

## АЛГОРИТМЫ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ СЕТЕВЫХ СИСТЕМ НА ОСНОВЕ (*MIN*,+) ФИЛЬТРАЦИЙ

Чубейко С.В.<sup>1</sup>

<sup>1</sup>ФГБОУ ВПО «Ростовский государственный университет путей сообщения», Ростов-на-Дону, Россия (344038, Ростов-на-Дону, пл. им. Ростовского Стрелкового полка Народного Ополчения, 2), e-mail: greyc@mail.ru

---

Статья посвящена новому подходу к разработке имитационных моделей сетевых систем с *IP*-трафиком. Основное внимание уделено методам оценки производительности и качеству обслуживания в *IP*-сетях и базирующихся на них распределенных сетевых системах. Описаны принципы архитектур интегрированного, *IntServ* и дифференцированного, *DiffServ* управления качеством обслуживания в компьютерных сетях. Представлен способ расчета характеристик системы с гарантированным обслуживанием в архитектуре услуг *IntServ*. Рассмотрен подход к разработке математических моделей сетевых систем с гарантированным обслуживанием на основе (*min*,+) алгебраических структур. Представлены методы «*network calculus*», являющиеся развитием алгебраического подхода к описанию систем обслуживания для оценки их производительности. Даны характеристики алгебраической структуры диоида, являющейся адекватным представлением моделей (*min*,+) фильтрации сетевого трафика. Представлена имитационная модель сетевой системы с (*min*,+) фильтрацией. Описаны операторы и конструкции языка функционального программирования для реализации предложенной имитационной модели. Выполнена программная реализация уравнения сетевого баланса на функциональном языке программирования.

---

Ключевые слова: качество обслуживания, теория массового обслуживания, имитационное моделирование, функциональное программирование

## ALGORITHMS AND SOFTWARE FOR NETWORK SYSTEMS SIMULATION BASED ON (*MIN*,+) FILTERING

Chubeyko S.V.<sup>1</sup>

<sup>1</sup>Rostov State Transport University, Rostov-on-Don, Russia, (344038, Rostov-on-Don, square n.a. Rostovskogo Strelkovogo polka Narodnogo Opolchenija, 2), e-mail: greyc@mail.ru

---

The article is devoted to the development of a new approach of simulation models of network systems with *IP*-traffic. Emphasis is placed on methods for assessing the performance and quality of service in *IP*-based networks, and they distributed network systems. The principles of integrated architectures, *IntServ* and differential, *DiffServ* *QoS* in computer networks is done. A method for calculating the characteristics of the system with a guaranteed service architecture services *IntServ* is provided. An approach to the development of mathematical models of network systems with guaranteed service on the basis of (*min*, +) algebraic structures is proposed. Methods of «*network calculus*», the development of an algebraic approach to the description of service systems to estimate their performance are presented. The characteristics of the algebraic structure dioid is adequate representation of the model (*min*, +) filtering of network traffic is done. A simulation model of the network system (*min*, +) filtering is presented. Operators and programming construction on functional programming language for the implementation of the proposed simulation model are described. A software implementation of the network balance equation on functional programming language is implemented.

---

Keywords: quality of service, queueing theory, simulation, functional programming

### Введение

Под сетевыми системами в рамках данной статьи будем понимать совокупность распределенных программно-управляемых систем обменивающихся информационным трафиком посредством стека *IP*-протокола. Такие сетевые системы предоставляют основной спектр нынешних Интернет-услуг и предъявляют различные требования к их качеству, которое принято называть *QoS* (*Quality of Service*, качество обслуживания). Современные механизмы *QoS* опираются на принципы либо интегрированной архитектуры обслуживания

(*Integrated Services, IntServ*), которая описана в руководстве *IETF RFC1633*, либо на принципы архитектуры дифференцированного предоставления обслуживания (*Differentiated Services, DiffServ*), которая описана в руководстве *IETF RFC2474*. Архитектура *IntServ* обеспечивает сетевое обслуживание в двух режимах: 1) контролируемой загрузки (*Controlled Load, CL*) для обеспечения гарантированного трафика к сетевым адаптивным приложениям «мягкого» реального времени; 2) гарантированного обслуживания (*Guaranteed Service, GS*) с обеспечением заранее заданной полосы пропускания к сетевым неадаптивным приложениям «жесткого» реального времени.

В случае *CL* для настройки услуг не используются управляющие параметры *QoS*, такие как уровень потерь пакетов и время задержки поступления данных. Доставка трафика сетевому приложению зависит лишь от времени задержки ретрансляции данных сетевым оборудованием, а также от всплескового характера самого трафика. Время минимальной задержки в режиме *CL*, будет стремиться к максимальной величине всплескового (в количественном отношении) объема трафика. В случае обеспечения уровня услуг *GS* сетевому приложению выделяется гарантированная полоса пропускания трафика и регулируется граничное значение времени задержки обслуживания. Величина максимального времени задержки зависит от параметров канала передачи данных, от дисциплины, интенсивности обслуживания и определена в руководстве *IETF RFC2212* и алгоритмом маркерной корзины (*Token Bucket*):

$$D = \begin{cases} \frac{(b-M)(p-R)}{R(p-r)} + \frac{(M+C_{tot})}{R} + D_{tot}, & p > R \geq r \\ \frac{(M+C_{tot})}{R} + D_{tot}, & r \leq p \leq R \end{cases} \quad (1)$$

где  $r$  – интенсивность поступления маркеров в корзину, байтов/с;

$p$  – максимальная интенсивность поступления данных, байтов/с;

$R$  – интенсивность обслуживания, байтов/с;

$b$  – размер маркерной корзины, байтов;

$M$  – максимальный размер поступающего блока данных (*IP*-дейтаграммы), байтов;

$C$  – максимальная девиация размера поступающего блока данных, байтов;

$D$  – максимальная девиация времени при обработке данных устройствами обслуживания, с.

В дополнение к рассмотренной, архитектура *DiffServ* осуществляет разделение трафика на классы и приоритеты. В целом, при управлении качеством обслуживания в *IP*-сетях основное внимание уделяется обеспечению минимальных (или максимальных) гарантированных значений параметров управления трафиком: полосой пропускания,

временной задержкой, девиацией временной задержки и уровнем потерь блоков данных. Поэтому одними из современных адекватных математических моделей сетевых систем с гарантированным обслуживанием являются модели, построенные на  $(min,+)$  или  $(max,+)$  алгебраических структурах, а развитие такого подхода для оценки производительности сетевых систем, называется «*network calculus*» [9]. Данное направление базируется на подходах теории алгебраических структур – полуколец с каноническим упорядочиванием (идемпотентных полуколец, диоидов) [8] идемпотентного анализа [3], моделирования дискретных динамических систем [6]. Моделированию и оценке гарантированной производительности и надежности сетевых систем посвящены работы с участием автора [1,2,4,5].

### Элементы теории: $(min,+)$ алгебра и *network calculus*

Основной алгебраической структурой в рассматриваемом подходе является структура, называемая «*диоид*» [8]. Диоид  $D$  определяется, как числовое множество  $E$ , снабженное двумя замкнутыми операциями: «сложение,  $\oplus$ » и «умножение,  $\otimes$ »,  $D = (E, \oplus, \otimes)$ . Операции в диоиде должны подчиняться следующим правилам: 1) ассоциативность по сложению и умножению; 2) коммутативность по сложению; 3) дистрибутивность по умножению, относительно сложения; 4) существование нейтральных элементов для обеих операций: нулевого (по сложению и поглощающего по умножению) и единичного; 5) идемпотентность сложения; 6) отсутствие обратных элементов.

В рамках статьи будем рассматривать операции диоида  $\otimes = \min$ ,  $\oplus = +$  следующим образом. Для функций  $F \in \mathbb{R}^+ \cup \{+\infty\}$ ,  $F = \begin{cases} y(s,t) \geq 0, & s \leq t; \\ y(s,t) = 0, & s > t. \end{cases}$ , а для  $a(s,t), b(s,t) \in F$ ,

$$(a(s,t) \oplus b(s,t)) = \min\{a(s,t), b(s,t)\}, \quad (a(s,t) \otimes b(s,t)) = \begin{cases} \min\{a(s,\tau) + b(\tau,t)\}, & s \leq t \\ & s \leq \tau \leq t \\ 0, & s > t \end{cases}.$$

Рассмотренная алгебраическая структура, диоид позволяет предложить достаточно эффективный подход к аналитическому и имитационному моделированию сетевых систем, наряду с теорией массового обслуживания (ТМО). Существенные отличия подходов, в целом, состоят в том, что в ТМО для расчета числовых характеристик необходимо построить аналитические или адекватные им приближенные модели входных потоков и элементов обслуживания, а в алгебраическом подходе «*network calculus*» исследуются граничные условия для кумулятивных функций, описывающих как входной поток, так и поток обслуживания. Таким образом, в последнем подходе, элемент обслуживания служит фильтром входного потока, придавая выходному потоку ту, или иную заданную форму.

Отметим, что функции, описывающие граничные условия могут быть как детерминированными, так и случайными [9].

Далее рассматриваем детерминированные функции фильтрации входного трафика. На рис. 1 показана иллюстрация к такому подходу, где  $x$  – входной фильтрованный поток (входная кумулятивная функция),  $y$  – поток обслуживания (выходная кумулятивная функция), а часть обозначений соответствует формуле (1). Буфер сетевых данных принимает входной поток  $p$ , который на физическом уровне IP-сети является последовательным байтовым потоком, составляющим накапливаемый (следовательно, кумулятивный) объем данных, и, в целом – входной трафик. Далее поток подвергается  $\alpha$ -фильтрации, то есть в техническом отношении при информационной перегрузке, переполнении входных буферов, различных неисправностях программно-аппаратного обеспечения, при выполнении функций *QoS*, разделении и трафика на классы и приоритеты происходит отсекаание части трафика, подобно прохождению сигналов через фильтры. Так формируется входной фильтрованный поток  $x$ , поступающий на устройство обслуживания, которое в свою очередь также имеет  $\beta$ -фильтр с характеристиками, зависящими от применяемого в сетевом устройстве алгоритма маркерной корзины (*Token Bucket*). В итоге формируется выходной поток  $y$ .

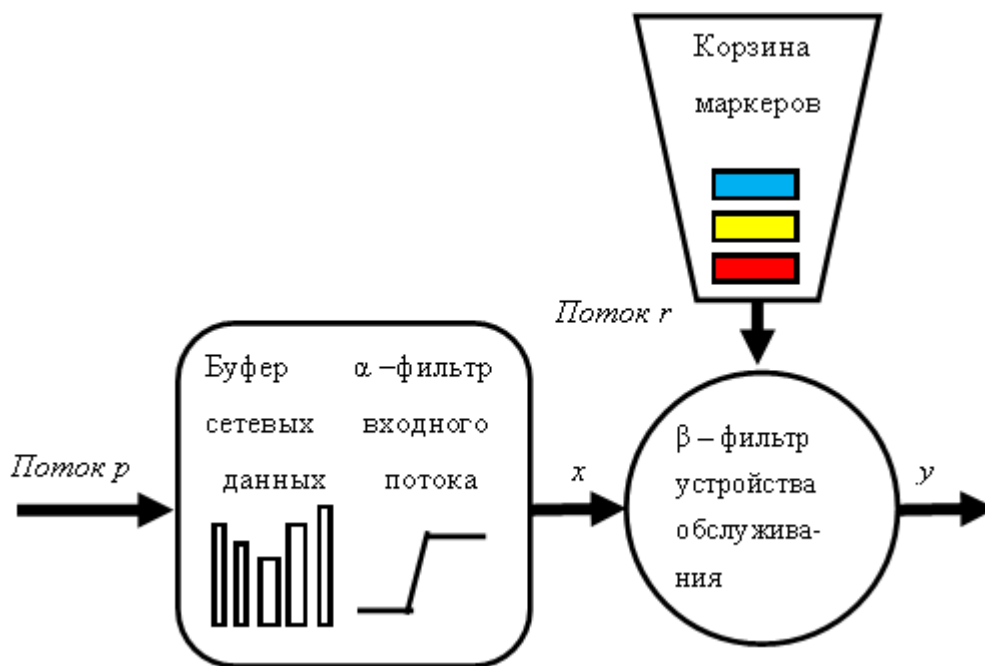


Рис. 1. Имитационная модель сетевой системы с  $(min,+)$  фильтрацией

Математической моделью рассматриваемого  $(min,+)$  подхода являются уравнения:

$$\begin{aligned}
 &\text{для } s, t \in \mathbb{Z}^+, s \leq t, \alpha: \mathbb{R}^+ \cup \{+\infty\} \\
 &x(t) - x(s) \leq \min_{0 \leq s \leq t} \{ \alpha(t-s) \}; \quad (2) \\
 &\text{для } \forall t \in \mathbb{Z}^+, \exists s \in \mathbb{Z}^+, s \leq t
 \end{aligned}$$

$$y(t) \geq \min_{0 \leq s \leq t} \{x(s) + \beta(t-s)\}. \quad (3)$$

Поиск решения уравнений (2) и (3) приводит к известной из теории цифровой обработки информации задаче о вычислении интеграла свертки:

$$(x \circ y)(t) = \int_{0 \leq s \leq t} x(s)y(t-s) ds, \quad (4)$$

а в терминах рассматриваемого подхода

$$(x \circ y)(t) = \min_{0 \leq s \leq t} \{x(s) + y(t-s)\}. \quad (5)$$

### Метод расчета $(\min,+)$ интегральной свертки

В работе автора [5] предложен подход к реализации алгоритмов *network calculus* на функциональном языке программирования *J* и были даны необходимы описания входных кумулятивных функций. Рассмотрим дальнейшее описание имитационных алгоритмов расчета  $(\min,+)$  интегральной свертки на функциональном языке программирования *J*.

В вышеуказанном языке программирования есть оператор  $\min$ , обладающая всеми необходимыми для названных ранее операций «сложение,  $\oplus$ » и «умножение,  $\otimes$ » свойствами. Оператор  $\min$  может быть применим, как к скалярным, так и к матричным величинам. Для скалярных величин можно в среде программирования *J* ввести с клавиатуры, например: **4 min 5, ((35 min 24) min 4)**. Можно проверить свойство дистрибутивности операции: **((3 min 2) + 4) = (3 + 4) min (2 + 4)**, результатом будет вывод на экран числа 1, что означает истинность выражения (если результат равен 0, то выражение не является истинным). Матричные операции в  $(\min,+)$  алгебре

$$\begin{pmatrix} p_1 \\ p_2 \end{pmatrix} \oplus \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = \begin{pmatrix} \min(p_1, q_1) \\ \min(p_2, q_2) \end{pmatrix} \text{ и } \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} \otimes \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix} = \begin{pmatrix} \min(p_1 + m_{11}, p_2 + m_{12}) \\ \min(p_2 + m_{21}, p_2 + m_{22}) \end{pmatrix}$$

в языке программируются соответственно: **P min"0 Q** и **P (min"1 .+) M**.

Для расчетов (поточечного) минимума двух функций  $(f + g)(t)$  и их свертки  $(f \circ g)(t)$  в языке *J* нужно запрограммировать соответственно: **min@(f@t, g@t)** и **min@(f@s, g@(t-s))**.

Теперь опишем главный результат статьи – программную реализацию основного уравнения баланса сетевой модели. Уравнение Линдли [7] (аналогично ТМО):

$$y(t+1) = (y(t) + x(t+1) - x(t) - c)^+$$

На функциональном языке *J* реализовано следующей программой:

```
cocurrent < 'LINDLEY'
ylist =: rhs0
c =: >@lhs1
xlist =: >@lhs2
```

```
ind =: <:@#@ylist
xnext =: ind { xlist
yprev =: {:@ylist
ynext =: max0 @ xnext + yprev + c
cocurrent <'base'
lindley_z_ =: ylist_LINLEY_,ynext_LINDLEY_
```

## Заключение

Анализ в области имитационного моделирования сетевых систем показал, что методы, построенные на основе алгебраического подхода с использованием кумулятивных функций входных и выходных потоков, представляются очень перспективными и вполне соответствуют технологическим особенностям оборудования, применяемого в современных *IP*-сетях. Выбор в качестве программной реализации весьма нераспространенного функционального языка программирования *J* и реализованные на нем алгоритмы имитационного моделирования также показали эффективность исполнения алгоритмов за счет внутренней рекурсивной организации самого языка программирования. В связи с этим, можно сделать вывод об адекватности применения математического и программного инструментария для имитационного моделирования сетевых систем для оценки их производительности.

## Список литературы

1. Бутакова М.А. Имитационное моделирование процессов возникновения ошибок для оценки надежности программного обеспечения / Бутакова М.А., Чубейко С.В. // Известия высших учебных заведений. Северо-Кавказский регион. Технические науки. – 2012. – №5 (168). – С.29-34.
2. Гуда А.Н. Прогнозирование надежности программного обеспечения на основе модели неоднородного пуассоновского процесса и бутстреп-методов / Гуда А.Н., Чубейко С.В. // Программные продукты и системы. – 2012. – №3. – С. 131
3. Маслов В.П. Идемпотентный анализ и его применение в оптимальном управлении / Маслов В.П., Колокольцов В.Н. – М.: ФИЗМАТЛИТ. – 1994. – 144 с.
4. Чубейко С.В. Модели и методы сетеметрии в задачах разработки высоконагруженных информационно-управляющих систем на транспорте // Труды Всероссийской научно-практической конференции «Транспорт-2011», май 2011 г. Часть 1. Естественные и технические науки. Рост. гос. ун-т путей сообщения. Ростов н/Д. – 2011. – С. 59-61.
5. Чубейко С.В. Моделирование систем с гарантированным обслуживанием: элементы теории и программная реализация // Сборник научных статей по итогам Международной

заочной научно-практической конференции «Теоретические и практические аспекты развития науки», 14-15 октября 2013 г. – СПб.: Изд-во «КультИнформПресс». – 2013. – С 113-120.

6. Baccelli F. Synchronization and linearity: An algebra for discrete event systems / Baccelli F., Cohen G., Olsder G.J., Quadrat J.-P. – Chichester: Wiley. – 1992. – 514 p.

7. Chang C.-S. Performance Guarantees in Communication Networks. – Springer-Verlag, London. – 2000. – 392 p.

8. Gondran M. Graphs, Dioids and Semirings. New Models and Algorithms. / Gondran M., Minoux M. – Springer Science+Business Media, LLC. – 2008. 384 p.

9. Jiang Y. Stochastic Network Calculus / Jiang Y., Liu Y. – Springer-Verlag, London. – 2008. – 240 p.

#### **Рецензенты:**

Павлов И.В., д.ф.-м.н., профессор, заведующий кафедрой высшей математики Ростовского государственного строительного университета, г. Ростов-на-Дону.

Чернов А.В., д.т.н., профессор, заведующий кафедрой прикладной математики и вычислительной техники Ростовского государственного строительного университета, г. Ростов-на-Дону.