

## РАЗРАБОТКА МОДЕЛИ АЛГОРИТМА «СКОЛЬЗЯЩЕГО» ОКНА ЦВЕТНЫМИ ВРЕМЕННЫМИ СЕТЯМИ ПЕТРИ

*Коннов Николай Николаевич*

*Канд. техн. наук, профессор кафедры «Вычислительная техника» Пензенского  
государственного университета, г. Пенза*

*E-mail: [knn@pnzgu.ru](mailto:knn@pnzgu.ru)*

*Домнин Александр Львович*

*аспирант кафедры «Вычислительная техника» Пензенского государственного  
университета, г. Пенза*

*E-mail: [a@pnzgu.ru](mailto:a@pnzgu.ru)*

## DEVELOPMENT OF MODEL THE “SLIDING” WINDOW ALGORITHM USING COLORED PETRI NETS

*Nikolai Konnov*

*candidate of engineering sciences, professor of Penza state university, Penza*

*Alexander Domnin*

*postgraduate of the computer engineering department, Penza state university, Penza*

### АННОТАЦИЯ

Приводится цветная временная сеть Петри, моделирующая алгоритм «скользящего» окна для измерения скорости трафика в сетях с коммутацией пакетов.

### ABSTRACT

Provides a timed colored Petri net modeling «sliding» window algorithm for measuring the speed of traffic in packet-switched networks.

**Ключевые слова:** моделирование; трафик; скользящее окно; сети Петри.

**Keywords:** modeling; traffic; «sliding» window; Petri nets.

Измерение скорости и величины пульсации потоков данных является обязательным компонентом реализации любой политики управления трафиком, призванной обеспечить соглашение об уровне обслуживания (SLA) в компьютерных сетях с поддержкой качества обслуживания (QoS). При профилировании трафика выполняется проверка скорости и пульсации трафика на соответствие SLA и его ограничение в случае необходимости. При шейпинге трафика выполняется придание трафику характеристик, соответствующих

соглашению об уровне обслуживания, по возможности - без его ограничения. Профилирование и шейпинг трафика являются механизмами обеспечения качества обслуживания (QoS) [1, 2].

Существуют следующие алгоритмы измерения скорости:

- усреднение на смежных временных интервалах, которое, однако, не позволяет оценивать скорость потоков с высокой пачечностью;
- алгоритмы «дырявого ведра» и «ведра токенов», относительно просто реализуемые и широко используемые для нужд профилирования и шейпинга, но, фактически, выполняющие не измерения текущей скорости потока, а лишь ее допусковый контроль;
- «скользящее» окно, позволяющее наиболее точно измерить текущее значение скорости потока, но для реализации требующее много вычислительных ресурсов.

Выбор наиболее эффективных методов управления качеством обслуживания в корпоративной сети, разработка новых механизмов QoS представляют собой сложную задачу, которая может быть решена средствами предварительного имитационного моделирования. Эффективным средством моделирования телекоммуникационных сетей, не привязанным к конкретному оборудованию, а основанным на математических моделях, является аппарат цветных иерархических временных сетей Петри [3—5]. В настоящей статье предлагается сеть Петри, моделирующая алгоритм «скользящего» окна, т. к. сети, моделирующие другие алгоритмы, рассмотрены в [6, 7].

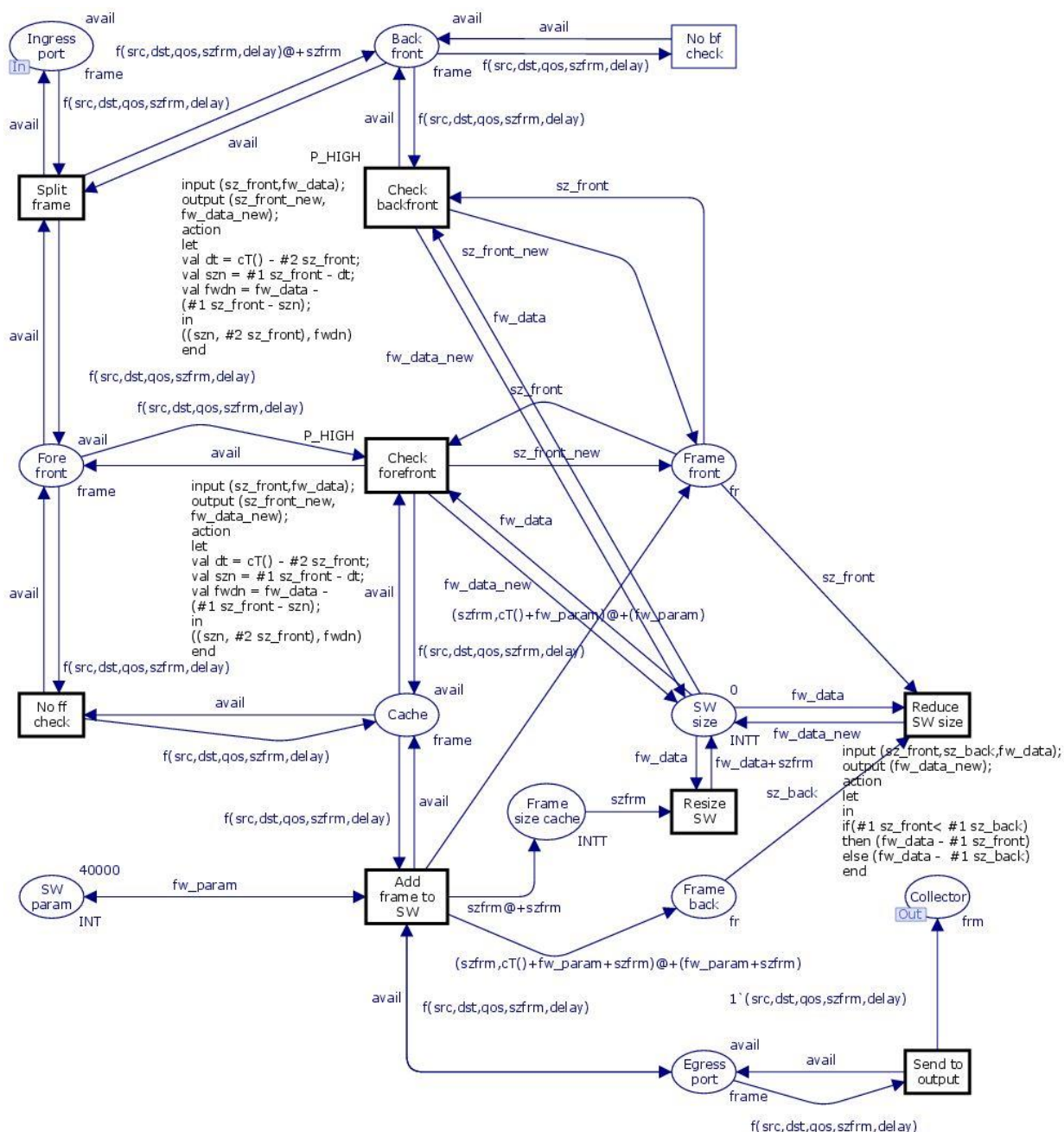
В классическом алгоритме «скользящего» окна требуется пересчет текущей суммы в каждый момент времени. Но для нужд измерения скорости трафика с целью его последующего профилирования и шейпинга не требуется постоянно производить вычисления. Они необходимы только в момент появления и получения очередного фрейма. Используем этот факт при разработке модели, что позволит значительно сократить объем вычислений при моделировании.

Модель алгоритма «скользящего» окна представлена цветными временными сетями Петри (рисунок 1). Разработка модели проводилась в системе CPN Tools, которая позволяет моделировать различные аспекты поведения сложных телекоммуникационных систем, благодаря своему функционалу и широкому набору инструментов.

Чтобы не возникало путаницы в терминах, определим, что длительность «скользящего» окна — это период времени в битовых интервалах, за который идет подсчет размеров попавших в этот период фреймов. В модели этот параметр является заданным изначально и не изменяется во времени. Размер «скользящего» окна — сумма размеров попавших в длительность этого окна фреймов на конкретный момент времени.

Поток фреймов моделируется маркерами, цвет которых соответствуют формату фрейма, описанному стандарте IEEE 802.1Q. Так как сеть временная, за каждым маркером закреплена метка времени, используемая в процессе моделирования, которая содержит время появления этого маркера в модели. Единица модельного времени равна одному битовому интервалу. В данном случае время связано с размером фреймов, которое задано в битах, а также — размером «скользящего» окна, указанное в битовых интервалах и др. Если возникнет необходимость вложить другой смысл в единицу модельного времени (байтовый интервал, миллисекунда, секунда), то следует пропорционально изменить связанные со временем количественные параметры модели.

Краткое описание реализации алгоритма следующее: сохраняем в определенных позициях информацию о фреймах, попадающих в «скользящее» окно. С течением времени информация в этих позициях изменяется — информация о предыдущих фреймах, уже не попадающих в длительность окна, удаляется, а информация о вновь поступивших — добавляется.



**Рисунок 1. Сеть Петри, моделирующая алгоритм скользящего окна**

Очередной маркер, моделирующий фрейм Ethernet, переходит в позицию *Ingress Port*. Затем происходит срабатывание перехода *Split frame*. В позицию *Forefront* переходит маркер, содержащий оригинальную информацию о фрейме, который будет использоваться для вычисления размера «скользящего» окна на момент появления фрейма (фрейм не будет учитываться в размере окна). А в позицию *Backfront* — маркер с меткой времени, увеличенной на размер фрейма. Это необходимо для вычисления размера «скользящего» окна на момент получения фрейма (фрейм будет учитываться в размере окна). После

этого для маркера в *Backfront* происходит срабатывание перехода *No bf check* или *Check backfront*, а для маркера в *Forefront* — срабатывает либо переход *No ff check*, либо *Check forefront*. Переходы *Check backfront* и *Check forefront* нужны для обработки случаев, когда при появлении очередного фрейма (*Check forefront*) или получении его (*Check backfront*) задний фронт «скользящего» окна попадает в период времени, в который происходил прием уже поступившего фрейма. В таких ситуациях размер «скользящего» окна на текущий момент должен быть дополнительно скорректирован.

Срабатывание переходов *No bf check* или *Check backfront* приводит к удалению маркера из позиции *Back Front*, а при срабатывании *No ff check* либо *Check forefront* маркер переходит в промежуточную позицию *Cache*. Далее, происходит срабатывание перехода *Add Frame to SW*, реализующего сохранение информации о фрейме в «скользящем» окне. С этим переходом связаны позиция *SW param*, хранящей длительность «скользящего» окна, позиция *Frame front*, в которой хранится информация о времени появлении фрейма и его размере, позиция *Frame back*, в которой хранится информация времени получения фрейма и его размере, позиции *Frame size cache*, используемой для отложенного на определенный промежуток времени изменения значения в позиции *SW size*, которая хранит размер «скользящего» окна на какой-то предыдущий момент времени. Так как размер окна увеличивается на размер появившегося фрейма только после приема этого фрейма, следовательно, необходимо смоделировать изменение значения в позиции *SW size* через период времени равный размеру фрейма. Это как раз реализуется позицией *Frame size cache* и переходом *Resize SW*.

Переход *Reduce SW size*, связанный с позициями *SW size*, *Frame front* и *Frame back* реализует удаление информации из этих позиций о фреймах, уже не попадающих в длительность «скользящего» окна.

Точный размер «скользящего» окна вычисляется в моменты времени поступления и получения очередного фрейма, так как основные вычисления

размера приходится именно на эти события. Размер окна не пересчитывается каждую единицу модельного времени.

Тестирование модели на равномерном (фреймы одинакового размера поступают через равные промежутки времени) и стохастическом (фреймы разного размера поступают через произвольные промежутки времени) видах трафика позволяют сделать вывод о корректности ее функционирования.

### **Список литературы:**

1. Вегешна Ш. Качество обслуживания в сетях IP. М.: Издательский дом «Вильямс», 2003.
2. Домнин А.Л., Кизилов Е.А., Пушкарев В.А. Моделирование механизмов qos в коммутаторах Ethernet цветными сетями Петри // Сборник статей Участников Всероссийского конкурса научных работ студентов и аспирантов «Телематика '2010: телекоммуникации, веб-технологии, суперкомпьютин» СПб: СПбГУ ИТМО, 2010.
3. Кучерявый Е.А. Управление трафиком и качество обслуживания в сети Интернет СПб.: Наука и техника, 2004.
4. Механов В.Б. Применение сетей Петри для моделирования механизмов обеспечения QoS в компьютерных сетях // Материалы междунаро. симпозиума “Новые информационные технологии и менеджмент качества (NIT&MQ’2010)” ФГУ ГНИИ ИТТ “Информика” М.: ЭГРИ, 2010.
5. Механов В.Б., Домнин А.Л. Моделирование алгоритмов управления полосой пропускания цветными сетями Петри // Труды IX Международ. научно-технической конференции “Новые информационные технологии и системы”: в 2 ч. Пенза: Изд-во ПГУ, 2010
6. Jensen K., Kristensen L. Coloured Petri Nets: modelling and validation of concurrent systems. Springer-Verlag, 2009.
7. Zaitsev D.A. Switched LAN simulation by colored Petri nets // Mathematics and Computers in Simulation, — vol. 65, — № 3, — 2004.