I. J. Computer Network and Information Security, 2012, 9, 1-9
Published Online August 2012 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2012.09.01

Methodology for Benchmarking IPsec Gateways

Adam Tisovský, Ivan Baroňák
Department of Telecommunications, Slovak University of Technology, Bratislava, Slovakia
tisovsky@ut.fei.stuba.sk, baronak@ut.fei.stuba.sk

Abstract — The paper analyses forwarding performance of IPsec gateway over the rage of offered loads. It focuses on the forwarding rate and packet loss particularly at the gateway's performance peak and at the state of gateway's overload. It explains possible performance degradation when the gateway is overloaded by excessive offered load. The paper further evaluates different approaches for obtaining forwarding performance parameters – a widely used *throughput* described in RFC 1242, *maximum forwarding rate with zero packet loss* and us proposed *equilibrium throughput*. According to our observations *equilibrium throughput* might be the most universal parameter for benchmarking security gateways as the others may be dependent on the duration of test trials. Employing *equilibrium throughput* would also greatly shorten the time required for benchmarking. Lastly, the paper presents methodology and a hybrid step/binary search algorithm for obtaining value of *equilibrium throughput*.

Index Terms — IPsec, benchmarking, throughput, offered load, forwarding rate, CPU utilization

I. Introduction

The significance of communication through the IP networks raises in all spheres of human activity. With increasing amount of transferred data and diversity of services grows also the amount of potential attacks and threats. The security standard is continually heightening and it is a challenge to fulfil the one when the amount of secured traffic is increasing at the same time. Process of securing the network traffic represented by encryption and/or authentication of packet flow is generally considered as computationally intensive. One of widely deployed security mechanisms for this purpose is IPsec – a suite of protocols, standards and rules, which can deliver these security services and also set up virtual private networks (VPNs) [1].

Evaluating the performance of IPsec gateway is important for vendors, service providers and customers. It is vital in device manufacturing and marketing, designing secure networks, VPNs, dimensioning traffic loads or in auditing the existing networks and devices. A demand for evaluating IPsec gateway performance is emphasized by the fact that forwarding performance of IPsec gateway expressed either in bits per second or in packets per second is not constant and not linear over the range of packet sizes [2]. The situation is illustrated in Figure 1. Moreover, as will be presented in the paper, forwarding performance of the gateway may decrease when it is overloaded by excessive offered load.

It is rather complicated to analytically calculate an actual performance of an IPsec gateway for particular nature of network traffic because the performance is determined by various parameters. These include performance of hardware components (e.g. processing units, memory, bus), software components (e.g. operating system kernel, cryptographic framework, algorithm library) and operational features (e.g. a/synchronous mode of operation, interrupt coalescing, etc.).

Measuring then becomes a reliable and convenient way for evaluating forwarding performance – on the devices either in-situ or in laboratory. To perform effective and accurate measurements we need to know all specific phenomena related to IPsec performance on a security gateway as well as to be equipped by a wise measuring tool. The tests should be performed in a confident, convenient and usually the fastest possible way.

In this paper we discuss methodology for measuring forwarding performance of IPsec gateway. We focus on how the forwarding rate and CPU utilization of IPsec process is dependent on the amount of offered load. Special attention is put on the peak of forwarding rate and also on the state of gateway overloading. We identify the differences between gateways equipped and not equipped by a separate hardware crypto accelerator.

Structure of the paper is following: First, the background of existing measurement methodology for IPsec gateways is presented, potential fields for enhancements are identified and the terminology that will be used in paper is defined. Proposed is also the parameter *equilibrium throughput* which should be more universal than existing benchmarking parameters. Next follows the analysis of forwarding rate and CPU utilization of IPsec gateway with and without hardware crypto accelerator over the range of offered loads. Explained is the reason of decreasing forwarding rate on overloaded gateway when no hardware crypto accelerator is present. Lastly, the methodology and algorithm for obtaining *equilibrium throughput* is proposed.

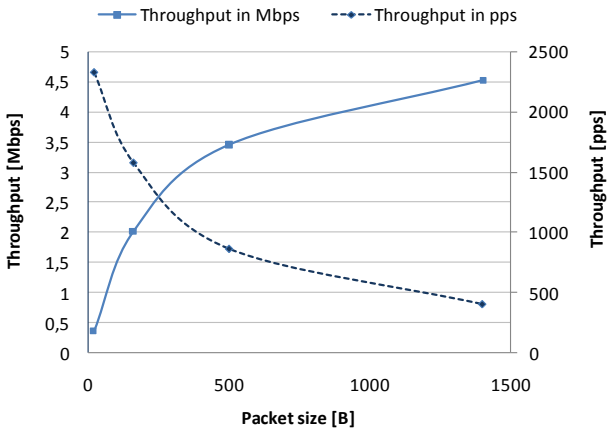

Figure 1: Throughput over the range of packet sizes; Cisco 1841, ESP-3des (software encryption)

II. Background of the Measurement Methodology and Motivation for the Work

Motivation for our work was the fact that despite number of performance studies of IPsec gateways there is to our knowledge not much literature dealing with the gateway performance dependency on the offered load, particularly with the scope on the performance peak and the gateway overloading, and also with the measurement methodology itself.

We also found out that the parameter *throughput* defined in RFC 1242 [3] with measurement methodology in RFC 2544 [4] may be in case of benchmarking security gateways dependant on duration of the test. In our opinion this might be a considerable drawback of its design. Since IETF drafts "Terminology for Benchmarking IPsec Devices" [5] and "Methodology for Benchmarking IPsec Devices" [6], which are directly focused on methodology of IPsec device testing, are operating with this very parameter, we decided to confront it with us proposed parameter *equilibrium throughput*.

Measurement methodology recommended by aforementioned IETF documents is to use an iterative search algorithm with adjusting offered load and employing UDP protocol. Importance of such search algorithm is supported by the fact that forwarding performance of a gateway that is not equipped by a separate cryptographic accelerator may decrease when the gateway is overloaded, as will be presented in the paper. To our knowledge none of the freely available measuring tools support such search algorithm, including up-to-date versions of widely used Iperf – version 2.0.5 [7], Netperf – version 2.5.0 [8], Hpcbench [9], D-ITG – version 2.8.0 [10], UDPmon – version 1.2.6 [11] and a list of others [12]. Therefore we present a proposal of such algorithm in a pseudo-code which may be implemented as a Bash script [13] and used altogether with Iperf.

III. Terminology

In the paper we use terminology for device performance testing introduced mainly in RFC 1242, RFC 2544, RFC 2285 [14] and aforementioned IETF drafts focused on IPsec benchmarking. The essential terms are:

- ***Offered load*** – the rate at which device under test receives the frames at a specified interface [14].
- ***Forwarding rate*** – the rate at which a device can be observed to successfully transmit to the correct destination interface in response to a specified offered load. It makes no explicit reference to frame loss [14].
- ***Throughput (RFC 1242 throughput)*** – the maximum rate at which none of the offered frames are dropped by the device [3, 5]. Important note is that this parameter is related to the offered load, i.e. it is interpreted as the maximum rate of frames sent to the device that leads to zero packet loss. For better comprehensibility of the terms in the paper we will refer it to as *RFC 1242 throughput*.
- ***Maximum forwarding rate (MFR)*** – the highest forwarding rate of a device taken from an iterative set of forwarding rate measurements, regardless of packet loss. *MFR* is often recorded when the device is little overloaded, but it should not be exploited to suggest that the device sustainably supports such rates of transmission [14]. It provides only supplementary information about performance of the device for a special situation.
- ***Maximum forwarding rate with zero loss (MFRZL)*** – parameter with concept [15] similar to *RFC 1242 throughput*. The difference is that it is related to the forwarding rate, i.e. to the maximal rate of traffic processed by the device with zero packet loss. It is sometimes confused with *RFC 1242 throughput* as both parameters are based on the packet loss evaluation. As will be shown in this paper, these parameters may diverse a lot, however.
- ***Equilibrium throughput*** – we propose this parameter as the highest forwarding rate of a device that is the same as offered load. Equivalent would be also inversed definition, i.e. the highest offered load that is the same as forwarding rate of device.

According to our observations *equilibrium throughput* is more suitable to universally describe the performance of security gateways than *MFRZL* or *RFC 1242 throughput*. The reason is that both last mentioned parameters based on evaluation of the packet loss may be dependent on duration of the test. This situation illustrates Figure 2.

When offered load exceeds the rate of *equilibrium throughput*, the device becomes overloaded. Offered load is now higher than forwarding rate and the system memory storing the packets (in this situation we can refer it to as an IPsec buffer) begins to fill. Despite this disparity, in a test of finite duration the buffer may cause that no packet is lost. During mild overload forwarding rate may be a little higher than during equilibrium state – higher density of arriving packets utilizes remaining CPU resources and leads to a higher probability for interrupt

coalescing [16, 17, 18]. This means that *MFRZL* and *RFC 1242 throughput*, which are based on the packet loss evaluation, will be measured higher than *equilibrium throughput*. The shorter the test or the larger the IPsec buffer, the higher would be their value. And vice versa, the longer the test or the shorter the IPsec buffer, the closer are both parameters to *equilibrium throughput*. Only in a test of infinite duration or when the IPsec buffer is of zero length all parameters would be equal. Mentioned dependency on duration of the test may be a considerable drawback of *MFRZL* and *RFC 1242 throughput* designs and therefore we propose *equilibrium throughput* as the most universal parameter in benchmarking and comparing IPsec devices.

On the other hand, *MFRZL* and *RFC 1242 throughput* would be useful to evaluate the capability of device to store excessive packets during a test trial of particular length. When presenting information based on these parameters duration of the test should be cited then.

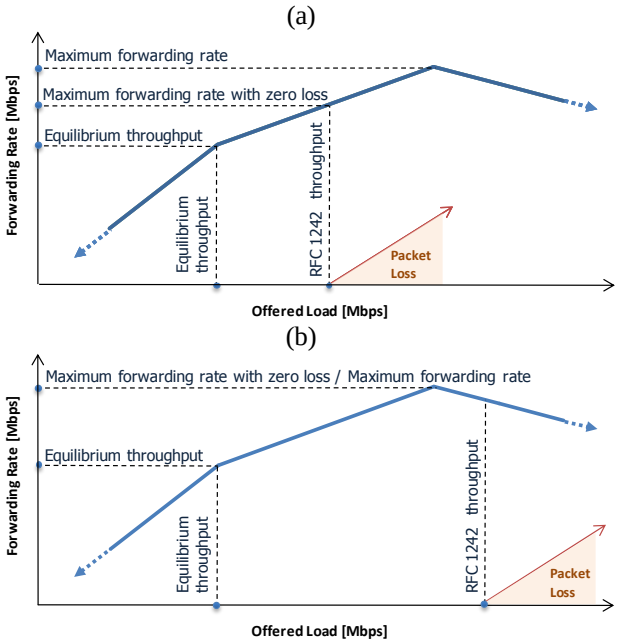

Figure 2: Explanation of forwarding rate behaviour at its peak: (a) *MFRZL* lies between *equilibrium throughput* and *MFR*. If the test was shorter or the buffer was larger, the packet loss would occur at the higher offered loads (b) The lower chart illustrates an example when in a finite-duration test the packet loss occurs only with offered load higher than the one associated to *MFR*. *RFC 1242 throughput* is then much higher than all other parameters, and *MFRZL* equals to *MFR*.

IV. Experiment Environment

We evaluated IPsec performance on Cisco 1841 Integrated Services Routers with integrated hardware crypto accelerator engine. Experiment environment was set as depicted in Figure 3. Two Cisco 1841 routers acted as the security gateways. All device connections were made using 100 Mbps Fastethernet links to ensure that forwarding bottleneck is IPsec processing on the router, not the link interface.

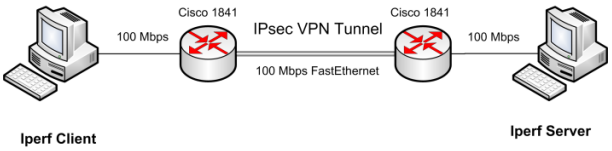

Figure 3: Scheme of experiment environment

As a traffic generator and measurement tool we used Iperf version 2.0.5 which was installed on two endpoints with operating system Linux Ubuntu version 10.04 – one acting as a client, i.e. the traffic transmitter, and the second as a server, i.e. the traffic receiver. We implemented a Bash script to trigger Iperf automatically with desired parameters.

As a transport protocol was used UDP because TCP employs flow and error control unneeded for our experiment. Important note – input and output values in Iperf are related to the payload on Layer 4 of OSI model excluding L4 header, thus all the rates and sizes in this paper are related to UDP datagram payloads, too. Default settings in the testing scenario were these:

- 500 bytes UDP payload datagram (528 bytes IP packet)
- constant packet inter-departure time
- IPsec in tunnel mode
- ESP protocol for encryption
- 3DES encryption algorithm
- 30 seconds duration of the tests
- 3 rounds of each test to calculate the average value

V. Forwarding Rate

A. Hardware crypto accelerator enabled

First we take a look at relation of forwarding rate of IPsec process and offered load when the hardware crypto accelerator is enabled. These values are plotted in Figure 4. The values of most interest (i.e. the peak of forwarding rate) were obtained by a step measurement with a step of 0.05 Mbps in interval from 18 Mbps to 19 Mbps of the offered load.

When continuously increasing offered load, forwarding rate is the same as offered load until it reaches *equilibrium throughput* at 18.15 Mbps. Until this point no packet is lost. Further increasing of offered load causes small increase of forwarding rate, but these values are not equal further. However, due to buffer the packets are not being discarded in a 30 second test until offered load is 18.40 Mbps with forwarding rate 18.36 Mbps. These values represent *RFC 1242 throughput* and *MFRZL*. At offered load 18.95 Mbps we reach the *MFR*, which is 18.83 Mbps. When the rate of offered load overruns *MFR* the forwarding rate decreases very slightly (minus 0.21%). With a further increase of offered load, however, it persists at the same level.

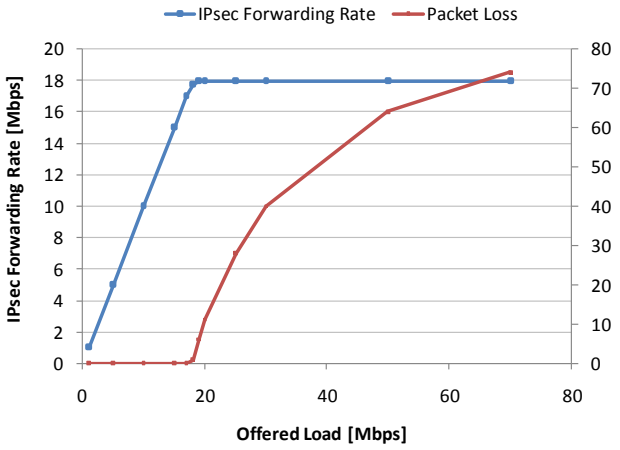

Figure 4: ESP-3des performance for 500 bytes UDP datagram; hardware encryption

Values are presented for the test duration of 30 seconds. Now we take a look at forwarding performance for different test durations. As results from Figure 5 and Table I, there is not a significant variance in *equilibrium throughput* as the duration of the test changes. In contrast, *RFC 1242 throughput*, *MFRZL* and also *MFR* show differences. In a case of *RFC 1242 throughput* the difference is 4.9% between maximal and minimal value, in a case of *MFRZL* the difference is 4.3% and in a case of *MFR* 1%. It is apparent that the shorter the test, the higher may be offered load that results into zero packet loss. Therefore *RFC 1242 throughput* may be reported relatively high. In the next section we introduce situation when the dependency on the test duration emerges even more.

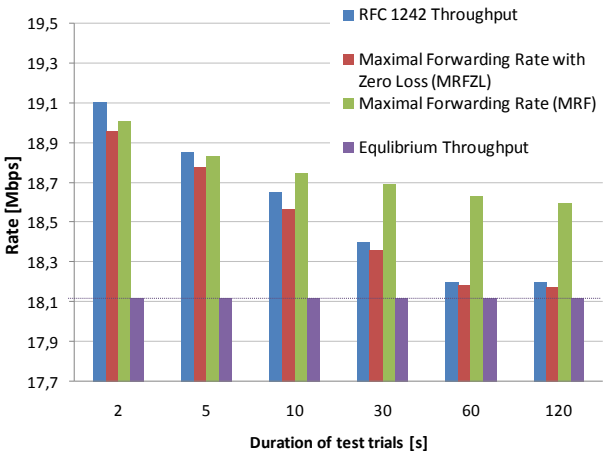

Figure 5: Variance of performance parameters depending on duration of the test trials; hardware encryption

TABLE I. VARIANCE OF PERFORMANCE PARAMETERS DEPENDING ON DURATION OF TEST TRIALS; HARDWARE ENCRYPTION

Duration of test [seconds]	Equilibrium throughput [Mbps]	RFC 1242 throughput [Mbps]	MFRZL [Mbps]	MFR [Mbps]
2	18.10	19.10	18.96	19.01
5	18.15	18.85	18.78	18.86
10	18.15	18.65	18.57	18.83
30	18.15	18.40	18.36	18.80
60	18.15	18.20	18.18	18.78
120	18.15	18.20	18.17	18.77

B. Hardware crypto accelerator disabled

When hardware crypto accelerator is disabled, behavior of forwarding rate is different. The dependency of forwarding rate and packet loss on offered load is plotted in Figure 6. The values of most interest are obtained by the step measurement with the step of 0.005 Mbps in interval from 3.4 Mbps to 4.4 Mbps of the offered load.

First difference is that after reaching maximal forwarding rate (at 3.481 Mbps in this case), further increase of offered load causes forwarding rate decrease. This behavior might be very tricky – the performance of security gateway can decrease significantly if the amount of IPsec traffic is overdesigned. Moreover, wrong information about the device performance could be obtained if an improper methodology of testing was implemented. Reason of described performance degradation will be discussed in the next section.

Second difference is that since the IPsec buffer is relatively high to the packet rate, device can sustain higher excessive offered load at the particular test durations with showing out a zero packet loss. During 30 second test the loss begins by offered load 3.590 Mbps, which is then *RFC 1242 throughput*, and it is 4% higher than *equilibrium throughput*.

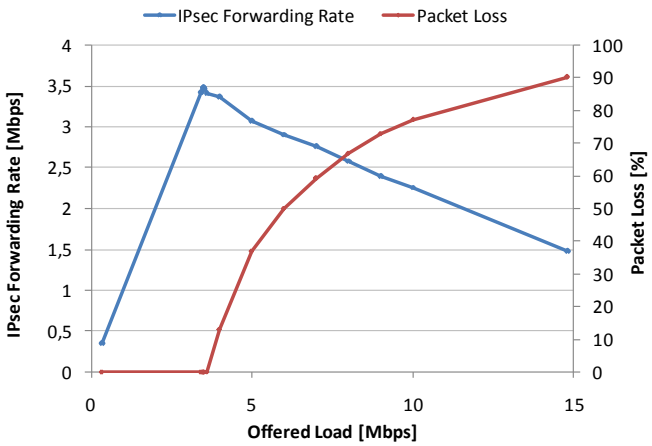

Figure 6: ESP-3des performance for 500 bytes UDP datagram; software encryption

In Figure 7 and Table II are presented forwarding performance parameters for different durations of the test. *Equilibrium throughput* shows omissible variance again. In contrast, *RFC 1242 throughput* shows almost 26% variability over the range of test durations, *MFRZL* 2.7% and *MFR* shows 2.5% variability.

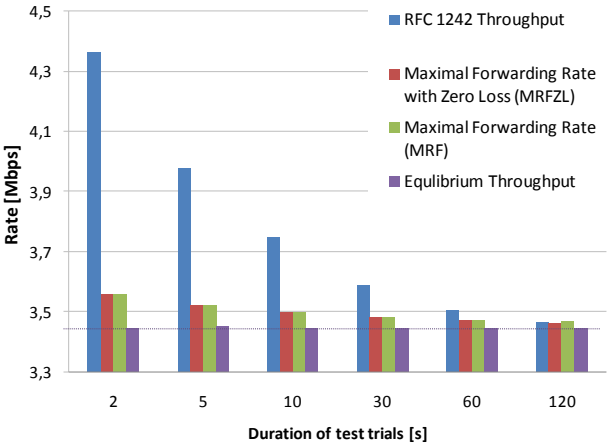

Figure 7: Variance of performance parameters depending on duration of the test trials; software encryption

TABLE II. VARIANCE OF DISCUSSED PERFORMANCE PARAMETERS DEPENDING ON DURATION OF TEST TRIALS; SOFTWARE ENCRYPTION

Duration of test [seconds]	Equilibrium throughput [Mbps]	RFC 1242 throughput [Mbps]	MFRZL [Mbps]	MFR [Mbps]
2	3.445	4.365	3.556*	3.556
5	3.450	3.980	3.523*	3.523
10	3.445	3.750	3.498*	3.498
30	3.445	3.590	3.481*	3.481
60	3.445	3.505	3.472*	3.472
120	3.445	3.465	3.461	3.468

* MFRZL equals to MFR. This situation was explained in Fig. 1b

Duration of the tests recommended by benchmarking methodology literature varies from 30 seconds [19] to 60 seconds [4, 6] and more. As can be seen from Table II and Figure 7, *RFC 1242 throughput* is quite fluxional over this interval. In the need for stable value we have to run the test at least for 120 seconds. This is inconvenient especially when we employ UDP throughput search algorithm which requires running of a number of test iterations. The parameter *equilibrium throughput* then emerges as a suitable alternative. It can evaluate performance of gateway universally with requirement of considerably shorter test trials.

VI. CPU UTILIZATION

To reveal the reason of mentioned IPsec forwarding rate behaviour we take a look at CPU utilization. Cisco IOS provides this information using command "*show processes cpu*", which covers utilization of last 5 seconds, 1 minute and 5 minutes. The utilization is split between operations on an interrupt level and on a process level. First we analyze CPU utilization of the gateway with hardware crypto accelerator enabled.

A. Hardware crypto accelerator enabled

Examining CPU utilization we find out that with enabled hardware crypto accelerator the IPsec forwarding performance is determined by the amount of operations executed on the interrupt level. The operations include mainly IPsec protocol stack execution, i.e. AH (Authentication Header) and ESP (Encapsulation Security Payload) headers processing, SPD (Security Policy Database) and SAD (Security Association Database) lookups, execution of cryptographic framework, which prepares transformation configuration for cryptographic accelerator, in SoftIRQ mode of a lower priority, and drivers execution in ISR (Interrupt Service Routine) mode of the highest priority [2]. On the process level, which has the lowest priority, there is no CPU utilization by the cryptographic operations, as they are done by the hardware crypto accelerator. Whole CPU utilization, let be maximum 99%, then consists of 96% utilization on the interrupt level and 3% on the process level. This minor utilization on the process level is the sum of common background not-intensive processes like *Pool Manager*, *Cisco Discovery Protocol*, *Per-minute Jobs*, *DHCP protocol*, *Load Meter*, *routing protocols*, etc.

In Figure 8 are presented values of CPU utilization on the range of different offered loads. For comparison, CPU utilization on the interrupt level for plaintext (non-IPsec) traffic is 12% at offered load 20 Mbps. As can be seen from the figure, trend of CPU utilization is not linear. The linearity appears only for offered load lower than about 10.3 Mbps, what is 58% of maximal measured forwarding rate – at this point the CPU utilization is 88%. With further increase of offered load the utilization rises slowly and finally reaches 99% at maximal measured forwarding rate 18.8 Mbps. Interesting to note is that this CPU utilization behaviour does not affect forwarding rate at all – it is strictly linear over the whole range of offered loads (until it reaches maximal forwarding rate).

This behaviour is not abnormal, however. Explanation is in interrupt coalescing which device applies when it is overloaded by incoming packets. In normal operation each incoming packet raises an interrupt which forces CPU to perform context switching – store all important data from the current process, then execute interrupt service routine of a higher priority, and finally restore the interrupted process. This concept lowers latency for the packets, but consumes additional CPU resources. If interrupts reach certain rate, operating system or network adapter driver in order to avoid exhausting CPU fully by the interrupts switches to an interrupt coalescence mode. Multiple packets are then served by a single interrupt as there is a high probability that more packets arrive in a short time interval. This saves CPU resources because less context switching is performed [17, 18].

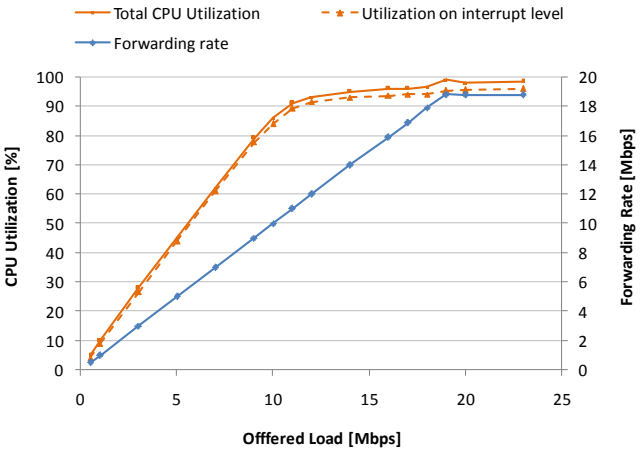

Figure 8: CPU utilization for ESP-3des 500 bytes UDP datagram; hardware encryption

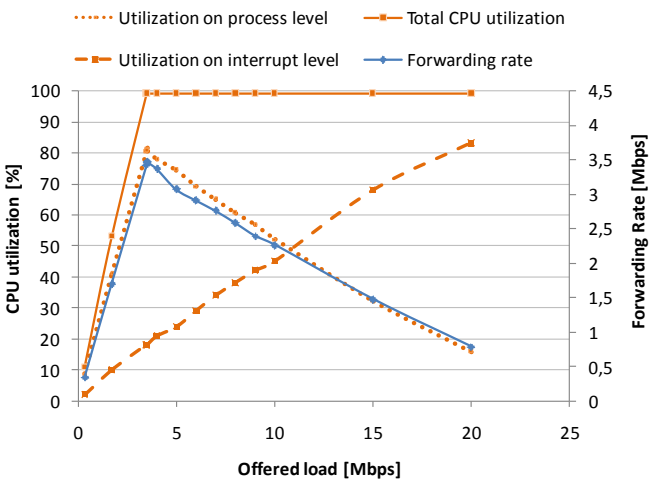

Table 9: CPU utilization for 500 bytes UDP datagram, ESP-3des; software encryption

For further understanding it is important to remind that maximal forwarding rate is achieved at almost full utilization of CPU on interrupt level. As a result, forwarding rate does not decrease when the device is overloaded (as was shown in Figure 4), because more packets cannot exhaust CPU by more interrupts. Instead, excessive packets are dropped from the overflowed receive buffer on the network adapter. This is in contrast with the case when crypto accelerator is disabled, as was shown in Figure 6 and will be also explained below.

B. *Hardware crypto accelerator disabled*

Total CPU utilization of IPsec processing without hardware crypto accelerator consists of two main components –interrupts handlers (ISR and SoftIRQ programs, explained in above section) and encryption process. The values on the range of different offered loads are shown in Figure 9.

Until *MFR* is reached, both main components of total CPU utilization (*bright orange*) – CPU interrupts (*orange*) and encryption process (*deep orange*) – are directly proportional to the offered load. Interesting to note is that interrupt utilization keeps linear trend over the whole inspected range – unlike the case with hardware crypto accelerator enabled. Maximum forwarding rate of IPsec processing is achieved at total CPU utilization of 99%, where 81.5% is consumed by the encryption process, 18% by interrupt handling and the rest by minor processes. If we continue increasing offered load beyond this point, more CPU resources will be consumed by interrupt handling and less will be available for the encryption process itself. This leads into decrease of forwarding rate, as shown in Figure 6 and Figure 9. This emphasizes the need for evaluating the performance of a device for various types of traffic and packet lengths, as its overloading would lead to performance degradation.

VII. Search Algorithm

We propose a hybrid step/binary search algorithm to measure *equilibrium throughput*. The pseudo-code is shown in Figure 10. This method is applicable for both types of IPsec gateway – with or without hardware crypto accelerator. Testing starts at chosen offered load (*Oload*). If measured forwarding rate (*FWrate*) equals to *Oload*, in next iteration is *Oload* increased by the step (*step)*. This increase repeats until the first difference between *Oload* and *FWrate* is recorded. On the contrary, if initial *Oload* was set too high, algorithm would decrease it continuously in iterations without step change until *Oload* equals *FWrate.* This is the "step search" phase of the algorithm. Now the algorithm knows that the peak of forwarding rate lies between the last two iterations. Therefore in every following iteration is the step halved and the algorithm decides whether it is added to or subtracted from an *Oload* depending on the equality or inequality of *Oload* and *FWrate*. This is the "binary search" phase of the algorithm. Testing ends when step is lower than certain ratio of *Oload*, say 0.2% (tolerance for approximation), or the iteration count has reached the limit, say 16 iterations.

The algorithm can be implemented in any programming language. We implemented it as a Bash script that employs Iperf application. Function *get_forwarding_rate()* in Bash 4.0 [13] would then comprise of script shown in Figure 11. The first command triggers Iperf with specified parameters and the second one extracts value of forwarding rate from the Iperf text output using consecutive *Sed* functions [21].

VIII. Enhancement of Search Algorithm

As was mentioned in the introduction, throughput of IPsec process expressed in bits or packets per second is dependent on the packet size. This means that we have to perform several measurements for different packet sizes to create a throughput profile for the device.

Methodology for benchmarking network devices in document RFC 2544 recommends measuring throughput for the packet sizes of 64, 128, 256, 512, 1024, 1280, 1420 bytes.

There is a discussion how to effectively set initial values of *Oload* and *step* when benchmarking a device for various packet sizes with no awareness of its performance. Normally, the values should be set to cover whole interval given by the link bit-rate, i.e. *Oload* should be set to 50 % and *step* to 25 % of the link bit-rate. In [20] is presented general mathematical model of IPsec throughput and a method for calculation the throughput for different packet sizes from known throughput for two packet sizes. The method is based on a calculation of two characteristic parameters of IPsec process – R_{alg} and t_{fix} :

$$R_{alg} = \frac{(L_{alg2} - L_{alg1})R_{M1}R_{M2}}{L_{M2}R_{M1} - L_{M1}R_{M2}} \qquad (1)$$

$$t_{fix} = \frac{L_{M1}}{R_{M1}} - \frac{L_{alg1}}{R_{alg}} \qquad (2)$$

where R_{M1} is measured IPsec throughput for packet size L_{M1} in which L_{alg1} bytes are secured, and analogously values R_{M2} , L_{M2} and L_{alg2} for different packet size. Then it is possible to calculate estimated throughput R_{calc} for any packet size L_{calc} in which L_{alg} bytes are secured:

$$R_{calc} = \frac{L_{calc}}{t_{fix} + \frac{L_{alg}}{R_{alg}}} \qquad (3)$$

We will assume that this method estimates throughput with certain inaccuracy, say 2 % at an average. To obtain precise values of IPsec throughput (i.e. with inaccuracy below 0.2%), aforementioned calculation can be integrated into the search algorithm. Such enhanced search algorithm will decrease required iteration counts for the tests.

Employment of the algorithm is following: First two measurements are performed with initial values set to cover whole link bandwidth. In every following measurement a calculated value R_{calc} will be used as an initial offered load with step of 2 % of this value. Advantage of hybrid step/binary concept is that even if the actual value of throughput does not lie within the initially set interval (i.e. the inaccuracy of calculated value is greater than was expected), algorithm will locate the peak in its step phase. Binary algorithm would not converge to the correct value at all.

Table III shows iteration counts for particular values of throughput when using normal setting of initial values and when employing throughput estimation method. For instance, IPsec throughput is 10 Mbps and normal setting of initial step is 25 Mbps – it takes 13 iterations to lower the step below 0.02 Mbps, what is considered as an approximation of the result. With estimation method initial step is 0.2 Mbps and it takes 6 iterations to approximate the result.

TABLE III. COMPARISON OF ITERATION COUNT FOR SEARCH ALGORITHMS

Actual throughput 2%) [Mbps]	Tolerance for approximation (0.2% of actual throughput) [Mbps]	Initial offered load [Mbps]	Initial step [Mbps]	Iteration count
Non-enhanced search algorithm				
1	0,002	50	25	16
10	0,02	50	25	13
30	0,06	50	25	11
90	0,18	50	25	10
Enhanced search algorithm				
1	0,002	1	0,02	6
10	0,02	10	0,2	6
30	0,06	30	0,6	6
90	0,18	90	1,8	6


```
begin
  set Oload;
  set step;
  set tolerance;
  for i := 1 to 16 step 1 do
    FWrate := get_forwarding_rate(Oload)
    if FWrate = Oload then
      if trend = decreasing then peak := 1 fi;
      if peak = 1 then step := step/2 fi;
      Oload := Oload + step;
      trend := increasing;
    else
      if trend = increasing then peak := 1 fi;
      if peak = 1 then step := step/2 fi;
      Oload := Oload - step;
      trend := decreasing;
    fi;
    if step < Oload*tolerance then
      print(FWrate);
      break;
    fi;
  od;
  print(FWrate);
end
```

Figure 10: Pseudo-code of hybrid step/binary search algorithm

```
iperf -c server_IP -u -b "$Oload"K -l 500B -t 30 -p 5001 -f
    k -T 1 -C > /home/user/output.txt

FWrate=$(sed -n '/KBytes/p' /home/user/output.txt | sed -n
    2p | sed 's/KBytes/&\n/;s/.*\n//;s/Kbits/\n&/;s/\n.*//')
```

Figure 11: Example of function *get_forwarding_rate()* in Bash scripting with employing Iperf

IX. Conclusion and Future Work

In the paper we presented IPsec gateway performance dependency on the offered load and proposed considerations for effective performance testing. We inspected the IPsec gateway when it is congested by excessive offered load. In case of absenting hardware crypto accelerator, i.e. when resources of CPU are shared between encryption process and protocol processing, there might be considerable forwarding performance degradation when the device is overloaded. This emphasises need for knowing performance limits of the device.

According to our observations *equilibrium throughput* is more suitable parameter to universally describe performance of security gateway. Benchmarking parameters based on packet loss evaluation, i.e. *RFC 1242 throughput* and *MFRZL*, may be dependent on duration of the test. Employing *equilibrium throughput* as a benchmarking parameter also greatly shortens the time required for device testing. On the other hand, *MFRZL* and *RFC 1242 throughput* might be useful to evaluate capability of device for buffering excessive offered load during particular time interval. When presenting information based on these parameters, duration of the test should be then cited.

We proposed a hybrid step/binary search algorithm to measure *equilibrium throughput*. A hybrid algorithm takes the best from both separate algorithms – a step algorithm in first phase roughly and quickly locates the peak, binary algorithm then converges to the peak very closely. Therefore it is not essential to set initial values of search algorithm wide enough to cover the peak. Too wide interval would lead to higher iteration count and to longer duration of the test.

The paper also outlined several interesting and potential fields of further research. It will be interesting inspect IPsec gateway performance for different nature of traffic, e.g. more parallel flows containing one or more packet lengths, or flows with not-constant packet inter-departure time distributions.

Acknowledgement

This work is a part of research activities conducted at Slovak University of Technology Bratislava, Faculty of Electrical Engineering and Information Technology, Institute of Telecommunications, within the scope of the project „Support of Centre of Excellence for SMART Technologies, Systems and Services II., ITMS 26240120029, co-funded by the ERDF“.

References

[1] S. Kent, K. Seo, “RFC 4301 - Security Architecture for the Internet Protocol”, IETF RFC, 2005

[2] G. Waters, K. Stammberger, “Understanding Crypto Performance in Embedded Systems,” Mocana design article, 2009. Available at: http://www.embeddeddeveloper.com/news_letter/files/CRYPTOWP_Rev2.pdf

[3] S. Bradner, “RFC 1242 - Benchmarking Terminology for Network Interconnection Devices”, IETF RFC, 1991

[4] S. Bradner, “RFC 2544 - Benchmarking Methodology for Network Interconnect Devices”, IETF RFC, 1999

[5] M. Kaeo, T. Van Herck, M. Bustos, “Terminology for Benchmarking IPsec Devices: draft-ietf-bmwg-ipsec-term-12”, IETF Draft, 2009

[6] M. Kaeo, “Methodology for Benchmarking IPsec Devices”, IETF Draft, 2009

[7] NLANR/DAST, “Iperf”, open-source project. Available at: http://sourceforge.net/projects/iperf/

[8] R. Jones, “Netperf”. Available at: http://www.netperf.org/netperf/

[9] B. Huang, M. Bauer, M. Katchabaw, "Hpcbench - a Linux-based network benchmark for high performance networks," High Performance Computing Systems and Applications, 2005. HPCS 2005. 19th International Symposium, pp. 65- 71, ISSN: 1550-5243, 15-18 May 2005

[10] A. Botta, A. Dainotti, A. Pescapè, "Multi-protocol and multi-platform traffic generation and measurement", INFOCOM 2007 DEMO Session, May 2007, Anchorage (Alaska, USA), D-ITG tool. Available at: http://www.grid.unina.it/software/ITG/
[11] R.E. Hughes-Jones, "Writeup for udpmon A Network Diagnostic Program", Oct. 2010. Available at: http://www.hep.man.ac.uk/u/rich/Tools_Software/udpmon/udpmon_v2.pdf
[12] C. Dovrolis, E. Goldoni, M. Schivi, "End-to-End Available Bandwidth Estimation Tools, an Experimental Comparison", PAM 2010 - Passive and Active Measurement Conference, 2010. Available at: http://pam2010.ethz.ch/TMA/papers/TMA2010_p13.pdf
[13] GNU Bash. Available at: http://www.gnu.org/software/bash/
[14] R. Mandeville, "RFC 2285 - Benchmarking Terminology for LAN Switching Devices", IETF RFC, 1998
[15] M. Castelino, F. Hady, Network Processing Forum, "Tutorial on NPF's IPsec Forwarding Benchmark," 2004. Available at: http://www.eetimes.com/design/communications-design/4009321/Tutorial-on-NPF-s-IPsec-Forwarding-Benchmark
[16] M. Zec, M. Mikuc, M. Žagar, "Estimating the Impact of Interrupt Coalescing Delays on Steady State TCP Throughput", Proceedings of the 10th SoftCOM 2002 conference, 2002
[17] M.G. Iatrou, A.G. Voyiatzis, D.N. Serpanos, "Network Stack Optimization for Improved IPsec Performance on Linux", In: SECRYPT 2009, Proceedings of the International Conference on Security and Cryptography, Milan, Italy, pages 83-91, INSTICC Press, 2009, ISBN 978-989-674-005-4, 7-10 July 2009
[18] K. Salah, "Integrated performance evaluating criterion for selecting between interrupt coalescing and normal interruption**",** International Journal of High Performance Computing and Networking, Volume 3 Issue 5/6, December 2005
[19] B. Hickman et al., "RFC 3511 - Benchmarking Methodology for Firewall Performance", IETF RFC, 2003
[20] A. Tisovsky, I. Baronak, "Analytical Model of IPsec Process Throughput," In: Advances in Electrical and Electronic Engineering (AEEE). ISSN 1336-1376. September 2012
[21] B. Barnett, "Sed - An Introduction and Tutorial". Available at: http://www.grymoire.com/Unix/Sed.html
[22] J. Dugan, "Iperf Tutorial", JointTechs 2010. Available at: http://www.es.net/assets/Uploads/201007-JTIperf.pdf

Adam Tisovský was born in Bratislava, Slovakia in February 1986. He received his master's degree from Slovak University of Technology, Bratislava in 2009. Since 2009 he is a postgradual student at Institute of Telecommunications, Slovak University of Technology, Bratislava. He focuses on network security, cryptographic algorithms, QoS, VoIP and modeling the network parameters with respect for real-time applications.

Ivan Baroňák was born in Žilina, Slovakia in July 1955. He received the electronic engineering degree from Slovak Technical University Bratislava in 1980. Since 1981 he has been a lecturer at Institute of Telecommunications, STU Bratislava. Nowadays he works as a professor at Institute of Telecommunications of FEI STU Bratislava. Scientifically, professionally and pedagogically, he focuses on problems of digital switching systems, ATM, Telecommunication management (TMN), NGN, VoIP, QoS, problem of optimal modeling of private telecommunication networks and services.

