

MIMA ALGORITHM THROUGHPUT MODELLING FOR CROSSBAR SWITCH WITH HOTSPOT LOAD TRAFFIC

Tasho Tashev

Institute of Information and Communication Technologies, Bulgarian Academy of
Sciences, Sofia, Bulgaria

tashotr@bas.bg

Abstract: *The task which includes computing of non-conflict schedule for crossbar switch of packet commutation is NP-hard. After checking their efficiency with throughput modelling by uniform load traffic, the check for non-uniform traffic is required. In this paper a family of pattern-matrices is used for the simulation of non-uniform traffic based on the hotspot model. The applicability check is executed through computer simulated commutation using our MiMa-algorithm. The simulation results permit to evaluate the MiMa's throughput as tending to 100%.*

Key words: *Network nodes, Packet messages, Crossbar switch, Modelling, Conflict elimination.*

МОДЕЛИРОВАНИЕ ПРОПУСКНОЙ СПОСОБНОСТИ MIMA-АЛГОРИТМА ДЛЯ ПАКЕТНОГО КОММУТАТОРА ПРИ ВХОДЯЩЕМ ТРАФИКЕ ТИПА „ГОРЯЧЕЙ ТОЧКИ”

T. Tassev

Институт информационных и коммуникационных технологий БАН, София,
Болгария, tashotr@bas.bg

Задача вычисления бесконфликтного расписания коммутации пакетов в коммуникационных узлах с матричным переключателем (crossbar switch) имеет неполиномиальной сложности. Проверка эффективности алгоритма вычисления проводится моделированием пропускной способности коммутатора при равномерно и неравномерно распределенном входящем трафике. В этой работе использовано семейство шаблонов-матриц для симуляции неравномерного трафика на базе модели “горячей точки” (Чао-модель). Проведено имитационное моделирование коммутации этого типа входящего трафика для определения пропускной способности предложенного автором MiMa-алгоритма. Результаты моделирования указывают на достижение в пределе 100% пропускной способности коммутатора.

1. Введение

Пакетный коммуникационный узел с матричным переключателем (crossbar switch node - коммутатор) - это устройство, чья основная задача есть максимальное ускорение передачи данных за счет параллельно существующих потоков между узлами коммуникационной сети [1]. В идеальном случае коммутатор передает пакеты между узлами, подключенными к его портам, с той скоростью, с которой узлы генерируют эти пакеты, не внося дополнительных задержек и не теряя ни одного пакета [2]. Это обеспечивается бесконфликтным расписанием для коммутации, вычисляемое управляющим блоком коммутатора (Scheduler – Рис.1).

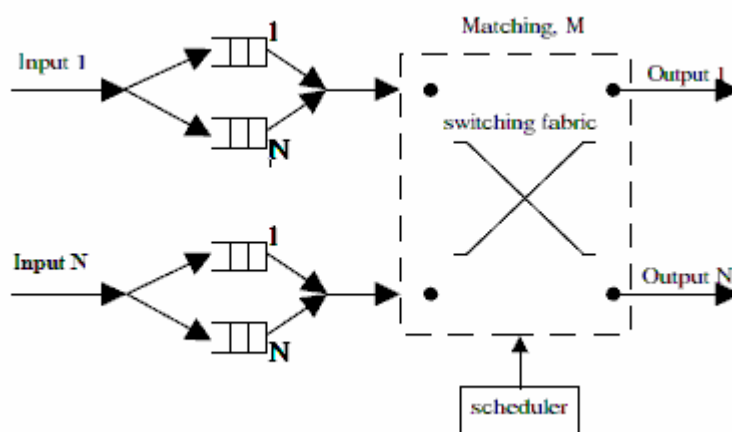


Рис 1. Логическая структура коммутатора с входящими буферами [1]

Математическим аппаратом доказано, что эта задача является NP-полной [3]. Существующие решения удовлетворяют часть целей [4], используя формализмы такие как теория массового обслуживания, генетические алгоритмы, матрицы-маски и др. [5, 6].

Научно-технический прогресс ведет к постоянному повышению скоростей и объемов коммуникационного трафика [7], что требует новых алгоритмов. Первый шаг при проверке эффективности нового алгоритма для расчета бесконфликтного расписания – это моделирование пропускной способности коммутатора при равномерно распределенном (uniform i.i.d. Bernoulli) входящем трафике. Следующий шаг требует проверки при неравномерно распределенном трафике.

В настоящей работы использовано семейство шаблонов-матриц для симуляции неравномерно распределенного трафика на базе модели “горячей точки” (hotspot - Чао-модель [8]). Возможности его проверены посредством компьютерной симуляции коммутации пакетов [9] с использованием

известного РИМ-алгоритма [10]. В данном докладе представлены результаты вычислительных экспериментов для нашего алгоритма MiMa, модель которого построена с использованием аппарата Обобщенных сетей [11].

Пропускная способность коммутатора при работе MiMa-алгоритма стремится к 100% при равномерно распределенном входящем трафике [12]. В этом докладе мы представим исследования работы MiMa- алгоритма при входящем трафике типа “горячей точки”. Они будут опорной базой для сравнения с результатами работы других алгоритмов.

2. Бесконфликтное расписание для входящего трафика

Проблема конфликтов общепринятая : коммутатор имеет n входных каналов и n выходных каналов связи, то есть n источников пакетов через коммутатор могут посылать их к n получателям [3]. Заявки на передачу пакетов образуют матрицу трафика T с размерностью $n \times n$, причем $T_{ij} = r$ ($r = 1, 2, \dots$) если число заявок от i -того источника к j -тому приемнику равно r , и $T_{ij} = 0$ если нет заявки. Для бесконфликтной коммутации надо вычислить последовательность бесконфликтных матриц Q_m (размерности $n \times n$, m принадлежит интервалу $[1, 2, \dots]$), сумма которых давала бы T . Каждая Q_m должна в любой строке или столбце иметь не более одной единицы. “Длина” последовательности является целое положительное число $k_{\text{реш}}$ – это число матриц в решении (последовательности).

Отметим, что матрица T представляет однонаправленный поток заявок – от входов к выходам коммутатора, представляющих пакетов с одной и той же длины. При этом ее формирование описанным способом означает, что используется входное буферирование с Виртуальными выходными очередями (crossbar VOQ - Virtual Output Queue) и заявки накапливались за число временных тактов (так называемых cell-time [8]), определяемое размером входного буфера коммутатора. Виртуальные выходные очереди продолжают привлекать внимание исследователей [13].

Эффективность работы коммутаторов на первом месте оценивают по использовании пропускной способности выходных каналов (throughput) – аналог Коэффициента полезного действия (Кпд). В нашем случае, если известно оптимальное решение $k_{\text{опт}}$ для бесконфликтной коммутации заданной матрицы T ($k_{\text{опт}}$ – положительное целое число), а алгоритм дает решение $k_{\text{реш}}$, то Кпд алгоритма равен отношению ($k_{\text{опт}} / k_{\text{реш}}$).

Кпд MiMa-алгоритма для равномерно распределенном входящем трафике мы получили в [12]. Продолжение исследований требует моделирование при неравномерном входящем трафике.

Модель Чао (hotspot traffic) для равномерно распределенного по входам (i.i.d.) и неравномерно распределенного по выходам трафика дается как [8] : $\lambda_{ij} = \rho * 0,5$ для $i = j$ и $\lambda_{ij} = \rho * 0,5/(n-1)$ в противном случае ($i, j \in 1, \dots, n$), где ρ - интенсивность нагрузки на одной входящей линии (i.i.d. Bernoulli). Для

модели Чао строим базовую Трафичную матрицу для трафика типа “горячей точки”, которую будем обозначать $Chao_1$ [9].

Матрица для шаблона $Chao_1$ можно развернуть в виде $Chao_i$, где $i=1,2,\dots$, как показано на Рис.2. Оптимальное решение для $Chao_i$ равно $k_{opt}=i*2*(n-1)$ при размерности T равной $n \times n$. Это является эквивалентом решения модели Чао для дискретных значений трафика в виде матрицы T .

$$T = \begin{matrix} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} & \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} & \dots & \begin{bmatrix} k-1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \dots & k-1 \end{bmatrix} & \dots \\ 2 \times 2 & 3 \times 3 & & k \times k & \end{matrix} \quad T = \begin{matrix} \begin{bmatrix} i & i \\ i & i \end{bmatrix} & \begin{bmatrix} i*2 & i & i \\ i & i*2 & i \\ i & i & i*2 \end{bmatrix} & \dots & \begin{bmatrix} i*(k-1) & \dots & i \\ \vdots & \ddots & \vdots \\ i & \dots & i*(k-1) \end{bmatrix} & \dots \\ 2 \times 2 & 3 \times 3 & & k \times k & \end{matrix}$$

Рис.2 Матрицы трафика $Chao_1$ и $Chao_i$ - размерности коммутатора до $k \times k$

Отметим, что число $i*2*(n-1)$ равно величине буфера одной входящей линии коммутатора – соответственно за столько же временных тактов (cell-time) накопилось „содержание” матрицы T . Поэтому развитие шаблона $Chao_1$ в виде семейства шаблонов $Chao_i$ позволяет, кроме получения значений для Кпд, исследовать влияние размера входного буфера на характеристики алгоритма для вычисления расписания.

3. MiMa-алгоритм для бесконфликтного расписания

Формально MiMa-алгоритм специфицирован с помощью аппарата ОС. Основными элементами ОС являются это переходы (моделирующие действия) и позиции (моделирующие состояния). В позициях находятся ядра, которые проходят через переходы соответственно с логических условий. Ядра имеют необходимые характеристики, которые могут изменяться при прохождении через переходов. Любая Обобщенно-сетевая модель имеет графичное представление, для MiMa-алгоритма оно дано в [12].

MiMa -алгоритм вычисляет ряд бесконфликтных матриц $Q_1, Q_2, \dots, Q_{k_{реш}}$ - пока все заявки матрицы T не получат назначение на коммутацию Тогда формирован искомый ряд матриц Q_m , и $k_{реш}$ определено. Представим неформально действие алгоритма.

Как начальные данные вводится матрица T . При представлении входящего трафика матрицей T , каждая Виртуальная выходная очередь являются ее ячейкой T_{ij} , причем число r 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 (числу входов равно число выходов коммутатора – n).

Вычисляем Вектор-колонну (колонна веса конфликтов), каждый элемент которого определяется числом конфликтов в каждой строке T . Вычисляем Вектор-строку (строка веса конфликтов), каждый элемент которого

определяется числом конфликтов в каждой колонне T . Вес конфликтов – это сумма заявок на коммутацию в колонне (строке) матрицы T – то есть сумма соответствующих элементов в колонне (строке) T .

В Вектор-строке выбираем максимальный элемент – он определяет колонну с максимумом конфликтов. В Вектор-колонне выбираем максимальный элемент – определяет строку с максимумом конфликтов. Если есть заявки в ячейке T пересечения выбранных колонны и строки, присваиваем эту заявку как элемент бесконфликтной матрицы Q_1 (и естественно уменьшаем на единицу значение в выбранной ячейке T). Если ячейка пустая, то выбираем элемент в Векторе-колонне, чье значение есть самое близкое к максимальному. Причем выбранный элемент в Векторе-строке остается прежний. Снова проверяем есть ли заявки в „точку” пересечения и т.д.

В результате в выбранной колонне T имеем заявку, выранную для коммутации (если такой существует). Строку и колонну, содержащие выранную заявку, исключаются от дальнейшего вычисления Q_1 . Следующие элементы Q_1 вычисляются повторением вышеописанной процедуры. В результате первая матрица бесконфликтного расписания Q_1 будет состояться из элементов (заявок) с (суб)-максимальном весе конфликтов в T . А стоимости выбранных ячеек T будут уменьшены на единицу.

Последующие бесконфликтны матрицы $Q_2, \dots, Q_{k_{\text{реш}}}$ вычисляются аналогичным образом. (Вычисляем Вектор-колонну (колонна веса конфликтов), ... и далее). Отметим, что для вычисления последней матрицы – $Q_{k_{\text{реш}}}$ – останутся только бесконфликтные заявки в (остаток) матрицы T .

4. Результаты вычислительных экспериментов

Переход от ОС-модели MiMa-алгоритма к программе для имитационного моделирования исполнен согласно [14]. Для программирования был использован программный пакет Vfort, предоставленный московским Институтом математического моделирования РАН для свободного пользования [15]. При вычислительных экспериментах использовалась структура BG01-IPP Института информационных и коммуникационных технологий Болгарской академии наук (www.grid.bas.bg). Входными данными для алгоритма является матрица входящих заявок на коммутирование – матрица T . При размерности $n \times n$ ($n = 3, 4, \dots \in N$) каждый ее элемент принимает значение сообразно выбранного $Chao_i$ -шаблона.

На Рис. 3 и Рис.4 показаны результаты работы алгоритма для шаблонов $Chao_1$, $Chao_5$ и $Chao_{10}$, $Chao_{100}$ соответственно. По горизонтальной оси координат показана размерность n входной матрицы T (от 3×3 до 130×130).

По левой вертикальной оси показано Кпд (throughput) работы алгоритма (приведенное к пропускной способности коммутатора- принятой за 1). По правой вертикальной оси – время для вычисления расписания (в секундах). Кривые соответствуют – графика влево на Рис.3 – результатам, полученными с шаблоном трафика $Chao_1$; влево на Рис.4 – трафик вида $Chao_{10}$.

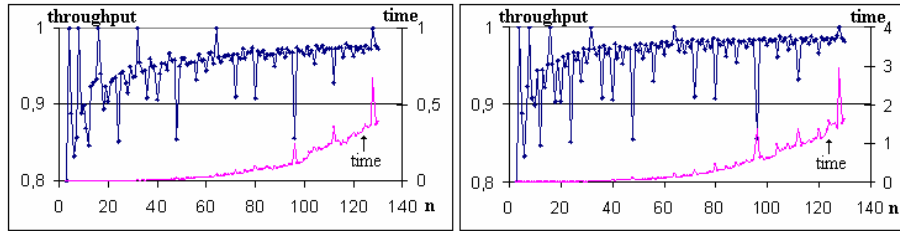


Рис.3. Пропускная способность и время для шаблонов вида Chao₁ и Chao₅

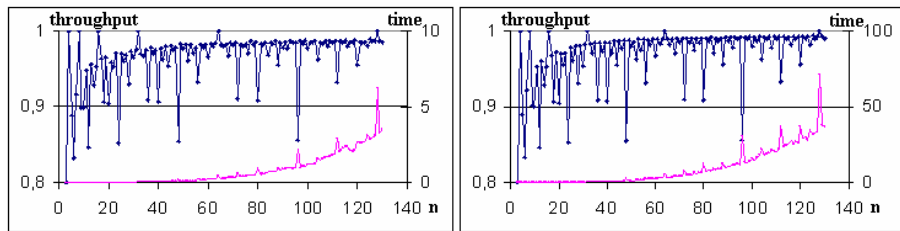


Рис.4. Пропускная способность и время для шаблонов вида Chao₁₀ и Chao₁₀₀

На Рис.5 показаны вместе кривые для времени вычисления решения для шаблонов Chao₁ (нижняя), Chao₁₀ (средняя) и Chao₁₀₀ (верхняя) в логарифмическом масштабе. Влево : y - это аппроксимация как полином для размерность n входной матрицы T от 30×30 до 130×130 ; R^2 - это корреляция данных.

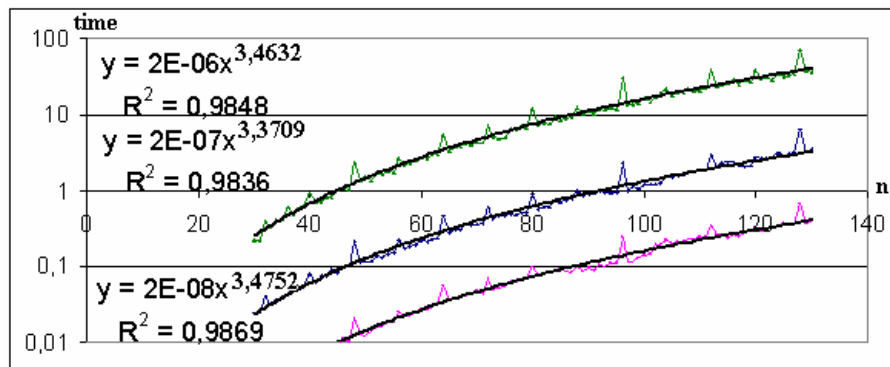


Рис.5. Линейная зависимость времени решения от индекса шаблона Chao_i

Полученные результаты (Рис.3 и Рис.4) позволяют заключить, что для исследованного трафика типа „горячей точки” пропускная способность MiMa - алгоритма стремится в пределе к максимальной (100 %).

Время вычисления, как следует от данных на Рис.5, имеет линейную зависимость от индекса шаблона - соответственно от величины входящего буфера ($Chao_1 \rightarrow 2 \cdot 10^{-8}$, $Chao_{10} \rightarrow 2 \cdot 10^{-7}$, $Chao_{100} \rightarrow 2 \cdot 10^{-6}$). А зависимость времени от размерности коммутатора пропорциональна степени $\sim 3,5$ ($Chao_1 \rightarrow 3,4752$ от n и т.д.). Это время включает в себя получение $i \cdot 2 \cdot (n-1)$ решений на коммутацию, поэтому временная сложность для одного решения (одну матрицу Q_m) надо понизить на одну степень. То есть для $Chao_i$ время выполнения MiMa-алгоритма пропорционально степени $\sim 2,5$ от размерности коммутатора, что совпадает с хорошей точности с известной теоретической оценки $o(N^{5/2})$ [8] сложности весовых алгоритмов, к которым принадлежит MiMa.

Отметим, что для размерности коммутатора равными степенями числа 2 (4,8,16,32,...) пропускная способность точно равна 100%. А для размерности коммутатора „посередине” степенями числа 2 (6,12,24,48,...) пропускная способность равна 85,5%. В обоих случаях имеем вдвое увеличенное время для вычисления. Это мы толкуем как наличие возможности для оптимизации MiMa-алгоритма по времени выполнения.

5. Заключение

Компьютерное моделирование показывает, что пропускная способность MiMa-алгоритма стремится к 100% для входящего трафика типа „горячей точки”, продемонстрировав линейную зависимость времени вычисления бесконфликтного расписания от использованного шаблона для входящего трафика. При этом для размерности коммутатора равными степенями числа 2 пропускная способность точно равна 100%. Полученные результаты указывают на возможность оптимизирования MiMa-алгоритма по времени.

Литература REFERENCES

1. Deb S., Shah D., Shakkottai S. Fast Matching Algorithms for Repetitive Optimization: An Application to Switch Scheduling // Information Sciences and Systems, 2006 40th Annual Conference on, 22-24 March 2006, P. 1266-1271.
2. Ермаков А.С. Модель балансировки коммутационных процессов // Proc. of Int. Workshop “DCCN-2007”, Sofia, Bulgaria, 2007. Moscow, Technosfera, 2007, P. 171-176.
3. Chen T., Mavor J., Denyer Ph., Renshaw D. Traffic routing algorithm for serial superchip system customisation // IEE Proc.- E.part, Jan 1990, V. 137, N.1, P. 65-73.

4. Dong Z., Rojas-Cessa R. Throughput analysis of shared-memory crosspoint buffered packet switches // Communications, IET (2012), VI.6, N.9, P. 1045-1053.
5. Raghupathikumar D., Bommanna K. A Genetic Algorithm based Scheduling of an Input Queued Switch // International Journal of Computer Applications (0975 – 8887), February 2012, V.39, N.6, P. 37-42.
6. Kolchakov K., Research on the algorithm with diagonal activation for non conflict schedule in case of a large size switching matrix // Proceedings of the Int. Conference “DCCN 2011”, October 26-28, 2011. Moscow, Russia. R&D Company “INT”, Moscow, Russia, P. 135-140.
7. Атанасова Т., Савченко Т., Головина Г., Баканов А. Интеллектуальная информационная среда обитания и субъективное восприятие качества жизни // Методы исследования психологических структур и их динамики: Вып. 5. / Под редакцией Т. Н. Савченко, Г.М. Головиной. - М.: Издательство “Институт психологии РАН”, 2010, стр. 164-170.
8. Chao-Lin Y., Chang C., Lee D. CR Switch: A Load-Balanced Switch With Contention and Reservation // IEEE/ACM Transactions on Networking, October 2007, V. 17, N.5, P. 1659–1671.
9. Ташев Т. Моделирование пропускной способности коммутационного узла при несбалансированном входящем трафике // Proceedings of the Int. Conference “DCCN 2011”, October 26-28, 2011, Moscow, Russia. R&D Company “INT”, Moscow, Russia, P. 96-102.
10. T.Anderson, S.Owicki, J.Saxe, and C.Thacker. High speed switch scheduling for local area networks // ACM Trans. Comput. Syst., Nov. 1993, V. 11, N.4, P. 319-352.
11. Atanassov K. Generalized Nets and System Theory // Akad. Press “Prof.M. Drinov”, Sofia, Bulgaria, 1997.
12. Tashev T., Atanasova T. Computer Simulation of MIMA Algorithm for Input Buffered Crossbar Switch // International Journal "Information Technologies & Knowledge", 2011, V.5, N 2, P.183-189.
13. Kang K., Park K., Sha L., Wang Q. Design of a crossbar VOQ real-time switch with clock-driven scheduling for a guaranteed delay bound // Real-Time Systems, January 2013, V. 49, Issue 1, P 117-135.
14. Ташев Т., Железов Р. Моделирование виртуальных выходных очередей для коммутационного узла Обобщенными сетями // Proc. of Int. Workshop "DCCN '2010" October 26-28 2010, Moscow, Russia. R&D Company ”INT”, Moscow, 2010. P. 198-203.
15. <http://www.imamod.ru/~vab/vfort/download.html>