

РАСЧЕТ ПРОПУСКНОЙ СПОСОБНОСТИ УЗЛА КРУПНОМАСШТАБНОЙ СЕТИ С ИСПОЛЬЗОВАНИЕМ LPF И МИМА-АЛГОРИТМА

Ташев Т.Д.,

Институт информационных и коммуникационных технологии БАН

ttashev@iit.bas.bg

Баканов А.С.

Институт психологии РАН

arsb2000@pochta.ru

Аннотация: Настоящая работа посвящена вопросам, связанным с оценкой производительности расчета пропускной способности (ПС) узла крупномасштабной распределенной информационной системы с использованием МИМА-алгоритма и LPF-алгоритма. В рамках проведенных исследований была разработана модель для компьютерного моделирования ПС, описаны полученные результаты для равномерного трафика. В результате сравнения был предложен вариант модификации МИМА-алгоритма с улучшенной ПС.

Ключевые слова: мониторинг, моделирование, крупномасштабные распределенные сети, пропускная способность.

Введение

Проблема вычисления бесконфликтного расписания для коммуникационного узла крупномасштабной информационной сети, с пакетной передачей, относится к важным задачам, которые необходимо решать на этапе проектирования крупномасштабной сети обмена данных [1,2,3,4,5]. Коммуникационный узел для коммутации пакетов с матричным переключателем (коммутатор - Crossbar Switch Node) должен максимально ускорять передачу пакетов данных, в своем коммутационном поле (switching fabric) [6]. В идеальном случае узел передает пакеты между узлами, подключенными к его портам, с той скоростью с которой эти узлы генерируют пакеты, не внося дополнительных задержек и не теряя ни одного пакета. Достижение этой цели обеспечивает бесконфликтное расписание коммутации, вычисляемое управляющим блоком (scheduler) коммутатора [7].

Математически доказано, что задача для вычисления бесконфликтного расписания является NP-полной. Существующие алгоритмы решения удовлетворяют требованиям частично, используя математический аппарат теории массового обслуживания [8], а также генетические алгоритмы [9], матрицы-маски [10], алгоритмы нейронных сетей [11] и др.

Увеличение пропускной способности связанных каналов, а также объемов коммуникационного трафика требует создания новых алгоритмов. Первый шаг при проверке эффективности нового алгоритма для расчета бесконфликтного расписания – это моделирование пропускной способности (ПС) коммутатора при равномерно распределенном входящем трафике (i.i.d. Bernoulli). На следующем шаге выполняется проверка при неравномерно распределенном трафике. Для коммутатора крупномасштабной сети такие проверки надо произвести для коммутационного поля большой размерности (больше 32x32). Это можно рассматривать как крупномасштабное моделирование по сравнению с классическим выбором размеров поля (типа 8x8, 16x16, 32x32) [6].

1 Описание алгоритмов и условий моделирования

Для вычисления бесконфликтного расписания был предложен новый алгоритм, названный МиМа-алгоритм (Minimum of Maxima). Он создан с использованием аппарата Обобщенных сетей [12]. Сущность описана в [2], а его формальное описание опубликовано в [13]. МиМа-алгоритм является представителем класса „весовых” (weight-match [8]) алгоритмов. Описание МиМа-алгоритма выполнено посредством аппарата Обобщенных сетей (ОС). Каждая Обобщенно-сетевая модель имеет графическое представление, для нашего алгоритма оно представлено на рис.1. Представим неформальное описание алгоритма.

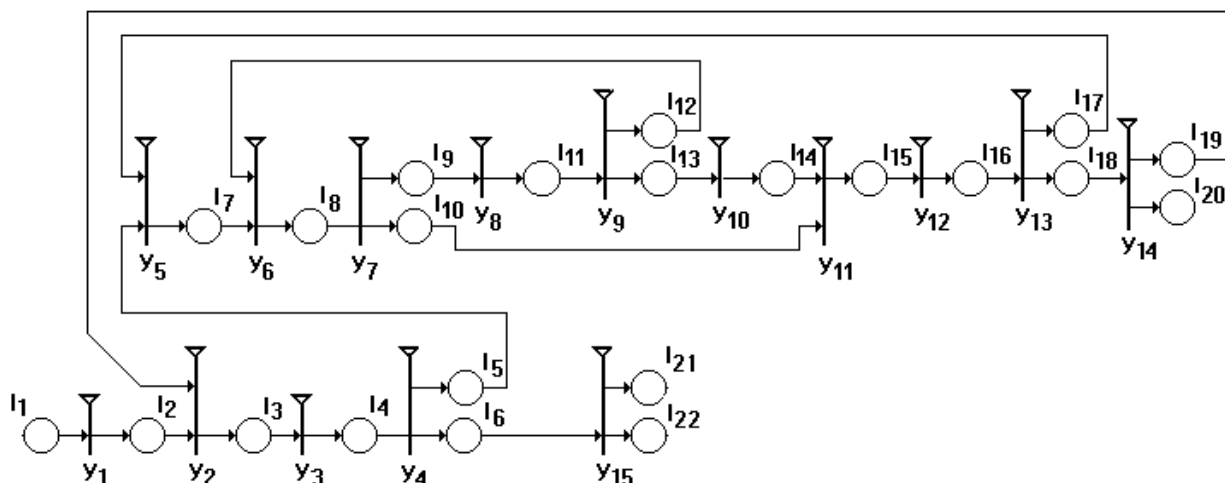


Рис. 1. Графическое представление Обобщенно-сетевой модели МиМа-алгоритма

В качестве начальных данных вводится матрица входящего трафика T , определенная с учетом выбранной модели (равномерно или неравномерно распределенного) трафика [5]. При представлении входящего трафика матрицей T , каждая виртуальная выходная очередь (VOQ) является ее ячейкой t_{ij} , причем значение (число) $t_{ij} = r$ ($r \in \{0, 1, 2, \dots\}$) определяется количеством заявок в виртуальной очереди (число заявок от i -того источника к j -тому приемнику), а число ячеек в строке T отвечает числу виртуальных очередей этого входа (число входов равно числу выходов коммутатора n). Состояние потока заявок на передачи пакетов коммутатором, переключающий $n \times n$ линий своего коммутационного поля, представляется некоторой $(n \times n)$ матрицей T . Ситуация конфликта возникает, когда в любом ряду (или столбце) матрицы T сумма значений его элементов есть число больше единицы [5].

МиМа-алгоритм должен вычислить ряд бесконфликтных матриц $Q_1, Q_2, \dots, Q_{\text{креш}}$ и в результате все заявки матрицы T получают назначение на коммутацию. Тогда формирование искомого ряда матриц Q_m , ($m = \text{креш}$) будет определено.

Работа алгоритма начинается вычислением Вектор-колонны (колонна или столбца „веса конфликтов”), любой элемент которой („вес”) определяется числом конфликтов в каждой строке T . Потом производится вычисление Вектора-строки (строка „веса конфликтов”), любой элемент которой определяется числом конфликтов в каждой колонне T . Вес конфликтов – это сумма заявок на коммутацию в колонне (строке) матрицы T – сумма соответствующих элементов в колонне (строке) T .

В Вектор-строке выбирается максимальный элемент. Он определяет колонну с максимумом конфликтов. В Вектор-колонне выбирается максимальный элемент – он определяет строку с максимумом конфликтов. При наличии заявки в ячейке T пересечения выбранных колонны и строки, присваиваем эту заявку как элемент бесконфликтной матрицы Q_1 (и уменьшаем на единицу значение в выбранной ячейке T). Если ячейка пустая, то выбирается элемент в Векторе-колонне, чье значение – самое близкое к максимальному. При этом выбранный элемент в Векторе-строке остается прежним. На следующем шаге проверяем, есть ли заявки в „точке” пересечения и т.д.

В результате в выбранной колонне T имеем заявку, выбранную для коммутации. Строка и колонна, содержащие выбранную заявку, исключаются из дальнейшего вычисления Q_1 . Следующие элементы Q_1 вычисляются повторением описанной процедуры. Тогда первая матрица бесконфликтного расписания Q_1 будет содержать элементы (заявки) с (суб)-максимальным весом конфликтов в T . Соответственно значения выбранных ячеек T будут уменьшены на единицу.

Следующие бесконфликтные матрицы $Q_2, \dots, Q_{\text{креш}}$ вычисляем аналогичным образом – вычисляем Вектор-колонну (колонна веса конфликтов), ... и так далее. Наконец, при вычислении последней матрицы – $Q_{\text{креш}}$ – останутся только бесконфликтные заявки в матрице T .

Формальное описание ММА-алгоритма приведено в [13]. Модель накапливает информацию о количестве переключений коммутационного поля, может собирать информацию о среднем количестве пакетов (переданных с помощью одного переключения), и др. Анализ модели подтверждает получение бесконфликтного расписания – ряд бесконфликтных матриц Q чья сумма равна матрице T . В настоящей работе производится сравнение МиМа-алгоритма с другими представителями этого класса алгоритмов. Для сравнения был выбран LPF-алгоритм [14].

Выбор LPF-алгоритма, обусловлен тем, что для него показано, что он является оптимальным как для класса „весовых” (weight-match), так и для класса „не-весовых” (size-match) алгоритмов.

Было проведено компьютерное моделирование пропускной способности узла с использованием LPF-алгоритма тем же способом, как и для МиМа-алгоритма [13], чтобы иметь сопоставимые результаты. Для расчета был использован программный пакет Vfort, предоставленный для свободного пользования Институтом прикладной математики РАН [15]. Расчеты выполнены на супер-компьютере "АВИТОХОЛ" Института Информационных и коммуникационных технологий Болгарской Академии Наук (www.hpc.acad.bg/system-1/). Он состоит из 150 HP Cluster Platform SL250S GEN8 servers (with 2 Intel Xeon E 2650 v2 CPUs and 2 Intel Xeon Phi 7120P coprocessors). Компилирование и выполнение производились локально на суперкомпьютере под Операционной системе Red Hat Enterprise Linux for HPC.

В этой работе мы будем использовать упрощенную модель входного трафика – идентичного и независимого по входам, равномерно распределенного по выходам (i.i.d. Bernoulli uniform) [6]. Модель определяется матрицей заявок на коммутацию $T^i(k \times k)$. Представление в виде семейства шаблонов для матрицы $T^i(k \times k)$ (согласно [13]) показано на рис.2. Имеем семейство шаблонов, которые обозначим как $U-1, U-2, \dots, U-i, \dots$. Каждый шаблон определяет матрицу для заданного количества входов/выходов коммутационного поля $(2 \times 2), (3 \times 3), \dots, (k \times k), \dots$. Обозначение $U-i$ используется для трафика типа T^i . Индекс i связан с величиной входного буфера.

На рис.2 показано увеличение размерности коммутационного поля первого шаблона $U-1$ (матрица T^1). Шаблон $U-i$ – общий вид семейства. Оптимальным решением для размерности $(n \times n)$ является: $k_{opt} = i \cdot n$. Пропускная способность вычисляется как $ПС = k_{opt} / \text{креш}$.

$$T_{(2 \times 2)}^1 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad T_{(3 \times 3)}^1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \dots \quad T_{(k \times k)}^1 = \begin{bmatrix} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \dots & 1 \end{bmatrix}$$

$$T_{(2 \times 2)}^i = \begin{bmatrix} i & i \\ i & i \end{bmatrix} \quad T_{(3 \times 3)}^i = \begin{bmatrix} i & i & i \\ i & i & i \\ i & i & i \end{bmatrix} \quad \dots \quad T_{(k \times k)}^i = \begin{bmatrix} i & \dots & i \\ \vdots & \ddots & \vdots \\ i & \dots & i \end{bmatrix}$$

Рис. 2. Семейство шаблонов равномерного трафика: $U-1$ и $U-i$ (i.i.d. Bernoulli uniform)

Что будет соответствовать поведению пользователей „каждый имеет информацию (пакеты) для каждого”. Это считается редкой ситуацией в крупномасштабных сетях, но традиционно любое исследование ПС коммутаторов начинается с рассмотрения этого случая [5].

2 Результаты компьютерного моделирования

В качестве входных данных для вычисления бесконфликтного расписания использовалась матрица заявок T для равномерно распределенного входящего трафика. Поскольку в настоящей работе описывается начальная стадия исследований, то для сравнения указанных алгоритмов, был выбран первый представитель семейства шаблонов для T см. [13]. В нашем случае это матрица, чьи элементы принимают значение 1 (Шаблон1).

На рис.3 и рис.4 показаны результаты вычислений LPF-алгоритма. Расчет был выполнен для размерности коммутационного поля $(n \times n)$ матричного переключателя узла от (3×3) до (100×100) . По горизонтальной оси показана размерность n входной матрицы T $(k \times k)$. По вертикальной оси

влево показана ПС (throughput) алгоритма, усредненная по 10 000 вычислений. По вертикальной оси (вправо) - время для вычисления расписания (в секундах, усредненное также по 10 000 вычислений). LPF-алгоритм выбирает случайным образом элемент для коммутации в случае, когда есть несколько элементов с одинаковым максимальным весом. Для получения необходимой точности значения ПС оказалось необходимым провести 10 000 вычислений.

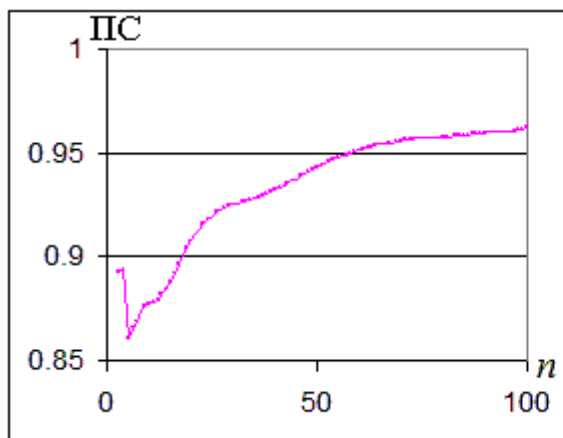


Рисунок 3. ПС для LPF-алгоритма

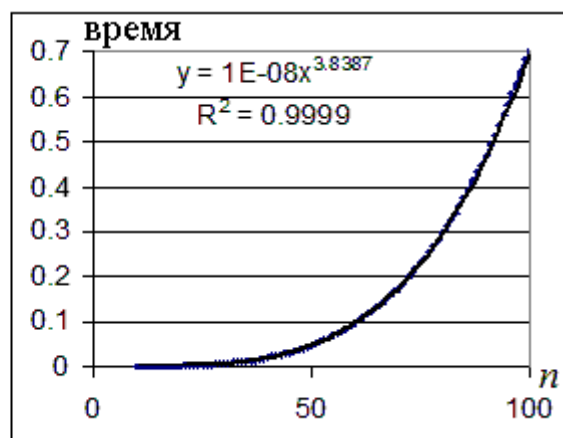


Рисунок 4. Время вычисления для LPF

На рис.5 и рис.6 показано сравнение результатов работы с Мима-алгоритма [12]. Качественные выводы достаточно определены – МиМа-алгоритм имеет меньшую ПС, но зато и меньшую вычислительную сложность. На правой границе участка ПС ($n=100$) у МиМа-алгоритма она хуже на ~2%. Можно высказать предположение, что при возрастании n разность в ПС должна будет уменьшаться, так как обе ПС стремятся к одному значению (100%).

Для проверки этого утверждения мы построили аппроксимации ПС обоих алгоритмов в виде степенной функции $a \cdot x^b$. Полученный результат показан на рис.7. Выбор диапазона до (70x70) связан с аппроксимацией ПС МиМа-алгоритма – до этого значения мы считаем ее удовлетворительной. На рис.8 представлены численные значения аппроксимаций, а также их разность (вертикальная ось вправо: 5% - 0,05 ПС). Видно, что разность уменьшается с увеличением n . При увеличении размерности коммутационного поля, после прохождения некоторого значения n у МиМа-алгоритма ПС будет практически совпадать с оптимумом, предоставляемым LPF-алгоритмом (например разность будет менее 1%), при (значительно) меньших затратах на время вычисления.

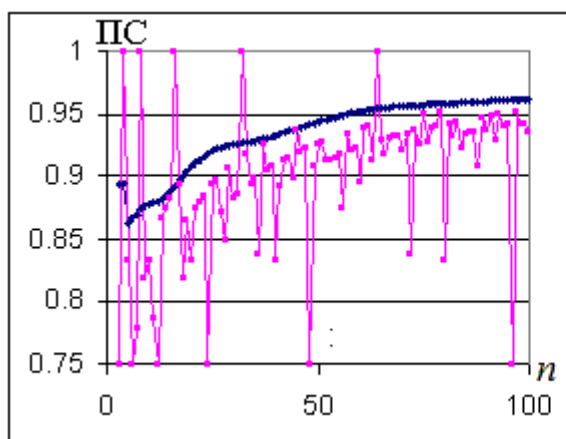


Рисунок 5. Сравнение ПС МиМа-алгоритма

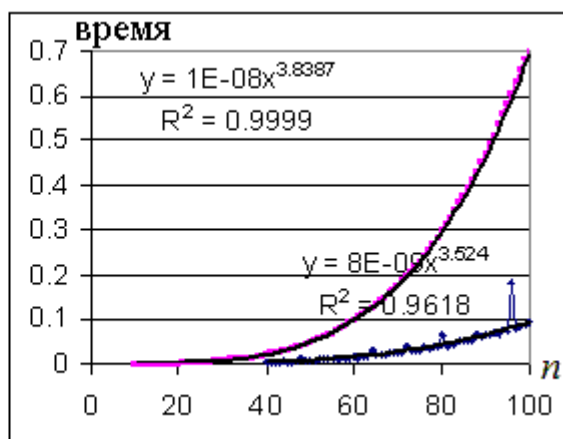


Рисунок 6. Время вычисления для двух алг-ов.

Более интересен вопрос о „прыгающем” характере ПС и времени у МиМа-алгоритма. Согласно представленным расчетам, МиМа-алгоритм работает быстрее, но не всегда делает „правильный” выбор, что обусловлено детерминированной дисциплиной выбора элемента для коммутации (в случае нескольких элементов с равными максимальными значениями в весовых векторах). А LPF-алгоритм выбирает случайным образом один из элементов с равными максимальными значениями в весовых векторах, и достигает большего числа „правильных” выборов. Следовательно LPF-

алгоритм в пространстве „выборов” равновероятно т.е. делает „правильные” и „неправильные” выборы, а детерминированная „политика” выбора у МиМа-алгоритма чаще ведет к „неправильным” выборам.

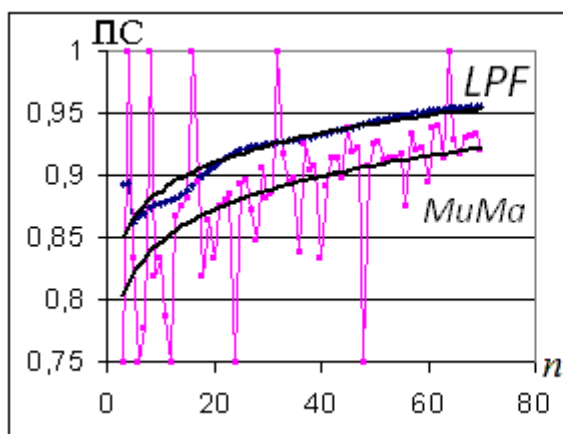


Рисунок 7. Аппроксимация ПС типа $a \cdot x^b$

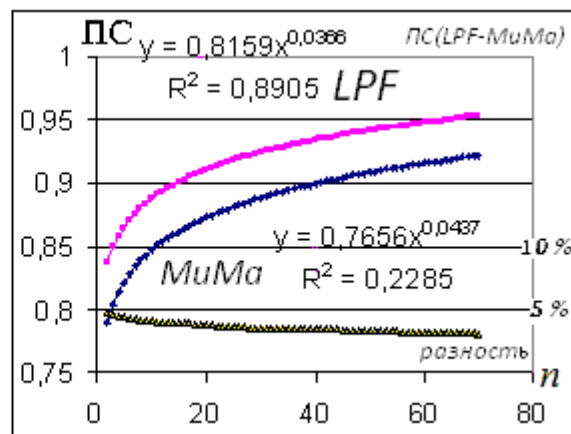


Рисунок 8. Разность двух аппроксимаций

Самой простой вариант модификации – изменить дисциплину выбора на „противоположную” при вычислении очередной матрицы Q_i . Т. е. для Q_1 имеем дисциплину „северо-западного угла”, как в оригинале. Для Q_2 имеем дисциплину „юго-восточного угла”, и чередуем дисциплину далее. Мы рассчитываем, что это улучшит показатели МиМа-алгоритм по сравнению с оригинальным вариантом. На данном этапе, важно не усложнять дисциплину выбора, чтобы не увеличивать время вычисления решения (бесконфликтного расписания).

Выводы

В этой работе представлены результаты компьютерного моделирования и расчетов на суперкомпьютере "АВИТОХОЛ" ИИКТ-БАН. Расчеты проводились в целях получения значений пропускной способности пакетного коммутатора под управлением LPF-алгоритма для вычисления бесконфликтного расписания. Полученные значения ПС сравнивались с соответствующими значениями для МиМа-алгоритма. Были определена временные затраты LPF-алгоритма – порядка $O(n(\exp(3.8)))$. Пропускная способность у двух алгоритмов стремится к максимальной, а по сравнению затрат на время вычисления - МиМа-алгоритм показывает лучшие результаты. Это валидно для равномерно распределенного входящего трафика. При будущих исследованиях планируется провести моделирование при больших n (примерно 100×100 и более) с целью определения значения, при котором разность в ПС будет менее 1%.

Литература

1. Баканова Н.Б. Использование программно-технических комплексов для повышения эффективности контроля в системах документооборота // «Электросвязь». 2007. № 6. С. 51–53.
2. Ташев Т.Д., Баканов А.С. Разработка модели пропускной способности сети с использованием МИМА- алгоритма // «Электросвязь» 2017 №8 С. 32-35.
3. Цвиркун А.Д. Основы синтеза структуры сложных систем. – М.: Наука, 1997. – 256с.
4. Цвиркун А.Д., Акинфиев В.К., Соловьев М.М. Моделирование развития крупномасштабных систем. – М.: Экономика. 1983. – С.35-38.
5. Chao, H., B. Lui, High performance switches and routers. John Wiley & Sons, 2007.
6. Hu, B., K. Yeung, C. He. On Iterative Scheduling for Input-queued Switches with a Speedup of $2-1/N$. Proceedings of 15th IEEE Int. Conf. HPSR 2014, July 1-4, 2014, Vancouver, Canada, pp.26-31.
7. Gupta P., N. McKeown. Designing and Implementing a Fast Crossbar Scheduler. IEEE Micro, Jan-Feb 1999, pp. 20-28.
8. Rojas-Cessa R. Interconnections for Computer Communications and Packet Networks. CRC Press, 2017.

9. Raghupathikumar D., Bommanna K. A Genetic Algorithm based Scheduling of an Input Queued Switch. *International Journal of Computer Applications* (0975 – 8887). – February 2012. – Vol. 39, № 6. – P. 37–42
10. Kolchakov K. Research on the algorithm with diagonal activation for non conflict schedule in case of a large size switching matrix. *Proc. of the Int. Conference “DCCN 2011”, 26–28 October 2011.* – Moscow, Russia. R&D Company “INT”.– 2011. – P.135–140.
11. Polivka, M., V. Skorpil. Modeling of network switch controlled by neural network. *Proc. of the 35th International Conference on Telecommunications and Signal Processing (TSP), 3-4 July 2012, Prague, Czech Republic.* p.165-168
12. *Atanassov, K. Generalized Nets and System Theory.* Prof. M.Drinov Acad. Publ. House, Sofia, Bulgaria, 1997.
13. *Tashev T., M. Marinov, V. Monov, R. Tasheva.* Modeling of the MiMa-algorithm for crossbar switch by means of Generalized Nets. *Proc. of Intelligent Systems (IS), 2016 IEEE 8th International Conference on, Sofia, p. 593-598, 2016.*
14. *Mekkittikul A., N. McKeown,* A practical algorithm to achieve 100% throughput in input-queued switches. *Proceedings of IEEE INFOCOM’98, March 1998.* pp. 792-799.
15. *Vabishchevich P.* VFort. <http://www.nomoz.org/site/629615/vfort.html> (last checked June 1, 2019).