Parametric study of the control system in the TCP network

Tatyana R. Velieva
Department of Applied Probability and Informatics
Peoples' Friendship University of Russia (RUDN University)
6 Miklukho-Maklaya St, Moscow, 117198, Russia
velieva_tr@rudn.university

Anna V. Korolkova
Department of Applied Probability and Informatics
Peoples' Friendship University of Russia (RUDN University)
6 Miklukho-Maklaya St, Moscow, 117198, Russia
korolkova_av@rudn.university

Dmitry S. Kulyabov
Department of Applied Probability and Informatics
Peoples' Friendship University of Russia (RUDN University)
6 Miklukho-Maklaya St, Moscow, 117198, Russia
Laboratory of Information Technologies
Joint Institute for Nuclear Research
Joliot-Curie 6, Dubna, Moscow region, 141980, Russia
kulyabov_ds@rudn.university

Sergey A. Abramov
Federal Research Center Computer Science and Control
of the Russian Academy of Sciences
44/2, Vavilova street, Moscow, 119333, Russia
sergeyabramov@mail.ru

***Abstract*—Self-oscillating modes in control systems of computer networks quite negatively affect the characteristics of these networks so the investigation of parameters of self-oscillations as well as self-oscillations areas is actual. But due to the non-linear nature of usually constructed mathematical models the study of self-oscillations areas and parameters are extremely labor-intensive. It is of interest to obtain a so-called parametric portrait describing the zones of occurrence of self-oscillations depending on the value of the parameters: one parameter (two-dimensional graph), two parameters (three-dimensional graph), and so on. Such a parametric portrait allows us to purposefully manage the characteristics of the investigated control system. The paper describes a parametric study technique based on the method of harmonic linearization because in the standard mathematical model based on ordinary linearization by Taylor expansion a self-oscillation regime disappears (due to Taylor expansion linearization). To verify the theoretical results obtained, simulation is used. In addition, it is proposed to use the computer algebra system for analytical calculations. For this, the criteria for choosing software were formulated. Based on these criteria, a set of software for analytical and numerical calculations was proposed.**

***Index Terms*—active queue management, simulation, NS2, Julia, SymPy, self-oscillating**

I. Introduction

The study of characteristics of technical systems with control as well as the study of influence of system parameters on behaviour of these characteristics is often required while modelling real technical systems with control with such a parasitic phenomenon as global synchronization. Global synchronization manifests itself as a self-oscillatory mode. In computer networks, in which TCP is the main transport protocol, the phenomenon of global synchronization occurs during traffic management [1], [2].

In this paper, we describe a technique for parametric study of a model with control based on the Krylov–Bogolyubov method [3], also known as harmonic linearization [4]. Also we discuss the choice of software to perform this research, as well as an approach to verifying the results obtained.

The structure of the paper is as follows. In the section II, the actual study plan is given. In the section III, the brief introduction to the RED algorithm is presented. In the section IV, we describe the criteria for selecting software tools. In the section V, we describe the method of harmonic linearization for nonsymmetric oscillations. In the section VI, we demonstrate the application of the harmonic linearization method for the RED algorithm. In the section VII, we show the example of the parametric portrait for the active traffic management module. In the section VIII, we describe the verification of the results of theoretical calculations.

II. Investigation workflow

The study are performed in accordance with the scheme on Fig. 1.

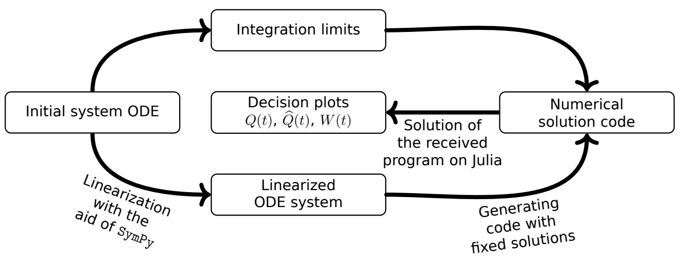

Fig. 1. Investigation workflow

The mathematical model constructed via the system of ordinary differential equations (2) is under investigation. To find the self-oscillation parameters, we assume that the system of equations will be linearized. We also should keep in mind that with the standard linearization, we lose the oscillatory structure of the system [5]. As an alternative, it is proposed to use the so-called harmonic linearization (see section V) [6]. After linearization, the system breaks up into several parts (see Fig. 2).

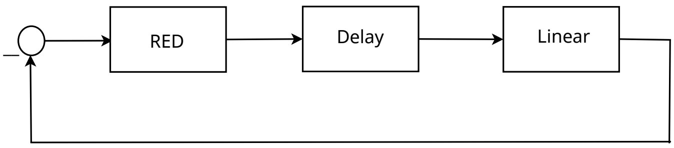

Fig. 2. Scheme of linearized RED

Harmonic linearization procedure is not a rocket science, however, it consists of a large number of trivial operations. For carrying out calculations it is proposed to use the SymPy computer algebra software (see section IV).

When a harmonic linearization occurs, a family of models appears (see section VI). Each model is obtained using a computer algebra system. Using the computer algebra system, we get for each submodel a set of files in the Julia language (see section IV). Further, by setting different values of the parameters, one can obtain the parametric portrait of self-oscillations (see section VII). It should be noted that calculations must be made for all submodels, although the solution will exist only for one submodel.

For verification, the NS2 simulation [7] is used (see section VIII).

III. The RED congestion adaptive control mechanism

The RED algorithm [8], [9], [10], [11] uses the probability of incoming packets drop with a weighted queue length as a parameter of the drop probability function. The value of drop function (the probability of packets drop) increases (see Eq. (1)) as the exponentially-weighted average queue length grows and depends on two threshold values of the average queue length.

$$p(\hat{Q}) = \begin{cases} 0, & 0 < \hat{Q} \leqslant Q_{\min}, \\ \dfrac{\hat{Q} - Q_{\min}}{Q_{\max} - Q_{\min}} p_{\max}, & Q_{\min} < \hat{Q} \leqslant Q_{\max}, \\ 1, & \hat{Q} > Q_{\max}. \end{cases} \tag{1}$$

Here $p(\hat{Q})$ is the drop function (the packet drop probability), $\hat{Q}$ is the exponentially-weighted queue size moving average, $Q_{\min}$ and $Q_{\max}$ are the thresholds for the exponentially-weighted queue length moving average, $p_{\max}$ is the the maximum value of drop function.

In spite of efficiency of the RED algorithm (due to simplicity of its implementation in the network hardware) it has some drawbacks, for example, there is a steady oscillatory mode for a number of parameters of the system, and the quality of service (QoS) indicators are negatively affected by this steady oscillatory mode [5], [12], [2]. Unfortunately there are no clear selection criteria for determination of self-oscillation area for RED parameters values.

The mathematical continuous model (see [13], [14], [15], [16], [17], [18]) with following simplifying assumptions: the model is written in the moments; only the congestion avoidance phase for TCP Reno protocol is considered; and the drop is possible only after reception of 3 consistent ACK confirmations, will be used for the RED algorithm description.

$$\begin{cases} \dot{W}(t) = \dfrac{1}{T(Q,t)} - \\ \qquad - \dfrac{W(t)W(t-T(Q,t))}{2T(t-T(Q,t))} p(t-T(Q,t)); \\ \dot{Q}(t) = \dfrac{W(t)}{T(Q,t)} N(t) - C; \\ \dot{\hat{Q}}(t) = -w_q C \hat{Q}(t) + w_q C Q(t). \end{cases} \tag{2}$$

Here the following notation is used: W is the average TCP window size; Q is the average queue size; $\hat{Q}$ is the exponentially weighted moving average (EWMA) of the queue size average; C is the queue service intensity; T is the full round-trip time; $T = T_p + \frac{Q}{C}$, where T_p is the round-trip time for free network (excluding delays in hardware); $\frac{Q}{C}$ is the time which package spent in the queue; N is the number of TCP sessions; p is the packet drop function.

IV. Software choice

The RED model software implementation may be carried out in two stages. At the first stage, a computer algebra system was employed. With the help of this system the whole time-consuming processing of the formulas is carried out. The resulting expressions are used in the numerical programs generation and in the formulas transfer to the text of articles. In our work we use the SymPy system [19] of symbolic calculations. This system initially was developed as a library of symbolic calculations for the Python language, which has become a universal glue language with the explosive growth of related tools and libraries due to its application in a variety of projects. Therefore, SymPy developed along with it. Now this is a fairly powerful system of computer algebra. SymPy suits us for the following reasons:

1) It is convenient to use the Jupyter notepad (with REPL ideology), which is a component of the system iPython [20], as the interactive shell.
2) Python, as a glue language, allows to integrate different software products. In addition, within the SciPy library [21] is supported a large number of output formats.
3) The output of SymPy can be naturally transfered to the NumPy [22] library for further numerical calculations and to other programming languages.

Then the resulting formulas may be used for computational programs generation. We suggest to use the Julia language [23] as a numerical programming language. It is unlikely that this language is really a silver bullet. However, it has a number of interesting features. This language is positioned as a modern reincarnation of the FORTRAN language. It supports as the stage of prototyping as well as writing the program final version. This language is intensively developing. All these factors have attracted our attention to this language.

V. Harmonic linearization method

The method of harmonic linearization was proposed by N. N. Bogolyubov, N. M. Krylov [3] and H. Nyquist [4]. The core of this method consists of separating the 'slow' variables from the 'fast' variables. The essential difference between harmonic linearization and the usual linearization method is that the harmonic-linearized system depends on the amplitudes and frequencies of the periodic processes, and the usual linearization method leads only to purely linear expressions, which allows to investigate the basic properties of nonlinear systems.

The harmonic linearization method is applied for systems, which consist of a linear link H_l and a nonlinear link H_{nl}, given by the function $f(x)$. A static nonlinear element is usually considered.

To the nonlinear element input free harmonic oscillations are applied:

$$x(t) = x_0 + \tilde{x} := x_0 + A\sin(\omega t).$$

On the output of the nonlinear element $f(x)$ we get a periodic signal. Let's expand it in a Fourier series:

$$y = \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \sin(k\omega t) + b_k \cos(k\omega t)).$$

with the following form of the Fourier series coefficients:

$$\begin{aligned} a_0 &= \frac{1}{\pi}\int_0^{2\pi} f(x_0 + A\sin(\omega t))\,\mathrm{d}(\omega t)\,; \\ a_k &= \frac{1}{\pi}\int_0^{2\pi} f(x_0 + A\sin(\omega t))\sin(k\omega t)\,\mathrm{d}(\omega t)\,; \\ b_k &= \frac{1}{\pi}\int_0^{2\pi} f(x_0 + A\sin(\omega t))\cos(k\omega t)\,\mathrm{d}(\omega t)\,; \quad k = \overline{1,\infty}. \end{aligned}$$

The linear element is a low-pass filter, which suppresses higher harmonics for k increase.

Let's write the signal after the non-linear element:

$$\begin{aligned} y = y_0 + \tilde{y} &\approx \\ &\approx \varkappa_0(A,\omega,x_0) + [\varkappa(A,\omega,x_0) + \mathrm{i}\varkappa'(A,\omega,x_0)]\tilde{x}, \end{aligned} \tag{3}$$

$\varkappa_0$ is a constant shift, $\varkappa$ and $\varkappa'$ are the harmonic linearization coefficients:

$$\begin{aligned} \varkappa_0(A,\omega,x_0) &= \frac{1}{2\pi}\int_0^{2\pi} f(x_0 + A\sin(\omega t))\,\mathrm{d}(\omega t)\,; \\ \varkappa(A,\omega,x_0) &= \frac{a_1}{A} = \\ &= \frac{1}{A\pi}\int_0^{2\pi} f(x_0 + A\sin(\omega t))\sin(\omega t)\,\mathrm{d}(\omega t)\,; \\ \varkappa'(A,\omega,x_0) &= \frac{b_1}{A} = \\ &= \frac{1}{A\pi}\int_0^{2\pi} f(x_0 + A\sin(\omega t))\cos(\omega t)\,\mathrm{d}(\omega t)\,. \end{aligned} \tag{4}$$

In addition to (3), we will write

$$\begin{aligned} z = z_0 + \tilde{z} = (y_0 + \tilde{y})H_l(\omega), \\ x = x_0 + \tilde{x} = g(\omega) - (z_0 + \tilde{z}). \end{aligned}$$

Then the harmonic linearization equation is derived:

$$\begin{aligned} &\left[x_0 + H_l(\omega)\bigg|_{\omega=0} \varkappa_0(A,\omega,x_0)\right] + \\ &\quad + [1 + H_l(\varkappa(A,\omega,x_0) + \mathrm{i}\varkappa'(A,\omega,x_0))]\tilde{x} = \\ &\qquad = g(\omega) := g_0(\omega) + \tilde{g}(\omega). \end{aligned}$$

By separating for constant and harmonic components, it may written:

$$\begin{aligned} \left[x_0 + H_l(\omega)\bigg|_{\omega=0} \varkappa_0(A,\omega,x_0)\right] = g_0(\omega), \\ [1 + H_l(\varkappa(A,\omega,x_0) + \mathrm{i}\varkappa'(A,\omega,x_0))]\tilde{x} = \tilde{g}(\omega). \end{aligned}$$

During the study of self-oscillatory mode the additional assumption that there is no external signal ($g = 0$) was made.

VI. RED model harmonic linearization

The RED model linearization and the function H_l derivation are described in detail in the article [24], [6].

Let us compute the coefficients of harmonic linearization $\varkappa_0(A,\omega,x_0)$, $\varkappa(A,\omega,x_0)$ and $\varkappa'(A,\omega,x_0)$ (4) for the static nonlinearity P_{RED}:

$$\begin{aligned} \varkappa_0(A,\omega,x_0) &= \frac{1}{2\pi}\int_0^{2\pi} P_{\mathrm{RED}}(x_0 + A\sin(\omega t))\,\mathrm{d}(\omega t)\,; \\ \varkappa(A,\omega,x_0) &= \frac{1}{A\pi}\int_0^{2\pi} P_{\mathrm{RED}}(x_0 + A\sin(\omega t))\sin(\omega t)\,\mathrm{d}(\omega t)\,; \\ \varkappa'(A,\omega,x_0) &= \frac{1}{A\pi}\int_0^{2\pi} P_{\mathrm{RED}}(x_0 + A\sin(\omega t))\cos(\omega t)\,\mathrm{d}(\omega t)\,. \end{aligned}$$

The different limits of integration may be obtained depending on the relations between the thresholds $Q_{\min}$ and $Q_{\max}$, the shift x_0 and the amplitude A.

Here are some examples of the graphical method for finding the near-points of integration (see Fig. 3) depending on the relations between the constant shift x_0, the amplitude A, the thresholds $Q_{\min}$ and $Q_{\max}$.

The resulting harmonic linearization coefficients are used to generate the program by means of a computer algebra system.

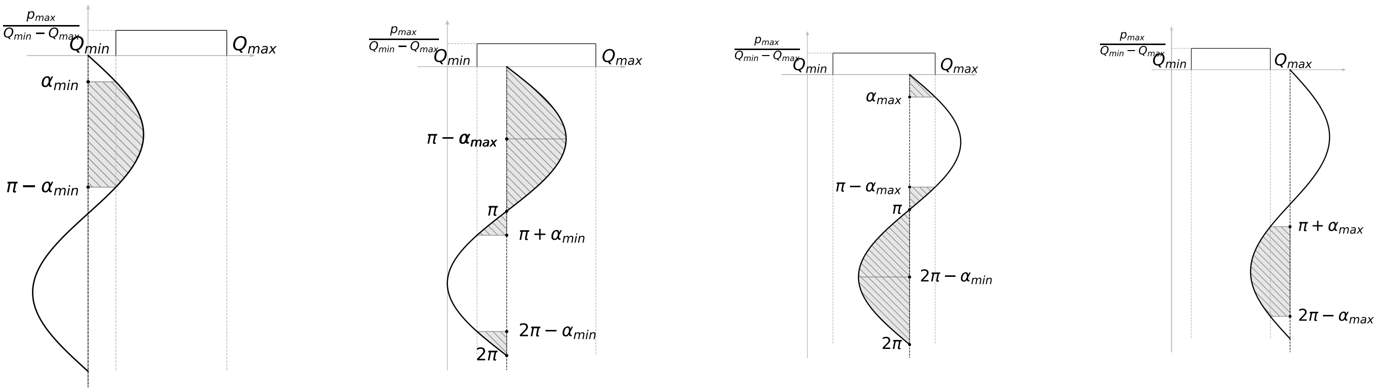

Fig. 3. Integration limits

VII. Influence of parameters on occurrence of self-oscillations

By using the developed algorithm, the dependence of self-oscillation regions on the RED algorithm parameters may be investigated. Naturally, we can consider other variants of RED-like algorithms.

The example [25] may be considered as the illustration. The following RED parameters are given: the number of sessions $N = 60$, round-trip time $T_p = 0.5$ s, thresholds $Q_{\min} = 75$ packages and $Q_{\max} = 150$ packets, drop probability $p = 0.1$, parameter $w_q = 0.002$. Let us investigate the dependence of self-oscillation on the link capacity C (see Fig. 4). The result is that the transition to the self-oscillatory regime occurs at $C_a = 15$ Mbps. That is, for $C \geqslant C_a$ the system will be in self-oscillation mode.

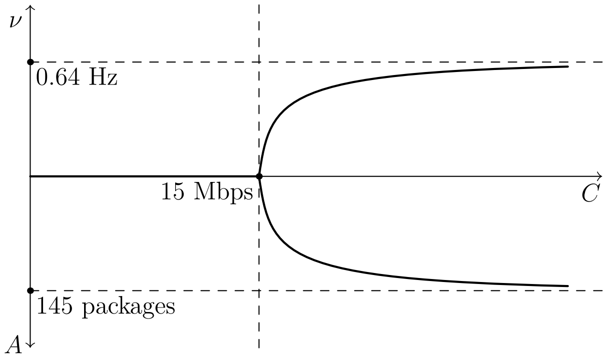

Fig. 4. Parametric portrait

VIII. Verification of results

The full-scale experiment often involves certain difficulties. For example, the real equipment is not always available. Also the use of a virtual stand is associated with high demands on computer equipment [16]. In addition, since the simulation takes place in real time, the whole process is extremely long.

To save resources and time, simulation tools are usually used. The package ns2 [7], [26] is a tool for network protocols simulating. This package was created as a reference modeling tool, so it is often used as an alternative to the full-scale experiment.

For the simulation we will use the so-called dumbbell topology (see Fig. 5, 6). Additional TCP sessions are emulated by addition of extra sources.

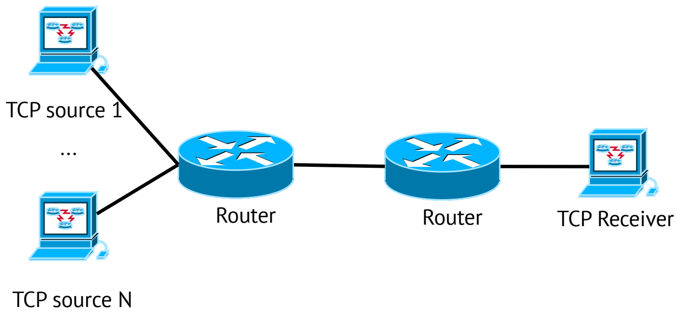

Fig. 5. Dumbbell topology

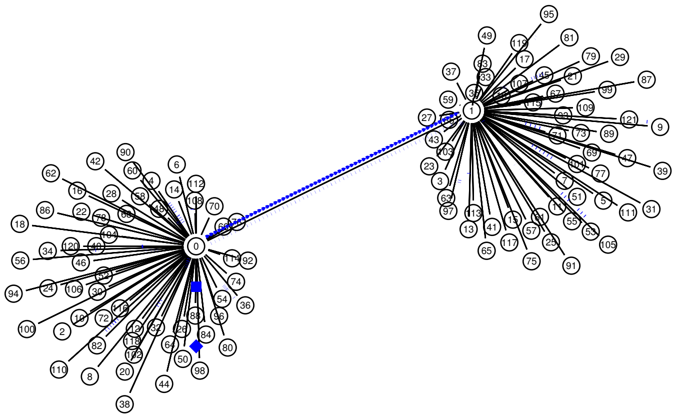

Fig. 6. Visualization of the simulation. Packets drop is shown

By using the output data the parameters of self-oscillations may be obtained. Here are the fragments of the program in the Julia language [23], in which the spectral portrait of the self-oscillatory mode is constructed on the basis of the Fast Fourier Transform algorithm [27].

Fragments of the program code for the ns2 simulation software for the model under study, as well as code fragments in the Julia language, responsible for constructing the spectral portrait of the self-oscillatory mode based on the Fast Fourier Transform algorithm, were presented in [28].

The Fig. 7 and Fig. 8 show the behavior of the average queue length for link capacity $C = 5$ Mbps and $C = 20$ Mbps. In the second case clearly shows the presence of the self-oscillation mode. Theoretically obtained characteristic of this mode: oscillation frequency $\nu = 0.6$ Hz, oscillation amplitude $A = 150$ packets.

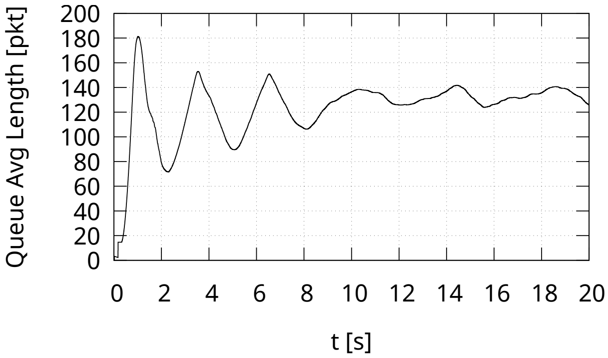

Fig. 7. Average queue length at link capacity $C = 5$ Mbps

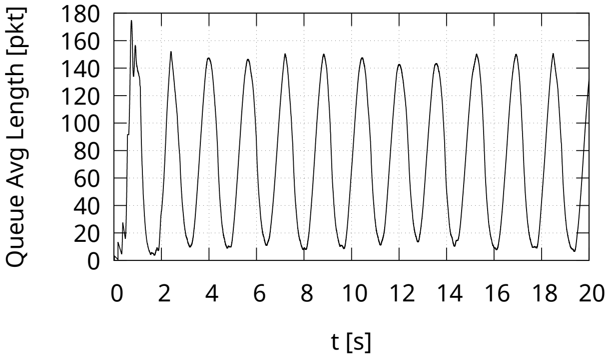

Fig. 8. Average queue length at link capacity $C = 20$ Mbps

In the spectral study of the results of the simulation, we obtained the following characteristics: the frequency of self-oscillations $\nu = 0.5$ Hz, the amplitude of the oscillations $A = 169$ packets (see Fig. 9 and Fig. 10). As can be seen, the theoretical and experimental results are very close. Thus, our program complex can serve the purposes of verification of theoretical studies of the self-oscillatory regime in control systems.

We conducted a parametric study of the active traffic control module type RED. On the basis of the study methodology parametric study was formulated. For the study a set of programs for analytical and numerical calculations has been created. Verification of theoretical results was carried out in the simulation system NS2.

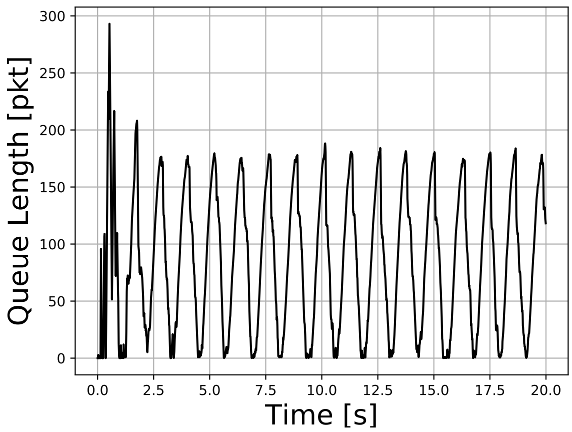

Fig. 9. Instantaneous queue length at link capacity $C = 20$ Mbps

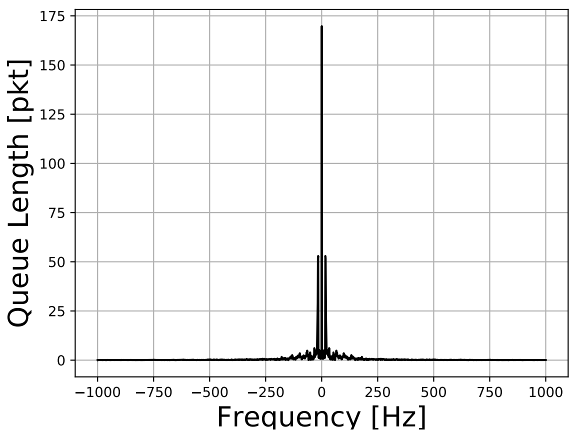

Fig. 10. Spectrum of self-oscillations of instantaneous queue length at link capacity $C = 20$ Mbps

Acknowledgment

The publication has been prepared with the support of the “RUDN University Program 5-100” and funded by Russian Foundation for Basic Research (RFBR) according to the research project No 16-07-00556.

References

[1] M. Allman, V. Paxson, and E. Blanton, “TCP Congestion Control,” Tech. Rep., sep 2009.

[2] W. Lautenschlaeger and A. Francini, “Global Synchronization Protection for Bandwidth Sharing TCP Flows in High-Speed Links,” in *Proc. 16-th International Conference on High Performance Switching and Routing, IEEE HPSR 2015*, Budapest, Hungary, 2015.

[3] N. Kryloff and N. Bogoliuboff, “Les méthodes symboliques de la Mécanique non Linéaire dans leur application à l’étude de la résonance dans l’oscillateur,” *Bulletin de l’Académie des Sciences de l’URSS. Classe des sciences mathématiques*, no. 1, pp. 7–34, 1934.

[4] H. Nyquist, “Regeneration Theory,” *Bell System Technical Journal*, vol. 11, no. 1, pp. 126–147, 1932.

[5] A. Jenkins, “Self-Oscillation,” *Physics Reports*, vol. 525, no. 2, pp. 167–222, apr 2013.

[6] D. S. Kulyabov, A. V. Korolkova, T. R. Velieva, E. G. Eferina, and L. A. Sevastianov, “The Methodology of Studying of Active Traffic Management Module Self-oscillation Regime,” in *DepCoS-RELCOMEX 2017. Advances in Intelligent Systems and Computing*, ser. Advances in Intelligent Systems and Computing, W. Zamojski, J. Mazurkiewicz, J. Sugier, T. Walkowiak, and J. Kacprzyk, Eds. Cham: Springer International Publishing, 2018, vol. 582, pp. 215–224.
[7] T. Issariyakul and E. Hossain, *Introduction to Network Simulator NS2*. Boston, MA: Springer US, 2012.
[8] S. Floyd and V. Jacobson, “Random Early Detection Gateways for Congestion Avoidance,” *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, 1993.
[9] V. Jacobson, “Congestion Avoidance and Control,” *ACM SIGCOMM Computer Communication Review*, vol. 18, no. 4, pp. 314–329, 1988.
[10] V. Kushwaha and R. Gupta, “Congestion Control for High-Speed Wired Network: A Systematic Literature Review,” *Journal of Network and Computer Applications*, vol. 45, pp. 62–78, 2014.
[11] R. Adams, “Active Queue Management: A Survey,” *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1425–1476, 2013.
[12] F. Ren, C. Lin, and B. Wei, “A Nonlinear Control Theoretic Analysis to TCP-RED System,” *Computer Networks*, vol. 49, no. 4, pp. 580–592, 2005.
[13] V. Misra, W.-B. Gong, and D. Towsley, “Stochastic Differential Equation Modeling and Analysis of TCP-Windowsize Behavior,” *Proceedings of PERFORMANCE*, vol. 99, 1999.
[14] ——, “Fluid-Based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED,” *ACM SIGCOMM Computer Communication Review*, vol. 30, no. 4, pp. 151–160, oct 2000.
[15] C. V. V. Hollot, V. Misra, D. Towsley, and Wei-Bo Gong, “On Designing Improved Controllers for AQM Routers Supporting TCP Flows,” in *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213)*, vol. 3. IEEE, 2001, pp. 1726–1734.
[16] T. R. Velieva, A. V. Korolkova, and D. S. Kulyabov, “Designing Installations for Verification of the Model of Active Queue Management Discipline RED in the GNS3,” in *6th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*. IEEE Computer Society, 2015, pp. 570–577.
[17] A. V. Korolkova, T. R. Velieva, P. A. Abaev, L. A. Sevastianov, and D. S. Kulyabov, “Hybrid Simulation Of Active Traffic Management,” *Proceedings 30th European Conference on Modelling and Simulation*, pp. 685–691, jun 2016.
[18] R. Brockett, “Stochastic Analysis for Fluid Queueing Systems,” in *Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No.99CH36304)*, vol. 3. IEEE, 1999, pp. 3077–3082.
[19] R. Lamy, *Instant SymPy Starter*. Packt Publishing, 2013.
[20] F. Perez and B. E. Granger, “IPython: A System for Interactive Scientific Computing,” *Computing in Science & Engineering*, vol. 9, no. 3, pp. 21–29, 2007.
[21] T. E. Oliphant, “Python for Scientific Computing,” *Computing in Science & Engineering*, vol. 9, no. 3, pp. 10–20, 2007.
[22] ——, *Guide to NumPy*, 2nd ed. CreateSpace Independent Publishing Platform, 2015.
[23] A. Joshi and R. Lakhanpal, *Learning Julia*. Packt Publishing, 2017.
[24] T. R. Velieva, D. S. Kulyabov, A. V. Korolkova, and I. S. Zaryadov, “The approach to investigation of the the regions of self-oscillations,” *Journal of Physics: Conference Series*, vol. 937, pp. 012 057_1–8, dec 2017.
[25] T. R. Velieva, A. V. Korolkova, A. V. Demidova, and D. S. Kulyabov, “Software Package Development for the Active Traffic Management Module Self-oscillation Regime Investigation,” in *Contemporary Complex Systems and Their Dependability: Proceedings of the Thirteenth International Conference on Dependability and Complex Systems DepCoS-RELCOMEX, July 2-6, 2018, Brunów, Poland*, ser. Advances in Intelligent Systems and Computing, W. Zamojski, J. Mazurkiewicz, J. Sugier, T. Walkowiak, and J. Kacprzyk, Eds. Cham: Springer International Publishing, 2019, vol. 761, ch. 48, pp. 515–525.
[26] E. Altman and T. Jiménez, “NS Simulator for Beginners,” *Synthesis Lectures on Communication Networks*, vol. 5, no. 1, pp. 1–184, jan 2012.
[27] K. R. Rao, D. N. Kim, and J. J. Hwang, *Fast Fourier Transform - Algorithms and Applications*, ser. Signals and Communication Technology. Springer, 2010.
[28] T. R. Velieva, A. V. Korolkova, M. N. Gevorkyan, S. A. Vasilyev, I. S. Zaryadov, and D. S. Kulyabov, “Software Package For The Active Queue Management Module Model Verification,” in *Proceedings 32st European Conference on Modelling and Simulation, ECMS 2018*, L. Nolle, A. Burger, C. Tholen, J. Werner, and J. Wellhausen, Eds. Wilhelmshaven: European Council for Modelling and Simulation, may 2018, pp. 498–504.