Multi-constrained Routing Optimization for Data Center Network

Feng-jun Shang and Xing Chen

ABSTRACT

Server-centric data center (SCDC) architecture is a new data center architecture with high throughput, scalable architecture and high fault tolerance. Some SCDC-based routing algorithms had been proposed, but these algorithms have the following disadvantages: 1) The most advanced routing algorithms in SCDC are only considering hop count in path selection; 2) Traditional QOS multi-constrained routing algorithms usually find one available path and is oriented to the switches; 3) The current multi-path algorithm cannot guarantee the quality of the selected path. Therefore, we propose a multi-constrained routing algorithm in the data center network. The SCDC topology is used to reduce the complexity of the algorithm, and it has high possibility to find the correct optimal path.

INTRODUCTION

In recent years, data centers have been widely used to meet growing business needs, and enterprises, service and content providers rely on data center resources for business operations and network services. The data center network (DCN) is an important part of the modern data center, which must have a high reliability and can provide satisfactory performance. First, traffic between servers in the data center dominates the traffic in DC. Second, the size of the data-center grow rapidly. So efficient routing in the data center becomes a necessary and challenging part of the DCN. In order to deal with the above problems, SCDC concept was proposed [2] [3]. Server is not only a terminal host but also a relay switch in SCDC. But the advanced SCDC routing algorithm is topological-related which is designed for a specific topology. Multi-constrained QOS routing algorithms are widely used in traditional networks such as Multi-Constrained Shortest Path (MCSP) [4], Multi-Constrained Optimal Path (MCOP) [5], etc., so the multi-constrained optimal path problem is introduced into SCDC to overcome the above problems.

Feng-jun Shang¹, Xing Chen²*
¹College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, China
²College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, China
Corresponding author: 1447924649@qq.com
RELATED WORK

Traditional data centers have five major defects: such as no performance isolation, limited management flexibility and so on. In order to overcome these problems, many network architectures and routing algorithms have been proposed. For example, the BCube topology-related routing algorithm allocates the server address according to the location feature. The algorithm finds the intermediate server by correcting a single digit of the previous server address. The topology-related routing algorithm works well only in a particular architecture and only selects paths based on hop count. In [7], the SPAIN multi-path selection algorithm needs to find the shortest path many times. It is only based on hop count every time which bring the result that the path quality cannot be guaranteed. In the general network routing field, many algorithms have been proposed to solve the multi-constrained QOS routing problem. H MCOP has a better performance than all previous multi-constraints algorithms. It can be very high probability to find a feasible path. Then this problem attracts a lot of people's attention. There are several algorithms for improving the performance of H MCOP, such as TS MCOP [9] and EH MCOP [10]. TS MCOP improved H MCOP to achieve the best results. These algorithms look for optimal paths well, but they can’t be used to find multi-path, so they cannot be used directly in a server-centric network environment.

FOUNDATION AND DEFINITIONS

Constraints will be divided into the following three constraints according to the different characteristics:

1) Additive constraints: such as hop count, delay, jitter, cost, and so on. The weights of the additional constraints are expressed as the sum of weights for all links (1)

\[ w_j(P) = \sum_{i=1}^{j} w_j(e_i) \]  

(1)

2) Multiplicative constraints: such as error rate, packet loss rate. By converting logarithm form of multiplicative constraints to additive constraints, the weight can be calculated with (2).

\[ w_j(P) = \prod_{i=1}^{j} w_j(e_i) = e^{\sum_{i=1}^{j} \ln(w_j(e_i))} \]  

(2)

3) Concave constraints: eg bandwidth. (3) is used to compute the concave constraint, which can be directly used as the constraint condition of the selected path.

\[ w_j(P) = \min \{w_j(e_1), w_j(e_2), ..., w_j(e_n)\} \]  

(3)

In order to calculate multiple constraints in a function, Jaffe [19] uses a linear function to represent the path cost, as in [4]:

\[ COST(P) = \sum_{i=1}^{k} d_i w_i(P) \]  

(4)

COST (P) represents the cost of path P and \(d_i\) is the coefficient of \(w_i\).

However, the linear function cannot be a good response to the real constraints. In order to fit the real constraint, nonlinear function (5) is used to calculate the path cost [6].

\[ COST(P) = \left(\sum_{i=1}^{k} \frac{w_i(P)}{e_i} \right)^\frac{1}{\gamma} \]  

(5)
When \( q \to \infty \)

\[
COST_i(P) = \max_{i \in P} \frac{w_i(P)}{C_i}
\]  

(6)

Definition 1 The optimal path: Mark all feasible paths from \( v_i \) to \( v_j \) as \( P_1, P_2, \ldots, P_L \), and use formula (5) to calculate the path cost. The optimal path \( P_0 \) satisfies:

\[
COST(P_0) \leq COST(P_i) \quad 1 \leq i \leq L.
\]

Definition 2 Neighbor node pair: In SCDC, if two servers are directly connected or indirectly connected through a switch, the two servers are neighbor node pairs.

Definition 3 Neighbor node matrix: In the SCDC network with \( N \) servers, the neighbor node matrix \( M_1 \) is a matrix of \( N^2 \). Each element \( v_{i,j} \) contains the number of hops, the weight vector and the path connecting \( v_i \) and \( v_j \). If \( v_i \) and \( v_j \) is not neighbor node, mark \( v_{i,j} \) as 0, else modified hop count.

Definition 4 Path length: If \( v_i \) and \( v_j \) are neighbor node pairs, we define the path length of \((v_i, v_j)\) as 1. If \((v_i, v_j) (v_j, v_k)\) is neighbor node pair, then there is a path \((v_i, v_j, v_k)\) between \( v_i \) and \( v_k \), this path length is 2. It can apply similar definition length x path.

**ALGORITHM DESIGN**

The basic idea of the algorithm is to find paths which path lengths is \( N + 1 \) by using paths of length 1 and paths of length \( N \). The algorithm uses the basic idea of Warshall to search for alternate path. It is an efficient algorithm to find the binary closure of relational relations. However, this algorithm can only be a network of two nodes of relevance, and with high complexity. The SCDC topology can help to reduce the time complexity.

---

**Figure 1. Dcell(2,1).**
We use a small network as an example to illustrate that a small DCell network model \((n = 2, k = 1)\) with six servers and three switches is created in Figure 1. We use four constraints \([c_1, c_2, c_3, c_4]\), \(c_1\) and \(c_2\) are additive constraints, \(c_3\) is the multiplicative constraint, and \(c_4\) is the concave constraint. So each path \(P\) has a corresponding four weights, the weight vector is:

\[
W_p = [w_1(P), w_2(P), w_3(P), w_4(P)]^T
\] (7)

If any two servers \(v_i, v_j\) \((1 \leq i \leq 6, 1 \leq j \leq 6)\) are neighbor node pairs, the path cost can be calculated by (5). Then the hop count and the related information (RI): weight vector and the cost value of \(v_{ij}\) are stored in \(M_1\). And then \(M_2\) is constructed according to \(M_1\) to record the feasible paths of length 2 and related information. Take \(v_1\) as an example. In order to find a viable path with path length of 2, we first search all neighbor nodes of \(v_1\) in \(M_1\), and then we get \(v_2\) and \(v_6\). We can reach \(v_1, v_4, v_5\) through \(v_2\) and \(v_6\) respectively. \((v_1,v_2,v_4)\) and \((v_1,v_6,v_5)\) can be get by remove duplicate paths and nodes. We calculate the path weight by using (1) (2) (3). Here we use \((v_1,v_2,v_4)\) as an example: because of the constraints that \(c_1\) and \(c_2\) are additive constraints, \(c_3\) is multiplicative constraints, \(c_4\) is concave constraints, so that:

If all weights of a path meet the constraint requirements, we calculate the path cost:

\[
COST(v_1,v_2,v_4) = \left\{ \sum_{i=1}^{4} \left[ \frac{w_j(v_i,v_{ij})}{c_i} \right] \right\}^{\frac{1}{q}}
\] (8)

If the path does not satisfy the restriction, the path is discarded. If the path \(P_i\) does not meet the requirement, the path containing the path of \(P_i\) is not satisfied. In the example, we assume that all the constraints are satisfied, we record the information in \(M_2\), and so on can be obtained \(M_x\). \(M_1\) and \(M_2\) are in Table 1.

The specific process of the algorithm is as follows:
1. for(u=1; T > T_min; u++)
2. for(i=1; i<=N; i++)
3. If(u==1) then Find all neighbor nodes{v_j1,v_j2,...} of v_i from its ID
4. If (v_i,v_jk) meets all constraints then Calculate_Cost(v_i,v_j)
5. Add[v_i,v_j] into Matrix[u,i,j]
6. End if
7. else
8. A=GetNeighbor(v_i)
9. v_y=any node in A
10. If[v_y,...,v_j] is in Matrix[u-1,i,j] then
11. Calculate_Cost(v_i,v_y,...,v_j) Add[v_i,v_y,...,v_j] into Matrix[u,i,j]
12. End if
13. End if
14. dE = Cost( P_Temporary optimal) - Cost( P_new ) ;
15. If(dE>=0) P_Temporary optimal=P_new
16. End for
17. T=r*T cooling
18. End for

Figure 2. Server-Centric Multi-Constrained Routing Algorithm pseudo-code.

The matrix Matrix [u, i, j] records all paths with length u from v_i to v_j, and their weight vectors and costs. The first loop is to find all paths and related information under simulated annealing. The tenth line finds all neighbors of v_i. Lines 6 and 14 calculate the path cost, and Lines 15 and Lines 16 determine whether the newly obtained path is superior to the previous best path. Then we get the optimal path and the suboptimal path.

ANALYSIS

We will prove that if there exists an optimal path satisfying the multi-constrained requirements, the algorithm can guarantee to find it. Assume that the source server is v_i, and the destination server is v_j. And there is an optimal path meeting multi-constrained requirements, assume it as (v_i, v_{m1}, v_{m2},..., v_{mn}, v_j). So this optimal path’s sub-paths((v_{m1}, v_{m2},..., v_{mn}, v_j), (v_{m2},..., v_{mn}, v_j), (v_{m3},..., v_{mn}, v_j),..., (v_{mn}, v_j)) all meet the multi-constrained requirements. Then node pairs(v_i,v_{m1}), (v_{m1}, v_{m2}),..., (v_{mn}, v_j) are all in neighbor node matrix M_1. For the reason that (v_{m,n-1}, v_{mn}) and (v_{mn}, v_j) are in M_1, (v_{mn,n-1}, v_{mn,v_j}) is in M_2. Due to (v_{m,n-2}, v_{m,n-1}) is in M_1, (v_{m,n-2}, v_{mn,n-1}, v_{mn}, v_j) is in M_3. And so on in a similar fashion, the path (v_i, v_{m1}, v_{m2},..., v_{mn}, v_j) must be in the matrix M_{n+1}. So the algorithm can guarantee us to find the optimal path if it exists.
CONCLUSION

MCMP problem is a very important problem in SCDC that has not been solved well. In this paper, a multi-constrained routing algorithm based on SCDC is proposed to solve MCMP problem. This algorithm uses SCDC topology to reduce the complexity of the algorithm and simplify the routing process, and there is a great possibility to find the optimal path. The cost of this optimal path is less than the cost of another routing algorithm optimal path.

ACKNOWLEDGMENTS

The author would like to thank the Chongqing Basic and Frontier Research Project under Grant NO. cstc2016jcyjA0590. The work is partly funded by the National Nature Science Foundation of China (No. 61672004).

REFERENCES