

УДК 681.31

ИССЛЕДОВАНИЕ И МОДЕЛИРОВАНИЕ ТАБЛИЦЫ ПОТОКОВ КОММУТАТОРА OPENFLOW В ПРОГРАММНО-КОНФИГУРИРУЕМЫХ СЕТЯХ

К. И. Никишин, к.т.н., старший преподаватель кафедры ВТ ПГУ, Пенза, Россия;
orcid.org/0000-0001-7966-7833, e-mail: nkipnz@mail.ru

*Современные сети становятся очень громоздкими, требуется быстрота передачи трафика по сети и для этих целей были введены программно-конфигурируемые сети (ПКС). Основным протоколом передачи данных в ПКС является OpenFlow. Рассмотрено распределение управления и данных в ПКС, описана передача данных от контроллера через OpenFlow. **Цель работы** – исследование и изучение протокола и коммутатора OpenFlow, таблицы потоков для корректной передачи кадров в ПКС на основе цветных временных иерархических сетей Петри и использованием пакета моделирования CPN Tools. Задачами исследования являются рассмотрение различных форматов кадров и таблиц потоков, разработка алгоритмы функционирования таблицы потоков и модернизация временных характеристик передачи кадров через данные таблицы с учетом заданных тайм-аутов. Модели позволили исследовать функционирование таблицы потоков и их временные характеристики в коммутаторе OpenFlow для дальнейшего изучения передачи разнородного гетерогенного трафика в ПКС.*

Ключевые слова: программно-конфигурируемые сети, контроллер, коммутатор, Ethernet, OpenFlow, таблицы потоков, тайм-ауты, сети Петри, CPN Tools.

DOI: 10.21667/1995-4565-2022-81-42-50

Введение

Сетевые технологии существуют давно в нашем мире и стали неотъемлемой его частью. Классической технологией передачи данных по сети была технология Ethernet. В дальнейшем были разработаны различные стандарты по обработке трафика в коммутаторе, это достигается за введения качества обслуживания [1].

Однако современные сети становятся очень громоздкими, требуется быстрота передачи трафика по сети, и для этих целей были введены ПКС. Таким образом, значительно упрощается логика управления коммутаторами в сети за счет разграничения управляющей логики и данных. Основным протоколом передачи данных от контроллера к сетевым устройствам и наоборот является открытый стандарт OpenFlow [2].

Теоретическая часть

Стандарт OpenFlow добавлен в современные коммерческие коммутаторы Ethernet, маршрутизаторы, беспроводные устройства, что позволяет создавать сети с самой разнообразной топологией и работой. В протоколе OpenFlow содержится управляющая логика пересылки трафика на уровне управления в ПКС, а данные непосредственно передаются сетевым устройствам, и они работают только с данными (рисунок 1) [3].

Таким образом, отдельно реализуются логика и работа с данными в ПКС. Управляющая логика контролируется контроллером, а сетевые устройства, коммутаторы отвечают за пересылку трафика.

Передача данных использует специальные таблицы, которые называются таблицами потоков Flow Table. Таблица содержит определенный перечень полей, по которым маскированные данные проверяются на соответствие. Протокол OpenFlow по таблице определяет действие с пакетом: направить непосредственно в определенный порт или в очередь, или

удалить пакет, или обратно переслать контроллеру. Канал связи, с которым общается контроллер по протоколу OpenFlow, является защищенным, как на рисунке 2.

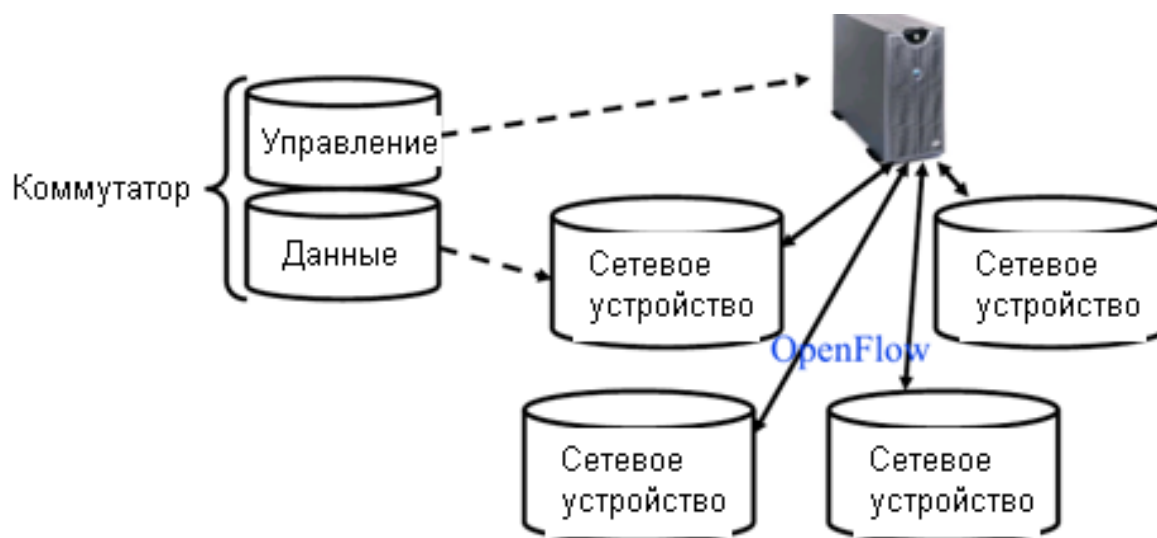


Рисунок 1 – Распределение управления и данных в OpenFlow

Figure 1 – Distribution of control and data in OpenFlow

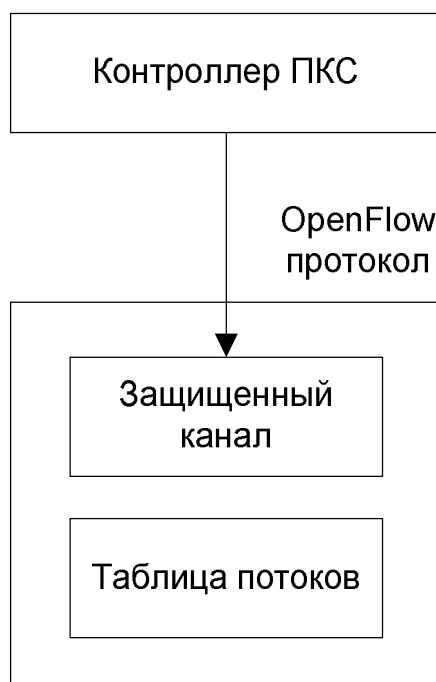


Рисунок 2 – Передача данных от контроллера через OpenFlow

Figure 2 – Data transmission from controller via OpenFlow

Коммутатор OpenFlow может состоять из нескольких таблиц потоков и групповой таблицы. По правилам пакета из этих таблиц осуществляется пересылка данных сетевым устройствам. На входной порт коммутатора OpenFlow поступает пакет, данный пакет сравнивается с таблицами потоков от 1 до N, после выполнения операции сравнения с одной из таблиц выполняется инструкция из таблицы, которая соответствует данному пакету, и принимается решение о передаче пакета в выходной порт, очередь, удалить. Алгоритм обработки пакета с таблицами потоков представлен на рисунке 3.



Рисунок 3 – Коммутатор OpenFlow

Figure 3 – OpenFlow switch

Таблицы пронумерованы с 0, и сравнение начинается с этой таблицы и далее. Если правило в таблице не найдено для пакета, такой пакет отправляется к следующей таблице через команду goto. Формат таблицы потока представлен на рисунке 4.

Поля соответствия	Приоритет	Счетчики	Инструкции	Таймауты	Cookie	Флаги
-------------------	-----------	----------	------------	----------	--------	-------

Рисунок 4 – Формат таблицы потоков

Figure 4 – Flow Table format

Выделяют следующие поля:

- поля соответствия для сравнения с пакетом, содержатся входящий порт и заголовок пакета с метаданными;
- приоритет;
- счетчики обновляются при сравнении пакета и регистрируют число пакетов, количество байт, удовлетворяющих условиям (таблица 1);
- инструкции выполняют определенное действие с пакетом. Выделяют следующие инструкции: «Output port_number» – передача на порт, «Group id» – передача пакета через групповую таблицу для дальнейшего сравнения, «Drop» – удалить пакет, «Set-Queue queue_id» – передача пакета в очередь коммутатора, дальнейшая отправка пакета решается механизмами и алгоритмами качества обслуживания QoS, «Meter meter_id» – получение метрик в сети;
- тайм-ауты – максимальное время, за которое может быть передан пакет устройству. Выделяют Idle timeout – удалить запись в таблице и пакет, если он не достигнет приемной стороны и Hard timeout – удалить принудительно запись в таблице и кадр, когда будет достигнуто заданное время;
- Cookie и флаги для управления таблицей потока.

Поля соответствия в таблице или заголовок пакета содержат следующие поля, как на рисунке 5: входящий порт, MAC адреса источника и приемника, VLAN ID – индекс виртуальной связи, приоритет пакета, IP адреса источника и приемника, IP ToS, порты источника и приемника на транспортном уровне согласно модели OSI.

Входящий порт	MAC адрес источника	MAC адрес приемника	VLAN ID	Приоритет	IP адрес источника	IP адрес приемника	IP протокол	IP ToS	Порт источника трансп. уровень	Порт приемника трансп. уровень
---------------	---------------------	---------------------	---------	-----------	--------------------	--------------------	-------------	--------	--------------------------------	--------------------------------

Рисунок 5 – Формат полей соответствия в таблице потоков

Figure 5 – Match field format in Flow Table

Таблица 1 – Пример счетчиков

Table 1 – Count Example

Счетчик	Число
Таблица потока	
Число пришедших пакетов	32
Число прошедших пакетов	32
На порт/ в очередь	
Число переданных пакетов	30
Число переданных байт	244
Число ошибок	2
Групповая таблица	
Число пришедших пакетов	12
Число прошедших пакетов	12

Пример таблицы потоков представлен в таблице 2, большинство полей из заголовка пакета маскируются, и определение происходит чаще всего по MAC или IP адресам приемника, IP протоколу и номерам портов транспортного уровня.

Таблица 2 – Пример таблицы потоков

Table 2 – Flow Table Example

Вх. порт	MAC ист.	MAC пр.	Приоритет	IP ист.	IP пр.	IP протокол	Порт ист.	Порт пр.	Действие	Счетчики
*	*	2A:4D :*	*	*	*	*	*	*	Порт1	100
*	*	*	*	*	192. 168. *	*	*	*	Порт2	251
*	*	*	*	*	*	*	21	21	Удалить	505
*	*	*	*	*	*	0x806*	*	*	Очередь1	821
*	*	*	*	*	*	0x1*	*	*	Контроллеру	53

Инструкции, ассоциированные с каждой записью в таблице потоков, могут описывать передачу и модификацию пакетов, обработку групповой таблицы и работу управляющей логики. Инструкции позволяют передать пакет последующим таблицам для продолжения обработки, а также допускают обмен метаданными между таблицами. Работа прекращается, когда набор инструкций, ассоциированный с определенной записью из таблицы, не указывает на следующую таблицу. Записи таблицы потоков могут указывать на групповые таблицы, которые позволяют переадресовывать пакеты на все порты или более сложные процедуры пересылки пакетов. Последняя таблица потоков не содержит команды goto.

Алгоритм работы таблицы потока по протоколу OpenFlow заключается в следующем (рисунок 6): при поступлении пакета очищается последнее совершенное действие и начинается старт с нулевой таблицы. Если пакет не соответствует правилу из таблицы потоков, то принимается одно из решений удалить пакет, переслать обратно контроллеру или перейти к следующей таблице потоков. Если пакет соответствует правилу, то обновляются счетчики, выполняются команды модификации, выполняются обновления набора команд, полей пакета и метаданных и происходит опрос к следующей таблице потоков. Если не нужно переходить к следующей таблице, то выполняются инструкция и действие с этим пакетом.

Для исследования и моделирования таблиц потоков в коммутаторе OpenFlow был выбран пакет CPN Tools, который позволяет исследовать различные параметры и характеристики компьютерных сетей с помощью аппарата цветных иерархических сетей Петри. Пакет хорошо зарекомендовал себя, и были исследованы компьютерные сети в статьях [4, 5].

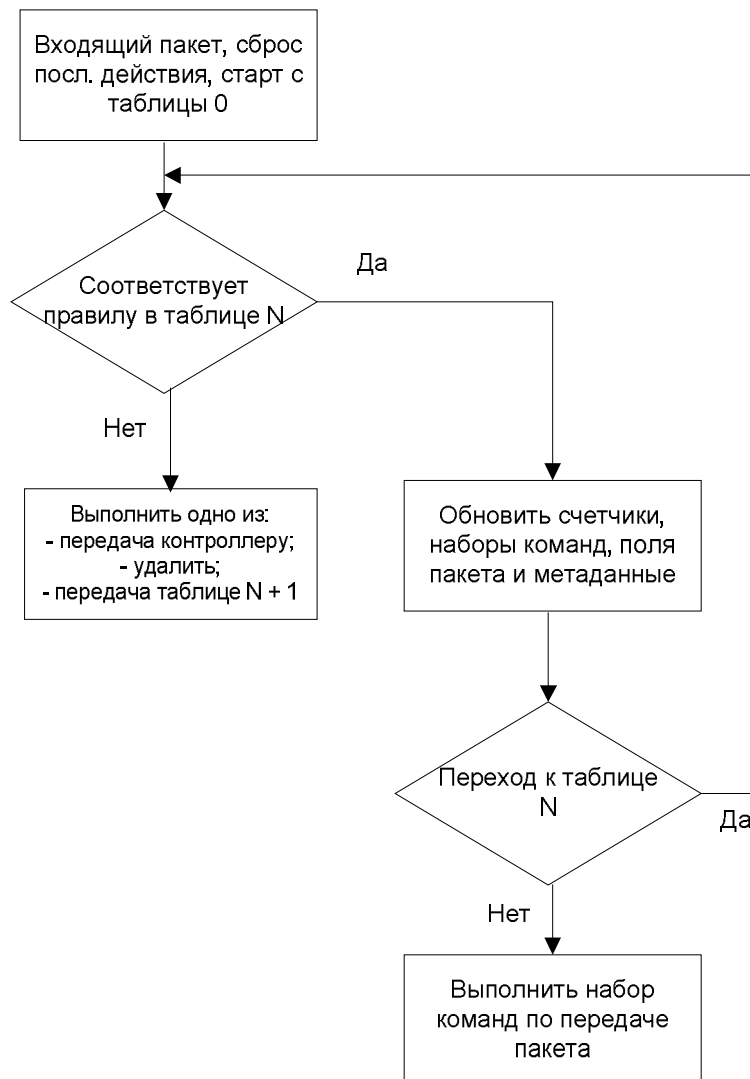


Рисунок 6 – Алгоритм передачи пакета в таблице потоков OpenFlow
Figure 6 – Algorithm of frame transmission in OpenFlow Flow Table

Экспериментальные исследования

В статье описывается работа с таблицами потоков на сетях Петри в виде реализованных подсетей. Алгоритм сравнения пакета с правилами в таблицах потоков и принятия решения по передаче пакета и дальнейшая классификация пакета не является целью исследования в данной статье.

На рисунке 7 представлена подсеть таблицы потоков 1. Входящий кадр передается через позицию Buffer In N цвета frame. Цвет frame реализован в виде кортежа frm, использующего перечень полей, как на рисунке 5, и представляется в виде заголовка кадра с некоторыми дополнениями или свободного состояния avail.

```
colset frm = product INT*InPort*MAC*MAC *VLID*QoS*IP*IP*INT*INTt*INTt declare input_ms.
```

Задаются следующие поля для цвета frm: номер поступившего кадра, входящий порт, MAC адреса источника и приемника, VLID индекс виртуальной связи, QoS приоритет качества обслуживания, IP адреса источника и приемника, размер кадра и задержки кадра.

Цвет InPort описывается как colset InPort = int with 1..10 timed для генерации и чтения входящего порта.

Цвет IP описывается как colset IP = int with 1..10 timed для генерации и чтения IP адресов. Остальные цвета полей представлены в статьях [6, 7].

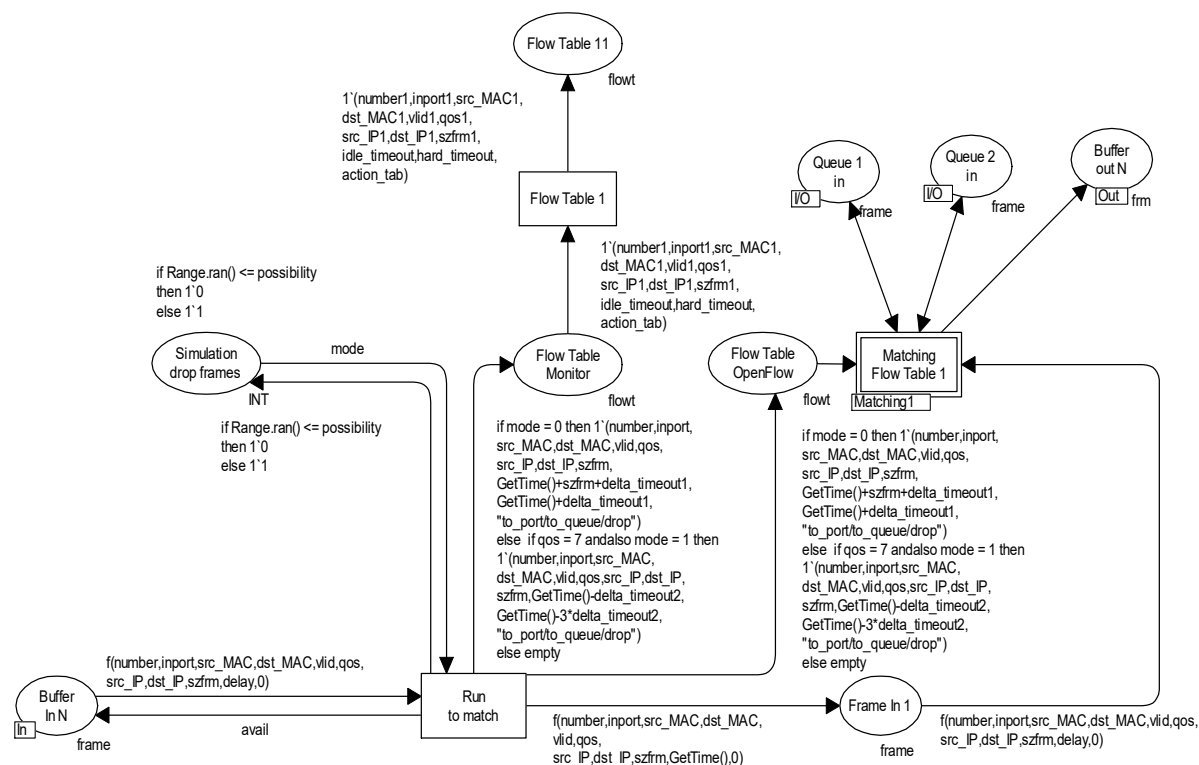


Рисунок 7 – Подсеть таблицы потоков 1

Figure 7 – Subnet of Flow Table 1

Кадр передается через позицию Run to match в виде кортежа с полями $f(\text{number}, \text{inport}, \text{src_MAC}, \text{dst_MAC}, \text{vlid}, \text{qos}, \text{src_IP}, \text{dst_IP}, \text{szfrm}, \text{delay}, 0)$. Заданы следующие переменные для кадра:

```
var number, delay: INT;
var inport, inport1: InPort;
var src_MAC, dst_MAC, src_MAC1, dst_MAC1: MAC;
var vlid, vlid1, vlid2: VLID;
var qos, qos1, qos2: QoS;
var src_IP, dst_IP, src_IP1, dst_IP1: IP;
var szfrm: BitSize;
```

Позиция Simulation drop frames служит для эмуляции в модели определенной доли кадров, которые будут удаляться из сети. Для этих целей в начальном значении позиции записано следующее выражение: `if Range.ran() <= possibility then 1'0 else 1'1`, где *possibility* – это статическая переменная, которая задает вероятность при генерировании диапазона. Через переменную *mode* будет происходить обработка данного события переходом Run to match.

Переходом Run to match генерируется таблица потоков 1 в позиции Flow Table 1, эта позиция имеет цвет flowt и описывается в виде colset `flowt = product INT*InPort*MAC*MAC*VLID*QoS*IP*IP*INT*INTt*INTt*STRING declare input_ms`. Формат цвета flowt соответствует формату на рисунке 4.

Поля типа INTt отвечают за тайм-ауты в таблице Idle timeout и Hard timeout. Строковое поле STRING предназначено для хранения действия с пакетом, в данном случае оно заполняется общее для всех типов кадров в виде "to_port/to_queue/drop". Таким образом, эмулируется вариативная отправка в порт или очередь, или удалить кадр.

Позиция Flow Table 1 заполняется по условию на дуге, и условие выглядит следующим образом:

```
if mode = 0 then 1'(number,inport, src_MAC,dst_MAC, vlid,qos,src_IP,dst_IP,
szfrm,GetTime()+szfrm+delta_timeout1,GetTime()+delta_timeout1,"to_port/to_queue/drop")
```

```

else if qos = 7 andalso mode = 1 then 1\ (number,inport,src_MAC, dst_MAC, vlid,qos,
src_IP,dst_IP,szfrm,GetTime()-delta_timeout2,GetTime()-3*delta_timeout2, "to_port/to_queue/ drop")
else empty.

```

Если установлен режим моделирования без удаления кадров $mode = 0$, то передается маркер цвета $flowt$, в котором заполнены тайм-ауты следующим образом: $Idle\ timeout = GetTime() + szfrm + delta_timeout1$ и $Hard\ timeout = GetTime() + delta_timeout1$. Функция $GetTime()$ генерирует машинное время компьютера. Таким образом, $Idle\ timeout$ является суммой функции $GetTime()$, размера кадра в битах и небольшого отклонения для тайм-аута $delta_timeout1$, в $Hard\ timeout$ не участвует размер кадра.

Если установлен режим моделирования для исследования удаляемых кадров $mode = 1$ и передается трафик реального времени, у которого приоритет качества обслуживания максимальный $qos = 7$, то генерируется меньшее время для тайм-аутов $Idle\ timeout$ и $Hard\ timeout$, где участвует статическая переменная $val\ delta_timeout2 = 10$.

Входящий кадр помещается в позицию $Frame\ In\ 1$, где фиксируется время поступления кадра в таблицу $f(number,inport,src_MAC,dst_MAC,vlid,qos,src_IP,dst_IP,szfrm,GetTime(),0)$. Кадр передается в позицию $Matching\ Flow\ Table\ 1$ подсети $Matching\ 1$, в этот же переход передается таблица потоков 1 в виде $1\ (number1,inport1, src_MAC1,dst_MAC1, vlid1,qos1, src_IP1, dst_IP1,szfrm1,idle_timeout,hard_timeout,action_tab)$.

По результатам работы перехода $Matching\ 1$ возможны следующие ситуации: помещение кадра в очередь 1 для трафика реального времени (позиция $Queue\ 1\ in$), помещение кадра в очередь 2 для стохастического (обычного) трафика (позиция $Queue\ 2\ in$) и передача кадра в выходной порт N (позиция $Buffer\ out\ N$).

С помощью специального средства «монитора» в среде CPN Tools были написаны предикат и его условие выполнения для записи в файл и дальнейшего его исследования. Монитор реализован на переходе $Flow\ Table\ 1$, который записывает значение из таблицы потоков 1 в виде $1\ (number1,inport1,src_MAC1,dst_MAC1,vlid1,qos1,src_IP1,dst_IP1,szfrm1,idle_timeout, hard_timeout,action_tab)$.

Функция предиката разработанного монитора описана ниже:

```

fun pred (bindelem) = let
  fun predBindElem (Flow_Table1'Flow_Table_1 (1,
    {action_tab, dst_IP1, dst_MAC1,hard_timeout, idle_timeout,inport1, number1, qos1,src_IP1,src_MAC1,szfrm1,vlid1})) = true
  | predBindElem _ = false
in  predBindElem bindelem
end

```

Функция обработки и записи в файл разработанного монитора описана ниже:

```

fun obs (bindelem) = let
  fun obsBindElem (Flow_Table1'Flow_Table_1 (1,
    {action_tab,dst_IP1,dst_MAC1, hard_timeout,idle_timeout, inport1, number1,qos1,src_IP1,src_MAC1,szfrm1,vlid1})) =
    Int.toString(number1)^", "^Int.toString(inport1)^", "^Int.toString(src_MAC1)^", "^Int.toString(dst_MAC1)^",
    "^Int.toString(vlid1)^", "^Int.toString(qos1)^", "^Int.toString(src_IP1)^",
    "^Int.toString(dst_IP1)^", "^Int.toString(szfrm1)^",          "^Int.toString(idle_timeout)^",
    "^Int.toString(hard_timeout)^", action_tab++\n"
  | obsBindElem _ = ""
in  obsBindElem bindelem end

```

Если входящий пакет не был найден в таблице потоков 1 , этот пакет перенаправляется в следующую таблицу потоков N через позицию $Frame\ to\ Next\ Flow\ Table$ (рисунк 8). И алгоритм повторяется аналогично таблице потоков 1 .

7. **Никишин К. И., Коннов Н. Н.** Генератор трафика Ethernet на основе цветных сетей Петри // Модели, системы, сети в экономике, технике, природе и обществе. 2016. № 1 (17). С. 299-307.

UDC 681.31

RESEARCH AND MODELING OF FLOW TABLE IN OPENFLOW SWITCH AND SOFTWARE DEFINED NETWORKS

K. I. Nikishin, Ph.D. (Tech.), lecturer, department of computer science, PSU, Penza, Russia;
orcid.org/0000-0001-7966-7833, e-mail: nkipnz@mail.ru

*Modern networks are becoming very cumbersome, the speed of traffic transmission over the network is required, and for these purposes Software Defined Networks (SDN) have been introduced. The main data transfer protocol in SDN networks is OpenFlow. The distribution of control and data in SDN networks is considered, data transmission from a controller via OpenFlow is described. **The aim of the work** is to investigate and research the OpenFlow protocol and switch, flow tables for correct transmission of frames in SDN networks based on color time hierarchical Petri nets and using CPN Tools modeling package. The objectives of the research are to consider various formats of frames and flow tables, to develop algorithms for the functioning of flow table and moderating time characteristics of frame transmission through these tables, taking into account specified timeouts. The models allowed us to investigate the functioning of flow table as well as flow temporal characteristics in OpenFlow switch for further investigation of heterogeneous traffic transmission in SDN networks.*

Key words: Software Defined Networks, controller, switch, Ethernet, OpenFlow, Flow Table, time-outs, Petri Nets, CPN Tools.

DOI: 10.21667/1995-4565-2022-81-42-50

References

1. *Opisanie standart IEEE 802.1q* [Elektronnyj resurs] – Rezhim dostupa: https://ru.wikipedia.org/wiki/IEEE_802.1Q.
2. **Kobayashi M., Seetharaman S., Parulkar G., Appenzeller G., Little J., Van Reijendam J., McKeown N.** Maturing of OpenFlow and Software-Defined Networking Through Deployments. *Computer Networks*. 2014, vol. 61, pp. 151-175.
3. **McKeown N., Anderson T., Balakrishnan H. et al.** Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*. 2008, vol. 38, no. 2, pp. 69-74.
4. **Nikishin K., Konnov N.** Schedule Time-Triggered Ethernet. International Conference on Engineering Management of Communication and Technology. *EMCTECH 2020*. DOI: 10.1109/EMCTECH49634.2020.9261540.
5. **Kizilov E., Konnov N., Nikishin K., Pashchenko D., Trokoz D.** Scheduling queues in the Ethernet switch, considering the waiting time of frames. *MATEC Web of Conferences*. 2016, vol. 44, pp. 01011-p.1–01011-p. 5.
6. **Nikishin K. I.** Mehanizm upravlenija trafikom real'nogo vremeni v kommutatore Ethernet. *Vestnik komp'juternyh i informacionnyh tehnologij*. 2015, no. 10, pp. 32-37. (in Russian).
7. **Nikishin K. I., Konnov N. N.** Generator trafika Ethernet na osnove cvetnyh setej Petri. *Modeli, sistemy, seti v jekonomike, tehnike, prirode i obshhestve*. 2016, no. 1 (17), pp. 299-307. (in Russian).