

ДИБРОВА М.А.,  
КОГАН А.В.,  
ВОРОБЬЕВА А.Л.

## СПОСОБ ФОРМИРОВАНИЯ МНОЖЕСТВА ПУТЕЙ В СЕТЕВЫХ ЦЕНТРАХ ДАННЫХ

В работе рассмотрена типовая структура сетевого центра данных. Обоснована целесообразность использования многопутевой маршрутизации в сетевых топологиях Fat Tree. Предложен способ повышения эффективности алгоритма поиска в глубину за счет учета свойства самоподобия топологии Fat Tree. Предложен модифицированный алгоритм поиска в глубину, приведено сравнение его с базовым алгоритмом.

The paper deals with the structure of a typical data center network. Substantiates the desirability of using multipath routing network topologies Fat Tree. A method for increasing the efficiency of the search algorithm in depth by taking into account the properties of self-similarity topology Fat Tree. A modified algorithm for depth-first search, compares it to the baseline algorithm.

### 1. Введение

В настоящее время в связи с возрастающими потребностями в вычислительной мощности и объемах информации актуальными становятся сетевые центры данных (DCN). Функциональным назначением DCN является высокоскоростная обработка и хранение большого объема данных. Сетевые центры данных (DCN) должны обеспечить высокую пропускную способность и надежность передачи информации. Это, в свою очередь, предъявляет высокие требования к конструированию трафика (TE) в таких системах.

При построении современных DCN в основном используется топология Fat Tree [1], которая является одной из наиболее распространенных топологий для построения распределенных систем, ориентированных на

решение высокопроизводительных задач. Топология Fat Tree представляет собой дерево, листьями которого являются вычислительные устройства, а узлами – коммутаторы. При этом у коммутаторов более высоких уровней пропускная способность каналов больше, т.е. связи с другими вершинами более «толстые». Поэтому эта топология и получила название Fat Tree (толстое дерево).

DCN при использовании топологии Fat Tree образуют четырехуровневую структуру [2], на верхнем уровне располагаются основные коммутаторы (Core switches), уровнем ниже располагаются агрегационные коммутаторы (Aggregation switches), затем располагаются граничные коммутаторы (Edge switches) к которым подключаются вычислительные устройства (Hosts) (рис. 1).

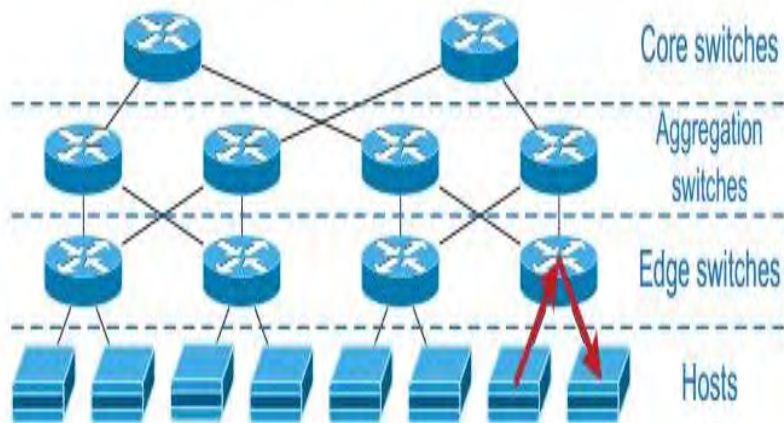


Рис. 1. Топология Fat Tree

В свою очередь, для DCN, состоящих из сотен и тысяч серверов большой размерности, топология Fat tree, на основе принципов

самоподобия, преобразуется в так называемую сеть Dragonfly [3]. Учет фрактального характера топологии сети Dragonfly позволяет

упростить процесс конструирования трафика, в частности процесс маршрутизации. Главными требованиями, предъявляемыми к маршрутизации в DCN, являются надежность, минимальная задержка и отказоустойчивость. Этим критериям в большей степени удовлетворяют способы многопутевой маршрутизации, которые нашли широкое применение в DCN [4, 5, 6]. Одним из основных недостатков комбинаторных алгоритмов многопутевой маршрутизации является значительная их временная сложность, которая, как правило, носит нелинейный характер. В работе [7] предложен волновой алгоритм формирования множества путей между произвольными вершинами с линейной временной сложностью. Дальнейшее уменьшение временной сложности связано с учетом топологии графа сети.

## 2. Модифицированный способ доступа на основе многопутевой маршрутизации

Для топологии Fat tree эффективным является метод поиска в глубину (DFS, сокращение от Depth-first search), в основе которого лежит рекурсивный способ обхода вершин графа [8]. В процессе обхода узлы отмечаются как посещенные и непосещенные, вначале все узлы являются непосещенными. Узел, который принимает управляющий пакет отмечается как посещенный и анализирует информацию о смежных с ним узлах. Затем данный узел согласно определенного правила передает пакет одному из неотмеченных смежных с ним узлов. Если у данного узла нет неотмеченных смежных с ним узлов, то он передает управляющий пакет обратно.

К недостаткам базового алгоритма следует отнести, что он не учитывает:

- 1) пропускную способность каналов связи;
- 2) особенности топологии сети.

Существуют три основных стратегии выбора каналов:

- Worst-Fit – выбирается канал с наибольшей доступной пропускной способностью;
- First-Fit – выбирается канал с любой из доступной пропускной способностью, которая отвечает требованиям;
- Best-Fit – выбирается канал с доступной пропускной способностью, которая наилучшим способом отвечает требованиям.

Следует отметить, что при формировании множества путей в DCN с помощью метода

DFS наиболее эффективной является стратегия Worst-Fit. Дальнейшее повышение эффективности метода DFS при формировании множества путей связано с учетом особенности иерархической организации DCN и пропускной способности каналов связи между узлами сети.

Метод DFS использует централизованный способ маршрутизации при котором центральный контроллер содержит всю информацию, необходимую для формирования маршрутов. Кроме топологии сети центральный контроллер содержит информацию о пропускной способности каналов связи, которая используется для выбора канала связи на более нижнем уровне или для возврата на уровень выше по дереву. После формирования очередного пути допустимая пропускная способность каналов связи, принадлежащих выбранному пути уменьшается на заданную величину.

На первом шаге алгоритма определяется уровень топологии на котором могут быть объединены узлы отправитель и получатель информации. Это делается на основании того, что центральный контроллер содержит информацию о топологии сети. Если узлы отправитель и получатель информации подключены к одному и тому же граничному коммутатору, то он напрямую связывает эти узлы. В противном случае путь проходит через несколько коммутаторов, количество которых зависит от взаимного расположения узлов.

Определение уровня соединения узлов позволяет упростить процесс маршрутизации. В этом случае исключаются лишние переходы между уровнями.

Для ускорения перехода между уровнями в модифицированном алгоритме вводится глобальная переменная next, которая указывает направление обхода узлов сети. При next=1 перемещение должно осуществляться вверх по дереву. При next=-1 следует выбирать путь вниз по дереву. При прохождении через уровень на котором могут быть связаны узлы отправителя и получателя значение переменной next становится равной -1. Это говорит о том, что обход достиг локального максимума и нужно двигаться вниз по дереву.

Если в результате обхода достигается крайний коммутатор, который не связан с получателем, переменная next принимает значение next=1, что бы вернуться на предыдущий уровень. Использование переменной next является ключевым при

реализации алгоритма DFS, так как позволяет учитывать особенности иерархической топологии.

Таким образом, модифицированный алгоритм DFS состоит из следующей последовательности операций:

1. Начало.
2. Определение уровня соединения пути.
3.  $next:=1$ .
4. Установить узел отправитель текущим узлом.
5. Добавить текущий узел к узлам пути.
6. Если текущий узел не является конечным коммутатором, то переход к пункту 9.
7. Если текущий узел является конечным коммутатором получателя, то переход к пункту 14
8. Установить предыдущий узел текущим.
9. Если достигнут верхний уровень, то установить  $next = -1$ .

10. Выбрать следующий узел среди смежных узлов.
11. Если не удалось выбрать следующий узел, то переход к пункту 8.
12. Установить выбранный узел текущим.
13. Переход к пункту 5.
14. Добавить узел получатель к пути.
15. Конец.

### 3. Моделирование и анализ эффективности предлагаемого способа формирования множества путей в DCN.

С целью анализа эффективности предлагаемого метода формирования множества путей была разработана специальная система моделирования. На рис.2 представлен пример формирования путей между 1 и 5 узлами DCN. Длина  $i$ -го пути ( $L_i$ ) определяется количеством переходов между исходной и конечной вершинами и равна  $N-1$ , где  $N$  – общее число вершин данного пути.

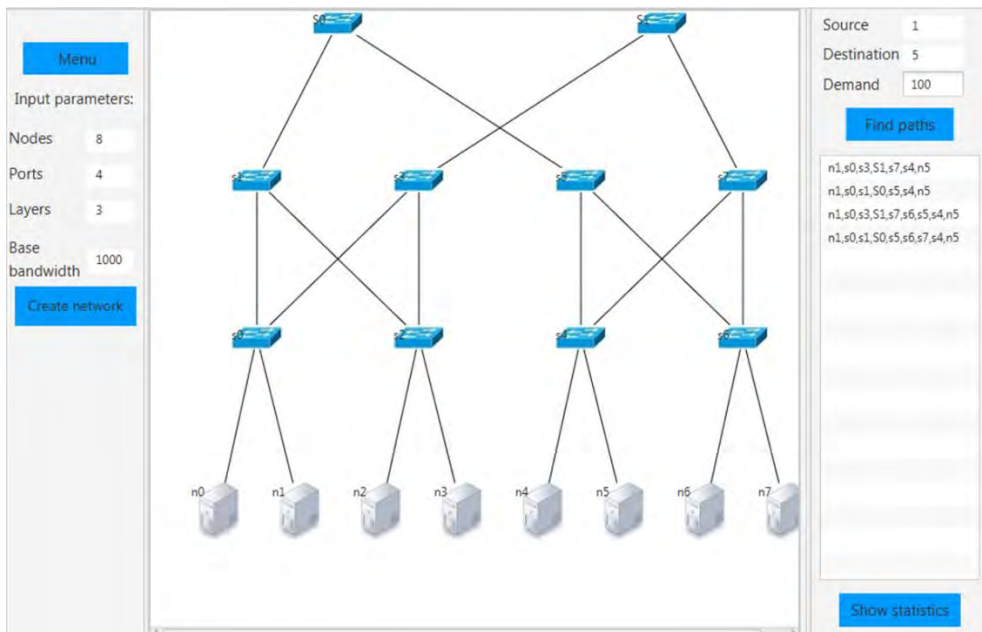


Рис.2. Формирование путей в DCN

В результате работы алгоритма формируется четыре пути между вершинами  $n_1$  и  $n_5$ :

1.  $n_1,s_0,s_3,s_1,s_7,s_4,n_5$ ;  $L_1=6$ .
2.  $n_1,s_0,s_1,s_0,s_5,s_4,n_5$ ;  $L_2=6$ .
3.  $n_1,s_0,s_3,s_1,s_7,s_6,s_5,s_4,n_5$ ;  $L_3=8$ .
4.  $n_1,s_0,s_1,s_0,s_5,s_6,s_7,s_4,n_5$ ;  $L_4=8$ .

В данном случае средняя длина пути  $L_{cp}=7$ .

Ниже представлен псевдокод модифицированного алгоритма DFS формирования пути в DCN:

```

DFS(G, a, b, d) { // G: network, a: source, b:
destination, d: demand
1 H=necessary-layer-to-connect(G, a, b);
2 path={};
3 u=a; // temp variable indicating current
location
4 next=1; // search direction flag, 1: upstream, -
1: downstream
5 return SEARCH(u, path, next);
}
SEARCH(u, path, next) {
1 path=path+u;

```

```

2 if(u=b) return true;
3 if ( layer-of(u)=H) next=-1;// reverse search
direction after reaching connecting layer
4 if(next=-1&& layer-of(u)=1) return false;//
failure at bottom layer
5 neighbors={v| layer-of(v)= layer-of(u)+next,
and available bandwidth of link(u, v)≥d};
6 found=false;
7 while (neighbors!=0&&found=false) {
8 v=worst-fit(neighbors); neighbors=neighbors\
{v};
9 found=SEARCH(v,path,next);
10 };
11 return found;}
    
```

В табл. 1 приведена зависимость длины путей от количества узлов DCN, полученная в результате моделирования базового и модифицированного алгоритма маршрутизации.

**Таблица 1. Зависимость длины пути от количества узлов DCN**

Количество узлов	Длина пути	
	Базовый алгоритм	Модифицированный алгоритм
8	9	7
16	15	7
32	27	7
64	51	7
128	99	7
256	195	7
512	387	7
1024	771	7

Из таблицы видно, что при базовом алгоритме, зависимость длины пути от количества узлов носит линейный характер, а для модифицированного метода остается постоянной. Это связано с тем, что при базовом алгоритме осуществляется последовательный перебор всех путей. При модифицированном

алгоритмы учитывается самоподобная топология сети, что позволяет каждый раз выбирать оптимальный путь и сократить количество итераций. В табл.2 приведена зависимость числа итераций при обычном и модифицированном алгоритме в зависимости от количества узлов DCN. Как видно из табл.2 преимущество предлагаемого алгоритма увеличится с ростом количества узлов в сети DCN.

**Таблица 2. Зависимость числа итераций от количества узлов DCN**

Количество узлов	Число итераций	
	Базовый алгоритм	Модифицированный алгоритм
8	8	6
16	14	10
32	26	18
64	50	34
128	98	66
256	194	130
512	386	258
1024	770	514

**5. Выводы по работе**

В работе предложен и обоснован алгоритм формирования множества путей в DCN. Данный алгоритм является модифицированным алгоритмом поиска в глубину. Предложенный алгоритм максимально учитывает особенности сетевой топологии Fat tree, что способствует более эффективному поиску множества путей передачи данных. В свою очередь применения режима worst-fit позволяет в рамках сетевой топологии Fat tree позволяет обеспечить линейную временную сложность формирования множества путей и обеспечивает лучшую балансировку нагрузки.

**Список литературы**

1. M. Al-Fares “A scalable, commodity data center network architecture,”/ M. Al-Fares, A. Loukissas, and A. Vahdat, // in Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication. New York, NY, USA: ACM, 2008, p. 6374. [Online]. Available: <http://doi.acm.org/10.1145/1402958.1402967>
2. Mysore R. N. Portland: a scalable fault-tolerant layer2 data center network fabric. [Text] / Mysore, R. N., A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, A. Vahdat // ACM SIGCOMM Computer Communication Review – SIGCOMM '09 – ACM New York, NY, USA, 2009. – URL: <http://dl.acm.org/citation.cfm?id=1592575>.

3. J. Kim “Technology-driven, highly-scalable dragonfly topology,”/ . J. Kim, W. J. Dally, S. Scott, and D. Abts, // in Proceedings of the 35th Annual International Symposium on Computer Architecture, ser. ISCA '08. Washington, DC, USA: IEEE Computer Society, 2008, p. 7788. [Online]. Available: <http://dx.doi.org/10.1109/ISCA.2008.19>
4. Eric Jo. A simulation and emulation study of sdn-based multipath routing for fat-tree data center networks. [Text] / Eric Jo, Deng Pan, Jason Liu, Linda Butler // Proceedings of the 2014 Winter Simulation Conference. IEEE Press Piscataway, NJ, USA, 2014. – URL: <http://dl.acm.org/citation.cfm?id=2694235>.
5. Omair Fatmi Distributed multipath routing for data center networks based on stochastic traffic modeling. [Text] / Omair Fatmi, Deng Pan.// Networking, Sensing and Control (ICNSC), IEEE 11th International Conference. – Miami, FL, USA, 2014. – URL: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6819683>.
6. Eun-Sung Jung Distributed Multipath Routing Algorithm for Data Center Networks/ Eun-Sung Jung, Venkatram Vishwanath, Rajkumar Kettimuthu// 2014 International Workshop on Data Intensive Scalable Computing Systems 2014 IEEE , S. – 49-56.
7. Кулаков Ю.А. Разработка и моделирование процесса безопасной многопутевой передачи информации в мобильных сетях / Кулаков Ю.А., Коган А.В., Пирогов А.А. // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: збірник наукових праць. – К.: Век+, 2011. – № 54. – С. 145-149.
8. Гуровиц В.М. Алгоритм поиска в глубину. [Электронный ресурс] – Информатика (Теория графов), 2014. – Режим доступа: <http://foxford.ru/wiki/informatika/algoritm-poiska-v-glubinu>