Multi-Functional Emulator for Traffic Analysis

Sándor Molnár, Péter Megyesi
High Speed Networks Lab., Dept. of Telecomm. and Mediainformatics,
Budapest Univ. of Technology and Economics, Budapest, Hungary
Email: {molnar, megyesi}@tmit.bme.hu

Géza Szabó
TrafficLab
Ericsson Research, Budapest, Hungary
Email: geza.szabo@ericsson.com

***Abstract*—We present the versatile functionality of our novel user behavior based traffic emulation system in this paper. We show the unique feature of the system, i. e., it is capable of working on different platforms (Windows, Android), on different access technologies (wired, WiFi, 3G) and as a remote controlled system on different sites (Europe, Asia, South America). Our examples exhibit some of the manifold traffic analysis possibilities as a result of this key functionality. We have also made our system available to the public [1].**

I. Introduction

Internet Service Providers (ISP) are interested in the everchanging traffic characteristics of the Internet to develop efficient traffic management methods, charging policies, etc. It is of crucial importance that applications are accurately identified and their properties are clearly understood. For the identification Deep Packet Inspection (DPI) is the key tool and to understand traffic properties an efficient traffic analysis is needed. One of the main bottleneck of developing high performance DPI tools is that the network data is the property of the operator and it results in a number of privacy issues. To avoid such problems network simulators are widely used but they are hardly able to accurately represent real traffic characteristics. It is partly due to the fact that simulation mechanisms are just approximately able to reproduce the actual traffic of emerging or even traditional applications and partly due to the difficulty that simulators continuously must be updated to the current versions of all applications which is practically impossible.

These obstacles motivated our research to develop our User Behavior based Emulator (UBE) which is capable of generating traffic with characteristics which is very close to the real traffic characteristics. It is due to the fact that we do not simulate or generate traffic based on some traffic model but rather we record typical user interactions with several applications on the Graphical User Interface (GUI) and construct application specific usage models which can be used later to emulate user interactions on remote controlled computers. It means that we extract typical user scenarios, e.g., used applications and their share, usage patterns, etc. from real measurements. We can generate traces and build a database when user actions e.g., mouse or keyboard events happened according to the emulated user scenarios. From these database we can construct arbitrary aggregate traffic mix having traffic characteristics very close to the real one. The generated traffic has realistic payload and traffic characteristics both in inter-packet and user level timescales and it has the advantage that it does not contain user sensitive data and can be distributed for wide audience.

The main contribution of this paper is to present the unique multifunction features of UBE and demonstrate their manifold use in traffic analysis with real measurement examples. It is shown that our system, beside that it generates traffic close to the real one, can be used in broad environments in many sense which makes traffic analysis much easier. Our system is capable of working on different platforms (Windows, Android), on different access technologies (wired, WiFi, 3G) and as a remote controlled system on different sites (Europe, Asia, South America).

The paper is structured as follows. In Section II a brief overview is given about state-of-the-art traffic generation techniques. An overview of UBE is presented in Section III. In Section IV we describe the functions integrated into UBE's publicly available website. In Section V traffic analysis examples are presented with discussions to demonstrate the key features of our UBE. Finally, Section VI concludes the paper.

II. Related Work

Traffic generation can be classified into *trace-based* and *model-based* approaches. The model-based approach is usually build on a stochastic traffic model where the model parameters are set based on traffic characteristics of measured data. The problem with this method that high accuracy can be achieved only with complex models having so many parameters which are practically impossible to set. In contrast, trace-based methods use real measurements getting the real payload and packet-trace information which makes accurately traffic reproduction possible. However, dealing with real traces arises serious privacy issues which often stop the applicability of this approach.

There are a number solutions which are widely used for testing network devices (e.g., [2], [3]) but they have the drawback of inaccuracy regarding payload and inter-packet timing since they lack of these information. These obstacles exclude them for using DPI testing.

A system with the main goal of high-performance traffic generation containing real application protocol signatures mixed with randomly generated data streams was proposed in [4]. Such method can be used for DPI tool testing. On the other hand for packet header based traffic classification systems e.g., [5], [6] the inter-packet timing of the generated

978-1-4673-3122-7/13/$31.00 ©2013 IEEE

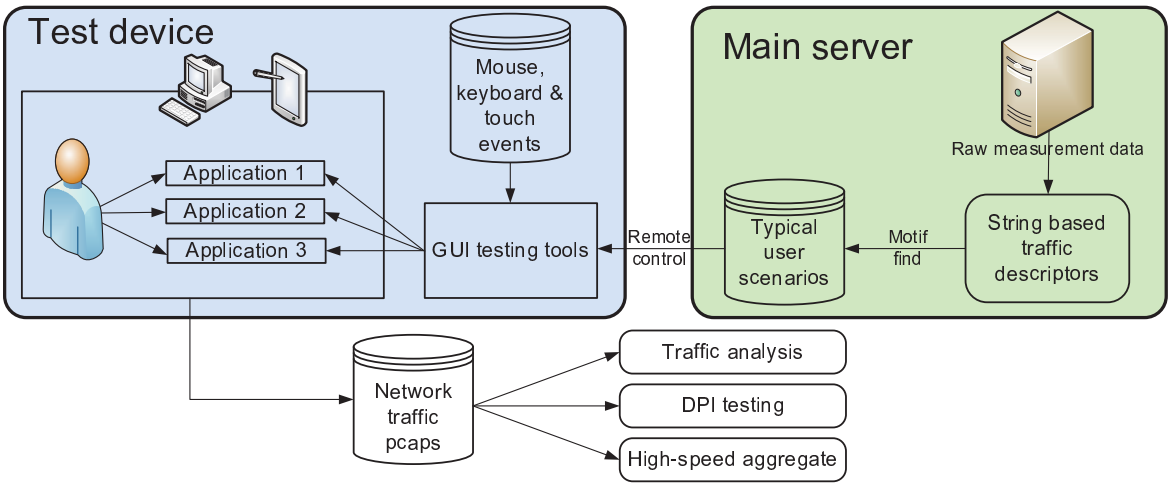


Fig. 1: The architecture of the User Behavior based Emulator

data contains no information as being completely synthetic. A solution is proposed in [7] in which real network traces are used to replay correct packet inter-arrival timing and ns2 simulator to mimic the TCP behavior. However, this solution provides no realistic packet payload.

Authors of [8] propose techniques for using statistical models of user behavior to drive real, binary, GUI-enabled application programs in place of a human user. The used applications that are connected to the Internet are Internet Explorer and Outlook mail client while they also use several offline applications as well. In contrast to this work we consider a much bigger variety of online applications in UBE. While their purpose was to demonstrate the utility of their techniques in an example experiment comparing the system resource consumption of a Windows machine running anti-virus protection versus an unprotected system the main goal of our system is to provide repeatable traffic generation in a more general network environment including mobile networks with various access technologies and smartphone OSes as well.

In [9] authors suggested an event-driven automata-synchronized replay system over WLAN with environments simulation. This system transforms the captured packet trace into a sequence of events that follow the IEEE 802.11 protocol and a three-level automata is used to achieve packet-replay control and synchronizes the environment effects.

In order to obtain both realistic traffic characteristic and realistic data payload actual measurements from operational networks are needed. A solution is presented in [10] where replay of real packet measurements recorded on OC-48 speed backbone network with commodity hardware was used. However, because of privacy issues these measurements cannot be available to the public.

In our system we target to use real measurements to have complete payload and realistic inter-packet timing data as well. The measurement what we generate completely lacks any kind of sensitive personal information which is a major issue with measurements recorded in operational networks. Further, the generated traffic is perfectly classified.

III. System Overview

Figure 1 sketches the architecture of the User Behavior based Emulator. The system can be separated into two parts: (A) the main server and (B) the test devices. The main idea behind our emulation system is that the main server logs into a test device and remotely executes previously written scripts which are able to run different applications and simulate mouse, keyboard or touch events on the device's GUI. This technique has the benefit that the generated network traffic will be similar to traffic generated by real users in many aspects (e.g. payload content, inter-packet timing, etc.). During the emulation process the network traffic is captured and stored in *tcpdump* format on the remote devices.

In order to focus on the emulation of user behaviors scenarios which appear in real environment UBE is able to process real measurement data to extract typical usage scenarios. In this procedure the framework uses special string based descriptors for representing user traffic [11]. For further details on this process see Section IV-A. The system is able to schedule the remote controlling procedure based on these string descriptors.

The environment of UBE with connected devices is presented in Figure 2. The main server is a commodity Linux PC which is located in our campus area and it is responsible for controlling and scheduling the emulation processes. A WiFi router is bridged through the server which is used to connect the local devices to the Internet. The test devices cover two different platforms: Windows computers and Android smart phones.

Android smart phones are connected to the main server via USB cable thus these devices are located in our campus site. The server uses the Android Debug Bridge (ADB) [12] to log into the smart phones and execute MonkeyRunner [13] scripts. MonkeyRunner is common tool used for stress testing Android applications as it can generate keystroke and touch events on

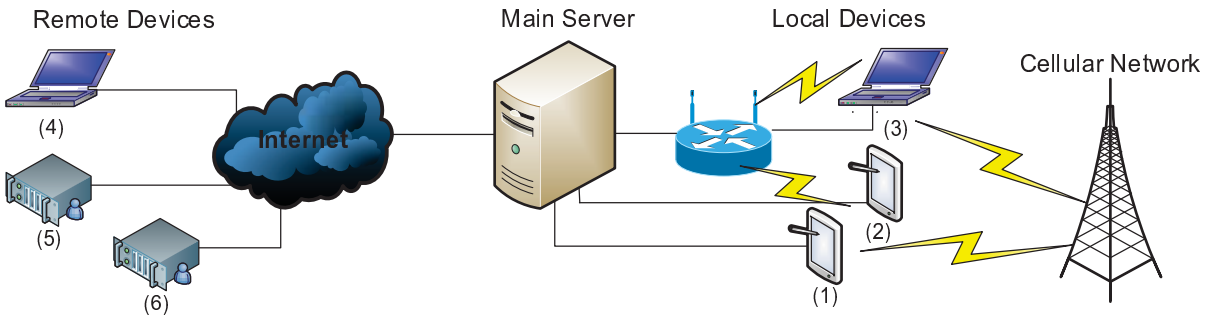


Fig. 2: The assembled environment for the User Behavoir based Emulator

an Android phone's GUI. During the emulation period the network traffic is captured using *tcpdump*. Currently we have two Android phones integrated into the emulation framework: an LG Optimus 540 and a Samsung Galaxy S II. The smart phones can access the Internet via WiFi link through the main server or via cellular network using 3G connection.

As desktop platform we installed Windows 7 to our test machines. The main server can connect to these devices using *telnet* and the GUI automation scripts are written in AutoIt language [14]. These scripts are executed via PsExec [15] for bounding the launched applications to a graphical session otherwise they could generate void errors. The generated traffic is recorded in *tcpdump* format using *windump*. Since the *telnet* connection is not limited to distance we are able install test machines in remote sites, outside our campus area. Currently one test computer is working in locally, and three in remote locations: a desktop PC in Budapest and two virtual machines, one in Tokyo, Japan hosted by the National Institute of Information and Communications Technology (NICT), the other in Recife, Brazil hosted by the Federal University of Pernambuco (UFPE). Table I summarizes the test devices integrated into UBE.

IV. Test System

The web interface of the User Behavior based Emulator is available for public testing [1]. In this section we present the features of UBE step-by-step which are accessible via the website. We would also like to encourage the research community to contribute to this research by sending new network measurements or GUI automation scripts of different applications after testing these functionalities.

TABLE I: Test devices in UBE

ID	Name	Location	Access
1	LG Optimus	Budapest	WiFi / 3G
2	Samsung Galaxy	Budapest	WiFi / 3G
3	Campus PC	Budapest	Wired / WiFi / 3G
4	Laptop	Budapest	Wired
5	NICT virtual	Tokyo	Wired
6	UFPE virtual	Recife	Wired

A. Real Measurement Processing

The first step using the website is to choose between the integrated measurements. We have collected several publicly available trace files which are frequently used by the research community. Due to privacy reasons these measurements contain packet header information only thus UBE uses a port base traffic classifier algorithm for converting them into string based descriptors. These strings represent the individual users' traffic in minute resolution. It also possible to create a new measurement case by uploading an input source file. UBE can process measurements in both *tcpdump* and *netflow* formats. The port based traffic classifier codes can be downloaded from the website.

The next step is the extraction of typical user behavior scenarios. In this procedure UBE searches for frequently occurring fixed time length patterns in the transformed input. Later the new results can be integrated into the user behavior scenario database which is globally defined for all measurements. UBE is able to parse these string descriptors and convert them into a remote controlling procedure. Note that the scope of this paper does not include the evaluation of the real measurement processing phase, we are only focusing on the emulation of previously defined user behavior scenarios. The web page also contains an interface for manual definition of user scenarios.

B. User Behavior Scenarios

UBE associates a remote controlling process to every defined user behavior scenario which describes the GUI automation scripts which have to be executed during the emulation phase. Since the framework can emulate one type of traffic with multiple applications we can specify one of them for every user scenario or the program can choose between them randomly. It is also possible to launch measurements via the the web page using a task manager interface. The trace files generated by the previously run emulation processes and the GUI automation scripts written in AutoIt and MonkeyRunner languages are also available for public download.

Note that the evaluation of the traffic of the stand-alone user behavior scenarios for validation purpose is unnecessary as the user emulation tool can perform the same actions as a user

does, e.g., clicks on icons, links, use the common functions of every application which is available on the GUI. From the traffic point of view the generated traffic of the emulated actions triggers the same traffic as any user. If we define typical user scenarios in which the user actions should be scheduled in an intelligent way to be similar to real world scenarios then the correctness of the scheduling strategy should be evaluated.

C. Aggregation of Trace Files

The third main feature of UBE is also available using the website. In the future we would like to construct realistic high-speed aggregated traffic streams using the recorded individual trace files. Note that this feature makes it possible to construct a real traffic trace as an aggregate mix with traffic from pre-defined applications. Such traces are hardly achieved in real measurements even if we neglect the privacy issues. As of now, UBE is able to merge multiple *tcpdump* files into one by an input descriptor file. The merge tool can modify the timestamp and IP address information of the packets creating an output were the individual trace files are played simultaneously and distinct IP addresses are associated to different users. This process does not change the timing information between an individual user's packets, it will follow the same statistics that it was recorded during the emulation phase of the given scenario. Detailed information about the aggregation tool can be found in [16].

V. Traffic Analysis

In this section we present analysis results of different traffic traces obtained by the User Behavior based Emulator. Due to limited paper size we do not intent to present every details but rather we demonstrate the capabilities of UBE by some selected user scenarios which are present in both PC and smart phone platforms. Each presented scenario was emulated at least a hundred times on the test device of UBE using every possible access type.

A. Web browsing

Recent studies showed that web traffic is dominant on smart phone platform [17], [18]. Towards understanding the differences between the same web browsing event in different circumstances we have emulated the same scenario on every test device. Figure 3 plots the cumulative distribution function (CDF) of the packet inter arrival times (IAT) in downstream direction.

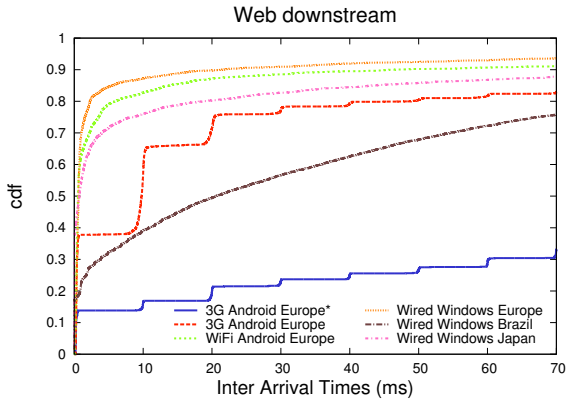
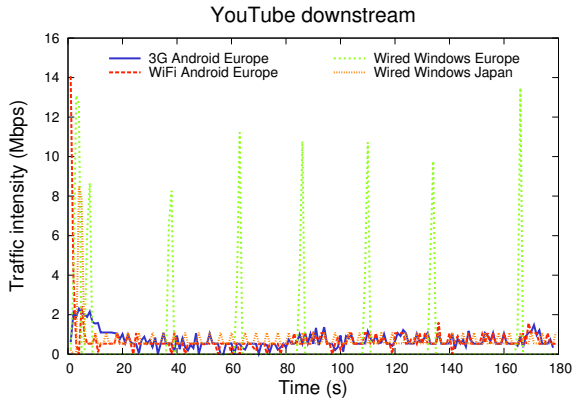


Fig. 3: Web Browsing Measurement Results

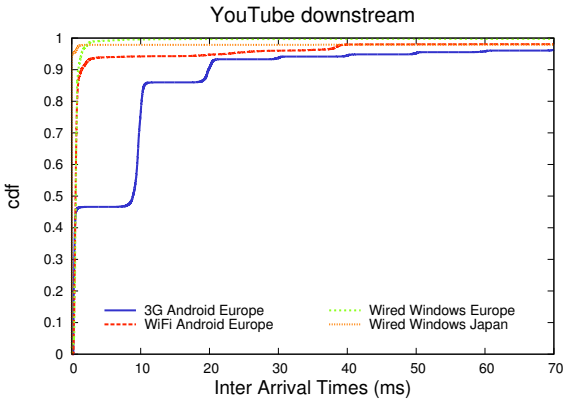
The CDFs related to the Windows platform and the Android using WiFi do not differ significantly. The reason of the difference is that desktop browsers download more data than smart phone's since many popular web sites have a mobile version which avoids using extra content e.g. flash based advertisements. However, using the 3G interface of the Android phones result in notably different values. In this case the IAT curve shows a tiered structure with 10 ms periodicity due to Node-B scheduling [19]. The deviation was even greater after the phone reached the monthly traffic limit and the operator limited its access speed to 128 kbps.

B. Media streaming

According to [20] media streaming was 52 percent of total mobile traffic in 2011 and it's projected to increase 25-fold to 2016 accounting for more than two-third of world's mobile data traffic. YouTube is undoubtedly the most popular media streaming website being third in the world's traffic rank



(a) Downstream Traffic Intensity



(b) Downstream Inter Arrival Times

Fig. 4: YouTube Measurement Results

[21]. It's position should remain strong in the future since it's parent company, Google is the developer of the Android platform therefore the YouTube mobile application is installed by default on every smart phone with such operating system.

A recent study [22] has revealed that YouTube uses an additional application level flow control over the traditional TCP mechanism. At the beginning of the streaming an initial buffering period takes place where the server is sending data as fast as possible. This phase is followed by a block sending procedure of 64 KB sized blocks where the application reduces the sending rate close to playback speed. Figure 4a shows the traffic intensity measurement results which present the different cases that the YouTube application block sending mechanism can flow.

The results we have measured in our campus site using a desktop Windows shows periodic and very bursty traffic. YouTube flow control shows this type of pattern if three conditions are fulfilled: (a) high speed access, (b) low round trip time and (c) no packet is lost due to congestion or buffer limit. Other Windows machines shows the normal flow of the 64 KB block sending period since a packet loss event occurred during the transmission. Android patterns show similar parameters but in case of 3G access the initial period is slower because of the limited bandwidth. These results confirm the statements presented in [22].

Figure 4b plots the CDF of IAT of the YouTube flows. WiFi and wired measurements show the same characteristics regardless of the platform while 3G results have 10 ms periodic tiered structure due to Node-B scheduling [19].

C. Skype

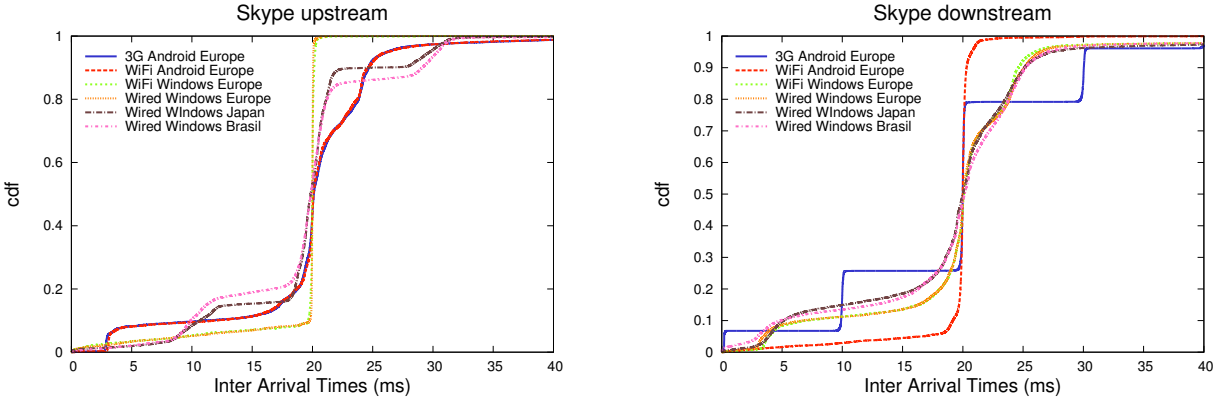
Although VoIP applications do not share major portions of the total Internet traffic, they are very popular among smart phone users by the reason of free voice or video calls. On the other hand, mobile operators are working on identifying these kind of traffic for applying different charging polices than regular data service. Therefor understanding the traffic characteristics of VoIP applications is a crucial objective. In this subsection we present the analysis results using the most popular VoIP application, Skype. During every emulation process UBE has remotely controlled Skype the application used its default wideband codec: SVOPC [23].

Figure 5a presents the CDF curve of the inter departure times of consecutive Skype packets in upstream direction. We have observed that the timing of these packets only depend on the platform and independent on the access type. In case of native Windows the timing is very precise to the 20 ms codec frame size, while on virtual Windows and Android platforms differ significantly from this value. The possible reason for this behavior is that while timing on native Windows machines generates interrupts by accurate hardware oscillators, virtual operating systems use software interrupts generated by the host OS, which can be delayed or lost completely [24]. Authors of [24] even showed that this skewness can be used for determining the host OS of a virtual machine. Similar explanation could be behind the Android based results as in that platform applications run inside the Dalvik Virtual Machine [25] thus they don't have direct access to the Linux kernel. Besides the voice packets, Skype also sends out *sync* packets after about 10 arrived frames. The timing of these frames seems random between the periodic voice packets which accounts for the linear slope at the beginning of the CDF curve.

In Figure 5b the CDF of IAT values on the other end of the conversation are plotted. This chart presents how different Internet routes can affect the downloaded stream. Measurement taken between close sites shows low jitter values, while between remote locations the skewness is higher. It is also interesting that the frame size of 20 ms overlaps the 10 ms periodicity of the Node-B scheduling using 3G access in one end of a conversation.

VI. Conclusion

We presented our developed User Behavior based Emulator (UBE) with some unique multifunctionality features in this paper. UBE uses real measurements to have complete payload and realistic inter-packet timing data as well and it emulates traffic accordingly avoiding all the obstacles coming



(a) Upstream Inter Arrival Times

(b) Downstream Inter Arrival Times

Fig. 5: Skype Measurement Results

from privacy issues of real measurements. We showed that UBE is capable of working on different platforms (Windows, Android), on different access technologies (wired, WiFi, 3G) and as a remote controlled system on different sites (Europe, Asia, South America). Some analysis examples are presented from the main application categories (web, media streaming, VoIP) to demonstrate that this multifunctionality opens unique traffic analysis possibilities including the comparisons of the traffic characteristics of the same investigated application on different platforms, technologies or geographical locations. These results also show that our framework is an effective tool of generating application signatures in a specified environment since the traffic characteristics of the same user behavior scenario could be fundamentally distinct under different circumstances. Our system is available to the public [1] and was demonstrated in [26].

ACKNOWLEDGMENT

The authors would like to thank the co-workers of NICT Tokyo, Japan and UFPE Recife, Brazil for supporting this research by hosting test machines in their campus site.

This research was supported by OTKA-KTIA grant CNK77802.

REFERENCES

[1] "User Behavior based Emulator Test Page," Sept 2012. [Online]. Available: http://ubetest2.hsnlab.tmit.bme.hu/

[2] S. Zander, D. Kennedy, and G. Armitage, "KUTE A High Performance Kernel-based UDP Traffic Engine," Centre for Advanced Internet Architectures, Swinburne University of Technology, Melbourne, Australia, Tech. Rep. 050118A, Jan 2005. [Online]. Available: http://caia.swin.edu.au/reports/050118A/CAIA-TR-050118A.pdf

[3] G. Antichi, A. D. Pietro, D. Ficara, S. Giordano, G. Procissi, and F. Vitucci, "Bruno: A high performance traffic generator for network processor," in *Performance Evaluation of Computer and Telecommunication Systems, 2008. SPECTS 2008. International Symposium on*, June 2008, pp. 526 –533.

[4] A. Santos, S. Fernandes, R. Antonello, P. Lopes, D. Sadok, and G. Szabó, "High-Performance Traffic Workload Architecture for Testing DPI Systems," in *Proc. IEEE GLOBECOM*, Houston, USA, Dec 2011.

[5] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian, "Traffic Classification On The Fly," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 2, pp. 23–26, 2006.

[6] F. Palmieri and U. Fiore, "A Nonlinear, Recurrence-based Approach to Traffic Classification," *Comput. Netw.*, vol. 53, pp. 761–773, April 2009.

[7] M. C. Weigle, P. Adurthi, F. Hernández-Campos, K. J. Kevin, and F. D. Smith, "Tmix: A tool For Generating Realistic TCP Application Workloads In ns-2," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 3, pp. 65–76, July 2006.

[8] C. V. Wright, C. Connelly, T. Braje, J. C. Rabek, L. M. Rossey, and R. K. Cunningham, "Generating client workloads and high-fidelity network traffic for controllable, repeatable experiments in computer security," in *Proceedings of the 13th international conference on Recent advances in intrusion detection, RAID'10*, Ottawa, Canada, Sept 2010, pp. 218–237.

[9] C.-Y. Ku, Y.-D. Lin, Y.-C. Lai, P.-H. Li, and K.-J. Lin, "Real traffic replay over wlan with environment emulation," in *Wireless Communications and Networking Conference (WCNC), 2012*, April 2012, pp. 2406 –2411.

[10] W. Feng, A. Goel, A. Bezzaz, W. Feng, and J. Walpole, "Tcpivo: A high-performance packet replay engine," in *Proc. of the ACM SIGCOMM workshop on Models, methods*, 2003, pp. 57–64.

[11] P. Megyesi and S. Molnár, "Finding typical internet user behaviors," in *Proc. 18th EUNICE Conference on Information and Communications Technologies*, Aug 2012, pp. 321–327.

[12] "Android Debug Bridge," Sept 2012. [Online]. Available: http://developer.android.com/tools/help/adb.html

[13] "MonkeyRunner," Sept 2012. [Online]. Available: http://developer.android.com/tools/help/monkeyrunner_concepts.html

[14] "AutoIt," retrieved: Sept, 2012. [Online]. Available: http://www.autoitscript.com/site/autoit/

[15] "PsExec," retrieved: Sept, 2012. [Online]. Available: http://technet.microsoft.com/en-us/sysinternals/bb897553

[16] B. Csatári, "Framework for comparison of traffic classification algorithms," in *Master Thesis*, 2011. [Online]. Available: http://www.crysys.hu/ szabog/publications/diplomazok/csatari-thesis.pdf

[17] G. Maier, F. Schneider, and A. Feldmann, "A first look at mobile handheld device traffic," in *Passive and Active Measurement*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2010, vol. 6032, pp. 161–170.

[18] H. Falaki, D. Lymberopoulos, R. Mahajan, S. Kandula, and D. Estrin, "A first look at traffic on smartphones," in *Proc. of the 10th ACM SIGCOMM conference on Internet measurement*. New York, NY, USA: ACM, 2010, pp. 281–287.

[19] H. Holma and A. Toskala, *HSDPA/HSUPA for UMTS*. John Wiley & Sons, Ltd, 2006. [Online]. Available: http://dx.doi.org/10.1002/0470032634.fmatter

[20] Ericsson, "Traffic and market report," June 2012. [Online]. Available: http://hugin.info/1061/R/1617338/516188.pdf

[21] "Alexa: Top 500 Global Sites," Sept 2012. [Online]. Available: http://www.alexa.com/topsites

[22] S. Alcock and R. Nelson, "Application flow control in youtube video streams," *SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 2, pp. 24–30, April 2011.

[23] J. Lindblom, "A sinusoidal voice over packet coder tailored for the frame-erasure channel," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 5, pp. 787 – 798, Sept 2005.

[24] X. Chen, J. Andersen, Z. Mao, M. Bailey, and J. Nazario, "Towards an understanding of anti-virtualization and anti-debugging behavior in modern malware," in *IEEE International Conference on Dependable Systems and Networks With FTCS and DCC 2008.*, June 2008, pp. 177 –186.

[25] "Dalvik Virtual Machine," Sept 2012. [Online]. Available: http://www.dalvikvm.com/

[26] S. Molnár, P. Megyesi, and G. Szabó, "Multi-functional traffic generation framework based on accurate user behavior emulation," in *Proc. IEEE INFOCOM (Demo)*, April 2013.