

## МОДЕЛИРОВАНИЕ АДАПТИВНОЙ ДИСПЕТЧЕРИЗАЦИИ ОЧЕРЕДЕЙ В КОММУТАТОРЕ С ПОДДЕРЖКОЙ QoS

*Е. А. Кизилов, Н. Н. Коннов, Д. В. Патунин*

## SIMULATION OF THE ADAPTIVE SCHEDULING QUEUES IN THE ETHERNET SWITCH SUPPORT

*E. A. Kizilov, N. N. Konnov, D. V. Patunin*

**Аннотация.** *Актуальность и цели.* Целью работы является повышение эффективности дефицитных циклических алгоритмов диспетчеризации очередей в коммутаторах Ethernet с поддержкой обслуживания (QoS). *Материалы и методы.* В качестве оценки эффективности алгоритмов предлагается использовать оценки вариации среднего и максимального значений джиттера трафика различных классов QoS, полученных методом имитационного моделирования алгоритмов диспетчеризации с использованием аппарата цветных временных сетей Петри. *Результаты.* Предложена сеть Петри, моделирующая адаптивный алгоритм дефицитного циклического обслуживания очередей с временной селекцией кадров (DRR-TSS). Проведен сравнительный анализ полученных статистических характеристик джиттера для предложенного алгоритма, классического алгоритма DRR и алгоритма временной селекции DTSS. *Выводы.* Результаты моделирования показали, что эффективность предложенного алгоритма в 1,3–5 раз выше, чем у известных.

**Ключевые слова:** коммутатор Ethernet, качество обслуживания, TSS-алгоритмы, джиттер, диспетчеризация.

**Abstract.** *Background.* The purpose of operation is increase in efficiency of deficit cyclic scheduling algorithms of queues in the switches Ethernet with support of service (QoS). *Materials and methods.* As an assessment of efficiency of algorithms it is offered to use estimates of a variation of average and maximum values of a jitter of a traffic of different classes QoS received by method of simulation modeling of algorithms scheduling of with use of the device of color temporal Petri nets. *Results.* The Petri net modeling the adaptive algorithm of deficit round robin servicing of queues with temporal selection of frames (DRR-TSS) is offered. The comparative analysis of the received statistical characteristics of a jitter for the offered algorithm and a classical algorithm of DRR and an algorithm of temporal selection of DTSS is carried out. *Conclusions.* Results of simulation showed efficiency of the offered algorithm in 1,3–5 times above than at known.

**Key words:** Ethernet switch, quality of service, TSS-algorithms, jitter, scheduling.

### **Введение**

В интеллектуальных коммутаторах Ethernet с поддержкой QoS [1] для диспетчеризации очередей широко применяются циклические алгоритмы, которые обычно реализуются в вариантах, позволяющих регулировать полосы пропускания для каждого канала:

- Weighted Round Robin (WRR) – вес задает количество кадров, пересылаемых из очереди за один сеанс обслуживания, при этом выделяемая полоса зависит от длины передаваемых пакетов;
- Deficit Round Robin (DRR) – вес задает количество байт данных, передаваемых из очереди за один сеанс обслуживания.

Алгоритм DRR более предпочтителен, так как позволяет более точно выделять полосу независимо от длины передаваемых кадров за счет применения механизма «дефицита»: если при передаче нескольких кадров возникает ситуация, когда квота очереди еще не выработана, но ее недостаточно для передачи очередного кадра, то очередной кадр передается, но объем данных, превышающих установленную для очереди квоту (дефицит), вычитается из выделяемой в следующем цикле обслуживания для этой очереди квоты [2].

Основным достоинством циклических алгоритмов является исключение монополизации каналов и простота реализации, а недостатком – неконтролируемая величина джиттера (разброса задержки) коммутируемых кадров разницы во времени ожидания кадров в различных очередях, что вызвано детерминистским порядком их обслуживания. Кроме того, задержка кадров определенного класса QoS, а также размеры соответствующих очередей зависят от выделенной полосы пропускания для конкретного класса. Это приводит к увеличению задержки кадров и перегрузки очередей в каналах с малой назначенной полосой пропускания при пульсирующем характере трафика.

Для решения этой проблемы был предложен алгоритм управления очередями, основанный на учете времени ожидания кадров в очередях различных классов и называемый в дальнейшем «обслуживание с временной селекцией кадров» (TSS – Time Selection Service) [3, 4].

Алгоритм DTSS работает следующим образом:

- каждый кадр, направляемый в соответствующую его классу QoS очередь, снабжается атрибутом, отмечающим время его постановки в очередь;
- после окончания обслуживания очереди диспетчер коммутатора анализирует атрибуты кадров, находящихся во главе очередей, и выбирает для дальнейшего обслуживания очередь с его наименьшим значением, т.е. очередь с самым «старым» на данный момент кадром;
- из анализа очередей исключается последняя обслуженная (кроме случая, когда все остальные очереди пустые);
- для регулирования полосы пропускания каждого класса трафика используется стандартный метод дефицита.

При этом в алгоритме DTSS так же, как в классических циклических сохраняется негативное влияние используемого метода управления полосой пропускания для трафика различного класса, группируемого по очередям, соответствующим полю QoS. Это связано с механизмом выделения для каждой очереди полосы пропускания выходного канала посредством установки квоты (порции) данных, считываемых из очереди за один сеанс обслуживания. Поскольку квота задает длительность сеанса непрерывной передачи очередной порции данных из очереди каждого класса, то это автоматически приводит к увеличению задержек в обслуживании других очередей.

Кроме того, следует отметить, что непрерывная передача больших порций данных трафика одного класса приводит к «пачечности» трафика, что в свою очередь увеличивает нагрузку на буфера последующих телекоммуникационных устройств. Простое пропорциональное уменьшение размеров выделяемых квот не решает проблему, так как оно ограничено физическими размерами кадров данных и необходимой точностью задания полосы пропускания.

### ***Постановка задачи***

Таким образом, можно сформулировать задачу диспетчеризации очередей, обеспечивающую одновременно справедливость, наличие дифференцированной, выделяемой для каждого класса трафика полосы пропускания и уменьшение «пачечности» (времени непрерывного обслуживания каждого кадра).

Ниже предлагается следующее решение поставленной задачи, одновременно сочетающее принципы диспетчеризации DRR с учетом времени ожидания обслуживания кадров при ограничении длительности обслуживания каждого канала:

- сохраняется логика циклической дисциплины обслуживания очередей с управлением полосой пропускания через установление квоты по методу «дефицита» для каждой очереди, что исключает монополизацию выходного канала и сохраняет высокую точность установки требуемой полосы;
- ограничивается длительность непрерывного обслуживания каждой очереди путем периодического переключения на передачу отдельного кадра, в качестве которого выбирается самый «старый» из всех находящихся в текущий момент в очередях.

Реализация предлагаемого алгоритма диспетчеризации заключается в выполнении следующих действий [5]:

- каждый кадр, направляемый в соответствующую его классу QoS очередь, снабжается дополнительной меткой, обозначающей время его постановки в очередь;
- после окончания обслуживания очереди диспетчер выбирает для дальнейшего обслуживания очередь, используя циклический алгоритм;
- выполняется подсеанс обслуживания выбранной очереди, в рамках которого проводится передача кадров в установленном объеме;
- по окончании подсеанса выбирается очередь по алгоритму, применяемому в TSS, т.е. содержащая кадр с наименьшим значением временной метки;
- выполняется передача одного разделительного кадра из выбранной очереди в выходной канал, при этом размер переданного разделительного кадра учитывается уменьшением текущего значения выделенной квоты той очереди, из которой взят разделительный кадр;
- после передачи разделительного кадра продолжается следующий подсеанс обслуживания ранее выбранной по алгоритму RR очереди до выработки выделенной квоты или подсеанса;

– для регулирования полосы пропускания каждого класса трафика используется дефицитный алгоритм.

Для сравнения различных алгоритмов диспетчеризации очередей необходимо определить критерии, которые позволяли бы оценить эффективность управления ими. В качестве критерия предлагается использовать вероятностные характеристики джиттера, а именно вариацию средних и максимальных значений трафика различных классов QoS. Эти характеристики можно рассматривать в качестве количественной меры справедливости обслуживания очередей, так как они отображают разброс времени ожидания кадров различного типа трафика, который обеспечивается применяемым алгоритмом диспетчеризации.

Задача определения эффективности на основе оценки вероятностно-временных характеристик обработки трафика в связи с вычислительной сложностью аналитических расчетов может быть решена только с помощью имитационного моделирования [6].

### *Сеть Петри, моделирующая алгоритм диспетчеризации*

Для описания моделей алгоритмов широко применяется аппарат иерархических временных раскрашенных сетей Петри, поддерживаемый пакетом CPN Tools [7–9]. Данный пакет имеет следующие достоинства по сравнению с другими программными средствами:

- анализ поведения моделей с помощью имитации динамики сети Петри;
- построение и анализ пространства состояний модели;
- журналирование поведения модели;
- гибкость, функциональность;
- поддержка мощного языка CPN ML.

Общая схема сети Петри, моделирующей исследуемый алгоритм, представлена на рис. 1.

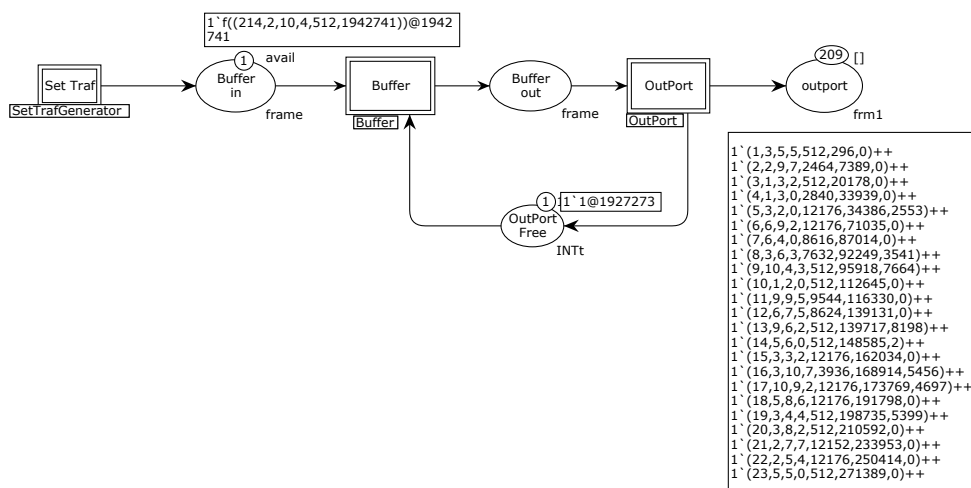


Рис. 1. Сеть Петри, моделирующая алгоритм диспетчеризации

Подсеть *SetTraffic* считывает из заранее сгенерированного файла последовательность маркеров, моделирующих трафик с заданными характеристиками, и обеспечивает их передачу в основную модель (подсеть *Buffer*) с учетом временных характеристик. Подсеть *Buffer* моделирует поведение буфера выходного порта и будет рассмотрена ниже. Подсеть *OutPort* моделирует работу выходного порта с учетом временных характеристик спецификации Ethernet. Выходная позиция *outport* используется для сбора статистики о параметрах выходного потока кадров после обработки в коммутаторе.

Обобщенная структура подсети буфера выходного порта, реализующего алгоритм DRR-TSS, представлена на рис. 2 и содержит следующие компоненты: классификатор кадров *Classifier*, четыре очереди *Queue1 ... Queue4*, узел сканирования очередей *scan*, четыре узла управления полосой пропускания *bend1 ... bend4*.

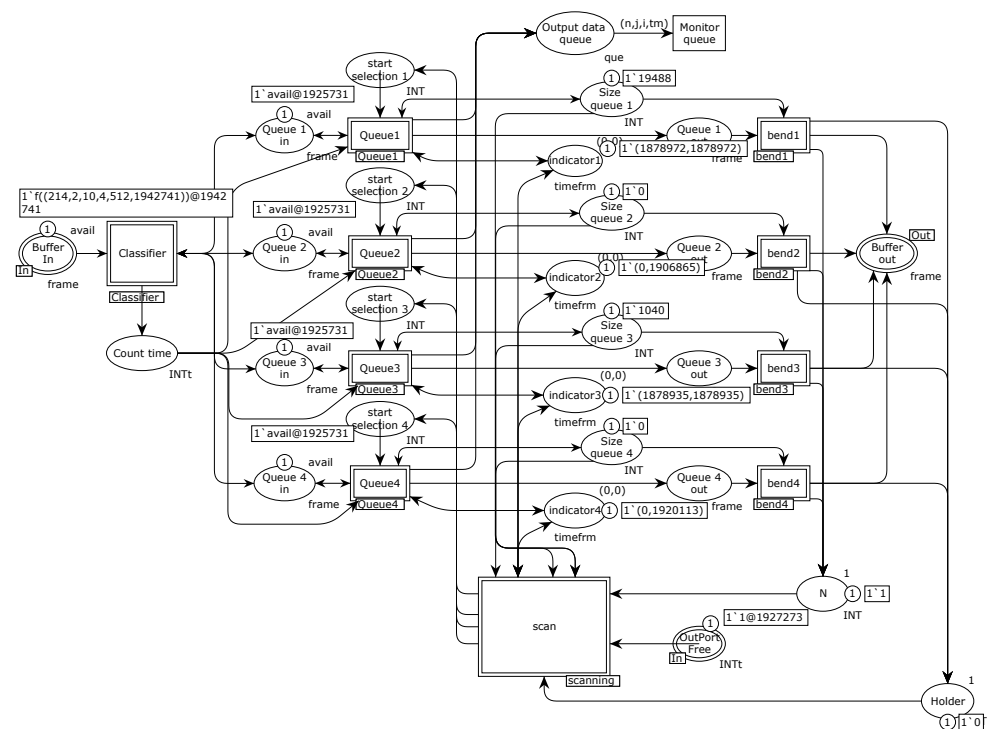


Рис. 2. Обобщенная структура подсети, моделирующей буфера выходного порта коммутатора

Сетевой трафик моделируется последовательностью маркеров, поступающих во входные порты и исходящих из выходных портов. Цвета маркеров образуют кортеж *ftm* полей, соответствующих формату кадров Ethernet 802.1 q/p [10]: адреса источника и приемника, поле тега с информацией о принадлежности определенному уровню приоритета, поле, указывающее длину полезной нагрузки кадра, а также служебные поля времени прихода и исхода кадров (используются для сбора статистики).

Используется также множество цветов frame, представляющее собой однонаправленный канал, который может быть либо занят передачей кадра frm, либо свободен avail, что позволяет выявлять и обрабатывать события, связанные с наличием или отсутствием кадра.

Кроме того, вводятся маркеры, цвета которых отражают служебные характеристики модели (размер очереди, количество переданных данных, признак очистки очереди и др.). Для описания цветов, переменных и функций используется язык CPN ML [7].

Маркер, моделирующий кадр трафика, после поступления в буфер выходного порта проходит через классификатор, затем поступает в очередь. Далее в узле сканирования очередей выбирается очередь, из которой извлекается кадр и передается в узел управления полосой пропускания, после чего кадр передается в выходной порт.

Подсеть классификатора представлена на рис. 3. После поступления кадра в позицию *Buffer in* переход *Classifier* проверяет, чтобы во всех позициях (*Queue 1 in* ... *Queue 4 in*), связанных с очередями, отсутствовали кадры. Если на входах очередей отсутствуют кадры, то классификатор на основе значения поля *qos* кадра направляет его в одну из очередей, при этом в позицию *Count time* помещается временная метка, соответствующая данному кадру. Далее данная метка передается в буфер вместе с кадром.

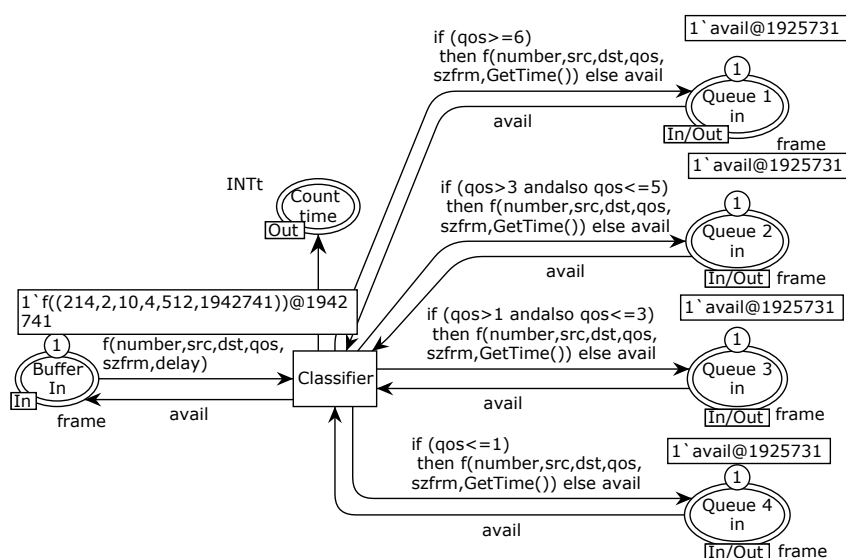


Рис. 3. Подсеть Петри, моделирующая поведение классификатора кадров

Подсеть, моделирующая поведение памяти, представлена на рис. 4. При поступлении кадра в позицию *Queue 1 in* проверяется наличие свободного места в очереди. Если свободное место есть, то в позиции *Size queue 1* измеряется количество хранимых кадров на значение поля *szfrm* кадра, в позицию *Buffer* в хвост очереди добавляется текущий кадр, в позицию *Buffer cnttime* в хвост очереди добавляется временная метка кадра из позиции *Count time*. Если в очереди отсутствовали кадры, то временная метка помещается в

позицию *indicator1*. Если в очереди нет свободного места, то кадр отбрасывается, помещаясь в позицию *Garbage collector*.

Когда в очередь поступает сигнал считывания кадра (позиция *start selection 1*), из позиции *Buffer* считывается кадр, стоящий в голове очереди, а в позиции *indicator1* выполняется сброс значений, и кадр помещается в позицию *Queue 1 out*.

Если значение позиции *indicator1* сброшено, то в нее из очереди *Buffer cnttime* помещается временная метка, соответствующая кадру в голове очереди в позиции *Buffer*.

Позиция *Output data queue* используется для сбора статистических данных о состоянии очереди.

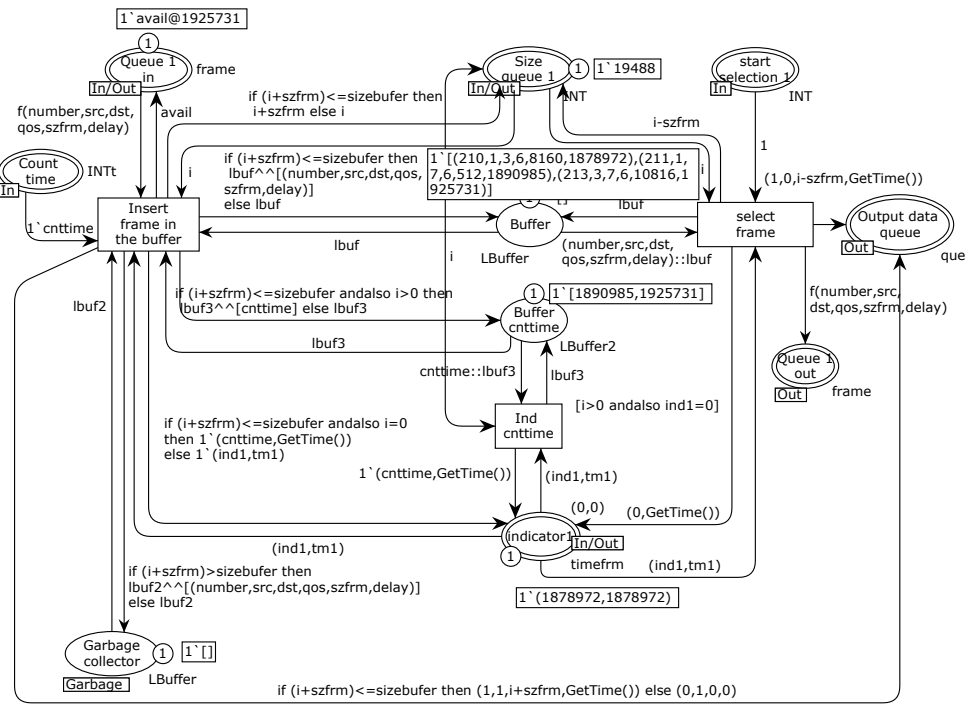


Рис. 4. Подсеть Петри, моделирующая поведение буфера очереди

Как уже было сказано, предлагаемый алгоритм комбинирует алгоритмы RR и TSS, поэтому модель узла сканирования очередей предлагается строить, совместив ранее рассмотренные модели DRR [11] и DTSS [4].

На рис. 5 представлена модель, обеспечивающая поочередное функционирование двух алгоритмов. При поступлении маркера в позицию *OutPort Free* в зависимости от значения маркера в позиции *RR* срабатывает либо переход, отвечающий за циклическое сканирование очередей *scan\_RR*, либо переход, отвечающий за обслуживание на основе временной селекции кадров *scan\_TSS*. Позиция *Number Queue* указывает, как очередь должна быть обработана при циклическом обслуживании. Позиции *indicator1* ... *indicator4*, *Size queue 1* ... *Size queue 4* содержат временные метки и размер буферов для оче-

редей с первой по четвертую соответственно. Позиции *start selection 1 ... start selection 4* связаны с очередями и передают маркер считывания кадра в выбранную очередь.

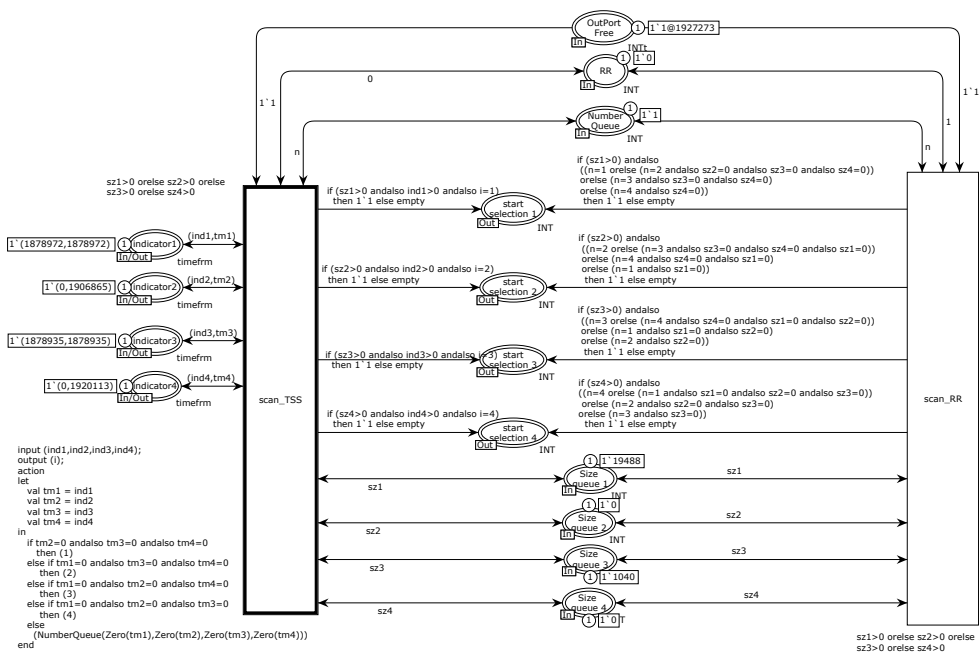


Рис. 5. Подсеть Петри, моделирующая алгоритм сканирования очередей DRR-TSS

Обнуление значения в позиции *counter paks* выполняется при условии, что текущим является алгоритм сканирования TSS. Если текущим является алгоритм RR, то обнуление выполняется при условии, что передана вся информация в рамках данного подсеанса или если очередь пуста или была выработана текущая выделенная квота.

Изменение значения маркера в позиции *RR* выполняется в двух случаях:

- если значение маркера равно 0, что соответствует алгоритму TSS, то значение меняется на 1;
- если выбран алгоритм сканирования RR, значение маркера равно 1, то обнуление выполняется, если выработан объем данных в рамках данного подсеанса.

Изменение значения в позиции *number queue* выполняется при условии, что текущий алгоритм сканирования – RR и была выработана выделенная квота данных, либо очередь пуста.

Изменение значения маркера в позиции *deficiency* выполняется независимо от алгоритма сканирования очередей, что позволяет обеспечить учет текущего дефицита при передаче как разделительного, так и очередного кадра.

Модель подсети регулирования полосы пропускания первой очереди представлена на рис. 6 и содержит следующие позиции:

- размер очереди в битах («Size queue 1»);
- передаваемый пакет («Queue 1 out»);
- дефицит очереди («deficiency»);



- номер следующей обрабатываемой очереди («Number Queue»);
- выходной порт коммутатора («Buffer out»);
- триггер применяемого алгоритма сканирования очередей («RR»);
- количество переданных данных в текущем подсеансе обслуживания очереди («counter paks»).

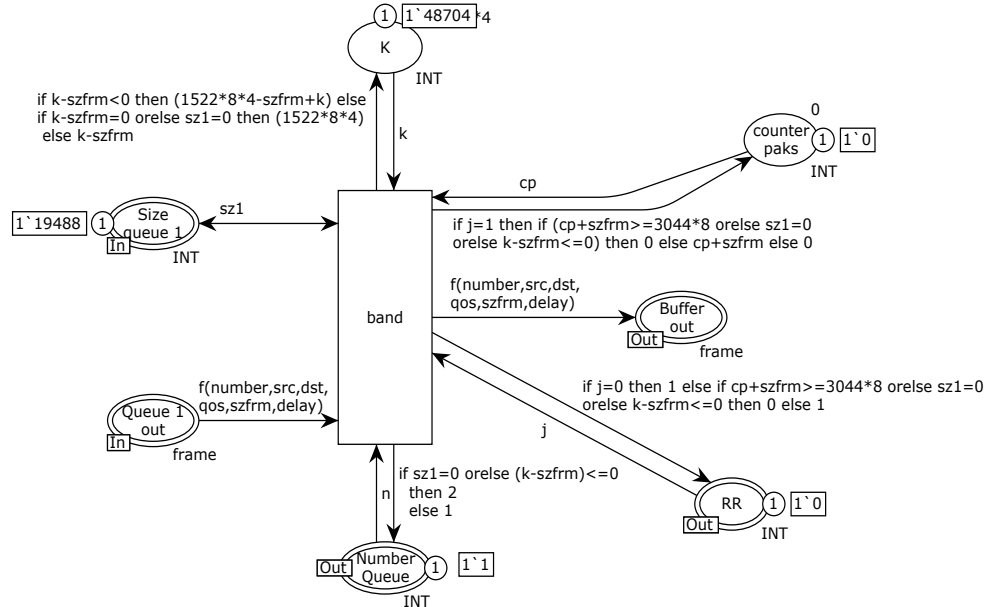


Рис. 6. Подсеть Петри, моделирующая узел регулирования полосы пропускания

После того, как кадр передается в позицию *Buffer out*, срабатывает переход *Port* подсети *OutPort* (рис. 7). При срабатывании перехода маркер из позиции *Buffer out* перемещается в позицию *outport*, значение маркера с позиции *pac's Count* увеличивается на 1, а в позицию *OutPort Free* помещается маркер, разрешающий сканирование очередей. Маркер в позицию *OutPort Free* передается с задержкой, равной  $96+64+\text{размер кадра TI}$ , где 96 – межпакетный интервал, 64 – заголовок кадра в стандарте IEEE 802.1q/p.

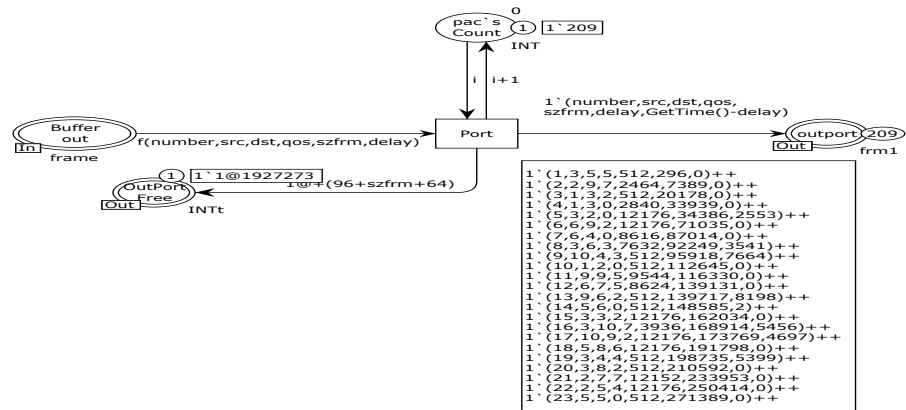


Рис. 7. Подсеть Петри, моделирующая поведение выходного порта

### Результаты моделирования алгоритма

Модель входного трафика записывалась в файл с помощью специальной сети Петри [12] и представляет собой комбинацию потока кадров, принадлежащих с равной вероятностью 4 классам QoS и следующих через временные интервалы, распределенные по экспоненциальному закону со средним значением периода поступления, и квазирегулярный поток с редкими кадрами максимальной длительности. Коэффициент загрузки канала равен 0.8. Модельное время выражено в тактовых интервалах TI, каждый из которых равен 100 нс для Ethernet или 10 нс для Fast Ethernet. Случайная длина кадров имеет следующее распределение, соответствующее реальным сетям: по 25 % от всех кадров имеют минимальную и максимальную длину (соответственно TI 512 и 12176), длина остальных равномерно распределена в оставшемся диапазоне.

Кадры имеют метки QoS от 0 до 7. Формируется 4 класса трафика, которым выделяются соответствующие очереди, и устанавливаются следующие значения полосы пропускания: очередь с QoS 0-1 – 10 %, QoS 2-3 – 20 %, QoS 4-5 – 30 %, QoS 6-7 – 40 % от выходной полосы, т.е. генерируемый трафик класса QoS 0-1 превышает выделенную для него полосу в два раза, QoS 2-3 согласован с выделенной полосой пропускания, а QoS 4-5 и 6-7 недогружают выделенные полосы. Параметры квот управления полосой пропускания заданы 12176, 24352, 36528 и 48704 TI для DRR-алгоритмов. Длительности подсеансов для DRR-TSS могут варьироваться.

Описанный выше трафик обрабатывался также с помощью подсетей, моделирующих алгоритмы DRR и DTSS.

Средние и максимальные значения джиттера для трафика различных классов QoS при различных алгоритмах диспетчеризации, а также их вариация сведены в табл. 1.

Таблица 1

Характеристики джиттера для алгоритмов DRR, DTSS, DRR-TSS

Классы трафика	Среднее значение джиттера, TI			Максимальное значение джиттера, TI		
	DRR	DTSS	DRR-TSS	DRR	DTSS	DRR-TSS
QoS 0-1	39 520	27 579	27 110	44 6705	196 943	201 899
QoS 2-3	23 460	26 714	25 049	166 584	199 584	187 168
QoS 4-5	18 976	22 439	24 721	107 521	141 382	179 313
QoS 6-7	18 972	21 691	23 041	89 353	135 998	159 643
Вариация средних значений	20 548	5528	4069	357 352	63 586	42 256

Анализ результатов, отображенных в табл. 1, показывает, что для алгоритмов DTSS и DRR-TSS по сравнению с алгоритмом DRR вариация средней задержки уменьшилась в 3,72 и 5,05 раза, а вариация максимальной задержки уменьшилась в 5,62 и 8,46 раза соответственно.

Таким образом, алгоритм диспетчеризации DRR-TSS обеспечивает более справедливое распределение полосы пропускания между трафиком различных классов по сравнению с TSS.

Было проведено исследование влияния длительности подсеанса непрерывной передачи данных на эффективность алгоритма DRR-TSS. Зависимость вариации средних значений джиттера от длительности подсеанса непрерывной передачи данных изображена на рис. 8. На основе полученных данных можно сделать вывод, что минимальные значения вариаций достигаются при длительности подсеанса непрерывной передачи данных, равной 6576 ТИ, что приблизительно равно среднему размеру кадра.

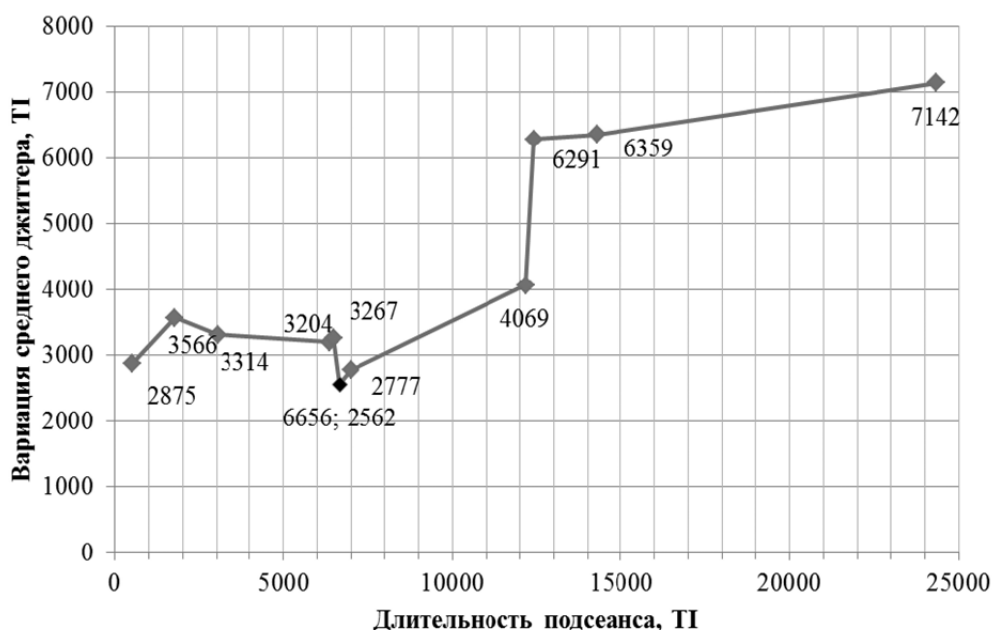


Рис. 8. Зависимость вариации среднего джиттера от длительности подсеанса

### Выводы

1. Разработана цветная иерархическая временная сеть Петри, моделирующая адаптивный алгоритм дефицитного циклического обслуживания с временной селекцией кадров DRR-TSS в коммутаторе с поддержкой QoS.

2. Проведенное с использованием разработанной модели сравнительное исследование эффективности алгоритма DRR-TSS и ранее известных алгоритмов показало, что новый алгоритм эффективнее алгоритма DRR более чем в 5 раз, а алгоритма DTSS – более чем в 1,3 раза, что подтверждено экспериментально.

Найдено оптимальное значение длительности подсеанса непрерывной передачи данных в алгоритме DRR-TSS, при котором достигается минимальная вариация средних и максимальных значений очередей, приблизительно равное средней длине кадра.

### **Библиографический список**

1. Олифер, В. Г. Компьютерные сети. Принципы, технологии, протоколы / В. Г. Олифер, Н. А. Олифер. – 4-е изд. – СПб. : Питер, 2010. – С. 943.
2. Вегешна, Ш. Качество обслуживания в сетях IP / Ш. Вегешна. – СПб. : Вильямс, 2003. – С. 368.
3. Кизиллов, Е. А. Учет времени поступления кадров для управления очередями в коммутаторе / Е. А. Кизиллов, Н. Н. Коннов, В. Б. Механов, К. И. Никишин // Телематика-2014 : труды XXI Всерос. науч.-метод. конф. – СПб. : Университет ИТМО, 2014. – С. 134–136.
4. Scheduling queues in the Ethernet switch, considering the waiting time of frames / E. Kizilov, T. Konnov, K. Nikishin, D. Pashchenko, D. Trokoz // MATEC Web of Conferences. – 2016. – Vol. 44. – P. 01011-p.1-01011-p 5.
5. Кизиллов, Е. А. Анализ алгоритмов динамической диспетчеризации очередей в коммутаторах с поддержкой QoS / Е. А. Кизиллов, Н. Н. Коннов, Д. В. Патунин // Новые информационные технологии и системы : сб. науч. ст. XIII Междунар. науч.-техн. конф. (г. Пенза, 23–25 ноября 2016 г.). – Пенза : Изд-во ПГУ, 2016. – С. 166–169.
6. Крылов, В. В. Теория телетрафика и ее приложения / В. В. Крылов, С. С. Самохвалова. – СПб. : БХВ, 2005. – С. 288.
7. Jensen, K. Coloured Petri Nets: modelling and validation of concurrent systems / K. Jensen, L. Kristensen. – Springer-Verlag, 2009. – P. 384.
8. Zaitsev, D. A. Switched LAN simulation by colored Petri nets / D. A. Zaitsev // Mathematics and Computers in Simulation. – 2004. – Vol. 65, № 3. – P. 245–249.
9. Kizilov, E. Modeling of QoS in the industrial Ethernet switches / E. Kizilov, N. Konnov, D. Pashchenko, D. Trokoz // The 5th International Workshop on Computer Science and Engineering-Information Processing and Control Engineering (WCSE 2015-IPCE). (April 15–17, 2015). – Moscow, Russia, 2015. – P. 185–190.
10. Олифер, Н. А. Базовые технологии локальных сетей. Центр информационных технологий / Н. А. Олифер, В. Г. Олифер // Библиотека портала CIT Forum. – URL: <http://citforum.ru/nets/protocols2>
11. Механов, В. Б. Моделирование цветными сетями Петри обслуживания очередей алгоритмом WRR / В. Б. Механов, Е. А. Кизиллов // Новые информационные технологии и системы : тр. IX Междунар. науч.-техн. конф. (г. Пенза, 9–10 ноября 2010 г.) : в 2 ч. – Пенза : Изд-во ПГУ, 2010. – Ч. 1. – С. 67–73.
12. 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.

---

**Кизиллов Евгений Александрович**

аспирант,

Пензенский государственный университет

E-mail: [jk6001@yandex.ru](mailto:jk6001@yandex.ru)

**Kizilov Evgeny Alexandrovich**

postgraduate student

Penza State University

**Коннов Николай Николаевич**

кандидат технических наук, профессор,

кафедра вычислительной техники,

Пензенский государственный университет

E-mail: [knnpnz@mail.ru](mailto:knnpnz@mail.ru)

**Konnov Nikolay Nikolayevich**

candidate of technical sciences, professor,

sub-department of computer engineering,

Penza State University

**Патунин Дмитрий Васильевич**  
студент,  
Пензенский государственный университет  
E-mail: dvpatunin@gmail.com

**Patunin Dmitry Vasilyevich**  
student,  
Penza State University

---

УДК 681.31

**Кизилов, Е. А.**

**Моделирование адаптивной диспетчеризации очередей в коммутаторе с поддержкой QoS** / Е. А. Кизилов, Н. Н. Коннов, Д. В. Патунин // Модели, системы, сети в экономике, технике, природе и обществе. – 2017. – № 2 (22). – С. 170–182.