Marek BOLANOWSKI, Andrzej PASZKIEWICZ, Michał WROŃSKI, Robert ŻEGLEŃ
RZESZÓW UNIVERSITY OF TECHNOLOGY, DEPARTMENT OF POWER ELECTRONICS, POWER ENGINEERING AND COMPLEX SYSTEMS,
Rzeszów, Poland

Representativeness analysis and possible applications of partial network data flows

Abstract

A new approach to statistical analysis of network flows and its possible application to statistical anomaly detection in high bandwidth communication networks are presented in the paper. The whole data stream was divided into smaller flows using Link Aggregation Control Protocol (LACP). A statistical analysis of the resulting flows shows that a single stream separated from the overall network traffic is representative when it comes to statistical anomaly detection. Such an approach allows the reduction of hardware resources needed to detect anomalies, and makes such a detection possible in high traffic communication systems.

Keywords: link aggregation control protocol, network security, network anomaly detection.

1. Introduction

Rising number of treats coming from computer networks forces to employ a multistage network traffic analysis to detect them. One of the places, where such an analysis is rarely performed is the contact point between Tier 2 networks and a local provider. The main features of such points of aggregations are, inter alia, high connection speeds and significant flow sizes. Decomposition of the data stream to Protocol Data Units (PDUs), specific to higher layers of the Open System Interconnection (OSI) model, is often required for classical methods of treat detection, thus limiting its usage. Such processing adds unacceptable delays to the transmission. Instead of using methods which require analysis of information on higher layers of the OSI model, methods of network traffic analysis using statistical signatures of treats have become widely used. Currently there are employed many methods of Dos/DDOS and BOT attack analysis based on: exceedance analysis, Hurst parameter, statistics, and entropy analysis [1-5]. The long range network time series analysis [6, 7] can yield probably best results in identifying network traffic anomalies, but it requires a lot of hardware resources, particularly memory and CPU time. Likewise, the intrusion detection method is based on statistical analysis [8]. Figure 1 shows a general operation diagram of such an analysis.

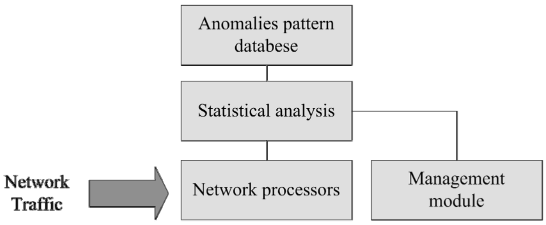

Fig. 1. General operation diagram of network traffic anomaly detection

The attack on the network causes many different anomalies, i.e.: increase in bandwidth saturation, TCP packets percentage change in the whole traffic, or statistical parameters change. Network Processors task is to directly catch information of interest from the port (i.e.: error rate, TCP/s, speed …) and forward it to Statistical Analysis module. The Network Processor has distributed architecture and is located in network devices which communicate with the rest of the modules using: NetFlow, OpenFlow, and PortMirroring protocols. Those modules are located in a dedicated management station. The Statistical Analysis module analyzes chosen parameters of the network traffic in real time and compares them with Anomalies Pattern Database. In the case of an attack, the Management Module undertakes specified actions (traffic blocking, further analysis, ...).

With increasing bandwidth, copying the data from link to an analysis tool starts to be a problem, along with the computational complexity of the algorithms used. There are three possible approaches: to ignore the problem and represent the whole traffic as time series and analyze them, to sample the data with a certain interval of Δt (and thus reduce resources demands, but deliver less accurate results), and to choose representative time series. The following part of the paper focuses on the latter method using LACP protocol.

2. Using the LACP protocol to divide the data

The LACP protocol, standardized by IEEE is a protocol which allows increasing the bandwidth of a communication channel by merging (aggregating) physical communication links. At first the protocol was standardized by 802.1ad, and currently by 802.1ax. Aggregation can be realized by using any of the three lowest layers of the OSI model.

The network traffic with a bandwidth of $t = n$ is directed to a single port of Network Device (ND) 1. Next, the traffic is transferred between ND1 and ND2 using m ports with the bandwidth $n_1, n_2, \dots, n_m$, where $\sum_{i=1}^{m} n_i > n$. The sub-flows have usually equal bandwidth when $\sum_{i=1}^{m} n_i \le n$, but when $\sum_{i=1}^{m} n_i > n$, the sub-flows $n_1, n_2, \dots, n_m$ can differ. Our test shows that quantitative differences between flows do not influence statistical parameters, especially when statistical parameters used to network anomaly detection are considered. Usually the Round Robin algorithm is used to distribute the traffic between ports, classifying the traffic on the basis of PDU from the lowest layers of the OSI model. Current LACP implementations add only small delays (microseconds) to the traffic handled by the devices (this was tested with a traffic generator JDSU TS-170 and standard LACP implementation, besides the logical links created with this protocol are widely used in high throughput networks).

A question which led to creation of this paper arises: Is statistical analysis of only one link aggregated by LACP representative enough to detect anomalies in the overall network traffic handled by the device?. The test environment shown in Figure 2 was created in a real production network of the network provider to verify this hypothesis.

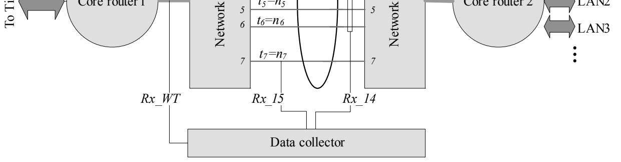

Fig. 2. Network topology used in the study

Two network switches, linked together by 7 links aggregated by the LACP protocol into a single connection, were inserted into the topology. Next, using the port mirroring protocol, all incoming network traffic (Tx), traffic going through link with number 7 (Rx_15) and aggregated traffic from links 1 to 6 (Rx_14) was copied. The data collected this way was analyzed to verify if there were any visible differences between statistical properties of Tx, Rx_15, Rx_14 data samples.

3. Data analysis

One of the first tests performed when one tries to analyze a time series is the test for stationarity. In brief, a stationary process is a process whose statistical properties (in particular moments of the distribution) are constant in time. The nature of stationarity and its applications are wider described in [9, 10]. Data stationarity in real-life cases is more of exception than a rule, but useful one – it is a necessary condition for many estimators and formal statistical analysis.

Over 80000 data samples, gathered from a real-life communication system were used in the study. The data was collected from devices working for ISP (Internet Service Provider) with over 600 clients – both individual and business. Variations in network connection load were studied for 29 days with a 30 second interval. Three data sets were prepared: one consisting of information about the overall incoming network traffic, two more about the traffic sent to two outgoing links with ratio of 86% / 14% load on each link (Rx_14 and Rx_15 respectively). Next, the stationarity analysis was made by computing and observing behavior of two first moments of probability distributions in each dataset. The arithmetic average was computed by definition and variance using the following equation: $s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$ where $\bar{x}$ is average for each day – because the data was divided into 24-hour long blocks. Such division was dictated by the shape of original data plots, showing seasonal trends with the period of one day.

Figure 3 shows the dynamics of average changes, while Figure 4 shows the dynamics of variance changes for all the three data sets. All statistics were plotted using the absolute scale, which depicts percentage changes from the mean values of the analyzed variables for each block. As shown in Figures 3 and 4, all the data share similar fluctuations. Except a few points, all are located below 40% fluctuations for the average and 70% for the variance. One can conclude that all the examined processes, though not stationary, share similar dynamics.

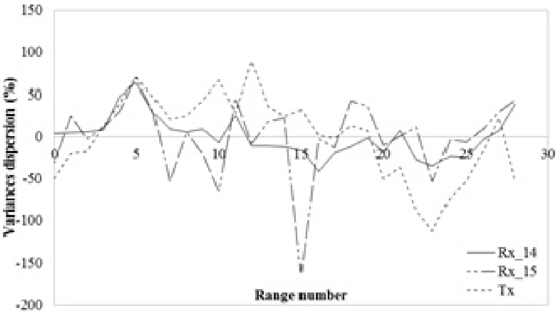

Fig. 3. Dynamics of variance in absolute scale

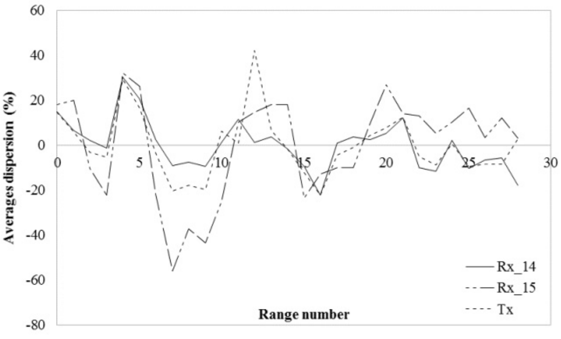

Fig. 4. Dynamics of the average in absolute scale

Another important test for statistical properties of a time series is autocorrelation analysis. Autocorrelation indicates how the dataset in question is correlated to itself with a certain lag l. Figure 5 shows the normalized autocorrelation analysis for lags from 1 to 160000 for each dataset: Tx, Rx_14 and Rx_15 respectively. As Figure 5 shows, the autocorrelation plots are indistinguishable.

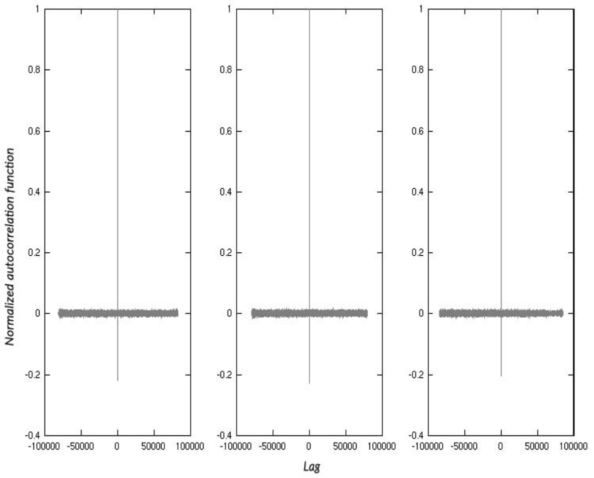

Fig. 5. Normalized autocorrelation function values for Tx, Rx_14 and Rx_15 datasets respectively. All plots for lags from –80000 to 80000

The Hurst exponent, which, as is well known, can reveal a lot of information about statistical properties of a process, was used to study processes dynamics further.

The Hurst exponent can be associated with how „jagged" the time series is – its low value suggests jagged plot shape, with a lot of quick changes, and higher values lead to smooth plots with less fluctuations.

Four methods were used to estimate the Hurst exponent for the data: residuals of regression, absolute values method, aggregated variance method and periodogram [11].

The first three methods share the same first step: division of the data into blocks with size m. Next, each algorithm differs. In the case of the residuals of regression, the partial sum for each block equals:

$$Y(i) = \sum_{k=1}^{\infty} \mathrm{x}_t, \tag{1}$$

for i = 1,2 ... N/m, where N stands for the original data set size. Next, a trend line is fitted to obtain statistic using the least squares method. Finally, variances of residuals are computed:

$$\mathrm{Var}(i) = \frac{1}{m}\sum_{i=1}^{m}(Y(i) - ia - b)^2, \tag{2}$$

where a and b are coefficients of the trend line. The obtained results are then averaged. This procedure, repeated for different block sizes, yields the log-log plot proportional to m^{2H}, where H denotes the Hurst exponent.

In the absolute values method, there was used the following equation for divided series:

$$X^{(m)}(k) = \sum_{i=(k-1)(m+1)}^{km} \mathrm{x}_i, \tag{3}$$

where k is a block index. Next the computation step involves generating statistic given by the following equation:

$$Y(m) = \frac{1}{N/m}\sum_{k=1}^{N/m}\left|X^{(m)}(k)\right|, \tag{4}$$

Such an operation, repeated for different block sizes, should create a line with a slope equal to 1 – H on a log-log plot.

The aggregated variance method differs from the above only by the last step, which uses the following equation instead of (4):

$$Y(m) = \frac{1}{N/_m}\left(\sum_{k=1}^{\frac{N}{m}}\left(X^{(m)}(k)\right)^2\right) - \left(\frac{1}{N/_m}\sum_{k=1}^{\frac{N}{m}}\left(X^{(m)}(k)\right)\right)^2 \qquad (5)$$

A periodogram is created differently. It is generated by the following equation (which is a discrete Fourier transform of the data):

$$I(\lambda) = \frac{1}{2\pi N}\left|\sum_{j=1}^{N} X_j e^{ij\lambda}\right|, \qquad (6)$$

where i denotes the imaginary unit and λ stands for subsequent frequencies in frequency domain.

Such statistic is called periodogram, and should be proportional to $|\lambda|^{1-2H}$ close to the origin, and from that, using the graphical method (fitting line with the least squares method), the Hurst exponent can be estimated. The estimation results are shown in Table 1.

Tab. 1. Results of Hurst exponent estimation

	H – Residuals of regression	H – Absolute values	H – Aggregated Variance	H - Periodogram
Rx_14	0.325994	0.3254	0.253629	0.151268
Rx_15	0.313059	0.30752	0.233878	0.207254
Tx	0.312013	0.322177	0.267055	0.151610

The results presented indicate the anti-persistent nature of the data, and the aggregated variance and periodogram show higher scale of anti-persistent behavior. In the case of reference data, on which the software used was tested, the differences between the methods were below ±0.02, which is clearly not met with real-life data. Such differences arise from estimational nature of the methods – they are based on moments of distributions of the analyzed data, which are not stationary, so the best estimate needs to be chosen. The periodogram stands out the most, despite that it is the most accurate method in the case of the reference data (also different reference data than used in testing the software). The residuals of regression and absolute values methods yield similar results, differing less than in the case of reference data, so the assumption can be made that those methods estimate the Hurst exponent best in the case studied.

There is another interesting thing in the results – lower estimated values for nearly all the methods except the periodogram for Rx_15. This can lead to a conclusion that this link has different dynamics than the rest of connections, but the differences are so small, especially knowing that the Hurst exponent estimates are not very accurate overall, that we can formulate another conclusion: though the reduced number of packets going through link Rx_15, statistical properties of the traffic observed on this link are not very different from those of the overall traffic on all interfaces of the device and from the second, more traffic-loaded link.

Entropy of the data was also computed, using the following equation:

$$H(X) = -\sum_i \frac{p_i}{N} log_2(\frac{p_i}{N}) \qquad (7)$$

where N is the number of data points and n_i is the frequency of sample number i in the data.

Skewness, another important statistical parameter, was computed from the following equation:

$$\gamma = E\left[\left(\frac{X-\mu}{\sigma}\right)^2\right] \qquad (8)$$

The results of the computations are shown in Table 2.

Tab. 2. Results of entropy and skewness computation

Data flow	Rx_14	Rx_15	Tx
Entropy	12.993	11.741	12.918
Skewness	–0.017699	0.084138	–0.017699

The results support the hypothesis that both of the partial flows are similar in statistical dynamics. The difference of entropies by one bit and skewness by 0.1 is insignificant, taking into account how much the flows differ in scale.

4. Conclusions

The usage of statistical methods to detect anomalies in computer networks depends heavily on choosing right statistical parameters, and constructing correct time series. Use of statistical methods to detect treats is not 100% effective. It can however be a good complement to traditional methods.

The paper shows that the analysis of a partial data stream (divided using LACP protocol) can be enough to use many statistical methods, i.e. anomaly detection methods in computer networks, but are not limited to them, because even a small partial flow is representative to the whole traffic. The approach presented can speed up the process of statistical analysis and, in consequence, allow using more complex, refined, and thus more resource demanding methods. It should be emphasized that possible implementation of the presented approach is based on network protocols already installed on the devices.

5. References

[1] Oshima S., Nakashima T.: Computational Complexity of Anomaly Detection Methods. Seventh International Conference on Broadband, Wireless Computing, Communication and Applications; Victoria, BC, Canada: IEEE. pp. 664-649, 12-14 Nov 2012.

[2] Feinstein L., Schnackenberg D., Balupari R., Kindred D.: Statistical approaches to DDos attack detection and response. DARPA Information Survivability Conference and Exposition; Washington, DC, USA: IEEE vol. 1, pp. 303–314; 22-24 Apr 2003.

[3] Nychis G., Sekar V., Andersen D., Kim H., Zhang H.: An empirical evaluation of entropy-based traffic anomaly detection. 8th ACM SIGCOMM Conference on Internet measurement. Vouliagmeni, Greece: ACM New York, NY, USA, pp. 151–156; 20-22 Oct 2008.

[4] Allen W., Marin G.: On the self-similarity of synthetic traffic for the evaluation of intrusion detection systems. 2003 Symposium on Applications and the Internet. Orlando, FL, USA: IEEE, pp. 242-248; 27-31 Jan 2003.

[5] Ciftlikli C., Gezer A.: Comparison of Daubechies wavelets for Hurst parameter estimation. Turk J Elec Eng & Comp Sci; vol. 18; pp. 117-128; 2010.

[6] Cyriac J., Hema A.: Decoupling Non-Stationary and Stationary Components in Long Range Network Time Series in the Context of Anomaly Detection. 37th Annual IEEE Conference on Local Computer Networks. Clearwater Beach, FL, USA: IEEE, pp. 76-84; 22-25 Oct 2012.

[7] Ciftlikli C., Gezer A., Ozsahin T.: Packet traffic features of IPv6 and IPv4 protocol traffic. Turk J Elec Eng & Comp Sci; vol. 20; pp. 727-749; 2012.

[8] Manikopoulos C., Papavassiliou S.: Network Intrusion and Fault Detection: A Statistical Anomaly Approach. IEEE Communications Magazine; vol. 40; pp. 76-82; 2002.

[9] Priestley M.: Spectral Analysis and Time Series. Academic Press 1981.

[10] Priestley M.: Non-linear and Non-stationary Time Series Analysis. Academic Press, 1988.

[11] Taqqu M., Teverowsky V., Willinger W.: Estimators for Long-Range Dependance: an Empirical Study. Fractals; vol. 4; pp. 785-788; 1995.

Received: 18.10.2015 *Paper reviewed* *Accepted: 03.12.2015*

Marek BOLANOWSKI, PhD, eng

He received a PhD degree in Computer Science from Lodz University of Technology in 2009. His current research interests focus on computer system and network design and interconnection network performance. He works at the Department of Distributed Systems Rzeszow University of Technology as an assistant professor.

e-mail: marekb@prz.edu.pl

Andrzej PASZKIEWICZ, PhD, eng

He received a PhD degree in Computer Science from Lodz University of Technology in 2009. His current research interests focus on widely under-stood processes in computer networks. He works at the Department of Power Electronics, Power Engineering and Complex Systems Rzeszow University of Technology as an assistant professor.

e-mail: andrzejp@prz.edu.pl

Michał WROŃSKI, MSc, eng

Born in 1986, since 2011 working in Rzeszów University of Technology in Rzeszów, Poland. Main scientific interests include non-extensive statistical mechanics, computer graphics, operating systems design and architecture. Master degree in computer science.

e-mail: mwronski@prz.edu.pl

Robert ŻEGLEŃ, MSc, eng

Robert Żegleń currently works as IT Manager at Citibank, supporting bank's Mainframe infrastructure. He earned his Masters of Engineering at Rzeszow University of Technology. His main interest area is computer network traffic engineering.

e-mail: robert@zeglen.eu