УДК 004.72 Научная статья

https://doi.org/10.32603/2071-8985-2023-16-5-49-58

Метод комплексного контроля передачи трафика в программно-конфигурируемых сетях

К. И. Никишин

Пензенский государственный университет, Пенза, Россия
nkipnz@mail.ru

Аннотация. Современные требования, предъявляемые к обработке трафика, должны обладать свойствами мобильности, быстроты, упрощения администрирования сетевого оборудования, что не обеспечивалось в классических сетях Ethernet с поддержкой качества обслуживания. Все эти требования привели к появлению программно-конфигурируемых сетей (ПКС). В ПКС основным протоколом служит OpenFlow, который позволяет обрабатывать разнородный трафик и управлять им. Однако существующий в протоколе OpenFlow ПКС метод обладает рядом недостатков, для устранения которых и предназначен метод, разработанный в статье. Метод комплексного контроля передачи трафика в ПКС основывается на алгоритме ранней диагностики потерь трафика реального времени при передаче разнородного трафика в коммутаторе OpenFlow и алгоритме передачи трафика реального времени с использованием планировщика и функцией контроля доставки в ПКС. Разработаны цифровые автоматы Мура предложенных алгоритмов передачи, представлено описание систем канонических уравнений и выходных функций по цифровым автоматам. С помощью формализации разработанных цифровых автоматов и таблиц переходов было выполнено кодирование состояний цифровых автоматов и проведен синтез цифровых автоматов на основе D-триггеров. Формализация разработанного метода и алгоритмов передачи трафика в ПКС на основе цифровых автоматов позволила спроектировать управляющие автоматы и алгоритмы на ПЛИС типа FPGA семейства Spartan 6 и на более поздних семействах.

Ключевые слова: программно-конфигурируемые сети, OpenFlow, Ethernet, трафик реального времени, планировщик расписания, контроль доставки трафика, цифровые автоматы, таблицы переходов состояний, сети Петри, ПЛИС, CPN Tools

Для цитирования: Никишин К. И. Метод комплексного контроля передачи трафика в программно-конфигурируемых сетях // Изв. СПбГЭТУ «ЛЭТИ». 2023. Т. 16, № 5. С. 49–58. doi: 10.32603/2071-8985-2023-16-5-49-58.

Original article

Method of Complex Control of Traffic Transmission in Software Defined Networks

K. I. Nikishin

Penza State University, Penza, Russia

✉ nkipnz@mail.ru

Abstract. Modern requirements for traffic processing should possess the properties of mobility, speed, and simplification of network equipment administration. However, the existing classical Ethernet networks with Quality of Service failed to meet these requirements, which led to the emergence of software defined networks (SDN). In SDN, the main protocol is OpenFlow, which enables processing and managing heterogeneous traffic. This article aims to propose a methodology for eliminating the disadvantages inherent in the OpenFlow protocol. A method for complex control of traffic transmission in SDN was developed based on the algorithm of early diagnosis of real-time traffic losses during the transmission of heterogeneous traffic in the OpenFlow switch and the algorithm of real-time traffic transmission using a scheduler and a delivery control function in SDN. Moore digital automata of the proposed transmission algorithms were developed. A description of systems of

canonical equations and output functions for digital automata was presented. Tables of transitions of digital automata were constructed, states of digital automata were encoded, and digital automata were synthesized based on D-triggers. The formalization of the developed method and algorithms of traffic transmission in SDN based on digital automata made it possible to design control automata and algorithms on FPGA of the Spartan 6 family and later familie.

Keywords: Software Defined Networks, Openflow, Ethernet, real-time traffic, scheduler, delivery control of traffic, digital automata, state transition tables, Petri nets, FPGA, CPN Tools

For citation Nikishin K. I. Method of Complex Control of Traffic Transmission in Software Defined Networks // LETI Transactions on Electrical Engineering & Computer Science. 2023. Vol. 16, no. 5. P. 49–58. doi: 10.32603/2071-8985-2023-16-5-49-58.

Введение. Существующие сетевые компьютерные парадигмы основаны на технологии Ethernet [1], которая позволяет передавать разнородный трафик по сети. Со временем из-за усложнения телекоммуникационных алгоритмов и обработки трафика было введено понятие «качества обслуживания» (Quality of Service, QoS) [2], [3]. QoS позволяет передавать разнородный трафик в зависимости от его приоритета, т. е. в коммутаторе Ethernet происходит классификация трафика [4].

В число современных требований, предъявляемых к обработке трафика, входят мобильность, быстрота, упрощение администрирования сетевого оборудования, что не обеспечивалось существующими классическими сетями Ethernet с поддержкой QoS. Эти требования привели к появлению программно-конфигурируемых сетей (ПКС) [5]–[7].

В ПКС основным протоколом служит OpenFlow, он позволяет обрабатывать разнородный трафик и управлять им [8]. Однако существующий метод в протоколе OpenFlow ПКС обладает рядом недостатков.

Первый недостаток заключается в значительном увеличении времени поиска правила для кадра в таблицах потоков [9]. Это приводит к информированию контроллера [10] о необходимости удаления кадра из ПКС на более позднем этапе диагностики.

Следующий недостаток состоит в отсутствии быстрой передачи разнородного трафика, поскольку на поиск таймаутов потока тратятся значительные временные и аппаратные ресурсы. Еще один недостаток – отсутствие контроля функций доставки трафика на входных портах коммутаторов OpenFlow. Из-за этого затруднительно определить доставку трафика в реальном времени получателю в необходимые моменты времени.

Следующим недостаток – то, что возникает необходимость прерывания трафика в случае временного конфликта между различными видами трафика в коммутаторе.

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

Цель исследования – разработка метода комплексного контроля передачи трафика в ПКС.

Задачи исследования заключаются в разработке цифровых автоматов Мура предложенных алгоритмов передачи, в описании систем канонических уравнений и выходных функций по цифровым автоматам, в построении таблиц переходов цифровых автоматов, кодировании состояний цифровых автоматов и синтезе цифровых автоматов на основе *D*-триггеров.

Описание метода. Первый алгоритм устраняет часть отмеченных замечаний, обеспечивает раннюю диагностику потерь трафика реального времени, а также снижает задержку при повторной передаче трафика реального времени за счет раннего информирования контроллера ПКС [11]. Более подробное описание предложенного алгоритма и его моделирование представлены в [12], [13].

Эффективность второго алгоритма заключается не только в том, что остается постоянной загрузка коммутатора, но и также и в том, что появляется возможность досрочной передачи эластичного трафика на протяжении времени блокировки коммутатора трафиком реального времени. Таким образом, коммутатор не простаивает и ожидает времени наступления доставки кадра

реального времени, а может в этот же промежуток времени передавать кадр эластичного трафика, что позволяет повысить пропускную способность сети и снизить задержку. Эти показатели играют решающую роль при передаче данных по сети конечному пользователю. На основе предложенного алгоритма была разработана модель [14].

Полученные алгоритмы передачи трафика в ПКС позволяют получить метод комплексного контроля передачи трафика. Метод состоит из следующих этапов.

Этап 1. Состоит в настройке и установке разнородного трафика на основе различных характеристик. Далее трафик поступает в коммутатор ПКС. К таким характеристикам относятся длина межкадрового интервала, интенсивность поступления эластичного трафика, учет различных вариаций длин кадров и приоритетов, передача трафика реального времени с различными постоянными периодами.

При этом распределение длины поля данных эластичного трафика имеет ярко выраженный бимодальный характер. Таким образом, до 50 % всех кадров имеют минимальную или максимальную длину, длина остальных считается равномерно распределенной в диапазоне от минимальной до максимальной.

Этап 2. Администратор сети ПКС может настроить требуемый алгоритм передачи трафика с его контролем в ПКС или автоматически устанавливает соответствующий контроль аппаратурой сети.

Этап 3. Выбирается вид контроля передачи трафика в ПКС. Если требуется жесткий контроль на основе временных окон таймаутов и с защитником при передаче трафика реального времени и раннего оповещения контроллера об удаляемых кадрах, то осуществляется передача на основе алгоритма ранней диагностики потерь трафика реального времени.

Если требуется менее жесткая передача трафика во времени, то она основывается на основе гибкого расписания планировщика, а не жестких интервалов времени таймаутов. Кроме этого появляется возможность досрочной передачи эластичного трафика, пока не наступило время доставки трафика реального времени. Таким образом, осуществляется контроль на основе алгоритма передачи трафика реального времени с использованием планировщика и функцией контроля доставки. При этом для данного алгоритма администратор ПКС должен проводить рассылку и настройку расписания на уровне приложения ПКС через API-интерфейсы и функции сети, синхронизировать с каждым устройством расписание при помощи коммутатора.

Если требуется комплексный контроль передачи трафика на основе двух предложенных алгоритмов, то они последовательно включаются в работу ПКС. Первым включается в работу алгоритм на основе временных окон таймаутов для контроля таймаутов при передаче трафика. После успешного выполнения контроля первым методом осуществляется контроль на основе планировщика и функции контроля доставки для трафика реального времени. При этом также появляется возможность досрочной передачи эластичного трафика в сочетании с комплексным контролем предложенным методом.

Этап 4. Администратор ПКС или аппаратура сети, отвечающие за контроль передачи, при необходимости могут изменить выбранный метод передачи и принять соответствующее решение из-за ограничения времени передачи трафика, задержки в ПКС, загрузки коммутатора, очередей коммутатора, ограничения полосы пропускания канала в коммутаторе.

В качестве математического аппарата для формализации алгоритмов передачи трафика и разработанного метода были выбраны цифровые автоматы, поскольку они – промежуточное звено для исследования алгоритмов с помощью имитационного моделирования. От цифровых автоматов можно перейти к сетям Петри, более подробно описанным в [15].

Первое преимущество сетей Петри заключается в том, что появляется возможность динамического исследования модели (количество удаленных кадров из ПКС, задержка в ПКС, загрузка коммутаторов OpenFlow).

Следующее преимущество состоит в декомпозиции модели, чтобы она не была слишком громоздкой и ею можно было управлять не только на уровне одного большого узла. Не происходит быстрого разрастания дерева достижимости состояния, пространства состояний модели и синтеза самой модели, как с помощью аппарата цифровых автоматов. Поэтому, исходя из вышеописанного, аппарат сетей Петри наилучшим образом подходит для имитационного моделирования.

В качестве среды имитационного моделирования выбран пакет моделирования CPN Tools [16]. CPN Tools обладает всеми преимуществами

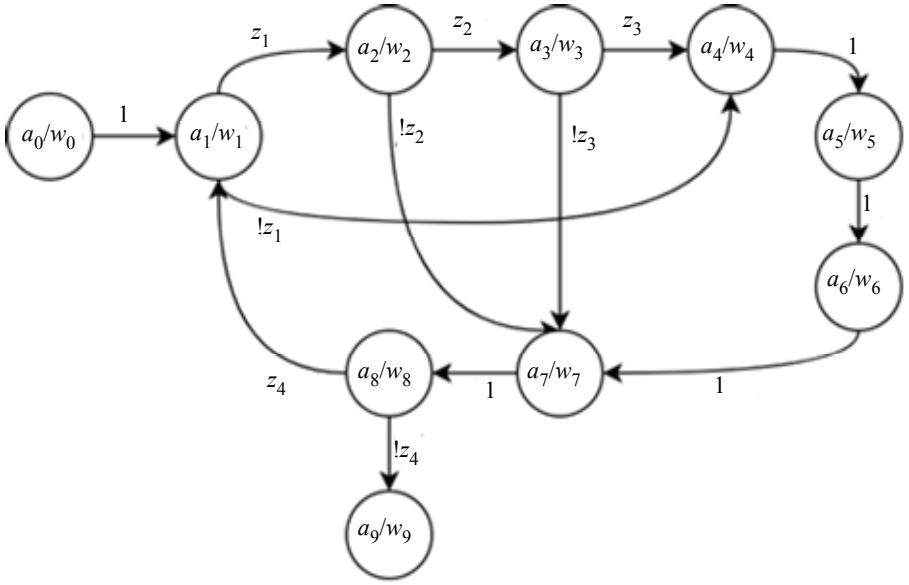

Рис. 1. Граф цифрового автомата Мура алгоритма 1
Fig. 1. Graph of the Moore digital automata of algorithm 1

аппарата сетей Петри, позволяет строить различные виды этих сетей, наилучшим образом подходит для исследования компьютерных сетей, их критериев, метрик и задержки в сети. Пакет обладает максимальной производительностью для исследования новых методов и алгоритмов в компьютерных сетях.

Кроме того, имеется встроенный язык программирования CPN ML, позволяющий строить сложные элементы, типы данных (структуры, списки, кортежи) и функции на сетях Петри, верифицировать модель на тупиковые состояния, свойство живости и анализ пространства состояний.

По описанным методу и алгоритму ранней диагностики потерь трафика реального времени был разработан цифровой автомат Мура, который 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 напишем систему канонических уравнений (СКУ) и систему выходных функций (СВФ). СКУ является аналитической интерпретацией таблиц переходов автоматов и определяет функции переходов цифрового автомата. СВФ – аналитической интерпретацией таблиц выходов автоматов, определяет функции выходов цифрового автомата.

СКУ:
$a_1(t+1) = a_7 z_4$;
$a_2(t+1) = a_1 z_1$;
$a_3(t+1) = a_2 z_2$;

СВФ:
$w_1(t) = a_1$;
$w_2(t) = a_2$;
$w_3(t) = a_3$;

Табл. 1. Прямая таблица переходов цифрового автомата Мура
Tab. 1. Direct transition table of the Moore digital automata

Начальное состояние a_m	Конечное состояние a_s	Выходной сигнал w_{a_s}	Входной сигнал $z_{a_m a_s}$
a_1	a_2	w_2	z_1
a_1	a_4	w_4	$!z_1$
a_2	a_3	w_3	z_2
a_2	a_7	w_7	$!z_2$
a_3	a_4	w_4	z_3
a_3	a_7	w_7	$!z_3$
a_4	a_5	w_5	1
a_5	a_6	w_6	1
a_6	a_7	w_7	1
a_7	a_8	w_8	1
a_8	a_1	w_1	z_4
a_8	a_9	w_9	$!z_4$

Табл. 2. Отмеченная таблица переходов цифрового автомата Мура
Tab. 2. Marked transition table of the Moore digital automata

Входной сигнал	Выходной сигнал								
	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9
	Состояния								
	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9
1	–	–	–	a_5	a_6	a_7	a_8	–	–
z_1	a_2	–	–	–	–	–	–	–	–
$!z_1$	a_4	–	–	–	–	–	–	–	–
z_2	–	a_3	–	–	–	–	–	–	–
$!z_2$	–	a_7	–	–	–	–	–	–	–
z_3	–	–	a_4	–	–	–	–	–	–
$!z_3$	–	–	a_7	–	–	–	–	–	–
z_4	–	–	–	–	–	–	–	a_1	–
$!z_4$	–	–	–	–	–	–	–	a_9	–

$a_4(t+1) = a_3 z_3 \vee a_1(!z_1)$; $w_4(t) = a_4$;
$a_5(t+1) = a_4$; $w_5(t) = a_5$;
$a_6(t+1) = a_5$; $w_6(t) = a_6$;
$a_7(t+1) = a_6 \vee$ $w_7(t) = a_7$;
$\vee\, a_3(!z_3) \vee a_2(!z_2)$;
$a_8(t+1) = a_7$; $w_8(t) = a_8$;
$a_9(t+1) = a_8$. $w_9(t) = a_9$.

По СКУ и СВФ была построена прямая таблица переходов цифрового автомата Мура алгоритма 1 (табл. 1).

Построим отмеченную таблицу переходов цифрового автомата Мура (табл. 2) и попытаемся минимизировать автомат. При минимизации полностью определенных цифровых автоматов Мура вводится понятие 0-эквивалентности состояний и разбиение множества состояний на 0-эквивалентные классы. К 0-эквивалентным относятся одинаково отмеченные состояния.

Если два состояния цифрового автомата Мура 0-эквивалентны и под действием одинаковых входных сигналов попадают в 0-эквивалентные состояния, то они называются 1-эквивалентными, и т. д. Видно, что в заданном цифровом автомате отсутствуют 0-эквивалентные состояния, а значит, мы имеем минимальный автомат.

Был также разработан цифровой автомат Мура алгоритма 2, который состоит из большего количества состояний и переходов. Поэтому рассмотрим только СКУ и СВФ разработанного автомата:

СКУ:

$a_1(t+1) = 0$;
$a_2(t+1) = a_1$;
$a_3(t+1) = a_2 \vee (a_{10} \vee a_{11} \vee a_{e1})(z_5 z_6 z_9) \vee$
$\vee\, (a_{12} \vee a_{13} \vee a_{14})\, z_9$;
$a_4(t+1) = a_3$;
$a_5(t+1) = a_4$;
$a_6(t+1) = a_5$;
$a_7(t+1) = a_6$;
$a_8(t+1) = a_7$;
$a_9(t+1) = a_8$;
$a_{10}(t+1) = a_8$;
$a_{11}(t+1) = a_9$;
$a_{12}(t+1) = a_5(!z_0) \vee a_9 z_2$;
$a_{13}(t+1) = a_{e0}(!z_4) \vee a_9(!z_2\ !z_3\ z_4) \vee a_{e1}(z_5 z_6 z_7)$;
$a_{14}(t+1) = (a_{10} \vee a_{11} \vee a_{e1})(z_5 z_6 z_9)$;
$a_{15}(t+1) = (a_{10} \vee a_{11} \vee a_{e1})(z_5 z_6\ !z_9) \vee$
$\vee\, (a_{12} \vee a_{13} \vee a_{14})(!z_9)$.

СВФ:

$w_1(t) = a_1$;
$w_2(t) = a_2$;
$w_3(t) = a_3$;
$w_4(t) = a_4$;
$w_5(t) = a_5$;
$w_6(t) = a_6$;
$w_7(t) = a_7$;
$w_8(t) = a_8$;
$w_9(t) = a_9$;
$w_{10}(t) = a_{10}$;
$w_{11}(t) = a_{11}$;
$w_{12}(t) = a_{12}$;
$w_{13}(t) = a_{13}$;
$w_{14}(t) = a_{14}$.

Табл. 3. Отмеченная таблица переходов цифрового автомата Мура
Tab. 3. Marked transition table of the Moore digital automata

Входной сигнал	Выходной сигнал																
	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_e	w_{10}	w_{11}	w_e	w_{12}	w_{13}	w_{14}	w_{15}
	Состояния																
	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{e0}	a_{10}	a_{11}	a_{e1}	a_{12}	a_{13}	a_{14}	a_{15}
1	a_2	a_3	a_4	a_5	–	a_7	a_8	–	–	–	–	–	–	–	–	–	–
z_0	–	–	–	–	a_6	–	–	–	–	–	–	–	–	–	–	–	–
$!z_0$	–	–	–	–	a_{12}	–	–	–	–	–	–	–	–	–	–	–	–
z_1	–	–	–	–	–	–	–	a_9	–	–	–	–	–	–	–	–	–
$!z_1$	–	–	–	–	–	–	–	a_{10}	–	–	–	–	–	–	–	–	–
z_2	–	–	–	–	–	–	–	–	a_{12}	–	–	–	–	–	–	–	–
$!z_2\, z_3$	–	–	–	–	–	–	–	–	a_{11}	–	–	–	–	–	–	–	–
$!z_2\, !z_3\, z_4$	–	–	–	–	–	–	–	–	a_{13}	–	–	–	–	–	–	–	–
$!z_2\, !z_3\, !z_4$	–	–	–	–	–	–	–	–	a_{e0}	–	–	–	–	–	–	–	–
z_4	–	–	–	–	–	–	–	–	–	a_{13}	–	–	–	–	–	–	–
$!z_4$	–	–	–	–	–	–	–	–	–	a_{e0}	–	–	–	–	–	–	–
$!z_5$	–	–	–	–	–	–	–	–	–	–	a_{e1}	a_{e1}	a_{e1}	–	–	–	–
$z_5\, z_6\, z_9$	–	–	–	–	–	–	–	–	–	–	a_3	a_3	a_3	–	–	–	–
$z_5\, z_6 !z_9$	–	–	–	–	–	–	–	–	–	–	a_{15}	a_{15}	a_{15}	–	–	–	–
$z_5\, z_6\, z_7$	–	–	–	–	–	–	–	–	–	–	a_{13}	a_{13}	a_{13}	–	–	–	–
$z_5\, z_6\, !z_7\, z_8$	–	–	–	–	–	–	–	–	–	–	a_{14}	a_{14}	a_{14}	–	–	–	–
$z_5\, z_6\, !z_7\, !z_8$	–	–	–	–	–	–	–	–	–	–	a_{13}	a_{13}	a_{13}	–	–	–	–
z_9	–	–	–	–	–	–	–	–	–	–	–	–	–	a_3	a_3	a_3	–
$!z_9$	–	–	–	–	–	–	–	–	–	–	–	–	–	a_{15}	a_{15}	a_{15}	–

Построим отмеченную таблицу переходов цифрового автомата Мура алгоритма 2 (табл. 3) и попытаемся минимизировать автомат.

В данном цифровом автомате Мура также отсутствуют 0-эквивалентные состояния, следовательно, он также минимальный.

Результаты формализации метода. С помощью формализации разработанных цифровых автоматов и таблиц переходов было выполнено кодирование состояний цифровых автоматов и проведен синтез цифровых автоматов на основе *D*-триггеров.

Табл. 4. Кодирование состояний автомата
Tab. 4. Encoding of automaton states

Состояние	Цифровой код			
A	Q_1	Q_2	Q_3	Q_4
a_1	0	0	0	0
a_2	0	0	0	1
a_3	0	0	1	1
a_4	0	0	1	0
a_5	0	1	1	0
a_6	0	1	0	0
a_7	0	1	0	1
a_8	0	1	1	1
a_9	1	0	0	0

Так как у абстрактного автомата *S* алгоритма 1–9 состояний, то структурный автомат *S* будет иметь 4 элемента памяти D_1–D_4; четыре абстрактных входных (z_1–z_4) и 9 выходных (w_1–w_9) сигналов. Кодирование состояний автомата *S* представлено в табл. 4.

Поскольку для синтеза цифрового автомата в качестве элементов памяти будут использоваться *D*-триггеры, в число функций табл. 5 записываем только D_i, соответствующие триггерам, которые необходимо установить в состояние «1», чтобы обеспечить переход в состояние с кодом K_{a_s}.

По табл. 5 запишем кодирование сигналов и состояний, необходимых для синтеза цифрового автомата на *D*-триггерах:

$D_1 = !z_4\, !Q_1\, Q_2\, Q_3\, Q_4$

$D_2 = !z_2\, !Q_1\, !Q_2\, !Q_3\, Q_4 \vee !z_3\, !Q_1\, !Q_2\, Q_3\, Q_4 \vee !Q_1\, !Q_2\, Q_3\, !Q_4 \vee !Q1\, Q_2\, Q_3\, !Q_4 \vee !Q_1\, Q_2\, !Q_3\, !Q_4 \vee !Q_1\, Q_2\, !Q_3\, Q_4$

$D_3 = !z_1\, !Q_1\, !Q_2\, !Q_3\, !Q_4 \vee z_2\, !Q1\, !Q_2\, !Q_3\, Q_4 \vee z_3\, !Q_1\, !Q_2\, Q_3\, Q_4 \vee !Q_1\, !Q_2\, Q_3\, !Q_4\, v\, !Q_1\, Q_2\, !Q_3\, Q_4$

$D_4 = z_1\, !Q_1\, !Q_2\, !Q_3\, !Q_4 \vee z_2\, !Q_1\, !Q_2\, !Q_3\, Q_4 \vee !z_2\, !Q_1\, !Q_2\, !Q_3\, Q_4 \vee !z_3\, !Q_1\, !Q_2\, Q_3\, Q_4 \vee !Q_1\, Q_2\, !Q_3\, !Q_4 \vee !Q_1\, Q_2\, !Q_3\, Q_4$

Табл. 5. Обратная таблица переходов цифрового автомата Мура
Tab. 5. Reverse transition table of the Moore digital automata

Начальное состояние a_m	Начальный код Q_1–Q_4	Конечное состояние a_s	Выходной сигнал w_{a_s}	Конечный код Q_1–Q_4	Входной сигнал $z_{a_m a_s}$	Функция F
a_1	0000	a_2	w_2	0001	z_1	D_4
a_1	0000	a_4	w_4	0010	$!z$	D_3
a_2	0001	a_3	w_3	0011	z_2	D_3, D_4
a_2	0001	a_7	w_7	0101	$!z_2$	D_2, D_4
a_3	0011	a_4	w_4	0010	z_3	D_3
a_3	0011	a_7	w_7	0101	$!z_3$	D_2, D_4
a_4	0010	a_5	w_5	0110	1	D_2, D_3
a_5	0110	a_6	w_6	0100	1	D_2
a_6	0100	a_7	w_7	0101	1	D_2, D_4
a_7	0101	a_8	w_8	0111	1	D_2–D_4
a_8	0111	a_1	w_1	0000	z_4	–
a_8	0111	a_9	w_9	1000	$!z_4$	D_1

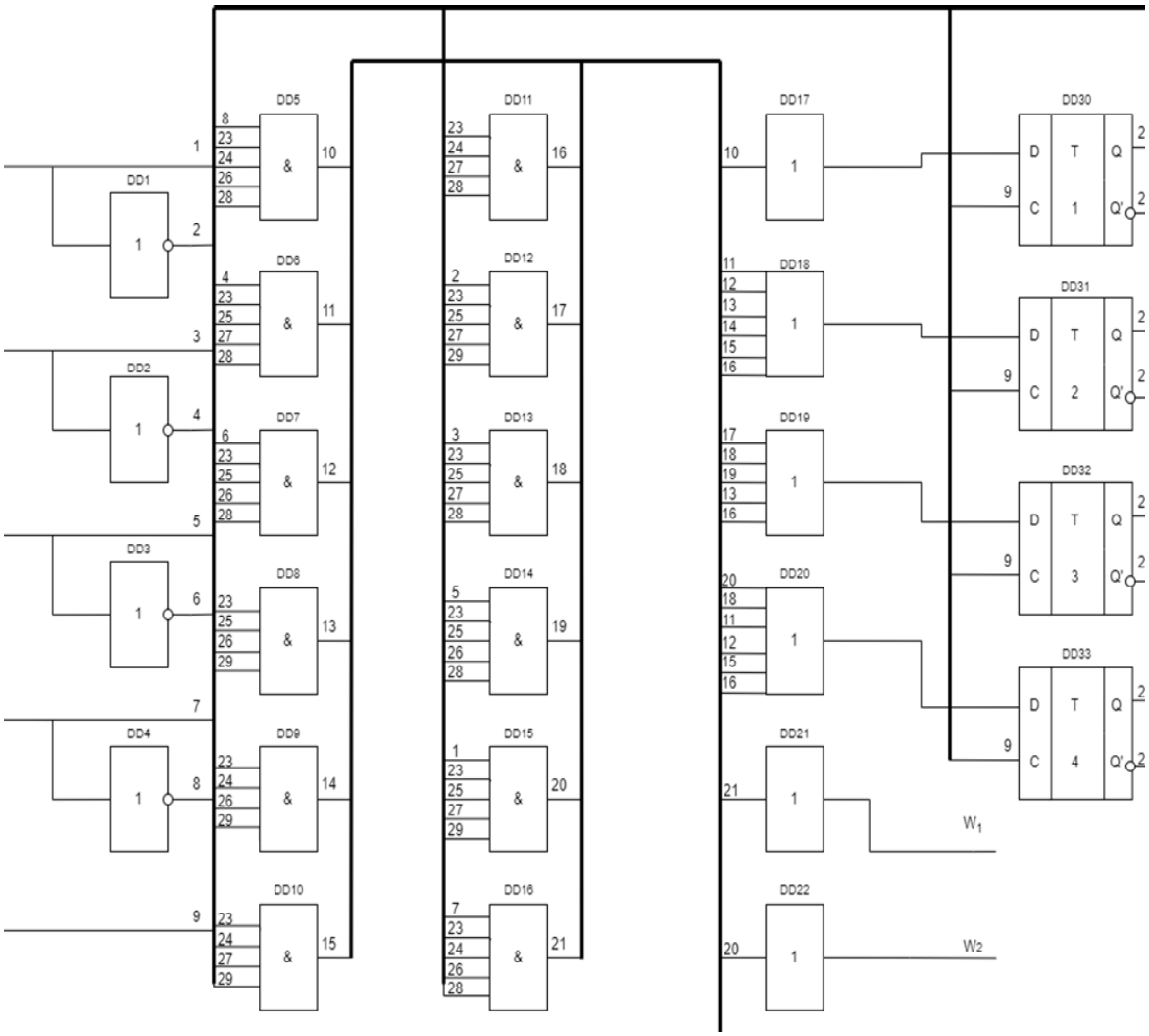

Рис. 2. Синтез цифрового автомата Мура алгоритма 1 на *D*-триггерах
Fig. 2. Synthesis of the Moore digital automata of algorithm 1 on *D*-triggers

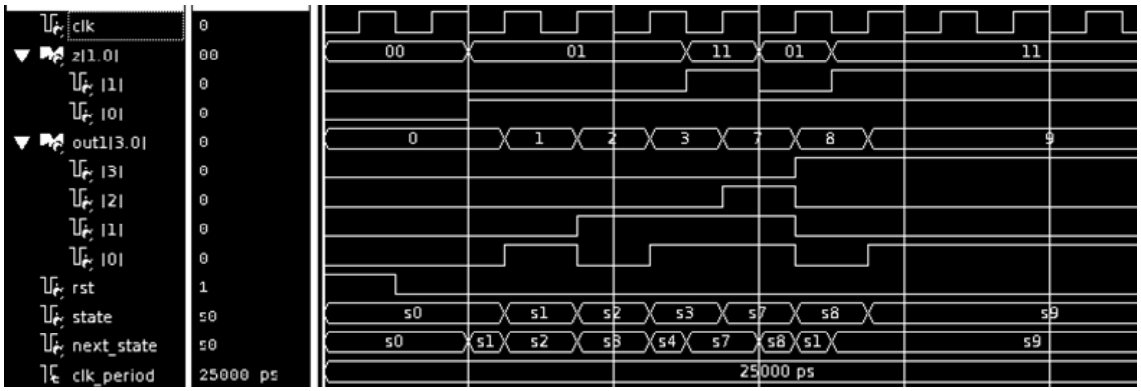

Рис. 3. Моделирование управляющих автоматов на ПЛИС в среде Xilinx ISE
Fig. 3. Simulation of control automata on FPGA in Xilinx ISE environment

$w_1 = z_4\ !Q_1\ Q_2\ Q_3\ Q_4$
$w_2 = z_1\ !Q_1\ !Q_2\ !Q_3\ !Q_4$
$w_3 = z_2\ !Q_1\ !Q_2\ !Q_3\ Q_4$
$w_4 = !z_1\ !Q_1\ !Q_2\ !Q_3\ !Q_4 \vee z_3\ !Q_1\ !Q_2\ Q_3\ Q_4$
$w_5 = !Q_1\ !Q_2\ Q_3\ !Q_4$
$w_6 = !Q_1\ Q_2\ Q_3\ !Q_4$
$w_7 = !z_2\ !Q_1\ !Q_2\ !Q_3\ Q_4 \vee !z_3\ !Q_1\ !Q_2\ Q_3\ Q_4 \vee !Q_1\ Q_2\ !Q_3\ !Q_4$
$w_8 = !Q_1\ Q_2\ !Q_3\ Q_4$
$w_9 = !z_4\ !Q_1\ Q_2\ Q_3\ Q_4$

Так как у абстрактного алгоритма автомата S 2–17 состояний, то структурный автомат S будет иметь 5 элементов памяти D_1–D_5; девять абстрактных входных (z_1–z_9) и 16 выходных (w_1–w_e) сигналов. Синтез цифрового автомата на элементах памяти D-триггеров для алгоритма 2 строится так же, как для цифрового автомата алгоритма 1.

На рис. 2 представлена часть спроектированного цифрового автомата Мура алгоритма 1 на элементах памяти D-триггерах.

Формализация разработанного метода и алгоритмов передачи трафика в ПКС на основе цифровых автоматов позволила спроектировать управляющие автоматы и алгоритмы на ПЛИС типа FPGA семейства Spartan 6 и на более поздних семействах. Результат моделирования метода на ПЛИС представлен на рис. 3.

Заключение. Разработан метод комплексного контроля передачи трафика в ПКС, основанный на алгоритме ранней диагностики потерь трафика реального времени при передаче разнородного трафика в коммутаторе OpenFlow и алгоритме передачи трафика реального времени с использованием планировщика и функцией контроля доставки в ПКС.

Разработаны цифровые автоматы Мура предложенного метода передачи, представлено описание систем канонических уравнений и выходных функций по цифровым автоматам, выполнено построение таблиц переходов цифровых автоматов, кодирование состояний цифровых автоматов и синтез цифровых автоматов на основе D-триггеров.

Формализация разработанного метода и алгоритмов передачи трафика в ПКС на основе цифровых автоматов позволила спроектировать управляющие автоматы и алгоритмы на ПЛИС типа FPGA семейства Spartan 6 и на более поздних семействах.

Список литературы

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

2. Karakus M., Durresi A. Quality of service (QoS) in software defined networking (SDN): A survey // J. of Network and Comp. Appl. 2017. Vol. 80. P. 200–218.

3. Scheduling queues in the Ethernet switch, considering the waiting time of frames / E. Kizilov, N. Konnov, K. Nikishin, D. Pashchenko, D. Trokoz // MATEC Web of Conf. 2016. Vol. 44. P. 01011-p.1–01011-p. 5.

4. Никишин К. И. Механизм управления трафиком реального времени в коммутаторе Ethernet // Вестн. компьютерных и информационных технологий. 2015. № 10. С. 32–37.

5. McKeown N., Anderson T., Balakrishnan H. Openflow: enabling innovation in campus networks // ACM SIGCOMM Comp. Communication Rev. 2008. Vol. 38, no. 2. P. 69–74.

6. Advanced study of SDN/OpenFlow controllers / A. Shalimov, D. Zuikov, D. Zimarina, V. Pashkov, R. Smeliansky // Proc. of the 9th Central & Eastern European Software Engin. Conf. in Russia. ACM. Moscow, 2013. P. 1–6. doi: 10.1145/2556610.2556621.

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. Maturing of OpenFlow and Software-Defined Networking through deployments / M. Kobayashi, S. Seetharaman, G. Parulkar, G. Appenzeller, J. Little, J. Van Reijendam, N. McKeown // Comp. Networks. 2014. Vol. 61. P. 151–175.

9. Никишин К. И. Исследование и моделирование таблицы потоков коммутатора Openflow в программно-конфигурируемых сетях // Вестн. Рязанского гос. радиотехнического ун-та. 2022. № 81. С. 42–50. doi: 10.21667/1995-4565-2022-81-42-50.

10. 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. P. 58–61.

11. Никишин К. И. Моделирование контроллера и верификация процесса передачи данных в программно-конфигурируемых сетях // Вестн. Рязанского гос. радиотехнического ун-та. 2022. № 80. С. 75–83. doi: 10.21667/1995-4565-2022-80-75-83.

12. Никишин К. И. Метод ранней диагностики потерь трафика реального времени с контролем таймаутов в программно-конфигурируемых сетях // Изв. Юго-Западного гос. ун-та. 2022. Т. 26. № 2. С. 142–158. doi: 10.21869/2223-1560-2022-26-2-142-158.

13. Никишин К. И. Моделирование метода ранней диагностики потерь трафика реального времени в программно-конфигурируемых сетях на основе аппарата сетей Петри // Вестн. Поволжского гос. технол. ун-та. Сер.: Радиотехнические и инфокоммуникационные системы. 2022. № 2 (54). С. 47–60. doi: 10.25686/2306-2819.2022.2.4.

14. Никишин К. И. Моделирование процесса передачи трафика реального времени с использованием планировщика и функцией контроля доставки в программно-конфигурируемых сетях // Изв. СПбГЭТУ «ЛЭТИ». 2023. Т. 16. № 1. С. 53–65. doi: 10.32603/2071-8985-2023-16-1-53-65.

15. Никишин К. И. Автоматный подход к построению сетей Петри // Сб. науч. ст. XVII Междунар. науч.-техн. конф. «Новые информационные технологии и системы». Пенза: Изд-во ПГУ. 2020. С. 80–82.

16. Jensen K., Kristensen L. M. Coloured Petri nets. Modelling and validation of concurrent systems. Berlin: Springer, 2009. 384 p.

Информация об авторе

Никишин Кирилл Игоревич – канд. техн. наук, доцент, Пензенский государственный университет, ул. Красная, 40, Пенза, 440026, Россия.
E-mail: nkipnz@mail.ru
https://orcid.org/0000-0001-7966-7833

References

1. Olifer V. G., Olifer N. A. Komp'juternye seti. Principy, tehnologii, protokoly. 4-e izd. SPb.: Piter, 2010. 943 s. (In Russ.).

2. Karakus M., Durresi A. Quality of service (QoS) in software defined networking (SDN): A survey // J. of Network and Comp. Appl. 2017. Vol. 80. P. 200–218.

3. Scheduling queues in the Ethernet switch, considering the waiting time of frames / E. Kizilov, N. Konnov, K. Nikishin, D. Pashchenko, D. Trokoz // MATEC Web of Conf. 2016. Vol. 44. P. 01011-p.1–01011-p. 5.

4. Nikishin K. I. Mehanizm upravlenija trafikom real'nogo vremeni v kommutatore Ethernet // Vestn. komp'juternyh i informacionnyh tehnologij. 2015. № 10. S. 32–37. (In Russ.).

5. McKeown N., Anderson T., Balakrishnan H. Openflow: enabling innovation in campus networks // ACM SIGCOMM Comp. Communication Rev. 2008. Vol. 38, no. 2. P. 69–74. (In Russ.).

6. Advanced study of SDN/OpenFlow controllers / A. Shalimov, D. Zuikov, D. Zimarina, V. Pashkov, R. Smeliansky // Proc. of the 9th Central & Eastern European Software Engin. Conf. in Russia. ACM. Moscow, 2013. P. 1–6. doi: 10.1145/2556610.2556621.

7. Perepelkin D. A., Byshov V. S. Balansirovka potokov dannyh v programmno-konfiguriruemyh setjah s obespecheniem kachestva obsluzhivanija setevyh servisov // Radiotehnika. 2016. № 11. C. 111–119.

8. Maturing of OpenFlow and Software-Defined Networking through deployments / M. Kobayashi, S. Seetharaman, G. Parulkar, G. Appenzeller, J. Little, J. Van Reijendam, N. McKeown // Comp. Networks. 2014. Vol. 61. P. 151–175. (In Russ.).

9. Nikishin K. I. Issledovanie i modelirovanie tablicy potokov kommutatora Openflow v programmno-konfiguriruemyh setjah // Vestn. Rjazanskogo gos. radiotehnicheskogo un-ta. 2022. № 81. S. 42–50. doi: 10.21667/1995-4565-2022-81-42-50. (In Russ.).

10. 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. P. 58–61.

11. Nikishin K. I. Modelirovanie kontrollera i verifikacija processa peredachi dannyh v programmno-konfiguriruemyh setjah // Vestn. Rjazanskogo gos. radiotehnicheskogo un-ta. 2022. № 80. S. 75–83. doi: 10.21667/1995-4565-2022-80-75-83. (In Russ.).

12. Nikishin K. I. Metod rannej diagnostiki poter' trafika real'nogo vremeni s kontrolem tajmautov v programmno-konfiguriruemyh setjah // Izv. Jugo-Zapadnogo gos. un-ta. 2022. T. 26. № 2. S. 142–158. doi: 10.21869/2223-1560-2022-26-2-142-158. (In Russ.).

13. Nikishin K. I. Modelirovanie metoda rannej diagnostiki poter' trafika real'nogo vremeni v programmno-konfiguriruemyh setjah na osnove apparata setej Petri // Vestn. Povolzhskogo gos. tehnol. un-ta. Ser.: Radiotehnicheskie i infokommunikacionnye sistemy. 2022. № 2 (54). S. 47–60. doi: 10.25686/2306-2819.2022.2.4. (In Russ.).

14. Nikishin K. I. Modelirovanie processa peredachi trafika real'nogo vremeni s ispol'zovaniem planirovshhika i funkciej kontrolja dostavki v programmno-konfiguriruemyh setjah // Izv. SPbGJeTU «LETI». 2023. T. 16. № 1. S. 53–65. doi: 10.32603/2071-8985-2023-16-1-53-65. (In Russ.).

15. Nikishin K. I. Avtomatnyj podhod k postroeniju setej Petri // Sb. nauch. st. HVII Mezhdunar. nauch.-tehn. konf. «Novye informacionnye tehnologii i sistemy». Penza: Izd-vo PGU. 2020. S. 80–82.

16. Jensen K., Kristensen L. M. Coloured Petri nets. Modelling and validation of concurrent systems. Berlin: Springer, 2009. 384 p.

Information about the author

Kirill I. Nikishin – Cand. Sci. (Eng.), Associate Professor of Penza State University, Krasnaya st., 40, Penza, 440026, Russia.
E-mail: nkipnz@mail.ru
http://orcid.org/0000-0001-7966-7833

Статья поступила в редакцию 23.03.2023; принята к публикации после рецензирования 31.03.2023; опубликована онлайн 25.05.2023.

Submitted 23.03.2023; accepted 31.03.2023; published online 25.05.2023.