with no specific reference) a cooperation with a hardware manufacturers and ISPs around the world and more than 40 000 deployed measurement probes.

SamKnows provides the collected broadband performance information to the two groups of consumers. Firstly to the end-users running the local network where the probes are deployed and secondly to the ISPs and governments and regulators (such as Federal Communications Commission (FCC) in US or European Commission (EC) in Europe). End-users are provided with an accurate information about its broadband connection performance only for the price of the probe power supply and additional network traffic generated by the active measurement tools. On the other hand, ISPs or telecommunication regulators fund the projects SamKnows participate in and provides probes to the end-users and they are provided with the overall performance parameters of the broadband connections of the specific ISP or country.

[10] `http://www.samknows.com/`

3.4.1 Technical Details

The measurement probe is called Whitebox (despite the 3rd version is black). Technical parameters of the particular versions can be found in Table 3. Whiteboxes performs only the active monitoring utilizing the services of the closest SamKnows servers deployed around the world as well as a real-world Internet services (YouTube, Twitter, Skype, ...). The results are encrypted and sent to the SamKnows data collectors, where thay are analyzed and can be presented to the users, ISPs or government institutions.

Table 3. The SamKnows Whiteboxes technical parameters.

	Whitebox 1.0	**Whitebox 3.0**	**Whitebox 4.0**
Device	TL-WR741ND	TL-WDR3600	TL-WDR4900
CPU	AR7240 (MIPS), 400 MHz	AR9344 (MIPS), 560 MHz	P1014 (PowerPC), 800 MHz
RAM	32 MB	128 MB	128 MB
Flash	4 MB	8 MB	16 MB
USB	no	2x 2.0	2x 2.0
LAN	4x 100 Mbps	4x 1000 Mbps	4x 1000 Mbps
WAN	1x 100 Mbps	1x 1000 Mbps	1x 1000 Mbps
Meas. limit	100 Mbps	500 Mbps	850 Mbps

The software running on the Whiteboxes is based on OpenWRT with capability for the remote updates. On the other hand, users are not allowed to modify the system, running tools or any settings of the device. However, the modified source codes are available [14] under the GPL v2.0 license.

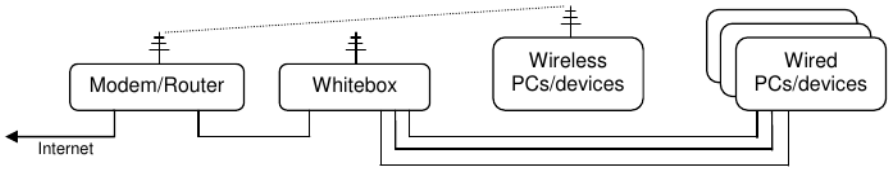

Figure 2. How to connect the Whitebox in the local network.

However the Whiteboxes are based on common (WiFi) routers available on today's market, they were modified to operate just as an Ethernet bridge, naturaly without Wi-Fi. Therefore, the Whiteboxes are supposed to co-exist with the user's router connecting the local network. If other wired devices are connected through

the Whitebox (Figure 2), it detects the network traffic and avoids running measurements when the user is using its connection. The WiFi capability of the Whitebox is used for the same purposes. More detailed description about the SamKnows technologies can be found at [15].

3.4.2 Leone

http://leone-project.eu/[11] The Leone is a project funded by the EU (FP7-ICT-317647). It is running from November 2012 until May 2015. The project implements and deploys various measurements according to the proposed standards defined by the IETF LMAP working group. SamKnows is one of the project participants, so the Leone project employs SamKnows Whiteboxes.

3.4.3 Measuring Broadband AmericaLeone

http://www.fcc.gov/measuring-broadband-america[12] SamKnows, and its Whiteboxes, also participate on a performance study of broadband service in the USA. The project Measuring Broadband America is covered by the FCC, that extended the project also to mobile devices [16].

3.5 Turris

http://www.turris.cz/[13] Project Turris is a service of protecting a local network. This project differs in several points from the other projects described so far. Users of the service obtain a Turris router (for technical parameters, see Table 4). Besides serving as a common router, it also passively monitors the passing network traffic. The main aim is not, as in case of other project, to get performance information, but a detection of security threads. The results of this monitoring is sent to the project collectors operated by CZ.NIC. As a reaction to the detected threads, Turris routers are updated to protect local networks of the participants.

Turris is probably the only currently running project performing the passive monitoring in home networks. The router is supposed to replace routers formerly connecting the participants' local networks. Furthermore, the router software is completely open source and CZ.NIC also encourages users to develop their own tools and extend the router functionality [17]. Besides the software source codes, the project also provides the design of the router hardware under an open source license. Basic technical parameter of the router are stated in the Table 4.

The router system is based on OpenWRT. The most important extension is distributed adaptive firewall which includes the ucollect framework working as a passively monitoring probe for a statistical analysis and detection of network anomalies.

[11] `http://leone-project.eu/`

[12] `http://www.fcc.gov/measuring-broadband-america`

[13] `http://www.turris.cz/`

Table 4. The Turris router technical parameters.

CPU	Freescale P2020, 1200 MHz
RAM	2 GB DDR3
Flash	16 MB NOR + 256 MB NAND
USB	2x 2.0
LAN	4x 1000 Mbps
WAN	1x 1000 Mbps

4 Applicable Tools

4.1 Netperf

http://www.netperf.org/netperf/[14] Netperf is a client-server system for measuring network performance parameters. The client runs various tests by active monitoring of its communication with the Netperf server. The client provides a number of various options for each test to customize the test behavior. Netperf supports both the IPv4 as well as IPv6 protocols.

Here is the list of the supported measurement tests. All network tests include several implementation types (sockets, DLPI, ATM API).

— Unidirectional TCP/UDP stream performance (bitrate).
— Request/response performance to get one way and round-trip average latency for both TCP and UDP.
— CPU utilization on both the client and server sides.

4.2 Distributed Internet Traffic Generator (D-ITG)

http://www.grid.unina.it/software/ITG/[15] D-ITG is a packet generator used to replicate network traffic of various Internet services (Telnet, VoIP, network games, etc.) following their stochastic models. As a side effect, the application is able to measure different performance characteristics. Namely, it provides information about bitrate, packet rate, one way delay, round-trip time, jitter and packet loss.

[14] http://www.netperf.org/netperf/
[15] http://www.netperf.org/netperf/

4.3 Iperf

https://github.com/esnet/iperf/[16] Iperf (specifically iperf3) is a network bandwidth measurement tool supporting TCP, UDP and SCTP. Besides the client-server implementation of the bandwidth measurement, the project also provides libiperf to access functionality from other applications written in C.

4.4 Paris Traceroute

http://www.paris-traceroute.net/[17] Paris traceroute is the improved version of the standard network diagnosis tool traceroute. It is able to obtain a more precise picture of the actual routes.

4.5 Network Diagnostic Test (NDT)

https://code.google.com/p/ndt/[18] The NDT is a complex client-server system providing network performance and configuration testing. The basic test is measuring the throughput between the server and the client. When the test is performed, the server sends the internal data to the client which analyze them to obtain further network parameters (packet loss, bottleneck link, etc.) and to detect various configuration issues (duplex mismatch, faulty hardware link, etc.).

4.6 Network Mapper (nmap)

http://nmap.org/[19] Nnmap is utility for network discovery and security auditing. It is able to determine what devices are available on the network, what services are available on the devices, what operating systems they are running and many other characteristics.

4.7 Network Analizer Sniffer Tool (NAST)

http://sourceforge.net/projects/nast.berlios/[20] Nast is a packet sniffer and a LAN analyzer. It sniff packets and save data in files, checks NIC in promisc mode.

— Build LAN hosts list, find LAN internet gateways, discover promiscuous nodes
— Follow a TCP-DATA stream, reset an established connection
— Perform a single half-open portscanner, perform a multi half-open portscanner
— Find link type (hub or switch), catch daemon banner of LAN nodes, control arp answers to discover possible arp-spoofings
— byte counting with an optional filter, and write reports logging

[16] https://github.com/esnet/iperf/
[17] http://www.paris-traceroute.net/
[18] https://code.google.com/p/ndt/
[19] http://nmap.org/
[20] http://sourceforge.net/projects/nast.berlios/

4.8 tcptraceroute

http://www.braumeister.org/formula/tcptracerout[21] The regular traceroute usually uses either ICMP or UDP protocols. Unfortunately firewalls and routers often block the ICMP protocol completely or disallow the ICMP echo requests (ping requests), and/or block various UDP ports. By sending out TCP SYN packets instead of UDP or ICMP ECHO packets, tcptraceroute is able to bypass the most common firewall filters.

4.9 D-ITG (Distributed Internet Traffic Generator)

http://traffic.comics.unina.it/software/ITG/[22] Platform capable to produce traffic at packet level accurately replicating appropriate stochastic processes for both IDT (Inter Departure Time) and PS (Packet Size) random variables (exponential, uniform, cauchy, normal, pareto, ...). D-ITG supports both IPv4 and IPv6 traffic generation and it is capable to generate traffic at network, transport, and application layer.

4.10 SIPp

http://sipp.sourceforge.net/[23] SIPp is a free Open Source test tool / traffic generator for the SIP protocol. It includes a few basic SipStone user agent scenarios (UAC and UAS) and establishes and releases multiple calls with the INVITE and BYE methods. It can also reads custom XML scenario files describing from very simple to complex call flows.

4.11 sipsak

http://sourceforge.net/projects/sipsak.berlios/[24] sipsak is a command line tool which can send simple requests to a SIP server. It can run additional tests on a SIP server which are usefull for admins and developers of SIP enviroments.

4.12 IPTraf

http://iptraf.seul.org/[25] IPTraf is a console-based network statistics utility for Linux. It gathers a variety of figures such as TCP connection packet and byte counts, interface statistics and activity indicators, TCP/UDP traffic breakdowns, and LAN station packet and byte counts.

---

21 http://www.braumeister.org/formula/tcptraceroute

22 http://traffic.comics.unina.it/software/ITG/

23 http://sipp.sourceforge.net/

24 http://sourceforge.net/projects/sipsak.berlios/

25 http://iptraf.seul.org/

4.13 dnsperf, resperf

http://nominum.com/measurement-tools/[26] dnsperf is a DNS server performance testing tool. It is primarily intended for measuring the performance of authoritative DNS servers, but it can also be used for measuring caching server performance in a closed laboratory environment. For testing caching servers resolving against the live Internet, the resperf program is preferred.

4.14 psad: Intrusion Detection and Log Analysis with iptables

https://cipherdyne.org/psad/[27] psad is a collection of three lightweight system daemons (two main daemons and one helper daemon) that run on Linux machines and analyze iptables log messages to detect port scans and other suspicious traffic. A typical deployment is to run psad on the iptables firewall where it has the fastest access to log data.

4.15 Snort, Suricata

https://www.snort.org/[28], *http://suricata-ids.org*[29] An open source IDS, IPS and Network Security Monitoring engine for UNIX and Windows. Real-time traffic analysis and packet logging. Support for protocol analysis, content searching and content matching.

5 Conclusion and Recommendations for Future Work

This text provides an overview about possible approaches to the monitoring of end-user networks. It describes the currently running projects focused on this area as well as a set of tools to obtain various information about the network traffic. This research brings a base knowledge to our following work — we plan to develop a kind of an office network probe. With such a device, we want to increase accuracy and usability of our current tools for monitoring backbone networks and improve the security of networks and their users.

From the currently running projects, it is obvious that the device hardware is not crucial. With gained experiences and knowledge we can quite simply change the specific hardware of the probe. This is possible with a commodity devices (SoHo routers) and customizable operating system (OpenWRT). Besides the severity of developing a custom hardware device, the flexibility of changing commodity devices according to changing demands is the main reason to recommend to use a commodity devices instead of developing hardware probe on our own.

The key issue of our work is going to be development (or customization) of the tools to obtain and analyse data from the monitored network. The first decision should be about the type of monitoring we need to do in the local network. Both types, active and passive monitoring, requires different point where the probe is

[26] `http://nominum.com/measurement-tools/`

[27] `https://cipherdyne.org/psad/`

[28] `https://www.snort.org/`

[29] `http://suricata-ids.org`

supposed to be deployed. While in case of active monitoring, the probe is supposed to behave like an end-user device, in case of passive monitoring, we need to observe as much of network traffic passing the local network as possible. The latter best fits into the routers or modems usually connecting the local network to the ISP's network. On the other hand, this requires another hardware resources for standard work of such a device.

Another aspect of such a project is a motivation of users to participate. As mentioned, due to the less amount of data observed in local network, it is important to have distributed many probes. The users are not willing to provide information about their network traffic for free (in many cases they are not willing to provide this information for anything). The currently running projects provides two kind of motivation in principle:

— SoHo router for free,
— provision of detail information about the user network traffic and quality of services provided by their ISP.

Besides some benefits for users, there are some principles that must be respected implicitly. Privacy of user data is unexceptionable. Another rule is, that active nor passive measurement must not limit the users activities in any way. It means, that before (and ideally during) the active measurement it is necessary to detect the current users activity and rearrange the tests if necessary. On the other hand, besides the automatic and central controlled launching of the measurement, the user should be able to run any of the measurement manually and get the current information. From the trustworthiness of the project, it is also meaningful to allow user to switch of the automated measurement entirely. Such a situation can be easily detected and we can ask for reasons to this act.

References

[1] Žádník M. *Flow Measurement Extension for Application Identification* CESNET technical report 14/2009 Available online[30]

[2] Čeleda P., Krejčí R., Barienčík J., Elich M., Krmíček V. *HAMOC – Hardware-Accelerated Monitoring Center* CESNET technical report 9/2010 Available online[31]

[3] Bartoš V., Čeleda P., Kreuzwieser T., Puš V., Velan P., Žádník M. *Pilot Deployment of Metering Points at CESNET Border Links* CESNET technical report 5/2012 Available online[32]

[4] IETF BMWG *Working Group Documents* Available online[33]

[5] IETF IPPM *Working Group Documents* Available online[34]

30 http://archiv.cesnet.cz/doc/techzpravy/2009/flow-measurement-applications/

31 http://archiv.cesnet.cz/doc/techzpravy/2010/hamoc/

32 http://www.cesnet.cz/wp-content/uploads/2013/03/metering_points.pdf

33 http://datatracker.ietf.org/wg/bmwg/documents/

34 http://datatracker.ietf.org/wg/ippm/documents/

[6] IETF LMAP *Working Group Documents* Available online[35]

[7] IEEE *Mobile Broadband Network Performance Measurements* Available online[36]

[8] Broadband Forum *Broadband Access Service Attributes and Performance Metrics (WT-304)* Available online[37]

[9] M-Lab *Open Source Codes* Available online[38]

[10] BISmark *Open Source Codes* Available online[39]

[11] RIPE Atlas *Open Source Codes* Available online[40]

[12] Philip Homburg *Releasing RIPE Atlas Measurements Source Code* Available online[41]

[13] RIPE Atlas *Community Website* Available online[42]

[14] SamKnows *Open Source Codes* Available online[43]

[15] SamKnows *Technical Papers* Available online[44]

[16] Federal Communications Commission *Measuring Mobile Broadband Methodology – Technical Summary* Available online[45]

[17] CZ.NIC *TurrisCompetition or Show what only your Turris can do* Available online[46]

35 http://datatracker.ietf.org/wg/lmap/documents/

36 http://ieee802.org/16/mbnpm/index.html

37 https://www.broadband-forum.org/technical/technicalwip.php

38 https://github.com/m-lab/

39 https://github.com/projectbismark/

40 https://atlas.ripe.net/get-involved/source-code/

41 https://labs.ripe.net/Members/philip_homburg/ripe-atlas-measurements-source-code

42 https://github.com/RIPE-Atlas-Community

43 http://www.samknows.com/opensource

44 http://www.samknows.com/broadband/methodology

45 http://www.fcc.gov/measuring-broadband-america/mobile/technical-summary

46 https://www.turris.cz/en/news/turrisoutez