

## ГЕНЕРАТОР ТРАФИКА ETHERNET НА ОСНОВЕ ЦВЕТНЫХ СЕТЕЙ ПЕТРИ

*К. И. Никишин, Н. Н. Коннов*

## THE TRAFFIC GENERATOR OF SWITCH ETHERNET USING COLORED PETRI NETS

*K. I. Nikishin, N. N. Konnov*

**Аннотация.** *Актуальность и цели.* В последнее время наиболее широко распространенной технологией стала технология *Ethernet* с поддержкой качества обслуживания *QoS* (*Quality of Service*). Для того, чтобы исследовать и улучшать гарантированное качество услуг связи, необходимо изучить входной трафик коммутатора *Ethernet*. Цель данного исследования – моделирование источника потока кадров, характеристики которого были бы адекватны свойствам реального трафика *Ethernet*. *Материалы и методы.* Реализация задач была достигнута за счет использования математического аппарата иерархических временных цветных сетей Петри. Для построения моделей на основе сетей Петри использовался пакет *CPN Tools*. *Результаты.* В работе подробно описаны формат кадра *Ethernet*, основные функции работы модели генератора трафика. Исходя из разработанной модели, было проведено исследование регулярного и стохастического трафика сети *Ethernet*. *Выводы.* Моделирование трафика сети *Ethernet* с поддержкой *QoS* позволяет получить реальную картину генерации трафика и понять, какие параметры трафика влияют на дальнейшую обработку кадров в сети *Ethernet*.

**Ключевые слова:** коммутатор *Ethernet*, качество обслуживания, трафик, генератор.

**Abstract.** *Background.* In recent years the most widespread technology has become the technology of Ethernet with support for QoS (Quality of Service). In order to investigate and improve the guaranteed quality of communication services, it is necessary to examine input traffic to the Ethernet. The purpose of this study is the modeling flow source frames, the characteristics of which would be adequate to the properties of real Ethernet traffic. *Materials and methods.* The implementation of the research objective was achieved by the use of mathematical apparatus of hierarchical time colored Petri nets. The package CPN Tools was used for developing models based on Petri nets. *Results.* The format of the Ethernet frame, the main functions of the model generator of traffic were described in detail at the article, based on which a research of regular and stochastic traffic of the Ethernet network was conducted in the article. *Conclusions.* Modeling traffic of the Ethernet network with QoS support allows getting a real picture of generating traffic and understand the traffic parameters affect the further processing of frames in the Ethernet network.

**Key words:** switch Ethernet, quality of service, traffic, generator.

### **Введение**

Генератор трафика коммутатора *Ethernet* представляет собой сложную часть работы телекоммуникационной сети. Чем глубже будет исследован алгоритм генерации кадров *Ethernet*, работа смешивания трафика с различным приоритетом, тем лучше можно проанализировать работу коммутатора *Ethernet*.

Свободно распространяемый пакет *CPN Tools* был выбран в качестве инструмента имитационного моделирования, позволяющего исследовать различные аспекты поведения сложных телекоммуникационных систем, использующего математический аппарат иерархических временных цветных сетей Петри [1, 2]. Для того, чтобы исследовать работу коммутатора *Ethernet*, необходимо прежде всего начинать с генерации входящего трафика для коммутатора. Модель представляет собой обобщенную модель реального трафика, обладает своей гибкостью и может подстраиваться к различным конфигурациям коммутатора *Ethernet*.

### 1. Общая модель генератора трафика коммутатора *Ethernet*

Генераторы регулярного и стохастического трафика вырабатывают последовательность маркеров, соответствующих кадрам сетевого трафика. Пакеты «раскрашиваются» в соответствии с адресами источника и получателя, классами качества обслуживания, заданными распределениями длины пакета и времени его поступления. Смеситель обеспечивает агрегирование отдельных потоков кадров и контролирует длину их генерируемого потока. На рис. 1 представлено представление верхней иерархии модели генератора трафика.

Сгенерированный поток кадров записывается в файл. При эксперименте специальный блок считывает множество маркеров из файла и восстанавливает их временную последовательность передачи в исследуемую модель телекоммуникационного устройства. Предварительная фиксация сгенерированной реализации трафика в файле обеспечивает повторяемость экспериментов при смене или коррекции исследуемой модели и ускорение процесса моделирования.

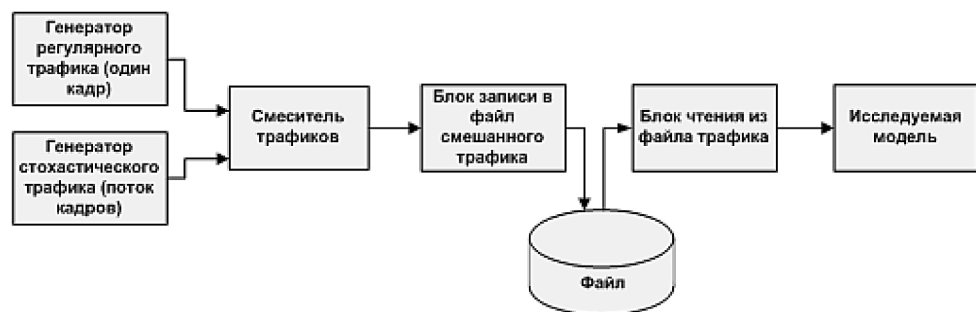


Рис. 1. Общая модель генератора трафика

### 2. Реализация модели генератора трафика на основе сетей Петри

Генератор трафика позволяет исследовать следующие важные характеристики кадров: межкадровый интервал, интенсивность поступления регулярного трафика, определение величины приоритета кадра на основе весов приоритета.

Для того, чтобы определить межкадровый интервал, необходимо определить количество интервалов. После нужно определить, какой процент интервалов относится к области минимальной длины кадра, к максимальной длине и к равно распределенной длине кадра. Конечный размер кадра полу-

чается, исходя из определения вышеописанной области. Например, в модели генератора трафика функция *SetSizeFull* выполняет роль формирования конечной длины кадра. Более подробный алгоритм данной функции представлен ниже.

Следующей важной характеристикой исследования трафика является интенсивность поступления регулярного трафика. Интенсивность позволяет задавать периодичность кадров реального времени, тем самым смешиваются регулярный и стохастический трафики. Кадры реального времени дальше анализируются в коммутаторе *Ethernet*, определяется эффективность обработки регулярного трафика и джиттер таких кадров. На примере модели интенсивность регулярного трафика задается в позиции *Period*.

Определение величины приоритета кадра на основе весов приоритета также является важной характеристикой генерации кадра. Для того, чтобы определить приоритет кадра, необходимо определить количество интервалов, как по аналогии с межкадровым интервалом, но здесь количество приоритетов ограничено. После для каждого приоритета определяется процент попадания приоритета в заданный интервал и вычисляется приоритет, попавший в минимальную область, в максимальную область и в равно распределенную область. Конечный приоритет кадра получается, исходя из определения вышеописанной области. На рис. 2 представлена подсеть определения приоритета кадра.

В позиции *Min Max QoS* задается вероятность попадания приоритета в минимальную и максимальную области. Функция *SetQoSInterval()* генерирует количество интервалов приоритета кадров. Конечный приоритет кадра вычисляет функция *SetQoSFull(x,vmin,vmax)*, по аналогии с функцией *SetSizeFull*.

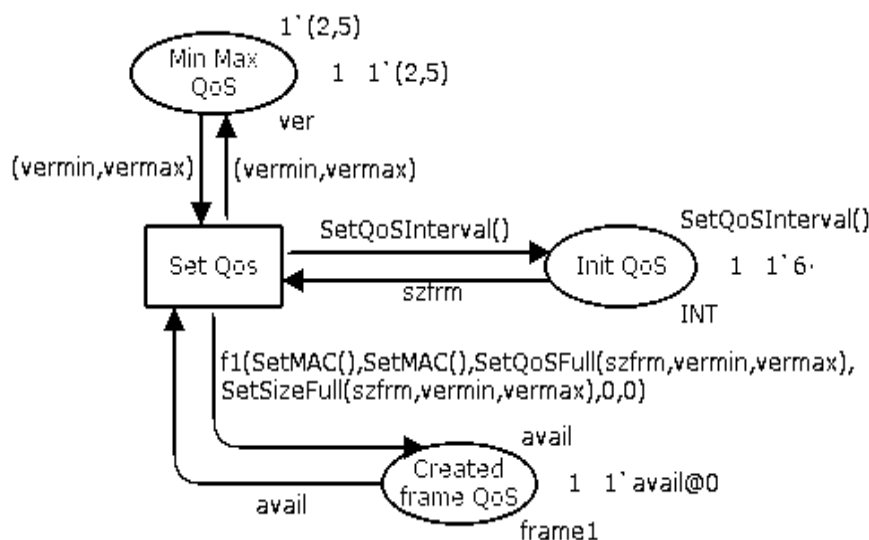


Рис. 2. Подсеть определения приоритета кадра

Для того, чтобы описать работу модели на сетях Петри в *CPN Tools*, необходимо прежде всего описать цвета, используемые в данной модели. В ней представлены следующие цвета:

– цвет *frm* отражает структуру кадра *Ethernet* и состоит из *MAC*-адресов источника и приемника, приоритета кадра, размера передаваемых данных и двух специальных полей задержек, которые, в свою очередь, предназначены для получения времени на входном и выходном порте коммутатора *Ethernet*:

$$colset\ frm = product\ MAC*MAC*QoS*INT*INTT*INTT;$$

– цвет *frame*, который может быть либо занят передачей кадра (*frm*), либо быть свободным (*avail*), что позволяет выявлять и обрабатывать события, связанные с наличием или отсутствием кадра.

Все поля модели являются настраиваемыми, что обеспечивает гибкость системы. Это подтверждается описанием следующих функций генерации полей формата кадра *Ethernet*. *MAC*-адреса источника и приемника можно задавать двумя способами: либо непосредственно в модели, либо получить через специальную функцию, которая выбирает случайным образом значение адреса. В модели использовался второй вариант генерации адреса в диапазоне от 1 до 10:

$$fun\ SetMAC() = MAC.ran().$$

В *Ethernet* коммутаторе приоритеты можно разделить на несколько частей, в зависимости от важности и скорости передачи кадра по телекоммуникационной сети. Чаще всего приоритеты разбиваются на два вида, но их может быть и больше. Общий алгоритм определения приоритета кадра на основе весов приоритета описан выше. В модели генератора трафика использовался частный случай, и было решено использовать два вида: высокий приоритет (характерен для кадров регулярного трафика) и обычный приоритет (характерен для стохастического трафика). Высокий приоритет имеет всегда значение 7, приоритет стохастического трафика генерируется по равномерному распределению в диапазоне от 0 до 6:

$$fun\ SetQoS() = discrete(0,6).$$

Регулярный и стохастический трафики модели являются гибкими и могут изменяться в зависимости от различных требований, применяемых к модели. Генератор регулярного трафика вырабатывает кадры с определенным постоянным периодом. Формируются одновременно 4 кадра регулярного трафика, причем, чтобы устранить одновременность прихода кадров в коммутатор, используется добавление небольшой задержки. Для первого кадра добавляется единица, для второго – двойка, и т.д. Методы устранения одновременного прихода кадров в коммутатор применяются и в реальных системах управления коммутатором *Ethernet*. Для того, чтобы перемешать 4 кадра между собой и сгенерировать различную очередность между этими кадрами, использовалась дополнительная функция *RandDelay*, умноженная на 10. Функция *RandDelay* принимает равно распределенное значение от 1 до 4:

$$fun\ RandDelay() = discrete(1,4).$$

Таким образом, конечная задержка будет равна  $RandDelay()*10 + \text{Номер кадра}$ . Длина кадра также может варьироваться, например, на рис. 3 длина кадра фиксирована и равна  $1522*8$  бит.

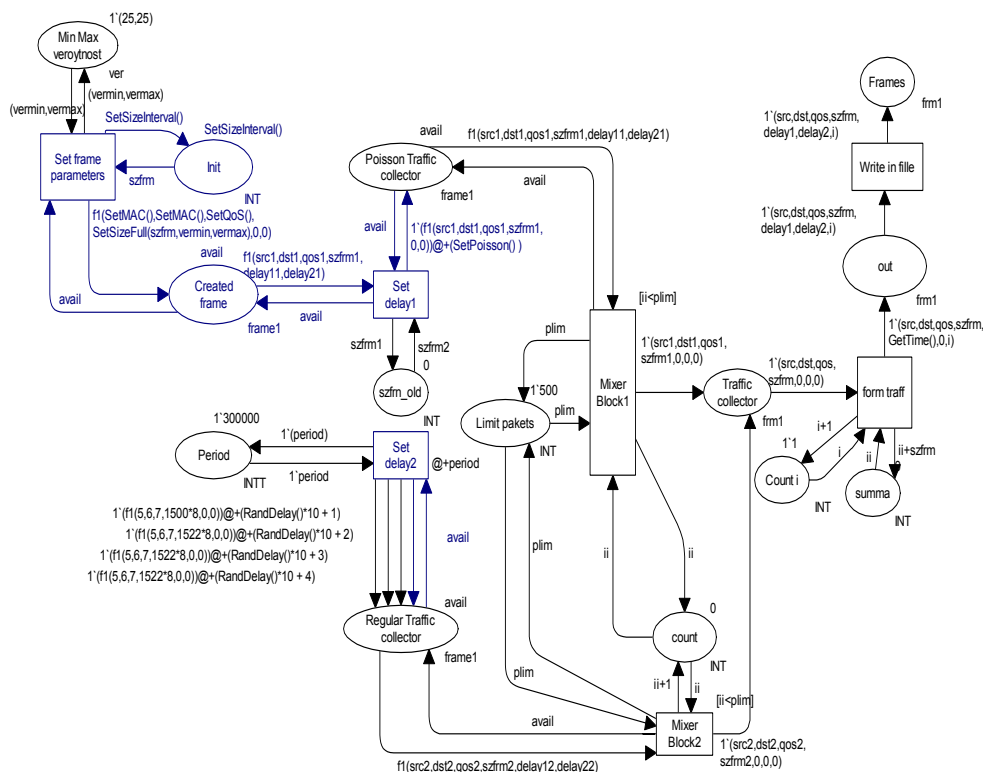


Рис. 3. Модель генератора трафика на сетях Петри

Генератор стохастического трафика обеспечивает формирование кадров через временные интервалы, распределенные по экспоненциальному закону (Пуассоновский поток). Функция Пуассоновского распределения представлена ниже, и через экспоненциальное распределение в функции *SetPoisson* можно регулировать загрузку канала коммутатора Ethernet:

$$\text{fun SetPoisson}() = \text{floor}(\text{exponential}(1.0/10000.0) + 0.5).$$

Как показывает эксперимент [3], распределение длины поля данных имеет ярко выраженный бимодальный характер: до 50 % всех кадров имеют минимальную или максимальную длину (64 и 1500 байт, соответственно), длину остальных можно считать равномерно распределенной в оставшемся диапазоне.

Распределение длины кадра стохастического трафика определяется следующим образом. Вначале в позиции *Min Max veroytnost* задаются вероятности кадров с минимальным и максимальным значением длины. Функция *SetSizeInterval* формирует дискретное значение в диапазоне от 0 до 99, диапазон можно также подстроить под требования различных конфигураций системы:

$$\text{fun SetSizeInterval}() = \text{discrete}(0,99).$$

Функция *SetSizeFull* вычисляет конечную длину кадра. В функцию *SetSizeFull* передается дискретное значение, полученное с помощью *SetSizeInterval*; это значение необходимо для определения вероятности попадания длины

кадра в определенную область. Таким образом, возникают три случая вероятности попадания длины кадра в одну из областей: минимальная, максимальная и средняя области. Затем после определения области генерируется длина кадра в битах. Размеры длины кадра являются также гибкими параметрами и могут быть перенастроены. Рассмотрим алгоритм функции *SetSizeFull* более подробно [4]:

- если полученная вероятность меньше переменной *vermin*, то длина кадра будет равна минимальному значению (в данной модели минимальное значение равно  $64 \cdot 8$  бит);

- если полученная вероятность больше  $100 - v_{max}$ , то длина кадра будет равна максимальному значению (в данной модели максимальное значение равно  $1522 \cdot 8$  бит);

- если полученная вероятность не попала ни в одну из ранее определенных областей, то размер длины кадра будет выбран случайным образом по равномерному распределению, диапазон которого равен от минимального  $65 \cdot 8$  до  $1521 \cdot 8$  бит.

Описание функции *SetSizeFull(x, vmin, vmax)* в пакете *CPN Tools* представлено ниже:

```
fun SetSizeFull(x,vmin,vmax) =
  if x < vmin then 64*8 else if x > (99-vmax) then (22+1500)*8
  else discrete(65,21+1500)*8.
```

После того как сформировалось значение длины кадра стохастического трафика, полученный кадр передается в смеситель трафика. Смеситель трафика обеспечивает смешивание двух видов трафика (стохастического и регулярного). Задержка регулярного трафика получается по Пуассоновскому распределению через функцию *SetPoisson*.

В модели в позиции *Limit packets* задается ограничение на количество генерируемых кадров. На рис. 3 в этой позиции стоит ограничение на 500 кадров регулярного трафика. Данная позиция может быть уменьшена или увеличена по требованию гибкости к системе. Позиция *count* осуществляет подсчет количества кадров регулярного трафика.

На выходе перехода смесителя трафика *Mixer Block1* изменяется цвет кадра формата *Ethernet*. Новый введенный цвет называется *frm1*, к которому, в отличие от цвета *frm*, добавляется еще одно поле – номер кадра. Это было сделано для того, чтобы при анализе статистических данных было легко распознать номер кадра. Позиция *Count i* осуществляет подсчет общего числа кадров. Позиция *summa* осуществляет подсчет размера передаваемых данных. Входная задержка кадра получается следующей функцией:

```
fun GetTime() = IntInf.toInt(time()).
```

Конечным блоком модели явился блок записи кадров в файл согласно цвету *frm1*. Запись в файл была осуществлена стандартными средствами пакета *CPN Tools*. Производится запись всех кадров, полученных после смесителя трафика. Запись в файл была сделана средствами монитора. Для этого на переходе *Write in file* был создан монитор с именем *MWritefromTraffGen* и расширением файла *txt*. Для того, чтобы записать поля кадра *Ethernet*, необходимо в теле функции *Observer* монитора записать поля цвета *frm*

(MAC-адреса источника и приемника, приоритет кадра, размер передаваемых данных и два специальных поля задержек), а также ввести номер кадра, поступившего в сеть *Ethernet*. Поля могут быть регулируемы, поэтому их можно дополнять или уменьшать при записи в файл.

Реализация монитора *MWritefromTraffGen* на языке *CPN ML* представлена ниже:

```
fun obs (bindelem) =
  let
    fun obsBindElem (MixerGenerator'Write_in_fille(1, {delay1, delay2, dst, i, qos, src, szfrm})) = "1`(^Int.toString(i)^, ^Int.toString(src)^, ^Int.toString(dst)^, ^Int.toString(qos)^, ^Int.toString(szfrm)^, ^ModelTime.toString(time())^",0)++\n"
    | obsBindElem _ = ""
  in
    obsBindElem bindelem
  end.
```

Для того, чтобы считывался сгенерированный трафик из файла, была сделана следующая подсеть на основе сетей Петри (рис. 4). Чтение реализовано на переходе *Initialise Packets* в сегменте кода. С помощью функции *TextIO.openIn*(«*MWritefromTraffGenWRR.txt*») осуществляется открытие файла [5], который был получен монитором *MWritefromTraffGen*. Все прочитанные кадры поступают в позицию *Read from file*. Для того, чтобы из позиции *Read from file* кадры опраивались строго по времени, на переходе *Transfer frame* стоит временная задержка  $@+(delay-\delta)$ . Затем кадр проходит в исследуемую модель.

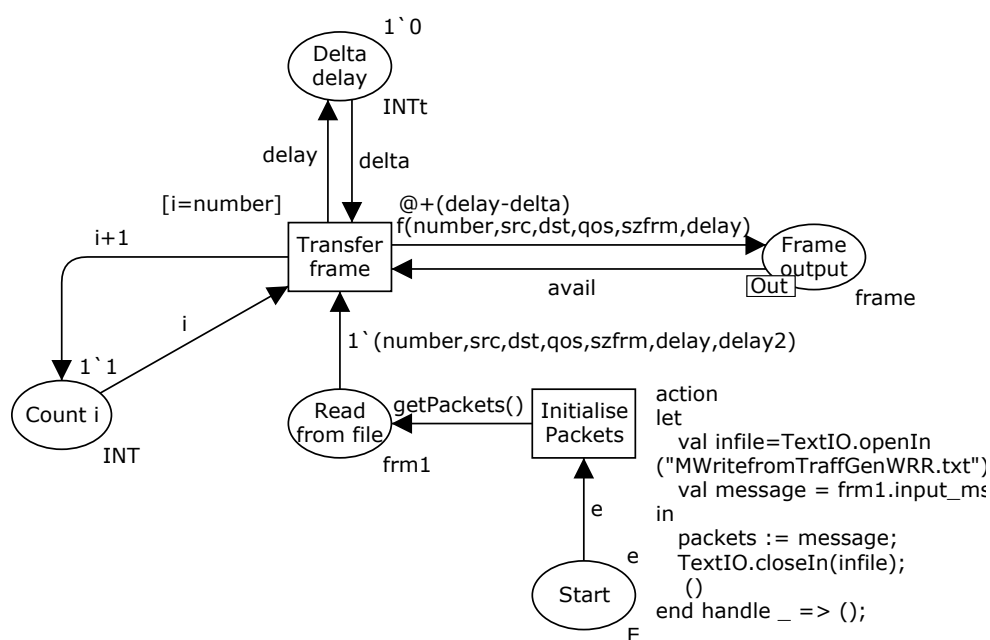


Рис. 4. Подсеть Петри чтения трафика из файла

### 3. Результаты работы модели

Трафик исследуемой модели сравнивался с теоретическим трафиком коммутатора *Ethernet*. На рис. 5 представлена гистограмма распределения длин кадров теоретического и экспериментального трафика, а на рис. 6 – гистограмма распределения интервалов между моментами поступления кадров.

Экспериментальный трафик подтверждает гибкость и эффективность модели. Также было экспериментально доказано сходство экспериментального и теоретического трафика с использованием аппарата сетей Петри.



Рис. 5. Гистограмма распределения длин кадров теоретического и экспериментального трафика



Рис. 6. Гистограмма распределения интервалов между моментами поступления кадров



### **Заключение**

Модель генератора трафика может быть использована для изучения особенностей входящего трафика коммутатора *Ethernet*. Кроме того, данные, полученные в результате анализа работы модели, могут быть использованы при разработке алгоритмов функционирования и программного обеспечения узлов телекоммуникационного оборудования с целью повышения качества обслуживания и эффективности обработки трафика.

### **Список литературы**

1. Jensen, K. Coloured Petri Nets: modelling and validation of concurrent systems / Kurt Jensen, Lars M. Kristensen. – Springer, 2009. – 384 p.
2. Зайцев, Д. А. Моделирование телекоммуникационных систем в CPNTools / Д. А. Зайцев, Т. Р. Шмелева. – Одесса : Одесская национальная академия связи им. А. С. Попова, 2008. – 68 с.
3. Петров, В. В. Структура телетрафика и алгоритм обеспечения качества обслуживания при влиянии эффекта самоподобия : автореф. дис. ... канд. техн. наук : 05.12.13 / Петров В. В. – М., 2004. – 198 с.
4. Никишин, К. И. Механизм управления трафиком реального времени в коммутаторе Ethernet / К. И. Никишин // Вестник компьютерных и информационных технологий. – М., 2015. – Вып. 10. – С. 32–37.
5. Описание Simple protocol example. – URL: [http://cpntools.org/\\_media/documentation/simpleprotocol.pdf](http://cpntools.org/_media/documentation/simpleprotocol.pdf).

---

**Никишин Кирилл Игоревич**

аспирант,  
Пензенский государственный университет  
E-mail: kirillnikmail@mail.ru

**Nikishin Kirill Igorevich**

postgraduate student,  
Penza State University

**Коннов Николай Николаевич**

кандидат технических наук, профессор,  
кафедра вычислительной техники,  
Пензенский государственный университет  
E-mail: knn@pnzgu.ru

**Konnov Nikolay Nikolaevich**

candidate of technical sciences,  
professor, sub-department  
of computer science,  
Penza State University

---

УДК 681.31

**Никишин, К. И.**

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