

ОРГАНИЗАЦИЯ ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

УДК 681.324

В. Д. Бобко, Ю. Н. Золотухин, В. Я. Пивкин

(Новосибирск)

ПОСТРОЕНИЕ ТАБЛИЦ УПРАВЛЕНИЯ МНОГОСТУПЕНЧАТЫМИ БЛОКИРУЮЩИМИ КОММУТАТОРАМИ СЕТЕВОГО ТИПА ДЛЯ ДВУХ КРИТЕРИЕВ ОПТИМИЗАЦИИ

Рассмотрена задача централизованного управления блокирующим коммутатором для двух критериев оптимальности: максимума удовлетворенных на каждом такте заявок и организации цикла состояний коммутатора минимальной длины, обеспечивающего гарантированное удовлетворение произвольной заявки. Предложены методы поиска приближенных решений для первого критерия и построения оптимальных циклов — для второго.

Введение. В настоящее время проводятся интенсивные исследования возможностей использования многоступенчатых коммутаторов в телефонных сетях, многопроцессорных вычислительных системах, интегральных сетях связи и т. д. [1, 2].

Многоступенчатый коммутатор с N входами и N выходами представляет собой структуру, каждая ступень которой состоит из группы элементарных 2×2 -коммутаторов (рис. 1). Управляя элементарными коммутаторами каждой ступени, можно соединять между собой нужные входы и выходы.

Коммутатор имеет память для хранения управляющей таблицы. Режим работы коммутатора синхронный, т. е. существует временной такт обновления управляющей таблицы (смены состояний элементарных коммутаторов).

Многоступенчатые коммутаторы по функциональным характеристикам делятся на блокирующие и неблокирующие с реконfigurацией. Блокировка возникает из-за того, что установка соединения i -го входа с j -м выходом коммутаторов этого вида обеспечивается фиксацией состояний цепочки элементарных коммутаторов, при этом становятся невозможными соединения, требующие для их установки использования каких-либо элементарных коммутаторов из этой цепочки в противоположных состояниях.

К неблокирующим коммутаторам относятся, например, сеть Бенеша, сеть «Мемфис», к блокирую-

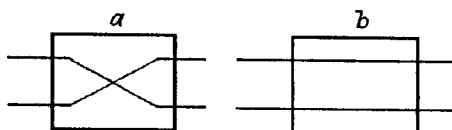


Рис. 1. Элементарный коммутатор:
 a — перекрестное соединение, b — прямое соединение

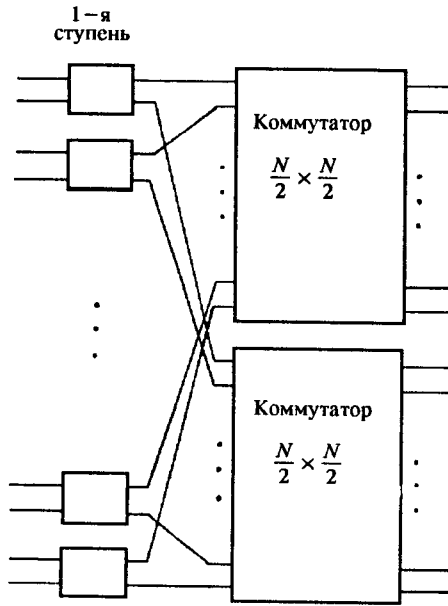


Рис. 2. Рекурсивная процедура построения коммутатора типа «сеть с базовой линией» размерностью $N \times N$:

первая ступень содержит $N/2$ элементарных коммутаторов

щим — «сеть с базовой линией», сеть «Омега», сеть Баньяна, n -кубическая сеть и другие [1].

Рассмотрим задачи управления блокирующим коммутатором для следующих двух критериев оптимальности:

- обеспечение максимума удовлетворенных заявок на каждом такте работы коммутатора (потактовая оптимизация);

- удовлетворение любой заявки в течение минимального по числу тактов цикла работы коммутатора (гарантированное время обслуживания).

В [3] показана топологическая эквивалентность перечисленных выше блокирующих коммутаторов. В связи с

этим, без потери общности, рассмотрим только коммутатор «сеть с базовой линией».

Коммутатор имеет $N = 2^r$ входов и $N = 2^r$ выходов, содержит $1/2 N \log_2 N$ элементарных коммутаторов, образующих $r = \log_2 N$ ступеней по $N/2$ коммутаторов в каждой. На рис. 2 показана рекурсивная процедура построения коммутатора (первая итерация). Полное построение требует $r - 1$ итераций.

На рис. 3 представлена топологическая структура коммутатора типа «сеть с базовой линией» размерностью 8×8 .

На коммутатор поступает поток заявок на соединение, который на каждом такте работы коммутатора задается матрицей требований $T = \|t_{ij}\|$ размерностью $N \times N$, где $t_{ij} = 1$ при наличии требования соединить i -й вход с j -м выходом и $t_{ij} = 0$ в противном случае.

Состояния элементарного коммутатора будем описывать булевой переменной, принимающей значение «1» при прямом и «0» при перекрестном соединениях коммутатора.

Конкретное состояние коммутаторов p -й ступени будем задавать вектор-столбцом $S_p = (\sigma_{p1}, \sigma_{p2}, \dots, \sigma_{pN/2})$, где $\sigma_{pj} \in \{0, 1\}$, $j = 1, 2, \dots, N/2$.

Состояние коммутатора в целом (управляющая таблица) задается вектором состояний ступеней $W = \{S_1, S_2, \dots, S_r\}$. Общее число состояний коммутатора равно $2^{1/2 N \log_2 N}$.

Потактовая оптимизация. По критерию максимума удовлетворяемых на каждом такте заявок задача оптимального управления коммутатором формулируется следующим образом:

найти состояние коммутатора $W_{\text{опт}}$, обеспечивающее удовлетво-

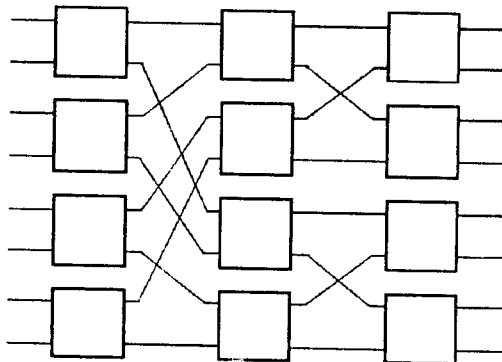


Рис. 3. Топологическая структура коммутатора «сеть с базовой линией» размерностью 8×8

рение наибольшего числа заявок при заданной матрице требований T .

Элементарному коммутатору p -й ступени, занимающему q -ю позицию, поставим в соответствие одноместную булеву функцию s_{pq}^σ , $\sigma \in \{0, 1\}$, $s_{pq}^0 = \bar{s}_{pq}$, $s_{pq}^1 = s_{pq}$, $p = 1, 2, \dots, r$; $q = 1, 2, \dots, N/2$.

Соединение i -го входа с j -м выходом коммутатора обеспечивает цепочка элементарных коммутаторов в определенных состояниях, содержащая по одному коммутатору каждой ступени. Эту цепочку будем называть маршрутом \tilde{t}_{ij} и представлять в виде конъюнкции

$$\tilde{t}_{ij} = s_{1k_1}^{\sigma_{1k_1}} s_{2k_2}^{\sigma_{2k_2}} \dots s_{rk_r}^{\sigma_{rk_r}},$$

где $(\sigma_{1k_1}, \sigma_{2k_2}, \dots, \sigma_{rk_r})$ — набор состояний элементарных коммутаторов, обеспечивающих соединение.

Поскольку между заявками из матрицы требований $T = \|t_{ij}\|$ и маршрутами \tilde{t}_{ij} существует очевидное взаимно однозначное соответствие (за исключением случая $N = 2$), будем считать, что поток заявок задан множеством

$$\tilde{T} = \{\tilde{t}_{ij}\}; i, j \in [1, N],$$

маршрутов, представленных в виде конъюнкций длины $r = \log_2 N$ каждая.

Справедливо следующее утверждение:

оптимальным по критерию максимума удовлетворенных на данном такте заявок является состояние коммутатора, обеспечивающее обращение в «1» наибольшего числа конъюнкций из $\tilde{T} = \{\tilde{t}_{ij}\}$.

Такой постановки будем придерживаться при дальнейшем рассмотрении задачи.

Два маршрута \tilde{t}_{nm} и \tilde{t}_{ki} будем называть совместимыми, если представляющие их конъюнкции не содержат общих переменных, соответствующих элементарным коммутаторам, находящимся в противоположных состояниях, и несовместимыми — в противном случае.

Заметим, что совокупность удовлетворенных заявок для любого из состояний коммутатора обладает свойством попарной совместимости их маршрутов.

Пусть $G(\tilde{T}, E)$ — граф, вершинами которого являются маршруты из \tilde{T} , и две вершины соединены ребром из E в том и только том случае, если соответствующие этим вершинам маршруты совместимы.

Нетрудно видеть, что задачу поиска оптимального состояния коммутатора можно представить как задачу поиска наибольшей клики графа $G(\tilde{T}, E)$, т. е. полного подграфа наибольшей размерности.

Задача поиска наибольшей клики NP -полна [4]. Несмотря на то что отсюда не следует NP -полнота рассматриваемой задачи, тем не менее есть основания считать, что она достаточно трудоемка. Поэтому в условиях дефицита времени между тактами смены управляющих таблиц целесообразно использовать методы поиска приближенных решений, достаточно близких к оптимальным.

Предлагаемый метод относится к методам градиентного типа.

Исходные данные — совокупность маршрутов-конъюнкций $\tilde{T} = \{\tilde{t}_{ij}\}$. Используемые на k -м шаге обозначения: \tilde{T}_k — совокупность рассматриваемых на k -м шаге маршрутов; $n(\tilde{t}_{ij})$ — число маршрутов из \tilde{T}_k , несовместимых с \tilde{t}_{ij} ; R — множество маршрутов, уже включенных в искомое решение.

Поиск решения осуществляется по следующей схеме.

1. Начальные значения: $k = 1$, $\tilde{T}_1 = \tilde{T}$, $R = \emptyset$.

2. Среди маршрутов из \tilde{T}_k определяется маршрут \tilde{t}_{pq} , для которого $n(\tilde{t}_{pq})$ минимально. Выбранный маршрут включается в R , т. е. $R := R \cup \tilde{t}_{pq}$. Формируется \tilde{T}_{k+1} , для этого из \tilde{T}_k удаляется \tilde{t}_{pq} и все несовместимые с \tilde{t}_{pq} маршруты.

Если $T_{k+1} = \emptyset$, то переходим к п. 3, иначе $k = k + 1$ и операции п. 2 повторяются.

3. По множеству R определяется искомое состояние коммутатора.

Для коммутатора размерностью $N \times N$ для поиска решения потребуется не более N шагов цикла п. 2.

Следует заметить, что предложенная схема не является готовым алгоритмом, а лишь декларирует идею поиска, основанную на понятии совместности.

При конкретной реализации можно учесть, например, что структурные особенности коммутатора позволяют выделить для каждого элементарного коммутатора его «зону влияния», т. е. подматрицу матрицы требований, содержащую $2N$ элементов и такую, что каждая заявка из этой подматрицы требует обязательного использования данного элементарного коммутатора в определенном состоянии.

Подматрицы, соответствующие элементарным коммутаторам, образуют покрытие матрицы требований, обладающее ярко выраженными структурными свойствами, которые можно использовать, например, для сокращения объема вычислений при определении подмножества заявок или маршрутов, несовместимых с данной заявкой или маршрутом. При этом в качестве исходных данных вместо множества маршрутов может использоваться непосредственно матрица требований.

Проиллюстрируем представление заявок маршрутами-конъюнкциями и итоги работы метода на примере коммутатора размерностью 8×8 для матрицы требований, изображенной на рис. 4, а.

Указанной матрице требований соответствует совокупность маршрутов $\tilde{T} = \{\tilde{t}_{18} = \bar{s}_{11}\bar{s}_{23}\bar{s}_{34}, \tilde{t}_{22} = \bar{s}_{11}\bar{s}_{21}\bar{s}_{31}, \tilde{t}_{25} = \bar{s}_{11}\bar{s}_{23}\bar{s}_{33}, \tilde{t}_{32} = \bar{s}_{12}\bar{s}_{21}\bar{s}_{31}, \tilde{t}_{35} = \bar{s}_{12}\bar{s}_{23}\bar{s}_{33}, \tilde{t}_{41} = \bar{s}_{12}\bar{s}_{21}\bar{s}_{31}, \tilde{t}_{51} = \bar{s}_{13}\bar{s}_{22}\bar{s}_{31}, \tilde{t}_{56} = \bar{s}_{13}\bar{s}_{24}\bar{s}_{33}, \tilde{t}_{61} = \bar{s}_{13}\bar{s}_{22}\bar{s}_{31}, \tilde{t}_{63} = \bar{s}_{13}\bar{s}_{22}\bar{s}_{32}, \tilde{t}_{67} = \bar{s}_{13}\bar{s}_{24}\bar{s}_{34}, \tilde{t}_{68} = \bar{s}_{13}\bar{s}_{24}\bar{s}_{34}, \tilde{t}_{72} = \bar{s}_{14}\bar{s}_{22}\bar{s}_{31}, \tilde{t}_{73} = \bar{s}_{14}\bar{s}_{22}\bar{s}_{32}, \tilde{t}_{78} = \bar{s}_{14}\bar{s}_{24}\bar{s}_{34}, \tilde{t}_{83} = \bar{s}_{14}\bar{s}_{22}\bar{s}_{32}, \tilde{t}_{86} = \bar{s}_{14}\bar{s}_{24}\bar{s}_{33}, \tilde{t}_{87} = \bar{s}_{14}\bar{s}_{24}\bar{s}_{34}\}$.

Результатом работы п. 2 является множество

$$R = \{\tilde{t}_{18} = \bar{s}_{11}\bar{s}_{23}\bar{s}_{34}, \tilde{t}_{22} = \bar{s}_{11}\bar{s}_{21}\bar{s}_{31}, \tilde{t}_{35} = \bar{s}_{12}\bar{s}_{23}\bar{s}_{33}, \tilde{t}_{44} = \bar{s}_{12}\bar{s}_{21}\bar{s}_{32}, \tilde{t}_{56} = \bar{s}_{13}\bar{s}_{24}\bar{s}_{33}, \tilde{t}_{61} = \bar{s}_{13}\bar{s}_{22}\bar{s}_{31}, \tilde{t}_{73} = \bar{s}_{14}\bar{s}_{22}\bar{s}_{32}, \tilde{t}_{87} = \bar{s}_{14}\bar{s}_{24}\bar{s}_{34}\}.$$

Конъюнкции, вошедшие в R , принимают значения, равные «1», при следующих значениях состояний элементарных коммутаторов: $s_{11} = 0, s_{12} = 0, s_{13} = 0, s_{14} = 1, s_{21} = 1, s_{22} = 1, s_{23} = 0, s_{24} = 1, s_{31} = 0, s_{32} = 0, s_{33} = 1, s_{34} = 0$.

Матрица удовлетворенных заявок и состояние коммутатора представлены на рис. 4, б, с.

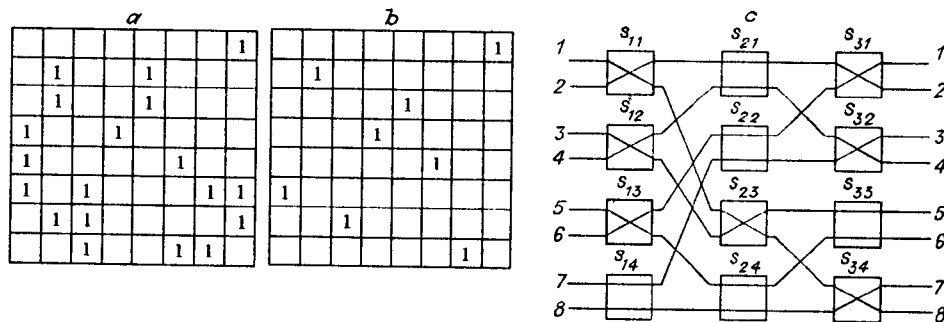


Рис. 4. Пример потактовой оптимизации:

а — исходная матрица требований, б — матрица удовлетворенных заявок, с — состояние коммутатора

Построение минимального цикла управляющих таблиц. Выше задача управления коммутатором рассматривалась с точки зрения критерия удовлетворения на каждом такте наибольшего числа заявок на соединение для заданной матрицы требований. Наряду со сложностью поиска оптимального решения, этот способ обслуживания обладает существенным недостатком: отсутствием гарантии для конкретной заявки быть удовлетворенной за фиксированное число тактов.

По критерию гарантированного времени обслуживания задача заключается в построении совокупности (цикла) состояний коммутатора $Z = \{W_0, W_1, \dots, W_m\}$, такой что:

- для произвольной заявки найдется состояние из Z , ее удовлетворяющее;
- число входящих в Z состояний минимально.

Поскольку максимально возможное число различных заявок равно N^2 , а за каждый такт обслуживается не более N из них, то, очевидно, N является минимальным числом тактов для гарантированного удовлетворения любой из возможных заявок.

Приведем достаточно простой алгоритм порождения совокупности из N состояний коммутатора, обеспечивающих оптимальное решение поставленной задачи.

Пусть $W_0 = \{S_1, S_2, \dots, S_r\}$ — некоторое состояние коммутатора. Здесь, как указывалось выше, S_i — вектор состояний элементарных коммутаторов i -й степени.

Обозначим через \bar{S}_i вектор состояний, противоположный S_i , т. е. если в S_i некоторый коммутатор находится в состоянии «1», то в \bar{S}_i его состояние равно «0», а если в S_i состояние коммутатора равно «0», то в \bar{S}_i оно равно «1».

Состояние W_0 порождает класс $Z(W_0) = \{W_0, W_1, \dots, W_{N-1}\}$ из N состояний, членами которого являются все состояния вида

$$\{S_1^{\sigma_1}, S_2^{\sigma_2}, \dots, S_r^{\sigma_r}\},$$

где $\sigma_i \in \{0, 1\}$, $S_i^0 = \bar{S}_i$, $S_i^1 = S_i$, $i = 1, 2, \dots, r$. Покажем, что класс $Z(W_0)$ при любом W_0 удовлетворяет требованиям выбранного критерия оптимальности. Для произвольного маршрута

$$\tilde{t}_{pq} = S_{1n_1}^{\sigma_{1n_1}} S_{2n_2}^{\sigma_{2n_2}} \dots S_{rn_r}^{\sigma_{rn_r}}$$

состояние W' , его удовлетворяющее, может быть сформировано из порождающего класса состояния W_0 с помощью следующей процедуры:

для каждого $j = 1, 2, \dots, r$, если σ_{jn_j} совпадает с состоянием этого коммутатора в S_j , то S_j входит в W' без изменения, в противном случае в W' входит \bar{S}_j .

Очевидно, что W' входит в класс $Z(W_0)$ и является единственным состоянием класса, удовлетворяющим заявку \tilde{t}_{pq} .

Заметим, что для произвольной пары состояний коммутатора порождаемые ими классы состояний либо совпадают, либо не пересекаются.

Любой из этих классов может быть основой для организации цикла смены состояний управляющей таблицы коммутатора, обеспечивающей гарантированное соединение для каждой из возможных заявок.

Примером такого цикла для коммутатора размерностью 8×8 является следующая последовательность состояний:

$$W_0 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}, \quad W_1 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \quad W_2 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \quad W_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix},$$

		Выходы							
		1	2	3	4	5	6	7	8
В х о д ы	1	0	1	2	3	7	6	4	5
	2	4	5	6	7	3	2	0	1
	3	6	7	4	5	1	0	2	3
	4	2	3	0	1	5	4	6	7
	5	5	4	7	6	0	1	3	2
	6	1	0	3	2	4	5	7	6
	7	3	2	1	0	6	7	5	4
	8	7	6	5	4	2	3	1	0

$$W_4 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}, \quad W_5 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix},$$

$$W_6 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad W_7 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}.$$

Работу коммутатора на протяжении цикла иллюстрирует таблица. Для каждой пары вход-выход на пересечении соответствующих строки и столбца указан номер такта работы коммутатора из сформированного цикла управления, обеспечивающего данное соединение.

Заключение. Предложенный в работе способ представления потока заявок совокупностями маршрутов-конъюнкций позволил достаточно естественно формализовать рассмотренные задачи оптимального управления многоступенчатыми коммутаторами сетевого типа и предложить методы их решения.

СПИСОК ЛИТЕРАТУРЫ

1. Амамия М., Танака Ю. Архитектура ЭВМ и искусственный интеллект. М.: Мир, 1993.
2. Степанян С. О. Коммуникационные сети в многопроцессорных ЭВМ (обзор) // Автоматика и вычисл. техника. 1987. № 3.
3. Chuan-Lin Wu, Tse-Yun Feng. On a class of multistage interconnection networks // IEEE Trans. on Computers. 1980. C-29. P. 694.
4. Гри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982.

Поступила в редакцию 23 марта 1995 г.