Hardware Acceleration for Cyber Security

Jiří Novotný
CESNET, z.s.p.o.
Zikova 4
160 00 Praha 6
Czech Republic
novotny@cesnet.cz

Pavel Čeleda
Masaryk University
Botanická 68a
602 00 Brno
Czech Republic
celeda@ics.muni.cz

Tomáš Dedek
CESNET, z.s.p.o.
Zikova 4
160 00 Praha 6
Czech Republic
dedek@liberouter.org

Radek Krejčí
Masaryk University
Botanická 68a
602 00 Brno
Czech Republic
radek.krejci@mail.muni.cz

ABSTRACT

These days the problem of cyber security is of utmost importance. Massive cyber attacks targeting government and mission critical servers can swiftly become an issue of national security. Various approaches for cyber defence and cyber security used to date have been based on software solutions without hardware acceleration. With the increasing number of network users, services and the current generation of multi-gigabit network links, the amount of transferred data has increased significantly. These facts have rendered many current solutions for network security obsolete. This paper presents hardware-accelerated system for cyber security. The time and performance critical parts are processed in hardware and only the relevant traffic parts are processed in software. Such approach allows us to use current security tools in multi-gigabit networks under worst-case scenarios like a distributed denial-of-service attacks.

1 INTRODUCTION

The role of cyber security become important in past decade. Successful example of cyber attack was attack against NATO member Estonia in 2007. NATO needs to develop effective countermeasures for such kind of attacks and threats in cyberspace.

In our research effort we concentrate on hardware-accelerated network security monitoring as an important part of cyber security. This paper presents hardware-accelerated system to support security tasks in high-speed networks. We target these most important issues of today's cyber security:

- How to deal with large amount of data passing through current generation of high-speed networks.
- Software-only monitoring solutions are not fast enough for the multi-gigabit networks.
- Many of special hardware appliances are not flexible enough to deal with newest cyber threats.
- Data coming from network devices have no sufficient quality.

We use hardware acceleration based on Field-Programmable Gate Array (FPGA) technology to provide sufficient computation power. The key component of our system is NIFIC probe - commodity PC server equipped with COMBOv2 acceleration card and Network Interface Filtering Card (NIFIC) firmware supporting 2x 10Gbps line rate processing.

The probe provides high quality data coming from more monitoring methods which benefits from hardware accelerated preprocessing (packet filtration, statistic counting, tail cutting, etc.) Time critical parts of monitoring algorithms are processed in hardware (working on multi-gigabit speed), while the complex algorithms (working on preprocessed data) are processed in host computer. This approach allow us to overcome issues mentioned above.

The NIFIC probes are supposed to be deployed to the observation points either at access gateways or in critical points of the network. They send security related information to the Security Operations Center

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **NOV 2010**	2. REPORT TYPE	3. DATES COVERED **00-00-2010 to 00-00-2010**

4. TITLE AND SUBTITLE **Hardware Acceleration for Cyber Security**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **CESNET, z.s.p.o.,Zikova 4,160 00 Praha 6,Czech Republic,**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

presented at the Information Systems and Technology Panel (IST) Symposium held in Tallinn, Estonia, 22-23 November 2010. U.S. Government or Federal Rights License

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**16**	

(SOC). The SOC observes security information provided by probes. The probes can be remotely reconfigured to focus in more detail to selected traffic or/and filter out malicious one. This integration of monitoring and executive features enables the operating staff - Computer Security Incident Response Team (CSIRT) [13] to efficiently solve the network incidents.

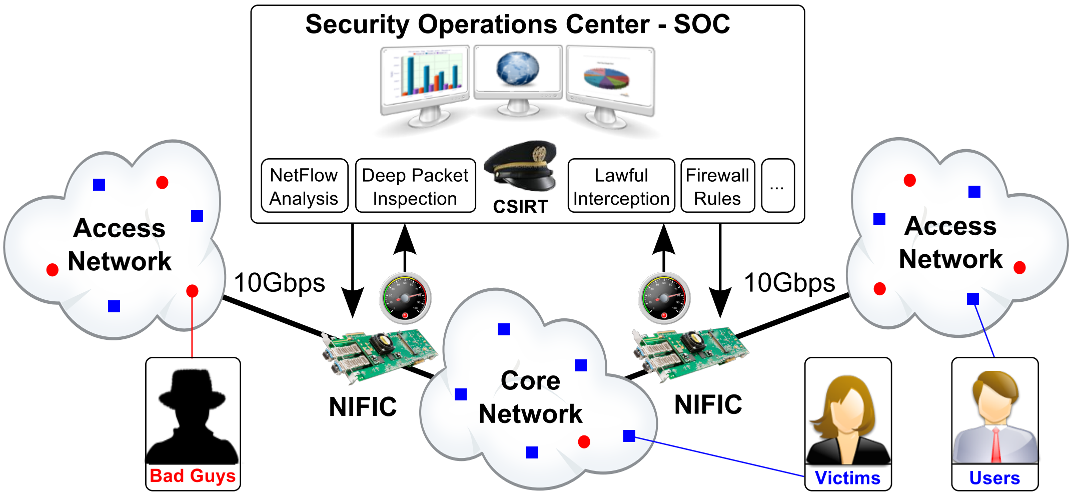

Figure 1: The system overview, with network security monitoring, acquisition and preprocessing layer at the bottom and security operations center on the top.

2 RELATED WORK

CSIRT teams need many various information sources for effective work. One of the most important sources are data collected from computer network in observation points via monitoring probes. Data used for security monitoring has to be of the highest quality. For example packet sampling could be sufficient for bussines oriented applications, but it is definitely not sufficient for anomaly detection [3].

Network security monitoring probes are either part of generic networking devices (routers, switches), specialized hardware appliances (hardware firewalls, IDS/IPS devices), commodity hardware boxes with appropriate software or PC based probes.

The monitoring extension in generic networking devices generate data with “sufficient quality” in standard conditions, but this extension needs extra processing power [8]. In time of attack they must provide packet forwarding and often have no power for network monitoring. The situation is even worse. They are in contrary with standalone boxes visible on the network and they are often targets of attackers. In many cases even mirrored data, without any preprocessing, have no sufficient quality [36] and the TAPs must be used for the standalone probes.

Specialized hardware appliances have in general sufficient power to work under attack. On the other hand it is hard to add extra functionality. PC based probes have advantage in their flexibility and relatively low cost but have performance issue in high-speed networks [1]. PF_RING solution [26] from Luca Deri can receive up to 1.2 Mpps in special conditions. The TNAPI [27] from the same author can receive up to 5 Mpps and up to 9 Gbps, which is not enough for packetloss processing on 10 Gbps lines. The performance lack can be solved using hardware accelerators based on FPGA technology [1], [32].

The commercially available FPGA accelerators are DAG cards [12] from Endace and network adapters from Napatech [23]. Platforms provided by research comunity are COMBO cards [4] from CESNET and NetFPGA [24] cards from Stanford. Endace and Napatech cards are focused on fast data transfers into computer memory with minor possibility to offload packet processing into card. CESNET offers NetCOPE [21] development kit for rapid development of network applications on COMBO cards. NetCOPE allows to develop hardware accelerated applications with offloading time critical parts of algorithms into FPGA. The NetFPGA cards are designed for educational purposes with many tutorials and test applications. Endace, Napatech and CESNET support 10 Gbps line rate. NetFPGA cards support 1 Gbps line rate.

There is available huge portfolio of monitoring tools for the PC based computers [31] with different functionality from simple tools to complex monitoring systems. The complex systems have lack of performance and setup is rather complicated. Many of CSIRT experts prefer to use combination of simple tools and join their output to achieve necessary monitoring functionality. Such approach can be very handy if the complex tools doesn't provide desired information.

Several software tools can be combined in one monitoring box for low speed networks. This approach fails for multi-gigabit networks and it is necessary to use more monitoring boxes in one observation point. Even that, some monitoring applications can still need more computational resources, than are available in commodity PC box. Main bottleneck is not computational power of processors (multicore/multiprocessor systems have sufficient computational resources), but missing support for traffic distribution among the processor cores and low throughput for short packets with using of commodity NIC cards [32].

Due to complexity of todays networks the monitoring probes are located in many places and produce huge amount of data. It leads to use of new approach to distributed data processing. The older works were based on Snort [11] and honeypots [16], the newer ones use flow monitoring [22], [28]. These concepts use distributed approach with just one monitoring method (data coming from packet inspection, flow, etc.) on the other hand we combine more monitoring methods together.

3 SYSTEM ARCHITECTURE

The presented system, as introduced on Figure 1, consists of one or more deployed NIFIC probe(s) connected to the SOC. Thanks to this approach, SOC is capable to get data from different observation points to increase information value of received data. The main contribution of using NIFIC probes is their ability to filter passing high-speed network traffic according to specific requirements. This guarantees that all information are based on complete i.e., not sampled data and conforms to real state of the network. In advance observed data is usually preprocessed on each probe by specific tools to speed up and simplify central data processing at SOC and to decrease amount of data transferred from probes to SOC. Decisions based on results of data analyses can be applied immediately, without any packet loss, to all deployed NIFIC probes.

Deployment of several NIFIC probes across the network allows SOC rapid response to current security threats. Observing data passing by links connecting monitored network with Internet and other outer networks helps to deal with outside attackers. But also taking care of the key points inside the network, like links to the data storages or database servers, is crucial since the most security threats come from internal network. The cooperation of involved devices is essential to prevent and to investigate security incidents.

Not only diversity of observation points improves applicability and accuracy of the network monitoring system. The possibility to apply different methods and tools to the same, or closely related, data is beneficial. Thanks to filtration and distribution of the observed data to the appropriate applications NIFIC probe allows to perform different approaches. It includes behavioral analysis, by means of NetFlow monitoring, as well as packet content analysis, so called Deep Packet Inspection (DPI).

Following sections describe hardware and software architecture of the NIFIC probe including remote configuration from SOC and probe deployment on the network. Finally also performance characteristics of the NIFIC probe are discused.

3.1 NIFIC Probe Architecture

The NIFIC probe architecture consists of several layers (see Figure 2). The low-level layers are optimized for line rate processing with zero packet loss. The upper layers are user oriented and support network traffic post processing, data visualization and propagation of SOC decisions in network behavior.

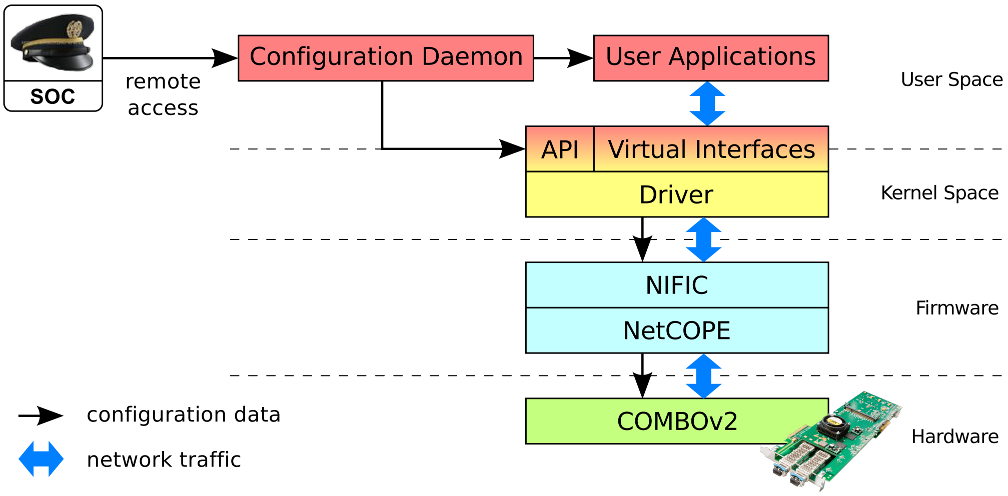

Figure 2: The functional layers of the NIFIC probe.

The NIFIC probe is a standalone packet filter, which is based on the commodity PC with COMBOv2 card (COMBO-LXT + COMBO-10G2) serving as hardware accelerator.

The NIFIC firmware is developed on NetCOPE platform and supports 2x10 Gbps line rate processing. The packet filter functionality is set up by rules in OpenBSD Packet Filter format. Every rule describes what should be done with the incoming packet, if the packet header matches the conditions used in the rule. The rules are ordered by priority. The rule with the higher priority is used, when packet match several rules.

One of the most important system features is that the rules could be changed on the fly without any packet loss. The packet could be classified accordingly to the MAC addresses, IPv4 address (supports prefixes), L4 protocol, TCP or UDP ports (supports range), TCP flags and input interface number.

Simple rule set, which configures NIFIC firmware to capture to software all packets incoming from the desired subnet, could look like this:

```
# pass all traffic from the desired subnet to software virtual interface 2
10 pass 1 2 on 0 from 147.251.21.0/24 to any
# pass the rest of the traffic through NIFIC
20 pass 0 on 1
30 pass 1 on 0
```

The previous example shows the usage of the **pass** action. The **block** action is also useful. For example, the following rule set configures the NIFIC probe to block all traffic incoming to the desired port.

```
# block the all traffic to port 80 on the interface 0
10 block on 0 from any to any port 80
# pass the rest of the traffic through NIFIC
20 pass 0 on 1
30 pass 1 on 0
```

These examples are very simple. The whole set of the supported header fields can be used in the rule to create a complex one.

Filtered data is provided to the user space by driver in a form of 14 special virtual interfaces accessed through the libsze2 library. Virtual interfaces are numbered from 2 to 15, while the lowest two numbers are reserved for physical ports. These numbers are used in filtration rules to describe data forwarding. Using the set of virtual interfaces allows to filter data for different types of applications or for the same applications intended for different purposes. Specification of data forwarded into each virtual interface fully depends on specified filtration rules including possibility to send one packet to more virtual interfaces.

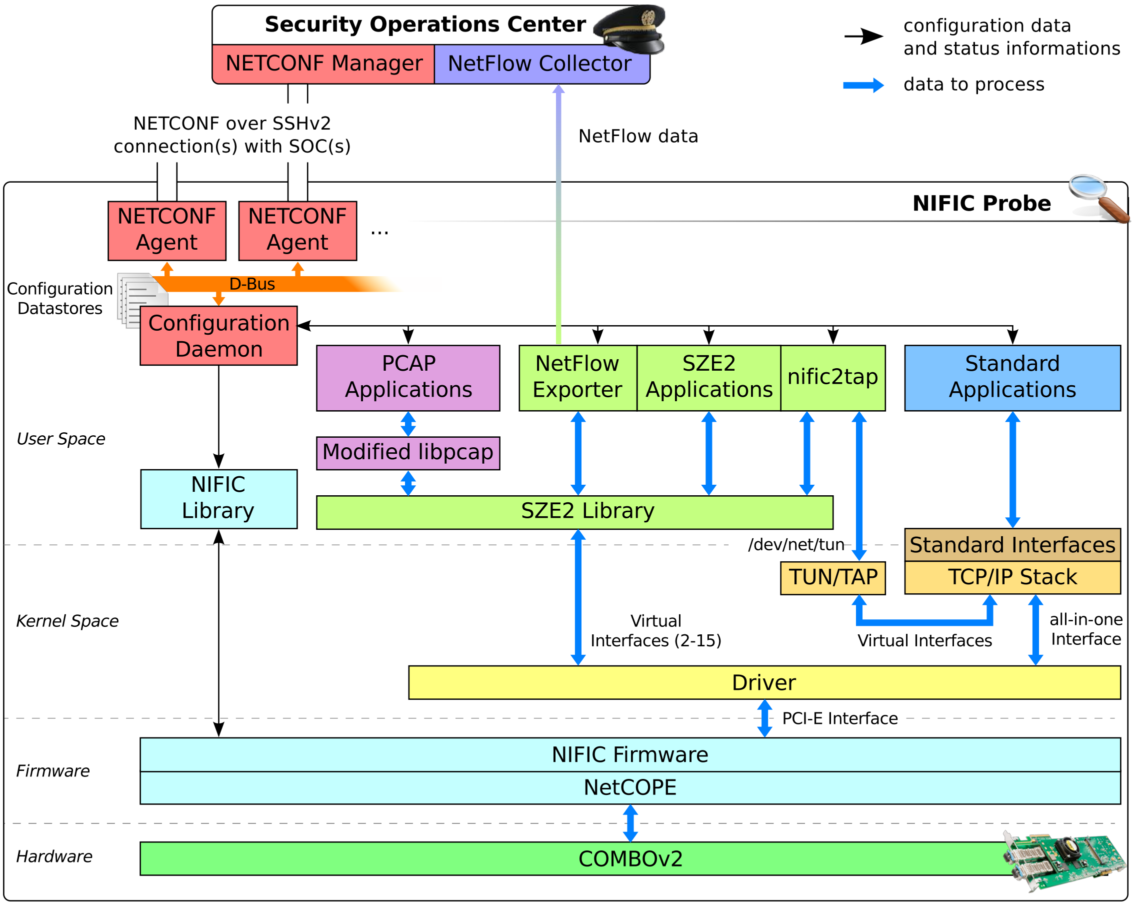

Figure 3: The NIFIC probe architecture schema.

Any application processing observed data is allowed to get data using following three approaches (see Figure 3).

- Applications read data using the Straight Zero Copy library (libsze2). This approach requires specific application's capability to use libsze2 Application Programming Interface (API). The example of such application is NetFlow exporter described in [5].

- We provide modified libpcap library using libsze2 API. This libpcap-sze library provides the same API as original libpcap library and this way it serves as a special layer between libsze2 and standard applications using libpcap [33].

- The last option is the most general but also the slowest one because data have to pass through the TCP/IP stack of the operating system. We use special *nific2tap* application that forwards data read from libsze2 to the TUN/TAP device driver [19]. This way observed data is accessible to all applications in a form of a standard network interface configured by *ifconfig* tool. *nific2tap* allows each NIFIC virtual interface to be accessed as a standalone network interface. By default the Combo driver provides one another standard network interface containing data from all virtual interfaces together.

Configuration changes made from SOC using remote configuration (see Section 3.3) are applied by NIFIC configuration daemon controlling both hardware and software parts of the NIFIC probe.

3.2 NIFIC Firmware Architecture

NetCOPE, the platform for rapid development of network applications was used for the NIFIC firmware development. NetCOPE provides a Network Module for 10 Gbps Ethernet interfaces and a Direct Memory Access (DMA) Module for fast packet transfer between the card and the host PC. The NIFIC firmware consists of classification and packet forwarding core (NIFIC core), which is connected to the Network and DMA Modules of the NetCOPE platform. The NIFIC firmware architecture is illustrated on Figure 4 and described further.

The packets are incoming to the card from network physical interfaces (2x10 Gbps) or from the PCI-E interface of the NIFIC probe. The packets from network are then processed by the Network Module, the packets incoming from PCI-E are processed by the DMA Module. In the Network Module and in the DMA module packets are transformed to the LocalLink like protocol, which is used as an input and also output to and from the classification and packet forwarding core.

The important fields from packet headers are parsed in Header Field Extractor (HFE) units and the whole packets are also stored to Packet Buffers. The extracted header fields serve as inputs for classification, where the packet is classified by the Classification Unit. The Classification Unit provides the action, which should be performed with the packet. The action is inserted as an additional information to packet read from the packet buffers in Header Insert units (HI) and packet continuous to the final stage, to Crossbar. In Crossbar, accordingly to the action the packet is either discarded or forwarded to one or more output interfaces. Finally, packets outgoing from Crossbar continue to the output part of the Network Module to be sent back to network (interface 0 or 1) or to the DMA module to be sent to NIFIC probe software interface (virtual interfaces from 2 to 15).

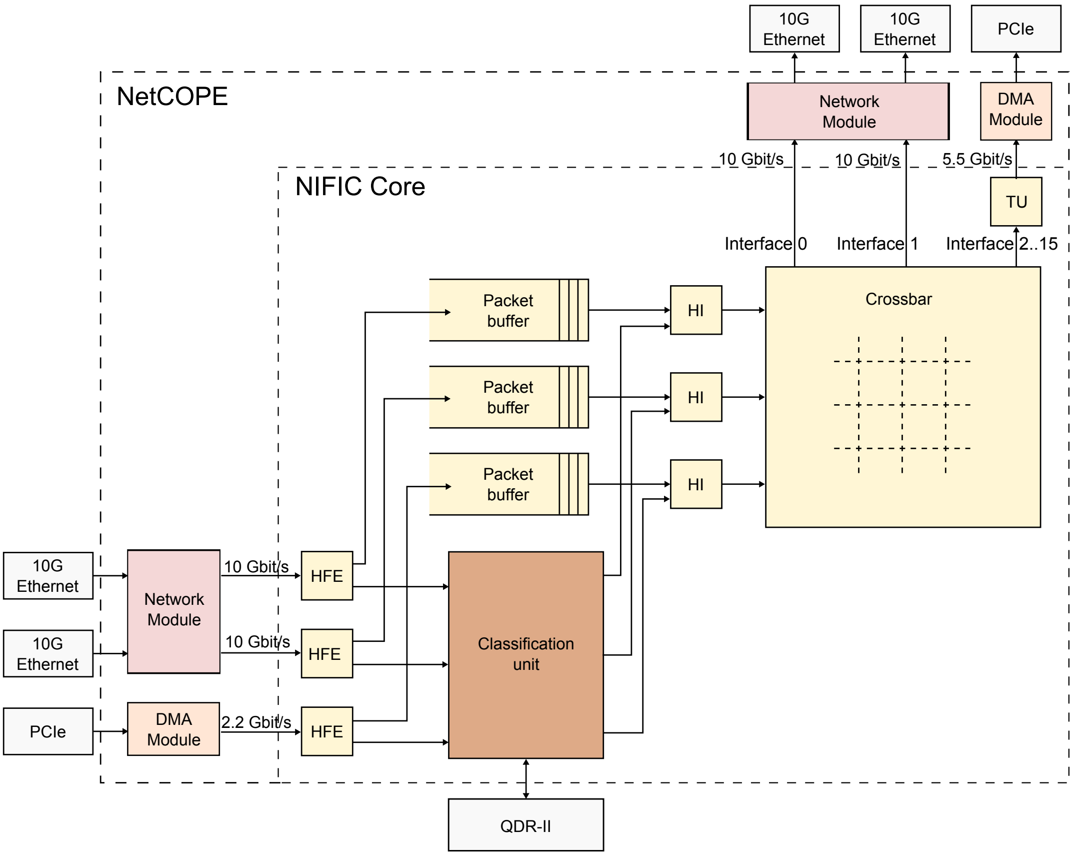

Figure 4: The schema of the NIFIC probe firmware.

The classification algorithm [29] is based on the problem decomposition, where techniques like Last Prefix Match and Perfect Hashing are used. The configuration of the classification is loaded to the Block-Ram memories and to the off-chip Quad Data Rate (QDR II) static RAM (SRAM) memory as the software preprocessed binary data. All memories are divided to the two contexts, where the first one is used for the actual configuration of Classification Unit and the second one is used for the future configuration. This two-context solution enables to change the rules (configuration of Classification Unit) on the fly without any packet loss, because all configuration data of the new rule set could be loaded to the Classification Unit when Classification Unit is still processing the packets on the basis of the old rules. When all new configuration data is written, the Classification Unit is simply switched to use the second context.

The current implementation of NIFIC firmware is focused on the full throughput of the network lines (2x10 Gbps). The maximum throughput of the host PC software interface is 2.2 Gbps for sending of packets from the software of NIFIC probe and about 5 Gbps for the receiving of the packets. The mentioned values are limited by hardware implementation, where timing and area constrains have to satisfied. Nevertheless, if there is a need the card with faster FPGA chip could be used or the architecture could be change to increase the throughput to NIFIC probe software.

However, the both mentioned limits are more than sufficient for all uses cases described in Section 4.

3.3 NIFIC Probe Remote Configuration

The remote configuration of the NIFIC probes is based on the NETCONF protocol [14] using SSHv2 protocol as its transport layer [34]. We use our own C implementation of the NETCONF protocol. It provides a simple mechanism to manipulate configuration data of target device as well as to retrieve relevant device status information. On Figure 5 you can see general schema of the NETCONF remote configuration system presenting connection of the NIFIC probes with SOC. Generic NETCONF protocol implementation consists of NETCONF manager providing user interface on the client (SOC) side and the NETCONF agent application that controls configuration datastores on the device side. On the device side, the agent is connected with specific NIFIC configuration daemon. This daemon is equipped with the capabilities applicable to the NIFIC probe. It includes ability to change filtration rules or to start or stop third-party applications processing observed data on specific virtual interface. Communication between configuration daemon and NETCONF agent is provided by D-Bus message system.

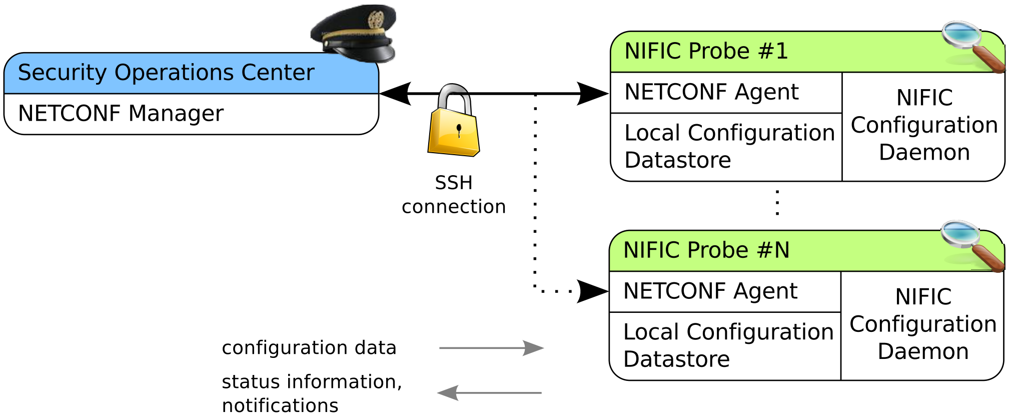

Figure 5: The remote NIFIC configuration using NETCONF protocol.

NETCONF uses simple Remote Procedure Call (RPC)-like approach to exchange messages between manager and agent application. This synchronous communication is used to push control commands and configuration data to the device. Using control commands, user is, for example, able to get status information on demand. In addition to this approach, NETCONF provides also asynchronous communication mechanism called NETCONF Notifications [7]. Notifications are messages sent to the management center when specific event occurs. Such event can be generally anything but in most cases they concern for changing configuration data or reaching some specific status value. To prevent receiving unwanted notifications, management application has to initially subscribe to receive messages about the specific kind of events which user is interested in.

Mandatory usage of SSHv2 as NETCONF's transport protocol provides confidence of carried configuration data and status information. NETCONF agent is started as SSH Subsystem so user authentication is performed directly by SSH. SSH user account is then used for authorization to access configuration data.

Protocol control messages as well as configuration data, including filtration rules and application settings, carried between manager and agent are encoded with the Extensible Markup Language (XML). Using XML enables further configuration data processing with common tools. Configuration data model specifies settings for all layers of NIFIC probe. NETCONF agent provides access into three types of configuration datastores where device configuration data is stored.

Startup datastore is default configuration datastore loaded at NIFIC probe startup. It contains verified configuration data specifying default device behavior.

Running datastore represents current NIFIC probe settings. Any change made to running configuration data is immediately applied to the NIFIC probe settings including all hardware and firmware settings as well as software applications parameters.

Candidate datastore is supposed to be some kind of sandbox datastore that is initially set as a copy of the running datastore. Operator can find out impact of different set of changes applied on current configuration data without any real impact to the device parameters. Performed changes are not applied to the device itself but only to the candidate configuration data. When performed changes seems right, the candidate configuration data can be committed into the running datastore and all prepared changes are applied.

It is common that one management center controls many devices (NIFIC probes). On the other hand, management center can be duplicated so multiple management centers can access and control deployed NIFIC probes. Concurrent access to the configuration data is solved by NETCONF's capability to lock configuration datastore.

3.4 NIFIC Probe Connection to User Network

The are several possibilities how to connect the NIFIC probe to the user network. For simple monitoring purposes, the mirror port of the switch could be used, but there could be problems with quality of data as was mentioned in Section 2, so it is better to use special TAP device. However, to fully utilize the filtering features of the NIFIC probe the suggested way of the NIFIC probe connection is directly to the user network line, where NIFIC probe acts as a network active device.

An example of a such connection is presented on the Figure 6. The COMBO-10G2 card is equipped with two XFP cages, where multi mode, mono mode, CWDM or copper transceivers can be used for connection to the user network.

The NIFIC probe also has to be configured with suitable rules. The rules on the Figure 6 configure the NIFIC probe to forward all traffic incoming from interface 0 to interface 1 and vice versa. In this connection and configuration the NIFIC probe acts as a network repeater.

From this starting point, user could configure the NIFIC probe to forward a traffic to the software interface or to block some incoming traffic. However, the two forwarding rules should be preserved to ensure the basic repeater functionality.

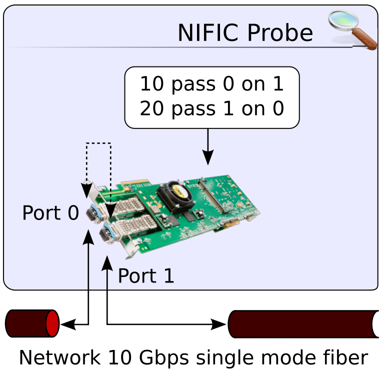

Figure 6: The NIFIC probe connection to user network.

3.5 NIFIC Throughput Test

To precisely measure the throughput of the NIFIC probe the probe was tested accordingly to the RFC 2544 [2]. The three scenarios were tested on the current NIFIC probe:

- Throughput among the physical Ethernet interfaces, where NIFIC probe was configured as a repeater.
- Throughput to the software of the NIFIC probe, where traffic was coming to both Ethernet interfaces and SZE2 tool was used to capture data.
- Same as the previous but tool using standard interface was used to capture data.

The results are summarized on the Figure 7. As could be easily observed, packets are passing through the NIFIC probe firmware without any packet loss, when whole traffic is forwarded from one port to another. This result is very significant, because the packet is not simply forwarded from one port to another. Every packet even for such a simple rules is classified by the classification algorithm, so the interpretation of the results is, that the NIFIC probe could classify packets at line rate for 2x10 Gbps lines without any packet loss. The feature of switching the rules without any packet loss was also positively tested.

The throughput to software has the hardware limitation at 5 Gbps level for all packet lengths as could be observed from the SZE2 tool graph. The throughput to the standard interfaces is limited for the short packets by the TCP/IP Stack of the NIFIC probe operating system. However, for the longer packets the throughput of standard interfaces is at the same level as for the SZE2 tool.

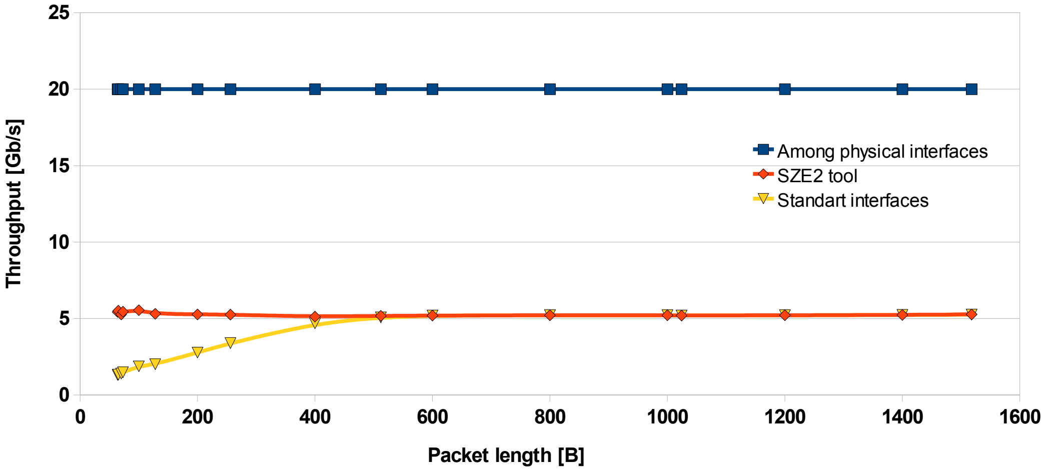

Figure 7: The current firmware of NIFIC probe test results.

For some special applications the throughput of the current NIFIC probe could be insufficient. Due to the fact, that firmware of the NIFIC probe could be easily re-loaded without need to take system down, the NIFIC probe firmware could be replaced with the NetCOPE based Network Interface Card (NIC) firmware. This firmware has the throughput to the software on the limit of whole PCI-E system. The measured throughput could be observed on the Figure 8.

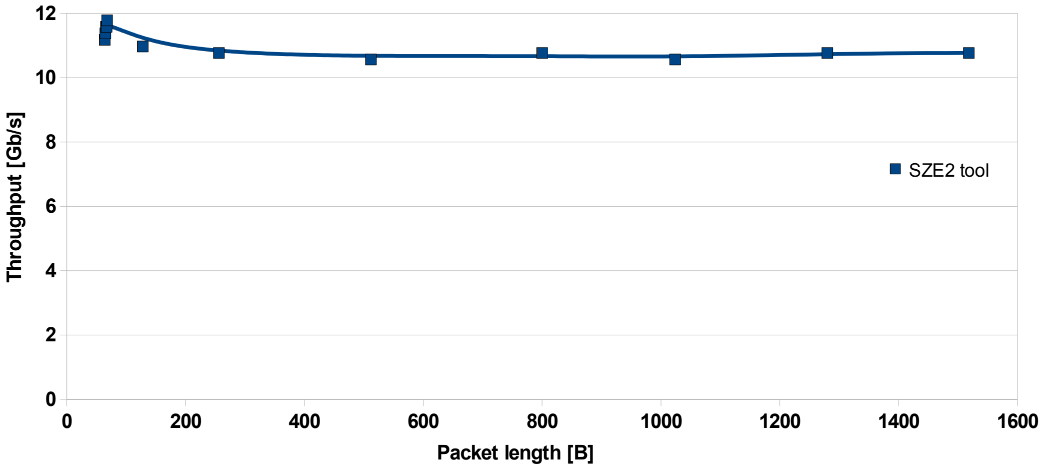

Figure 8: The NIC design with maximum throughput.

4 USE CASES AND DEPLOYMENT SCENARIOS

This section describes how to use and deploy NIFIC probes. Selected use cases are based on real life examples and cover areas from a passive network monitoring to an active traffic processing. We use 64-bit Linux operating system on hosts with COMBOv2 cards. The software applications using NIFIC include FlowMon NetFlow/IPFIX generator, Wireshark packet analyzer, iptables - Linux kernel firewall, deep packet inspection tools etc. Virtually any libpcap [33] based tool can be used with NIFIC see Figure 9.

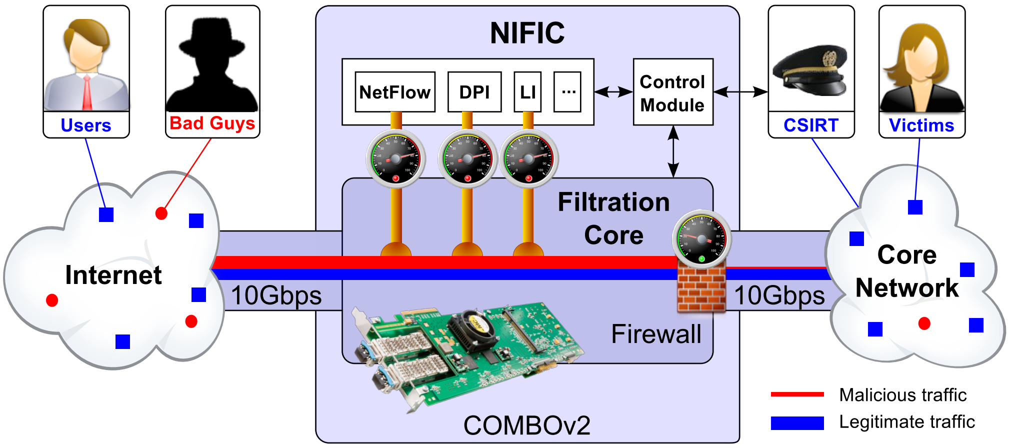

Figure 9: The virtual security monitoring center - generates NetFlow information, provides packet traces for deep packet inspection and protects network with built-in firewall.

4.1 Flow-based Network Traffic Monitoring

The NetFlow [9] and IP Flow Information Export (IPFIX) [10] statistics are used to get information about who communicates with whom, when, how long, how often, using what protocol and service and also how much data was transferred. The NetFlow or IPFIX export can be enabled on routers which constitute primary source of flow data today. On the other hand utilization of standalone dedicated systems such as FlowMon [6] seems to have several benefits. Offloading of resource intensive flow measurement to dedicated hardware probe is probably the most important one. For more details about hardware-accelerated flow measurement see [25].

We have modified FlowMon probe software to use NIFIC's interfaces. Packet header is read from hardware. Software creates a flow record and prepared flow record is inserted into flow cache. Expired flows are exported using NetFlow or IPFIX protocol.

Generated flow data is collected [17] and tools supporting network behavior analysis [30] are used to identify malicious traffic. Using network behavioral analysis in comparison with signature based approach allows us to recognize zero-day attacks, unknown viruses and new malware. Flow measurement is passive and we don't create or modify any traffic on the observed network. Even encrypted traffic (communication endpoints) can be monitored using flow technology.

4.2 Web Access Analyzer

The Hypertext Transfer Protocol (HTTP) traffic analysis is very important and provides a lot of interesting information. Typically HTTP is passed through firewalls. This feature is used by many applications to bypass firewall rules. The flow data doesn't provide necessary information in case of web server virtual hosting [15]. The web servers use to host more than one domain name on the same IP address see Figure 10.

```
Timestamp              Source-IP     Dest-IP          Method   Host
2010-03-18 20:35:09    172.16.30.2   147.251.21.139   >  GET   www.liberouter.org
2010-03-18 20:35:24    172.16.30.2   147.251.21.139   >  GET   canti.liberouter.org
```

Figure 10: The virtual host access logged with *httpry* application [18].

The HTTP protocol uses well known ports (e.g. 80, 3128 and 8080). Using NIFIC we can set a rule and send all HTTP traffic to a virtual interface. Further analysis can by performed with specialized packet analyzers like *httpry*, *tcpdump*, *wireshark* etc. The HTTP analysis is passive and we don't create or modify any traffic on the observed network.

4.3 VoIP Analyzer

Voice over Internet Protocol (VoIP) is a technology to deliver voice communications over IP networks. VoIP systems employ session control protocols (e.g. SIP, H.323 and H.248) and transport protocols (e.g. RTP, RTCP and SRTP). The call control protocols establish, modify and release connections. The transport protocols encode speech allowing transmission over an IP network as digital audio via an audio stream.

The lower cost and greater flexibility helped to increase number of deployed VoIP systems in past decade. VoIP systems are susceptible to attacks as are any network-connected devices. The hackers can institute denial-of-service attacks or collect sensitive call data. VoIP systems may have security weaknesses [20] which can affect confidentiality, integrity and availability of your VoIP environment.

The Wireshark packet analyzer [35] supports advanced VoIP analysis. Using Session Initiation Protocol (SIP) traffic filter in NIFIC (UDP or TCP with dst port 5060) we were able to trace VoIP handshakes at 10 Gbps backbone level. Hardware accelerated approach guarantees packet loss measurement of VoIP traffic. In software we process only VoIP related traffic (typically a small subset of whole traffic). The VoIP analysis is passive and we don't create or modify any traffic on the observed network.

4.4 Network Protector

The network protector works as a packet filter and it accepts or rejects packets based on user-defined rules. We set an Access Control List (ACL) as NIFIC's filtration rule to protect network against bad guys and unauthorized access. It is completely transparent (line rate compliant) for legitimate users and leakproof for bad guys (infected hosts, . . .). Malicious IP addresses are blocked to stop further propagation of unwanted traffic.

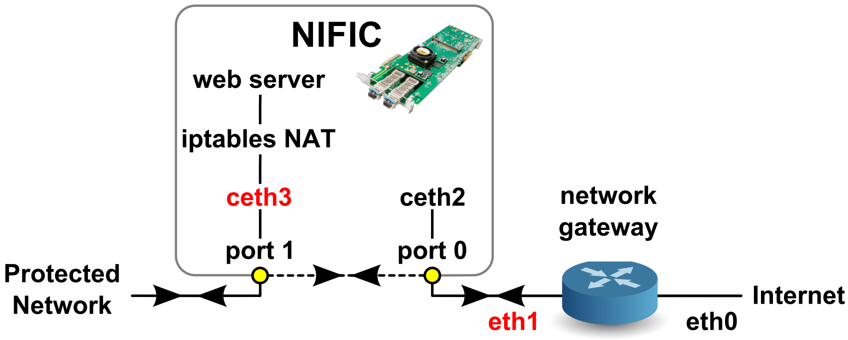

Figure 11: The network protector - connection schema.

The Figure 11 shows deployment of network protector. NIFIC is connected as last device before local network gateway that provides connection to the Internet. NIFIC port 0 is connected to the gateway and has no IP address. NIFIC port 1 connects the rest of local network and it duplicates gateway's interface for local network including IP address and, which is really important, its Media Access Control (MAC) address.

Legitimate hosts are transparently connected through NIFIC to gateway and they are not restricted by NIFIC in any way. Blocked hosts are redirected by NIFIC to ceth3 interface and HTTP traffic (port 80) is processed by NIFIC web server. Finally web page is send back to show message about blocking.

4.5 Traffic Limiter

iptables (Linux kernel firewall) supports a rich set of packet manipulation options. For instance to limit amount of ingress Internet Control Message Protocol (ICMP) traffic to your network we send them through *iptables* keeping forwarding of other protocols in hardware.

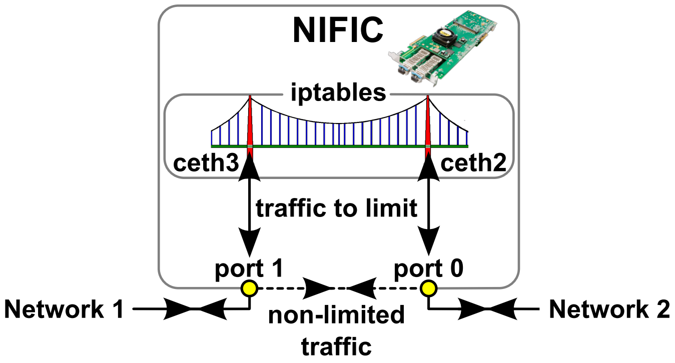

Figure 12: The traffic limiter - connection schema.

The Figure 12 shows deployment of traffic limiter. We setup NIFIC to behave like network bridge (IP addresses are not assigned to ceth2 and ceth3) and put it inline (e.g. between the edge router and the internal switch). NIFIC rule sends ICMP packets to *iptables* to shape traffic. Virtually we can limit any kind of traffic by modifying NIFIC and *iptables* rules.

5 SYSTEM CUSTOMIZATION

The Virtex5 FPGA on the COMBO-LXT card is fully programmable. So, the current NIFIC probe firmware could be simply modified in a several ways.

The current NIFIC firmware implementation supports up to 2000 various rules and about 1500 unique IP addresses. For example, if there occurs a need the NIFIC probe firmware could be modified to support more unique IPv4 addresses, more rules or to support currently unsupported fields of header packets as for example VLAN tags. There are only two limitation in a such modifications. First one is the working frequency of the whole NIFIC probe firmware, which could not decrease to ensure full throughput of the firmware. The second one is the limited number of resources on the Virtex5 FPGA.

We are currently working on some improvements, which will increase the maximum number of rules and also the number of unique IP addresses including the IPv6 addresses.

Moreover, not only the classification part of the NIFIC probe firmware could be modified. The throughput to software of the NIFIC probe could be increased, the outgoing packets could be modified. As an example of a such modification could be the request from our colleagues, when they began to use the NIFIC probe for their research. They needed to measure the amount of packets and bytes, which matched the concrete rule. On the basis of that request we extended the NIFIC probe firmware with the counters counting number of packets and bytes for each rule. Due to the fact, that NIFIC probe firmware is modular and could be modified easily the whole modification was implemented during a few days.

To summarize, any customer driven internal change to the firmware could be made even the classified ones in a very short time.

6 CONCLUSION AND FUTURE WORK

The Internet has highly dynamic nature. To successfully defend network against the various cyber attacks it is necessary to use combination of flow-based analysis, corelation of flows, payload inspection, line rate packet capture and many other techniques. The commodity PC boxes with software only applications can be used for non-critical, small and low speed networks, while the hardware acceleration is essential for the rest.

In this paper, we have described our work on system for security network monitoring with using of hardware acceleration. We use set of NIFIC probes based on commodity PC's with COMBOv2 acceleration card and NetCOPE development kit for the fast development of hardware accelerated applications. The probes provide data preprocessing (i.e. filtration, statistic counting, etc.) and traffic distribution among the processor cores directly in hardware.

We have described NIFIC probe, remote control of the probes together with system architecture, as well as several use cases of the applications and deployment scenarios for the network security monitoring.

The proposed system is tested on production networks of CESNET and Masaryk University. In future work we plan to add more monitoring applications to NIFIC probes. We will deploy more probes and test new scenarios. We aim at active defence of our networks (see use cases 4.4 and 4.5) to extend current "monitoring only" approach.

ACKNOWLEDGEMENT

This work is supported by the Research Intent of the Czech Ministry of Education MSM6383917201 and also supported by the Czech Ministry of Defence under Contract No. SMO02008PR980-OVMASUN200801.

REFERENCES

[1] Amiel A. Heyde. Investigating the performance of endace dag monitoring hardware and intel nics in the context of lawful interception, 2008. caia.swin.edu.au/reports/080222A/CAIA-TR-080222A.pdf.

[2] McQuaid Bradner. *Benchmarking Methodology for Network Interconnect Dev – RFC 2544*. IETF, Network Working Group, 1999. http://www.ietf.org/rfc/rfc2544.txt.

[3] Daniela Brauckhoff, Bernhard Tellenbach, Arno Wagner, Martin May, and Anukool Lakhina. Impact of packet sampling on anomaly detection metrics. In *IMC '06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, pages 159–164, New York, NY, USA, 2006. ACM.

[4] CESNET, z.s.p.o. *Description of COMBO Cards*. http://www.liberouter.org/hardware.php.

[5] CESNET, z.s.p.o. *FlowMon Probe Project Web Page*. http://www.liberouter.org/flowmon/index.php.

[6] CESNET, z.s.p.o. *FlowMon Probe Project Web Page*. http://www.liberouter.org/fflowmon.

[7] Sharon Christolm and Hector Trevino. *NETCONF Event Notifications – RFC 5277*. IETF, Network Working Group, 2008. http://www.ietf.org/rfc/rfc5277.txt.

[8] Cisco, 2010. http://www.cisco.ac/en/US/technologies/tk543/tk812/technologies_white_paper0900aecd802a0eb9.html.

[9] B. Claise. Cisco Systems NetFlow Services Export Version 9. RFC 3954 (Informational), October 2004.

[10] B. Claise. Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information. RFC 5101 (Proposed Standard), January 2008.

[11] DirkGeschke. Fast Logging Project for Snort, 2003. http://freshmeat.net/projects/flop/.

[12] Endace Measurement Systems, 2010. http://www.endace.com/.

[13] ENISA, 2010. http://www.enisa.europa.eu/act/cert/support/guide2/introduction/what-is-csirt.

[14] Rob Enns. *NETCONF Configuration Protocol – RFC 4741*. IETF, Network Working Group, 2006. http://www.ietf.org/rfc/rfc4741.txt.

[15] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616 (Draft Standard), June 1999. Updated by RFC 2817.

[16] Chunming Rong Geng Yang and Yunping Dai. A distributed honeypot system for grid security. In *Grid and Cooperative Computing*, pages 1083–1086. Springer Berlin / Heidelberg, 2004.

[17] Peter Haag. *NfSen - NetFlow Sensor*, 2010. http://nfsen.sourceforge.net.

[18] Jason Bittel. *HTTP logging and information retrieval tool*, 2010. http://dumpsterventures.com/jason/httpry/.

[19] Maxim Krasnyansky and Maksim Yevmenkin. *Universal TUN/TAP Device Driver*, 2000. `http://vtun.sourceforge.net/tun/`.

[20] R. Kuhn, T. Walsh, and S. Fries. Security Considerations for Voice Over IP Systems. NIST Special Publication 800-58, January 2005.

[21] Tomas Martinek and Martin Kosek. Netcope: Platform for rapid development of network applications. In *Proc. of 2008 IEEE Design and Diagnostics of Electronic Circuits and Systems Workshop*, pages 219–224. IEEE Computer Society, 2008.

[22] Cristian Morariu. A Distributed Architecture for IP Traffic Analysis, 2010. `http://www.ibr.cs.tu-bs.de/projects/nmrg/meetings/2008/munich/nmrg-25-morariu.pdf`.

[23] Napatech, 2010. `http://www.napatech.com`.

[24] NetFPGA, 2010. `http://www.netfpga.org`.

[25] Jiri Novotny, Pavel Celeda, and Martin Zadnik. Hardware-Accelerated Framework for Security in High-Speed Networks. In *RTO-MP-IST-076*, 2008.

[26] Ntop, Luca Deri , 2010. `http://www.ntop.org/PF_RING.html`.

[27] Ntop, Luca Deri , 2010. `http://www.ntop.org/TNAPI.html`.

[28] SCRITP project. Scalable and Robust Decentralized IP Traffic Flow Collection and Analysis, 2010. `http://www.csg.uzh.ch/research/script`.

[29] Viktor Pus and Jan Korenek. Fast and scalable packet classification using perfect hash functions. In *FPGA '09: Proceedings of the 17th international ACM/SIGDA symposium on Field programmable gate arrays*, New York, NY, USA, 2009. ACM.

[30] Martin Rehak, Michal Pechoucek, Martin Grill, Jan Stiborek, Karel Bartos, and Pavel Celeda. Adaptive multiagent system for network traffic monitoring. *IEEE Intelligent Systems*, 24(3):16–25, 2009.

[31] Stanford Linear Accelerator Center, 2010. `http://www.slac.stanford.edu/xorg/nmtf/nmtf-tools.html`.

[32] Sven Ubik, Petr Žejdl. Passive monitoring of 10 gb/s lines with pc hardware, 2008. `tnc2008.terena.org/core/getfile63f0.pdf?file_id=371`.

[33] Tcpdump Team. *libpcap - library for packet capture*, 2010. `http://www.tcpdump.org`.

[34] Margaret Wasserman and Ted Goddard. *Using the NETCONF Configuration Protocol over Secure SHell (SSH) – RFC 4742*. IETF, Network Working Group, 2006. `http://www.ietf.org/rfc/rfc4742.txt`.

[35] Wireshark Team. *Wireshark - packet analyzer*, 2010. `http://www.wireshark.org`.

[36] Jian Zhang and Andrew W. Moore. Traffic Trace Artifacts due to Monitoring Via Port Mirroring. In *Proceedings of the Fifth IEEE/IFIP E2EMON*, pages 1–8, 2007.