

МЕТОДИКА КОДИРОВАНИЯ УПРАВЛЯЮЩИХ АВТОМАТОВ ДЛЯ ПЕРЕДАЧИ ТРАФИКА В ПРОГРАММНО-КОНФИГУРИРУЕМОЙ СЕТИ

К.И. Никишин

Пензенский государственный университет, г. Пенза, Россия

Аннотация: программно-конфигурируемые сети (ПКС) – новая парадигма в области сетевых телекоммуникаций. Отличием от классических сетей является то, что с инфраструктурным уровнем взаимодействует контроллер ПКС на уровне управления. Однако классический метод передачи разнородного трафика на основе протокола OpenFlow имеет ряд недостатков. Для устранения недостатков передачи трафика направлены новые методы передачи трафика, методика комплексного контроля передачи трафика, методика кодирования управляющих автоматов для передачи трафика в ПКС. Методика кодирования управляющих автоматов основывается на цифровых автоматах Мура и алгоритмах ранней диагностики потерь трафика реального времени при передаче разнородного трафика в коммутаторе OpenFlow и передачи трафика реального времени с использованием планировщика и функцией контроля доставки в ПКС. Разработаны управляющие автоматы в соответствии с методикой и цифровыми автоматами Мура. Проведено моделирование работы управляющих автоматов на ПЛИС семейства Spartan 6 с помощью временных диаграмм. Оценены аппаратные затраты на ПЛИС управляющих автоматов. Спроектированные управляющие автоматы могут войти в состав телекоммуникационного оборудования, контроллеров или коммутаторов OpenFlow по средствам ПЛИС или ASIC для контроля и передачи трафика в ПКС с помощью управляющих сигналов и команд

Ключевые слова: программно-конфигурируемая сеть, OpenFlow, разнородный трафик, цифровые автоматы, сети Петри, управляющий автомат, ПЛИС

Введение

Программно-конфигурируемые сети (ПКС) – новая парадигма в области сетевых телекоммуникаций [1-2]. Основной отличительной особенностью от классической сети Ethernet [3] с поддержкой «качества обслуживания» [4] является то, что с инфраструктурным уровнем взаимодействует контроллер ПКС на уровне управления. Таким образом, контроллер ПКС является важной частью в сети [5].

В ПКС основным протоколом является OpenFlow, протокол позволяет обрабатывать разнородный трафик и управлять им [6-7]. Однако классический метод передачи разнородного трафика на основе протокола OpenFlow имеет ряд недостатков, более подробно недостатки передачи разнородного трафика описаны в статье [8]. Рассмотрим некоторые из этих недостатков.

К первому недостатку можно отнести отсутствие быстрой передачи разнородного трафика, поскольку тратятся значительные временные и аппаратные ресурсы для поиска таймаутов потока.

Следующим недостатком является то, что отсутствует контроль функций доставки трафика на входных портах коммутаторов OpenFlow. Из-за этого затруднительно определить доставку трафика реального времени получателю в необходимые моменты времени.

В связи с этими недостатками автором статьи были предложены и разработаны новые методы передачи и контроля трафика в ПКС. Разработан метод ранней диагностики потерь трафика реального времени при передаче разнородного трафика в коммутаторе OpenFlow (метод 1) [8]. Метод 2 заключается в передаче трафика реального времени с использованием планировщика и функцией контроля доставки в ПКС.

Описание методики

Автором статьи была разработана и описана методика комплексного контроля передачи трафика в ПКС. Данная методика основывается на новых методах передачи трафика, таких как метод ранней диагностики потерь трафика реального времени при передаче разнородного трафика в коммутаторе OpenFlow и метод передачи трафика реального времени с использованием планировщика и функцией контроля доставки в ПКС.

При этом администратор сети ПКС может настроить требуемый метод передачи трафика с его контролем в ПКС или автоматически устанавливает соответствующий контроль аппаратура сети. То есть, возможен выбор одного из методов передачи трафика в ПКС. Передаваемый трафик также проверялся на валидацию в соответствии с сетевыми стандартами при помощи специально разработанного программного обеспечения [9].

Кроме этого возможен комплексный контроль передачи трафика в ПКС, тогда предложенные методы включаются последовательно в работе ПКС.

На основе предложенной методики были разработаны цифровые автоматы Мура, описаны с помощью систем канонических уравнений и выходных функций, построены различные таблицы переходов состояний цифровых автоматов.

Формализация алгоритмов и методики передачи трафика в ПКС в виде цифровых автоматов связана с тем, что цифровые автоматы являются промежуточным звеном для исследования алгоритмов с помощью имитационного моделирования (OMNET++, CPN Tools, Cisco Packet Tracer [10]). От цифровых автоматов можно перейти к сетям Петри.

В качестве среды имитационного моделирования выбран пакет моделирования CPN Tools [11]. CPN Tools обладает всеми преимуществами аппарата сетей Петри, позволяет строить различные виды сетей Петри.

Пакет CPN Tools наилучшим образом подходит для исследования компьютерных сетей, их критериев, метрик и задержки в сети. Он обладает максимальной производительностью по исследованию новых методов и алгоритмов в компьютерных сетях.

Но описанной формализации на цифровых автоматах и моделирования передачи трафика в ПКС на основе сетей Петри было недостаточно. Возникла необходимость спроектировать управляющие автоматы на основе заявленных методов и методики передачи трафика в ПКС.

В аппаратном обеспечении, микропроцессорах разделяют управляющие и операционные автоматы. Операционный автомат исполняет определённую команду или операцию.

Основную роль выполняет управляющий автомат, который посылает соответствующие команды, синхросигналы для исполнения в

оборудовании. Поэтому управляющие автоматы должны быть разработаны к привязке к конечному оборудованию, аппаратуре.

Язык описания схем VHDL позволяет осуществить кодирование управляющих автоматов на ПЛИС типа FPGA. Таким образом, работу управляющих автоматов можно проверить на реальном оборудовании ПЛИС и получить необходимые результаты. В качестве примера было выбрано современное семейство ПЛИС Spartan 6, можно проектировать управляющие автоматы и на более поздних семействах ПЛИС.

Рассмотрим методику кодирования управляющих автоматов для передачи трафика в ПКС. Методика состоит из нескольких этапов.

Этап 1 - на основе методики комплексного контроля и методов передачи трафика получаем цифровые автоматы Мура с заданным количеством состояний. Получили прямые таблицы переходов состояний цифровых автоматов.

Этап 2 - кодируем в соответствии с прямой таблицей переходов текущие и следующие состояния, входные сигналы.

Этап 3 - устанавливаем временные задержки для формирования и смены состояний цифрового автомата. Производим моделирование управляющего автомата, получаем временные диаграммы работы управляющего автомата.

Этап 4 - проверяем работу управляющего автомата на ПЛИС с помощью временных диаграмм. Сравниваем значения, полученные на временных диаграммах с разработанной прямой таблицей переходов. В случае отклонения от работы цифрового автомата - необходимость скорректировать кодированием управляющего автомата и вернуться к этапу 2.

Этап 5 - спроектированные управляющие автоматы могут войти в состав телекоммуникационного оборудования, контроллеров или коммутаторов OpenFlow по средствам ПЛИС или ASIC для контроля и передачи трафика в ПКС с помощью управляющих сигналов и команд.

Проектирование управляющих автоматов на ПЛИС

Рассмотрим цифровой автомат Мура, который соответствует методу ранней диагностики потерь трафика реального времени и представлен на рис. 1. У цифрового автомата

имеются состояния ($a_1 - a_8$), входной сигнал (переходы $z_1 - z_4$), выходной сигнал ($w_1 - w_8$), безусловные переходы обозначаются «1».

Для полученного цифрового автомата Мура напомним систему канонических уравнений (СКУ) и систему выходных функций (СВФ). СКУ является аналитической

интерпретацией таблиц переходов автоматов и определяет функции переходов цифрового автомата. СВФ является аналитической интерпретацией таблиц выходов автоматов и определяет функции выходов цифрового автомата.

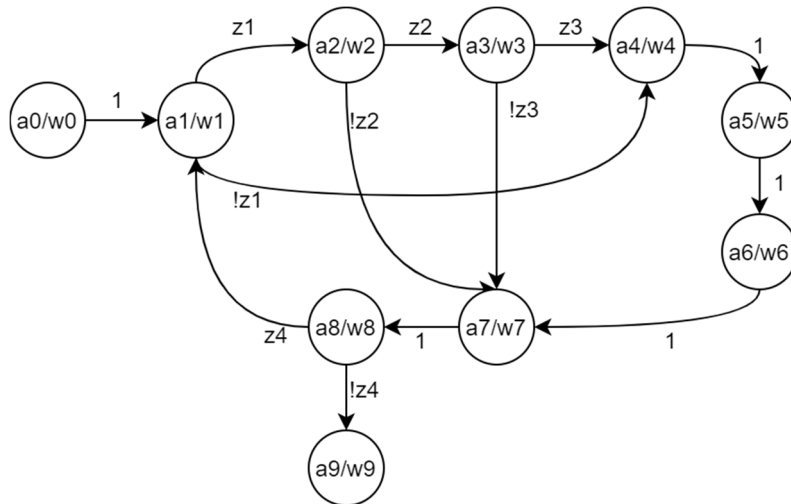


Рис. 1. Граф цифрового автомата Мура

СКУ:

$$a_1(t+1) = a_7z_4;$$

$$a_2(t+1) = a_1z_1;$$

$$a_3(t+1) = a_2z_2;$$

$$a_4(t+1) = a_3z_3 \vee a_1(!z_1)$$

$$a_5(t+1) = a_4$$

$$a_6(t+1) = a_5$$

$$a_7(t+1) = a_6 \vee a_3(!z_3) \vee a_2(!z_2)$$

$$a_8(t+1) = a_7$$

$$a_9(t+1) = a_8$$

СВФ:

$$w_1(t) = a_1$$

$$w_2(t) = a_2$$

$$w_3(t) = a_3$$

$$w_4(t) = a_4$$

$$w_5(t) = a_5$$

$$w_6(t) = a_6$$

$$w_7(t) = a_7$$

$$w_8(t) = a_8$$

$$w_9(t) = a_9$$

Построили прямую таблицу переходов состояний цифрового автомата. Схема работы управляющего автомата заключается в том, что на свой вход поступает код текущего состояния

цифрового автомата (state) и внешний сигнал (z). На выходе управляющего автомата получаем код следующего состояния (next_state). Кроме этого входящими сигналами являются синхросигнал и сигнал сброса.

Для описания состояний цифрового автомата необходимо использовать перечисляемый тип. Для этого описывается тип (state_values), значениями которого являются состояния цифрового автомата. Внутренний сигнал (state) будет иметь этот перечисляемый тип, в котором и сохраняются текущие состояния автомата. Их описание представлено ниже.

type state_values is (s0, s1, s2, s3, s4, s5, s6, s7, s8, s9);

signal state, next_state: state_values;

Если поступает сигнал сброса, то управляющий автомат находится в начальном состоянии, т.е. S0. При наступлении нарастающего фронта синхросигнала происходит изменение состояний управляющего автомата.

Для описания логики переходов цифрового автомата используется оператор case. Логика программируется в соответствии с таблицей переходов состояний цифрового автомата. При этом в зависимости от входящего сигнала автомат будет находиться в

различных состояниях. Пример описания переходов состояния управляющего автомата представлен ниже.

```
process (state, z)
begin
  case state is when s0 =>
    out1 <="0000";
    if z ="01" then next_state <= s1; end if;
  ...
end process;
```

Для проверки работы разработанного управляющего автомата необходимо разработать файл на языке VHDL для моделирования ПЛИС и получения временных диаграмм. В этом файле устанавливаются временные задержки для формирования и

смены состояний цифрового автомата и вся основная реакция управляющего автомата на входные сигналы. Фрагмент кода представлен ниже.

```
stim_proc: process
begin
  rst<='1'; wait for 25 ns;
  rst<='0'; wait for 25 ns;
  z<="01"; wait for 25 ns;
  z<="11"; wait for 25 ns;
end process;
```

END;

Временная диаграмма работы управляющего автомата метода 1 представлена на рис. 2.

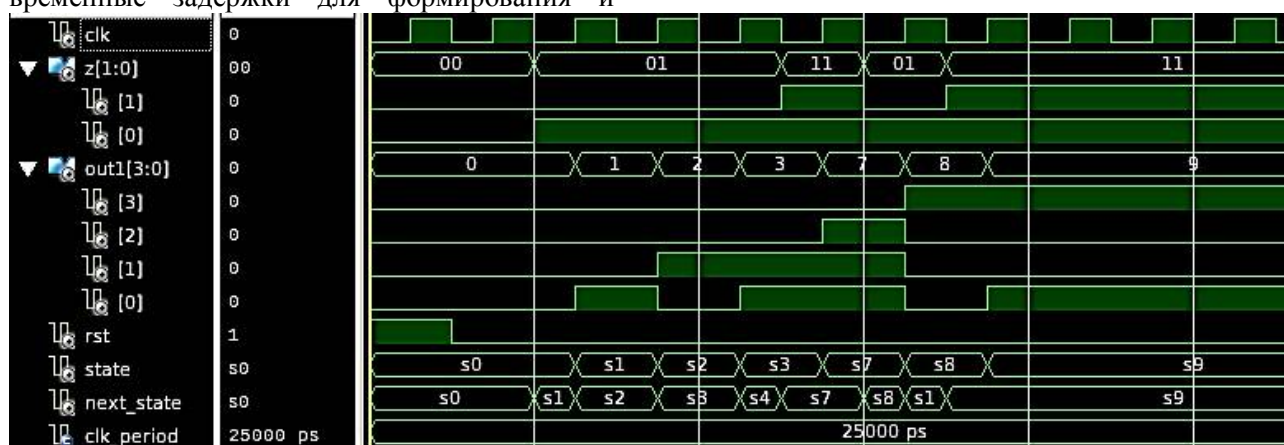


Рис. 2. Временная диаграмма работы управляющего автомата метода 1

Затраты ПЛИС управляющего автомата представлены в отчете ниже, показан процент занятых регистров, триггеров и логических таблиц.

Количество используемых регистров: 8 из 54,576 1%

Количество используемых триггеров: 4

Количество используемых логических таблиц: 4 из 27,288 1%

Аналогичным образом для метода 2 передачи трафика реального времени с использованием планировщика и функцией контроля доставки в ПКС был разработан цифровой автомат Мура, СКУ, СВФ, таблица переходов состояний.

С помощью разработанной методики кодирования аналогичным образом был разработан управляющий автомат. Данный управляющий автомат состоит из большего количества состояний и входных сигналов. Поэтому будет показана только временная диаграмма работы управляющего автомата метода 2 на рис. 3.

Затраты ПЛИС управляющего автомата представлены в отчете ниже, показан процент занятых регистров, триггеров и логических таблиц.

Количество используемых регистров: 10 из 54,576 1%

Количество используемых триггеров: 5

Количество используемых логических таблиц: 19 из 27,288 1%

Отличие данной статьи от предыдущих работ автора заключается в том, что разработана методика кодирования управляющих автоматов для передачи трафика в ПКС, состоящая из нескольких этапов и основанная на предложенных автором методах передачи трафика в ПКС. С помощью данной методики были спроектированы цифровые автоматы Мура, системы канонических уравнений и выходных функций. Разработанные управляющие автоматы позволили выполнить синтез цифровых автоматов и методики в составе реального оборудования на ПЛИС, которое в дальнейшем

может являться частью в составе контроллеров

или коммутаторов OpenFlow ПКС.

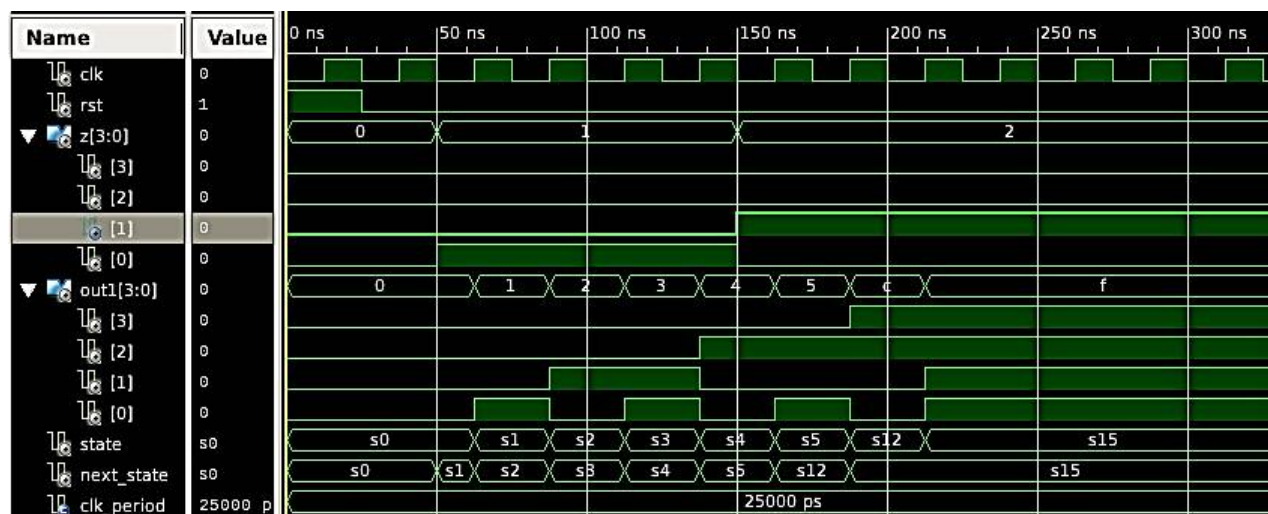


Рис. 3. Временная диаграмма работы управляющего автомата метода 2

Заключение

Описана методика кодирования управляющих автоматов для передачи трафика в ПКС. Методика кодирования управляющих автоматов основывается на цифровых автоматах Мура и алгоритмах ранней диагностики потерь трафика реального времени при передаче разнородного трафика в коммутаторе OpenFlow и передаче трафика реального времени с использованием планировщика и функцией контроля доставки в ПКС.

Разработаны управляющие автоматы в соответствии с методикой и цифровыми автоматами Мура. Проведено моделирование работы управляющих автоматов на ПЛИС семейства Spartan 6 с помощью временных диаграмм. Оценены аппаратные затраты на ПЛИС управляющих автоматов.

Спроектированные управляющие автоматы могут войти в состав телекоммуникационного оборудования, контроллеров или коммутаторов OpenFlow по средствам ПЛИС или ASIC для контроля и передачи трафика в ПКС с помощью управляющих сигналов и команд.

Литература

1. McKeown N., Anderson T., Balakrishnan H. Openflow: enabling innovation in campus networks // ACM SIGCOMM Computer Communication Review. 2008. Vol. 38. No. 2. pp. 69–74.
2. Advanced study of SDN/OpenFlow controllers/ A. Shalimov, D. Zuikov, D. Zimarina, V. Pashkov, R.

Smeliansky // Proceedings of the 9th Central & Eastern European Software Engineering Conference in Russia. ACM. Moscow, 2013. pp. 1–6. DOI - 10.1145/2556610.2556621.

3. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. 4-е изд. СПб.: Питер, 2010. 943 с.

4. Karakus M., Durresi A. Quality of service (QoS) in software defined networking (SDN): A survey // Journal of Network and Computer Applications. 2017. Vol. 80. pp. 200–218.

5. Maniu R. and Dumitru L.A. Exploring the possibilities of a self-regulating SDN controller // Scientific Bulletin «Mircea cel Batran» Naval Academy. 2015. Vol. 18. No. 1. pp. 58–61.

6. Maturing of OpenFlow and Software-Defined Networking Through Deployments/ M. Kobayashi, S. Seetharaman, G. Parulkar, G. Appenzeller, J. Little, J. Van Reijendam, N. McKeown // Computer Networks. 2014. Vol. 61. pp. 151–175.

7. Перепелкин Д.А., Бышов В.С. Балансировка потоков данных в программно-конфигурируемых сетях с обеспечением качества обслуживания сетевых сервисов // Радиотехника. 2016. № 11. С. 111–119.

8. Никишин К.И. Моделирование метода ранней диагностики потерь трафика реального времени в программно-конфигурируемых сетях на основе аппарата сетей Петри // Вестник Поволжского государственного технологического университета. Сер.: Радиотехнические и инфокоммуникационные системы. 2022. № 2 (54). С. 47–60. DOI - 10.25686/2306-2819.2022.2.4.

9. Никишин К.И. Разработка программного обеспечения для валидации сетевых стандартов // Вестник Воронежского государственного технического университета. 2022. Т. 18. № 6. С. 62–67. DOI – 10.36622/VSTU.2022.18.6.008.

10. Никишин К.И. Исследование передачи трафика в программно-конфигурируемой сети с использованием Cisco Packet Tracer // Вестник Воронежского государственного технического университета. 2022. Т. 18. № 5. С. 85–90. DOI – 10.36622/VSTU.2022.18.5.011.

11. Jensen K., Kristensen L.M. Coloured Petri Nets. Springer, 2009. 384 p.
Modelling and Validation of Concurrent Systems. Berlin:

Поступила 10.01.2023; принята к публикации 13.02.2023

Информация об авторах

Никишин Кирилл Игоревич – канд. техн. наук, доцент кафедры вычислительной техники, Пензенский государственный университет (440026, Россия, г. Пенза, ул. Красная, д. 40), e-mail: nkipnz@mail.ru

CODING METHODOLOGY OF CONTROL AUTOMATS FOR TRAFFIC TRANSMISSION IN A SOFTWARE DEFINED NETWORK

K.I. Nikishin

Penza State University, Penza, Russia

Abstract: software defined networks (SDN) are a new paradigm in the field of network telecommunications. The difference from classical networks is that the control panel controller interacts with the infrastructure layer at the control level. However, the classical method of transmitting heterogeneous traffic based on the OpenFlow protocol has a number of disadvantages. In order to eliminate the disadvantages of traffic transmission, new methods of traffic transmission, the methodology of complex control of traffic transmission in SDN, the coding methodology of control automats for traffic transmission in SDN have been directed. The coding methodology of control automats is based on Moore digital automats and algorithms for early diagnosis of real-time traffic losses during the transmission of heterogeneous traffic in the OpenFlow switch and the transmission of real-time traffic using a scheduler and a delivery control function in SDN. Control automats have been developed in accordance with the methodology and Moore's digital automats. The modelling of work of control automats on FPGAs of the Spartan 6 family using time diagrams is carried out. Hardware costs for FPGA control automats are estimated. The designed control automats can be part of telecommunications equipment, controllers or OpenFlow switches by means of a designed FPGA or ASIC for monitoring and transmitting traffic to the control system by means of control signals and commands

Key words: software defined network, OpenFlow, heterogeneous traffic, digital automata, Petri nets, control automat, FPGA

References

1. McKeown N., Anderson T., Balakrishnan H. "Openflow: enabling innovation in campus networks", *ACM SIGCOMM Computer Communication Review*, 2008, vol. 38, no. 2, pp. 69–74.
2. Shalimov A., Zuikov D., Zimarina D., Pashkov V., Smeliansky R. "Advanced study of SDN/OpenFlow controllers", *Proc. of the 9th Central & Eastern European Software Engineering Conf. in Russia. ACM*, Moscow, 2013, pp. 1–6. DOI - 10.1145/2556610.2556621.
3. Olifer V.G., Olifer N.A. "Computer networks. Principles, technologies, protocols" ("Komp'yuternye seti. Printsipy, tekhnologii, protokoly"), St. Petersburg: Piter, 2010, 943 p.
4. Karakus M., Durresi A. "Quality of service (QoS) in software defined networking (SDN)", a survey, *Journal of Network and Computer Applications*, 2017, vol. 80, pp. 200–218.
5. Maniu R., Dumitru L.A. "Exploring the possibilities of a self-regulating SDN controller", *Scientific Bulletin «Mircea cel Batran» Naval Academy*, 2015, vol. 18, no. 1, pp. 58–61.
6. Kobayashi M., Seetharaman S., Parulkar G., Appenzeller G., Little J., Van Reijndam J., McKeown N. "Maturing of OpenFlow and Software-Defined Networking Through Deployments", *Computer Networks*, 2014, vol. 61, pp. 151–175.
7. Perepelkin D.A., Byshov V.S. "Balancing data flows in software-defined networks with ensuring the quality of service of network services", *Radio Engineering (Radiotekhnika)*, 2016, no. 11, pp. 111–119.
8. Nikishin K.I. "Modeling a method for early diagnosis of real-time traffic losses in software-defined networks based on the apparatus of Petri nets", *Bulletin of Volga State Technological University (Vestnik Povolzhskogo gosudarstvennogo tekhnologicheskogo universiteta)*, 2022, no. 2 (54), pp. 47–60. DOI - 10.25686/2306-2819.2022.2.4.
9. Nikishin K.I. "Development of software for validation of network standards", *Bulletin of Voronezh State Technical University (Vestnik Voronezhskogo gosudarstvennogo tekhnicheskogo universiteta)*, 2022, vol. 18, no. 6, pp. 62–67. DOI – 10.36622/VSTU.2022.18.6.008.
10. Nikishin K.I. "Investigation of traffic transmission in a software-defined network using Cisco Packet Tracer", *Bulletin of Voronezh State Technical University (Vestnik Voronezhskogo gosudarstvennogo tekhnicheskogo universiteta)*, 2022, vol. 18, no. 5, pp. 85–90. DOI – 10.36622/VSTU.2022.18.5.011.
11. Jensen K., Kristensen L.M. "Coloured Petri Nets. Modelling and Validation of Concurrent Systems", Berlin: Springer, 2009, 384 p.

Submitted 10.01.2023; revised 13.02.2023

Information about the authors

Kirill I. Nikishin, Cand. Sc. (Technical), Associate Professor, Penza State University (40 Krasnaya str., Penza 440026, Russia), e-mail: nkipnz@mail.ru