Network Neutrality – Measures and Measurements: A Survey

Clemens H. Cap, Andreas Dähn, Thomas Mundt
Department of Computer Science
University of Rostock
Rostock, Germany
{clemens.cap, andreas.daehn, thomas.mundt}@uni-rostock.de

***Abstract*—Over the last five years, network neutrality (which means that network infrastructure is treating all data packets equally) has grown to a valuable research area which can be seen as application of anomaly detection. Neutrality violations result from a combination of traffic differentiation (either by statistical protocol identification or deep packet inspection) and infrastructure components which are capable of classification based packet handling. We examine some examples for neutrality violations and then go on to neutrality testing. Neutrality testing approaches can be divided into three categories: Active approaches (which usually utilize specialized testing peers), passive approaches (which monitor the incoming and outgoing network traffic at the user's computer or local network) and hybrid approaches (which combine both). In this article, we take a look at some implementations and at assets and drawbacks of both approaches and implementations. Some major drawbacks originate from ambiguous test results (such as a test reporting a neutrality violation for what really is a network congestion). Depending on the approach and the implementation, different testing programs have very different statements which shall not be compared without consideration of testing principle and implementation details. Aside from algorithmic testing, crowdsourcing approaches which use volunteers' observations have been developed recently.**

Keywords — ***Network neutrality; Network performance anomaly detection; User-oriented performance metrics; Intrusive and non-intrusive performance measurement mechanisms***.

I. Introduction

Network neutrality is the idea of a network treating all handled packets equally (a more detailed definition is provided in the following Subsection). Over the last years, an increasing number of network equipment became capable of traffic differentiation, lowering the barrier to violations of network neutrality. Subsequential, identification of network neutrality violations became a research topic and led to the development of several neutrality violation detection systems. In this survey, we provide an overview on current technical measures, which are used to violate neutrality, as well as measurement techniques. We focus on conceptual rather than on implementation details and cover active, passive, and hybrid neutrality violation detection techniques.

The article starts with a short terminology chapter. In the following Section, we provide a brief digest on the historical development and the debate surrounding network neutrality. Section 3 then contains technical details, covering symptoms of neutrality violations, traffic differentiation techniques, and neutrality violation detection approaches. Section 4 provides a short outline about Internet service providers' ways to tamper with measurements. Section 5 is practical oriented: It contains examples of observed neutrality violations and provides a review of currently available neutrality detection software. Section 6 finally concludes the article.

A. Terminology

When the term "network neutrality" is used in this article, we assume the following definition: *A network is neutral, if all data packets are processed equally, regardless of their origin, destination, protocol or content* [1] (translation A.D.). This definition has been chosen because of its shortness and clearness, although it needs a well defined reference point. The arising problems are discussed in detail in Section II. Other definitions relate neutrality violations to turning away from the "best effort" principle. Best effort commonly means that a infrastructure component works "first in, first out" with no guarantees regarding packet delivery or any quality of delivery. More thoughts on the definition by "best effort" can be found in [2].

We will use the term "network provider" in general as neutrality violations seem not restricted to Internet service providers nor other network carriers.

II. Development of network neutrality

One may ask whether the Internet has ever been neutral, since there has always been a relation between network quality and paid fee. In contrast to this observation, one detail has changed over the last ten years: Network infrastructure equipment became capable of traffic differentiation. According to the above definition of network neutrality, the Internet has been neutral as long as all infrastructure components worked best-effort.

Currently, two factors influence the network-neutrality-debate: One is the rising impact of next generation networks which unite television, telephone and Internet connection. The other concerned with media rumors is network providers changing their terms of service [3], the European Union taking a new approach on network neutrality evaluation [4] and activities of media companies working towards a "free

Internet" [5] as well as (probably other) companies working against piracy using filtering mechanisms [6]. Recent versions of service level agreements used by Internet service providers contain restrictions of throughput whenever a specified amount of data has been transferred using defined services. This change is probably due to the widespread practice of "overselling", which means that the ISPs sell e.g. more throughput to customers than they can theoretically provide if all customers would acquire the maximum capacity the same time. In this context one may take a look at contracts between Internet service providers and customers and ask whether general network access or even specific characteristics are sold. Do contracts assure minimum values for at least some of the network properties such as latency, throughput or jitter (as specified e.g in [7])?

Next generation networks feature their own problem regarding network neutrality. The question whether the wall socket or just the PC connection of the user's router shall be subject to the definition of network neutrality remains unanswered currently. In this article, we focus on the user's home network Internet uplink, not the wall socket.

III. Detecting network neutrality

Detecting network neutrality contains a basal problem: To prove a network connection to be truly neutral, testing connections to every possible target with every possible protocol would be necessary. As such a practice would obviously be impossible, tests scan for violations of network neutrality.

A. *Symptoms of network neutrality violations*

Violating network neutrality can result in four observable symptoms (respective to single data streams (i.e. the set of all data packets belonging to one transfer as seen from upper layers)):

1) unavailability of sites or services,
2) enhanced quality of service,
3) reduced quality of service,
4) low-level phenomena such as changed arrival times of data packets compared to each other.

The term "quality of service" is used as defined in [7], covering throughput, latency, jitter, and error rate.

This list reveals one of the problems making detection of network neutrality a difficult task: some of these symptoms can also be caused by other reasons than a violation of network neutrality.

B. *How network neutrality is violated*

Network providers violate network neutrality for three main reasons: Political, social, or economical reasons. A further discussion of network providers' motivation to violate network neutrality is beyond this article's scope.

To violate network neutrality, network providers distinguish data streams originating from the same IP address. We take a look at practical relevant methods: Deep packet inspection (DPI) and statistical protocol identification (SPID). The basic ideas of DPI and SPID are explained as follows. A detailed introduction with a review of current DPI implementation techniques can be found in [8] or [9]. General information on SPID can be found in [10], in which the use of Bayes' classifier is demonstrated. An example for the use of SPID to differentiate web applications is described in [11]. The results of these measures are subsequently used to apply policies to data streams. Such policies may contain modifications of transferred contents, denial of packet forwarding as well as enhanced or degraded priority.

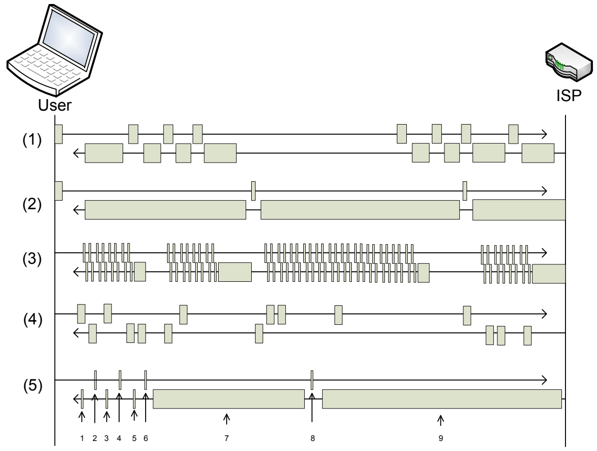
Both methods originate from network security systems, which scan for malicious data or suspect behavior, and have been developed and improved in this context.

Statistical protocol identification analyzes packet contents and the meta data surrounding a transmission. One of the SPID-methods is analyzing the byte-distribution within a data packet. Other methods analyze transmission frequencies or sizes. These methods lead to satisfying statements about the used upper layer protocol even if the payload is encrypted [12].

Figure 1 illustrates how SPID identifies protocols or applications using meta data of data streams: Different types of network usage generate different data exchange pattern. The first example might be a browsing session: The user load a page. The page refers to several other files (images, style sheets, ...) which are loaded subsequently. Once the user finished reading, he may open the next page. The second example depicts probably a download (without identifying the actual protocol): Only small acknowledgment packets originate from the client. Example (3) could result from an interactive shell session: Small packets represent single keystrokes as well as the appearing characters. Some keystrokes trigger longer responses, e.g. directory listings. Example (4) contains no obvious pattern except all packets having a comparable size; this pattern might probably indicate a chatting user. Example (5) resembles a POP3-session; the actual exchanged data is provided aside the drawing.

Deep packet inspection considers knowledge about upper layer protocols as their headers are necessarily included in the packet. Only encrypted (application-) protocol headers are not available to deep packet inspection. Consequently deep packet inspection needs to make assumptions about assignment between ports and protocols.

In a typical DPI use case, specific data can be found at specific offsets within a data packet (for example in a HTTP-request (in a TCP-packet without extra headers): The sender address starts at bit 96, the destination address at 128; the TCP-source-port at bit 160, the destination point at 176. The HTTP-request itself starts at bit offset 352). As the packet including its payload is analyzed, DPI tends to be a bottleneck in packet forwarding. Especially the deployment of a new rule set was a problem. This problem has triggered the development of new algorithms, e.g. [13]. Identified data streams can subsequently be marked as high- or low-priority or payload may get modified. It is also possible to silently drop the packet.



Client → Server		Server → Client
	1	+OK POP server ready H alicenet001
user alice@alice.net	2	
	3	+OK password required for user "alice@alice.net"
pass alicessecret	4	
	5	+OK mailbox "alice@alice.net" has 3 messages (138711 octets)
retr 1	6	
	7	Body of mail #1
retr 2	8	
	9	Body of mail #2

Fig. 1. Example for statistical protocol identification by packet sizes and frequencies.

C. Detection Approaches

To detect symptoms of neutrality violations, two major approaches have been developed. Some additional approaches have been made, e.g. to gain information on neutrality violations using crowdsourcing.

All approaches have two constraints: The number of false positives and false negatives shall be as small as possible. False positives would be scenarios in which a neutral (but perhaps congested) network is reported as non-neutral; a negative would be a neutrality-violating network reported as neutral. The fine tuning on these indices is usually done by means of statistical evaluation.

1) The Active Approach: The active approach tests connections explicitly for neutrality violations. During a test, data is exchanged between a testing client on the user's computer and one or more testing servers on one or more well-known hosts on the Internet. Both sides observe the data exchange carefully and apply statistical tests to it. This statistical evaluation results in a statement whether the test found neutrality violations – or not (or that the test results are inconclusive).

The typical setup for a measurement based on the active approach can be divided in two active parts: One on the user's computer located in his home network; the other on well-known testing servers. The intermediate routing can neither be influenced nor examined actively.

This approach features a huge drawback: As long as specific testing peers are necessary, tests can only state whether connections to *these* specific peers are neutral. Statements regarding the neutrality of connections to other targets can not be derived. The active approach has been implemented, e.g. in the project *Glasnost* [14].

2) The Passive Approach: The passive approach monitors the user's everyday network usage. A piece of software records a detailed statistic about exchanged network packets. Additional information has to be provided by the user. These information is used to evaluate data exchanges. Statistic methods are applied to distinguish discriminated from promoted traffic. Finally, a statement about the neutrality of the network uplink is made. Some implementations of the passive approach aggregate collected data from all users on central systems to boost the approaches efficiency. This collection of data is necessary to have a sufficiently large sample. Otherwise detection would rely purely on a single user's behavior – who probably would not generate enough data to allow statements related to each connection.

The network setup for a measurement using the passive approach needs only a monitoring client on the users computers and a server for central evaluation which has to be reachable through the Internet.

Blasting the restriction on testing peers is a great advantage of the passive approach. The approach provides an answer to the question "is the network uplink neutral regarding everything I do" (which is nothing else than "is the network uplink neutral" to a single user) and not "is a bunch of connections through my network uplink neutral". This advantage is bought by submitting detailed information about the network usage to a central (and potentially unsafe) server. Implementations of the passive approach react on this problem by allowing the user to disable data aggregation for a specified timeout or specific domains. However, this restriction hits exactly the big advantage: If the user decides to disable data aggregation while visiting some sites, neutrality violations applied to those sites can not be detected. The passive approach has been implemented, e.g. in the project *NANO* [15] which analyzes the network usage at a rather low level; other projects such as *Fanthom* [16] utilize a view from within the user's browser.

3) Hybrid Approaches: Combining the active and the passive to a hybrid approach is promising. It may combine the advantage of easy measurements (inherited from the active approach) while using all used network connections as view port (inherited from the passive approach). Two ideas of hybrid approaches seem feasible. They shall be described briefly.

A first approach would connect multiple instances of the passive approach. Whenever the central evaluation cannot decide whether something is a neutrality violation, additional instances of the measurement client are acquired to act actively. They would reproduce a connection whose neutrality cannot be decided. This supplement would allow quick tests whenever something seems to come up. However, this idea contains the possibility of abuse by its design: The design resembles a bot net. Furthermore, this part could even falsify a measurement: Imagine a site suffering from congestion. Its reduced performance is noted by a passive instance and submitted to the central evaluation. Additional instances are ordered to perform measurements (by opening additional connections to the target). This feedback loop causes additional traffic which intensifies the congestion.

Possibly due to the problem of feedback loops and abuse, to the authors knowledge this approach has not been implemented yet.

A different idea of an hybrid approach embeds a "black box" in the providers network. A measurement is performed by establishing an encrypted tunnel between the user's computer and the "black box". This setup enables differential measurements, as packets crossing the providers network and packets sent through the same network encrypted (and therefore possibly invisible to deep packet inspection) can be compared. The assumption of encrypted traffic to be indifferentiable to the provider may turn out to be a problem as current SPID algorithms also target encrypted data [18]. Thus, additional measures will be necessary to obfuscate the encrypted channel. Consequently, there will be an off-trade to the measurement's accuracy. This approach has been implemented in the "N00ter"-project [17], Figure 2 shows the network setup for such an hybrid approach. In contrast to the active and the passive approach, a "black box" within the network providers infrastructure is necessary.

4) Crowdsourcing: The previously sketched approaches base on technical (using algorithms and statistics) evaluation – crowdsourcing uses human resources instead. The basic idea is: Ask people browsing the Internet to submit noticed cases of the Internet behaving "abnormal", e.g. sites being unavailable. The costs are very low: basically such a service would only need a public communication channel such as a web site or an e-mail-address.

The drawbacks of this approach are the drawbacks of crowdsourcing: Users have varying ideas of "blocking a site", probably depending on their knowledge.

IV. Counter-Measurement-Measures

Obviously, Network providers may have less to no interest in customers proofing their networks to be non neutral. Thus they may implement strategies to tamper with measurements. This goal could be reached by changing policies applied to network uplinks (from non-neutral to neutral). Such a measure would need a trigger – at this point the differences between active and passive approaches become additionally important.

This approach of avoiding neutrality violation proofs may work with every measurement utilizing data packets to well-known testing targets: Traffic to these targets can be interpreted as indicator for an immanent (or ongoing) test and used as trigger for a policy change.

V. Examples

This Section starts with examples of neutrality violations which have been observed. Subsequently software for detection of neutrality violations shall be presented.

A. Neutrality violations

It is worth to mention that this Subsection shows possibilities of neutrality violations. The observed techniques may not have been used with the intention of violating network neutrality, the observations can also be due to misconfiguration. Please keep in mind that the described phenomena could also be used in more harmful scenarios, e.g. to filter contents for political statements.

1) Connection interception: If users search for the term "falun gong" using the Chinese search engine "baidu.cn", the connection will be intercepted. According listings are provided in [19].

The user's client receives TCP-packets with active reset-flag which cause the connection to terminate. It is not possible to determine the origin of those packets: The server at baidu.cn or some routing station may have injected them. Even if (in case of injection) the server keeps sending packets after a connection reset has been injected, those original packets would probably be dropped by every stateful firewall.

2) Content manipulation: Today's network infrastructure equipment is capable of changing the payload of redirected packets. We will show this capability in two real-world scenarios.

In our first example, network equipment manipulates SMTP-connections. The response to the command "ehlo" gets manipulated. We observed a manipulation which caused the server identification and the announcement of encrypted communication with "STARTTLS" to be obfuscated. This obfuscation results in mail clients assuming the absence of encrypted connections via "STARTTLS" (which is probably prompted to the user who will eventually switch back to the use of plain connections, allowing the network provider to read transmitted contents). The listings (modified and unmodified) can be found at [19].

In the second example, web site contents are modified massively. Our example was a HTML-file just embedding an image. When requested through an UMTS network connection provided by the local Internet Provider "1und1", the file contents change: JavaScripts are included and the location of the embedded image points now to a location at the virtual (mapped) IP address 1.1.1.1. The image file at this different location is a size-compressed version of the original image (showing more artifacts). One can assume this manipulation is due to short network capacities. Detailed listings of this example can be found in [19].

3) Manipulation of HTTP Transfers: A different method of neutrality violation utilizes IP address spoofing to impersonate other entities. The "BlueSocket" wlan-access-control system shall be described here as example for commercial use of IP address spoofing. A manipulation takes place whenever an unauthenticated user tries to request a web site. In this

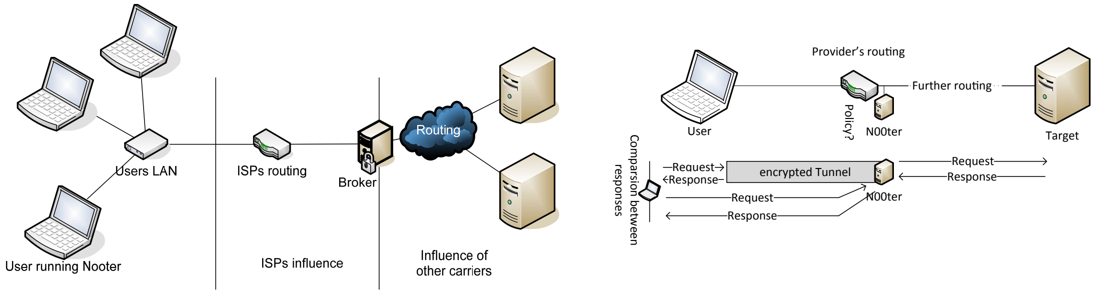


Fig. 2. Typical network setup for a hybrid measurement as proposed for N00ter [17] and illustration of N00ter's working principle: N00ter establishes an encrypted tunnel to a broker within the provider's network to perform differential measurements.

case, the answer does not originate from the queried server but from the BlueSocket-System: It redirects the user to its logon-page. To do so, it spoofs the original server address. Note that without domain knowledge it is not possible to differentiate whether the answer originates from the queried server or has been injected. A detailed dump can be found in [19].

B. Neutrality tests

This Section introduces some projects which aim to detect violations of network neutrality. Glasnost and NANO implement the active respectively passive approach. N00ter is a hybrid approach based on N00ter-boxes embedded in the network provider's infrastructure. ShaperProbe derives statements about traffic shaping from an evaluation of incoming data packets. Herdict represents an approach for detection of network blockages purely based on crowdsourcing.

1) Glasnost: Glasnost ([14], [20]) deploys the active approach. A test consists of several data exchanges, which are monitored by the server and the client application. Subsequently the data transfers are analyzed and a statement about neutrality is presented.

Glasnost was designed for easy usage. The end-user-part of Glasnost has been implemented as a Java applet embedded into a web page. The applet features only one user interface object: A "start"-button. The usability-thinking continues along the measurement: It has been designed to finish within a time which is short enough for the user to wait. Longer measurements would raise the method's precision, but testing showed that most users lost patience (or interest) whenever measurements took longer than 6 minutes [20].

According to [20], the statistical evaluation has been tuned to gain a false-positive rates about 0.7% – even in short tests.

To gain knowledge about traffic differentiation, Glasnost transfers two kinds of traffic. This data differs only in its contents, not in packet size or sequence. Consequently, timings and packet sizes remain the same allowing only deep packet inspection to differentiate between the dummy and the actual packets.

2) NANO: NANO ([15],[21]) represents the passive approach. An agent observes the network usage on a specified network interface. Additional information (e.g. the uplink media and a contact e-mail-address of the user) has to be provided during setup. NANO sends bundled data to an evaluation server using a secured channel. Currently all data is stored at the Georgia Institute of Technology. NANO is currently available for Linux users only; a Windows-Version had been announced.

Privacy concerns are considered in configurations: The user can disable the logging of traffic to specified hostnames. An additional piece of software may be used to suspend the monitoring service for a specified amount of time.

As described in [21], the accuracy of statements concerning traffic differentiation depends highly on the amount of analyzed data. Additional causal interferences make it difficult to provide an overall amount of false positives or false negatives.

3) N00ter: N00ter ([17]) follows a hybrid approach. As illustrated in Figure 2, the active part establishes an encrypted tunnel to a black box ("N00ter") within the Internet service providers network. Subsequently the N00ter acts as a proxy: It receives requests through the tunnel and forwards them to the (arbitrary) target. The N00ter receives the answer and sends it twice to the user's PC: Through the tunnel as well as through the ISPs plain network. The received answers are finally compared as the setup allows for differential measurements.

Additional measurements can be performed by sending the requests plain through the providers network, too.

4) ShaperProbe: ShaperProbe ([22], [23]) utilizes basal effects of traffic shaping: In typical scenarios, the activation of shaping algorithms can be easily noticed by tracking the times of incoming data packets. The effect is caused by some shaping algorithms: To limit the connection "speed" (packets per time or bytes per time) to a specified value, it needs to quantify its current value. To do so, an amount of time has to pass. Subsequently, packets get delayed.

This difference in packet timing between the first seconds and the following time (very fast start, long pause (to speed down), finally continuous amount of bytes/second) can be detected and evaluated.

After the actual network testing, statements about the existence of shapers on the data path are derived. Although the approach should be usable with arbitrary data transfers, ShaperProbe currently uses well-known targets.

5) Herdict: Finally we introduce Herdict ([24]) as representative of the crowdsourcing approach. It is quite straightforward: The user announces pages to be "accessible" or "inaccessible" and the site adds this entry (connected with the user's Internet service provider which is detected automatically) to its database. Entries can also be submitted to Herdict by Mail or twitter-message, although, as the Herdict-FAQ states, there exist exceptions: No sites exposing pornographic material will be accepted; additional the "Google SafeSearch"-filter is applied.

VI. Conclusion

Over the last five years, network neutrality violations became more frequent. This change led also to new development on the field of neutrality violation analysis. There are currently two ready-to-use testing methods, the active and the passive method. Still, neutrality violations can never provide absolutely trustworthy results: Active tests may be detected by network operators (and thus be manipulated), passive tests either suffer from a lack of raw data to evaluate or need to collect data of multiple users for central evaluation. Differentiation between intended neutrality violations and network congestions remain a difficult task.

A comparison of the different approaches' results is not useful: They test different network properties. Statements derived from active approaches concern well-known testing connections. While some active approaches enable the user to test multiple protocols and multiple test targets (Glasnost), other approaches rely on single targets (ShaperProbe). This difference is caused by different design tenets: ShaperProbe does not assume shaping to differentiate between different kinds of data streams – Glasnost does. Statements derived from crowdsourcing depend highly on users posting neutrality violation suspects. Statements generated by passive approaches depend on user's Internet usage. Therefore, this approach has to deal with noise, perhaps more than other approaches. Combination of these approaches leads to hybrid approaches (as N00ter), which finally allow clear statements as they use the same viewpoint as purely passive approaches extended to a second channel (which is assumed not to be influenced by the provider). This allows a direct comparison between data exchange through a provider's network while it may be influenced on one channel and not influenced on the other channels.

Although the perfect solution for network neutrality analysis is yet to be found, existing approaches provide a wide range of analytic tools. Existing approaches enable users to scan for (dumb) shapers, or to test singular protocols. Passive approach driven projects seem a promising field of future work as they solve the active approaches' problem of restricted viewpoints.

References

[1] G. M. Bullinger, "Netzneutralität: Pro und Contra einer gesetzlichen Festschreibung," *Deutscher Bundestag: Wissenschaftliche Dienste*, June 2010.

[2] J. Crowcroft, "Net Neutrality: The Technical Side of the Debate: A White Paper," *SIGCOMM Comput. Commun. Rev.*, vol. 37, pp. 49–56, January 2007.

[3] U. Mansmann, "Kabel Deutschland drosselt Filesharing für Bestandskunden," 2012. http://heise.de/-1652920, last accessed 2013-04-29.

[4] EC, DG Communications Networks, Content and Technology, "Online public consultation on "specific aspects of transparency, traffic management and switching in an Open Internet"," 2012. http://ec.europa.eu/information_society/digital-agenda/actions/oit-consultation/index_en.htm, last accessed 2013-04-29.

[5] A. Wilkens, "Große Internet-Untrnehmen formen Lobbyverband für ein 'freies Internet'," 2012. http://heise.de/-1653423, last accessed 2013-04-29.

[6] A. Wilkens, "Musikindustrie setzt weiter auf Websperren, Warnhinweise und Filter," 2012. http://heise.de/-1653013, last accessed 2013-04-29.

[7] A. S. Tanenbaum and D. Wetherall, *Computer Networks*. Pearson, 5. ed., 2011.

[8] A. Chaudhary and A. Sardana, "Software Based Implementation Methodologies for Deep Packet Inspection," in *Information Science and Applications (ICISA), 2011 International Conference on*, pp. 1 –10, april 2011.

[9] R. K. Lenka and P. Ranjan, "A Comparative Study on DFA-Based Pattern Matching for Deep Packet Inspection," in *Computer and Communication Technology (ICCCT), 2012 Third International Conference on*, pp. 255 –260, nov. 2012.

[10] A. Ali and R. Tervo, "Traffic identification using Bayes' classifier," in *Electrical and Computer Engineering, 2000 Canadian Conference on*, vol. 2, pp. 687 –691 vol.2, 2000.

[11] R. Archibald, Y. Liu, C. Corbett, and D. Ghosal, "Disambiguating HTTP: Classifying web Applications," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International*, pp. 1808 –1813, july 2011.

[12] W. Jiang and M. Gokhale, "Real-Time Classification of Multimedia Traffic Using FPGA," in *Field Programmable Logic and Applications (FPL), 2010 International Conference on*, pp. 56–63, 31 2010-sept. 2 2010.

[13] Kefu, X. and Deyu, Q. and Zhengping, Q. and Weiping, Z., "Fast Dynamic Pattern Matching for Deep Packet Inspection," in *Networking, Sensing and Control, 2008. ICNSC 2008. IEEE International Conference on*, pp. 802 –807, april 2008.

[14] "Glasnost: Test if your ISP is shaping your traffic." http://broadband.mpi-sws.org/transparency/bttest.php, last accessed 2013-04-29.

[15] Feamster, N. and Ammar, M. and Mukarram bin Tariq, M. and Motiwala, M., "GTNOISE Network Access Neutrality Project," 2011. http://gtnoise.net/nano/, last accessed 2013-04-29.

[16] M. Dhawan, J. Samuel, R. Teixeira, C. Kreibich, M. Allman, N. Weaver, and V. Paxson, "Fathom: A Browser-Based Network Measurement Platform," in *Proceedings of the 2012 ACM conference on Internet measurement conference*, IMC '12, (New York, NY, USA), pp. 73–86, ACM, 2012.

[17] D. Kaminsky, "Black Ops Of TCP/IP 2011," *Defcon*, 2011. http://dankaminsky.com/2011/08/05/bo2k11, last accessed 2013-04-29.

[18] C. Liu, G. Sun, and Y. Xue, "DRPSD: An novel method of identifying SSL/TLS traffic," in *World Automation Congress (WAC), 2012*, pp. 415 –419, june 2012.

[19] http://opsci.informatik.uni-rostock.de/index.php/NN2013, last accessed 2013-04-29.

[20] M. Dischinger, M. Marcon, S. Guha, K. P. Gummadi, R. Mahajan, and S. Saroiu, "Glasnost: Enabling End Users to Detect Traffic Differentiation," March 2010.

[21] M. B. Tariq, M. Motiwala, N. Feamster, and M. Ammar, "Detecting Network Neutrality Violations with Causal Inference," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, CoNEXT '09, (New York, NY, USA), pp. 289–300, ACM, 2009.

[22] P. Kanuparthy, "Shaperprobe." http://www.cc.gatech.edu/˜ partha/diffprobe/shaperprobe.html, last accessed 2013-04-29.

[23] P. Kanuparthy and C. Dovrolis, "ShaperProbe: End-to-End Detection of ISP Traffic Shaping using Active Methods," in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, IMC '11, (New York, NY, USA), pp. 473–482, ACM, 2011.

[24] "Herdict: Help spot web blockages." http://www.herdict.org/, last accessed 2013-04-29.