ACEAIT-5239

On IP Network Traffic Engineering Performance of Modified Branch Exchange and Interior Point Method

Kairat Jaroenrat*, Tichakon Sumrongsub, Thanachot Thitivichienlert and Natdanai Suksaeng
Department of Computer Engineering, Faculty of Engineering at KamphaengSaen, Kasetsart University, Malaiman Road, NakhonPathom, Thailand
kairat.j@ku.ac.th

Abstract
Because of a lot of computer networks, the communication over networks is increasing. A greater number of users causes the network performance to be reduced. We has pointed to the current study of the routing protocol commonly used most widespread in the Internet, which is the Open Shortest Path First (OSPF). Then, in order to make the better network performance, we apply the Branch Exchange and compare with the Primal-Dual Interior Point method in performance of determining the appropriate weight of the network links and processing time. We found that the performance of the Branch Exchange is based on the number of link chosen to exchange the weight with the others. If this number is high, the data flow performance of Branch Exchange method is nearly the Interior Point method But it takes much more processing time. If this number is low, the data flow performance of Branch Exchange method is also reduced slightly from Interior Point method but it takes much less processing.

Keyword: Branch Exchange, Primal-Dual Interior Point, Traffic Engineering

1. Introduction

Nowadays, computer networks have more important roles and the data exchange in networks has been widely used. Therefore, to increase the ability of the system to be more efficiently and reduce the delay of the overall system, we have to consider the weight-setting mechanism of the routing protocols. The most famous Routing Protocols in the system is Open Shortest Path First (OSPF). Because the OSPF routers use Shortest-Path algorithm to find its routes, routing of traffic flow does not get together with the traffic requirement of such system. Therefore, in order to increase efficiency and reduce cost of the system, we need to replace the default weight-setting mechanism with some good performance algorithm.

Traffic routing, often called traffic engineering, is to distribute load for given traffic demands over the installed link such that performances are optimized. The most important issue is that IP network is a datagram network, in which the routing protocols route traffic over path with shortest distance, i.e. sum of link weight. Most weight-setting algorithms are fairly complex. For example simple branch exchange algorithm [1] requires complexity of $O(N^5)$, where N is number of nodes, which is prohibitive for moderate to large size networks. However, link weight setting for an optimum routing pattern is also a complex problem or even unfeasible [2] [3] [4].

2. Background

2.1 Open Shortest Path First

OSPF is one of the most widely used routing protocol in the IP network with several good features. Using Dijkstra algorithm, OSPF router can route the data packet to the shortest path, which has a minimum total weight. Typically, we set the initial link's weight (default weight) with the value calculated from equation (1).

$$\text{Weight} = \frac{10^8}{\text{Bandwidth}} \qquad (1)$$

2.2 Objective Function

Consider a directed network graph G = (N, A) with a capacity c_a for each $a \in A$ and as define in previous section, d_{st} denote the amount of traffic flow between s and t. Let f_a^{st} indicate how much of the traffic flow from s to t over arc a, traffic load l_a over link $a \in A$ is the sum of all f_a^{st}. It is suggested in [3] to measure the performance of network by cost function:

$$\Phi = \sum_{a \in A} \phi_a(l_a, c_a), \qquad (2)$$

where $\phi_a(l_a, c_a)$ is an M/M/1 queuing theory style link cost function given by

$$\phi_a(l_a, c_a) = l_a / (c_a \quad l_a) \qquad (3)$$

With this function, it is more expensive to send flow along arcs whose loads approach capacity, which is what we want. However, the function does not deal with overloaded links, i.e. $l_a \geq c_a$. To overcome this problem, $l_a/(c_a \quad l_a)$ is approximated by a piece-wise linear function $\phi_a(0) = 0$ and derivative

$$\phi_a(l_a, c_a) = \begin{cases} 1 & \text{for } 0 \leq l_a / c_a < 1/3, \\ 3 & \text{for } 1/3 \leq l_a / c_a < 2/3, \\ 10 & \text{for } 2/3 \leq l_a / c_a < 9/10, \\ 70 & \text{for } 9/10 \leq l_a / c_a < 1, \\ 500 & \text{for } 1 \leq l_a / c_a < 11/10, \\ 5000 & \text{for } 11/10 \leq l_a / c_a < \infty. \end{cases} \qquad (4)$$

2.3 Optimum Solution

With piece-wise linear cost function define by (3), the general routing problem can be formulated as the following linear programming [2] [3].

$$\text{Min } \Phi = \sum_{a \in A} \phi_a \qquad (5)$$

Subject to:

$$\sum_{u:(u,v)\in A} f_{u,v}^{s,t} - \sum_{u:(u,v)\in A} f_{v,u}^{s,t} = \begin{cases} d_{st} \text{ if } v = t \\ -d_{st} \text{ if } v = 0 \\ 0 \text{ otherwise} \end{cases} \qquad v,s,t \in N, \qquad (6)$$

$$\phi_a \geq l_a \qquad a \in A, \qquad (7)$$

$$\phi_a \geq 3l_a - 2/3c_a \qquad a \in A, \qquad (8)$$

$$\phi_a \geq 10l_a - 16/3c_a \qquad a \in A, \qquad (9)$$

$$\phi_a \geq 70l_a - 178/3c_a \qquad a \in A, \qquad (10)$$

$$\phi_a \geq 500l_a - 1468/3c_a \qquad a \in A, \qquad (11)$$

$$\phi_a \geq 5000l_a - 19468/3c_a \qquad a \in A, \qquad (12)$$

$$l_a = \sum_{t \in N} f_a^{s,t} \qquad a \in A, \qquad (13)$$

$$f_a^{s,t} \geq 0 \qquad a \in A;\ t \in N. \qquad (14)$$

Constraint (5) are flow conservation constraints; constraints (6) – (11) describe the cost function; and constraint (12) define the load on each arc.

2.4 Branch Exchange

Branch Exchange [6], is a method to design distributed networks, start by connecting all nodes with initial connections and then try to add, drop or exchange the links to get better efficient network. Normally, the Branch Exchange method can be used to solve the network design problem by adding, reducing and exchanging network cables in the network. The method used in the exchange links is called the combination However, since the purpose of this research is to adjust the links' weight of the network, we modified the links exchange mechanism to be the links' weight exchange.

If the number of links in the network is n, there will be n! solutions. But combination is the objects order in a components set with the no sequence-priority consideration. Therefore, to reduce the number of solutions and speed up the process, we exchange only r links ($r \leq n$) which have the highest utilization. Then the number of solutions can be calculated in equation (15).

$$\binom{n}{r} = \frac{n!}{r!(n-r)!} \qquad (15)$$

when n is the number of links in the network,
r is the user-defined number that only r links with the highest utilization can be exchanged.

2.5 Primal-Dual Interior Point

N.K. Karmarkar proposed a linear programming solution called Interior point method [7] which has better speed of convergence than the traditional linear programming (Simplex method). For Simplex method, the variables' movement is in only the corners and the boundary of the possible answers until the best solution is founded. But for Interior point method, variable move inside the possible area of the answers until you get the best solution. So, this method uses less number of iterations than Simplex method to find the best solution.

Interior point method is a method that uses iterative technique to find the solution with three main steps in each iteration. First step is to calculate the starting point which is in the possible area of the answer. Next step is to evaluate the answer by using termination tolerance and maximum iteration values. Last step is to move to find the better answer with Karuh Kuhn-Tucker condition and Newton method.

3. Experiments

Before the main experiment, we compare the fast near-optimum method to the slow optimum method because we cannot use a slow simplex for a comparison to our modified branch exchange method. Then we compare the primal-dual interior point with constraint relaxation to the traditional simplex by testing 50 networks with C++ and GLPK[5]. The networks in this test are 10-nodes networks which are obtained by varying the following design parameters: 5 sets of different topology and node degree $\in \{1.0, 1.1, 1.2, 1.3, 1.5, 1.7, 1.8, 2.0, 2.2, 2.5\}$.

After that, we created a simulation software and synthesize 4 sets of each 12, 14 and 16 nodes networks with 4 different node degree: 1.3, 1.5, 1.7 and 2.0 on Intel Core i3, 2.13 GHz machine and then the traffic demand matrix for each set of nodes is also generated. By varying synthesis parameters, a total of 48 networks are generated using the full-duplex link of 10 Mbps capacity. Then we apply the modified Branch Exchange method to find the solution with 6 values of r-parameter by using C++ language and compare to the near-optimum solution which is solve by Interior point method.

3.1 Routing Performance

Routing cost is used to measure the efficiency of capacity assignment in term average network delay. Fortz and Thorup [2] defined the routing cost as shown in (5), then the routing performance can be calculated from (16).

$$\text{Routing Performance (\%)} = \frac{c_S * 100}{c} \qquad (16)$$

when c is the routing cost of the network which was weight calculated by user's methods and c_s is the routing cost of the network which was weight calculated by standard methods for a comparison.

3.2 Time Performance

As the routing cost, the processing time can be used to measure the efficiency of the traffic engineering algorithm. The time performance can be calculated from (17).

$$\text{Time Performance (\%)} = \frac{t \times 100}{t_s} \qquad (17)$$

when t is the processing timed of the user's methods and t_s is the processing time of the standard methods for a comparison.

4. Performance Evaluation

4.1 Linear Programming Solution

The performance comparisons between the primal-dual interior point with constraint relaxation and the traditional simplex method were shown in Fig.1 and Fig.2.

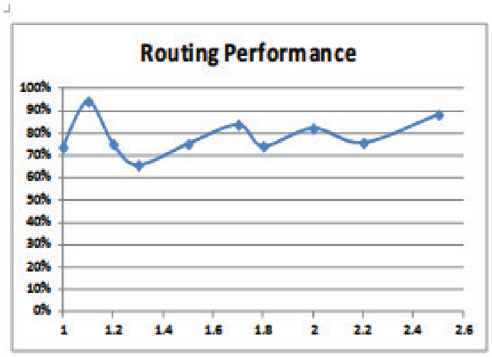

Figure 1: Routing performance of Interior Point method

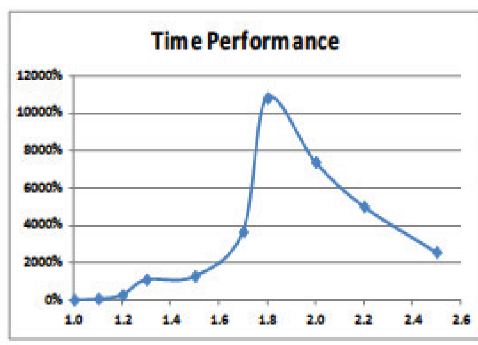

Figure 2: Time performance of Interior Point method

From Fig.1, we see that the routing performance of the interior point method, which we use in this research, is in the high range about 65 to 95% of simplex method. But the time performance shown in Fig.2 show that interior point method gives a very less processing time in more complex network.

4.2 Modified Branch Exchange Solution

The performance comparisons between our modified branch exchange method and the primal-dual interior point with constraint relaxation were averaged and shown in Table 1.

Table 1. Performance comparison between modified branch exchange method and the primal-dual interior point method.

	Routing Performance			Time Performance		
r	12 nodes	14 nodes	16 nodes	12 nodes	14 nodes	16 nodes
1	140.01%	58.72%	39.42%	2716468.59%	677134.03%	29143898.00%
2	150.50%	61.70%	44.81%	395647.87%	49105.36%	2898550.72%
3	158.58%	64.68%	48.63%	26671.11%	2675.81%	128307.94%
4	170.42%	67.35%	50.53%	1423.87%	134.34%	5204.94%
5	173.86%	70.11%	55.34%	104.08%	7.19%	323.82%
6	171.66%	70.75%	52.95%	6.49%	0.40%	7.07%

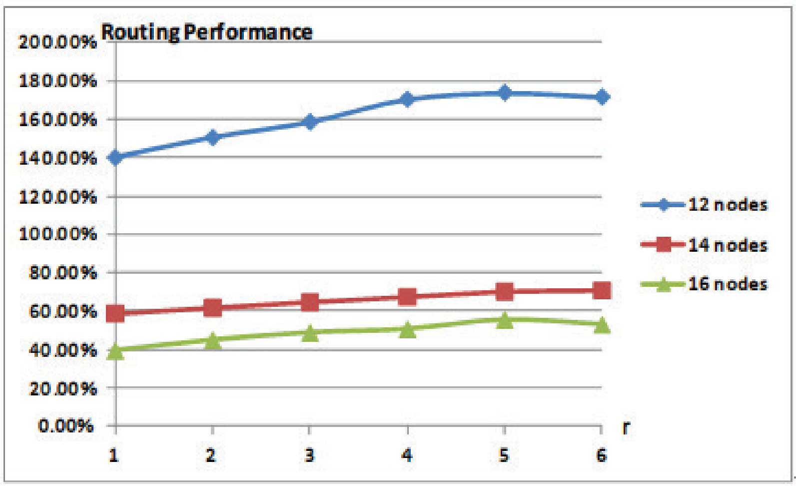

Figure 3: Routing Performance Comparison.

In fig.3, routing performance of 12, 14 and 16 nodes network are compared and plot versus "r" value. We can see that in small networks (such as 12 nodes network), the branch exchange method can give higher routing performance than the interior point method. But in the larger network such as 14 and 16 nodes network, branch exchange give worse result than the interior point method. Moreover, the higher "r" value give the higher routing performance.

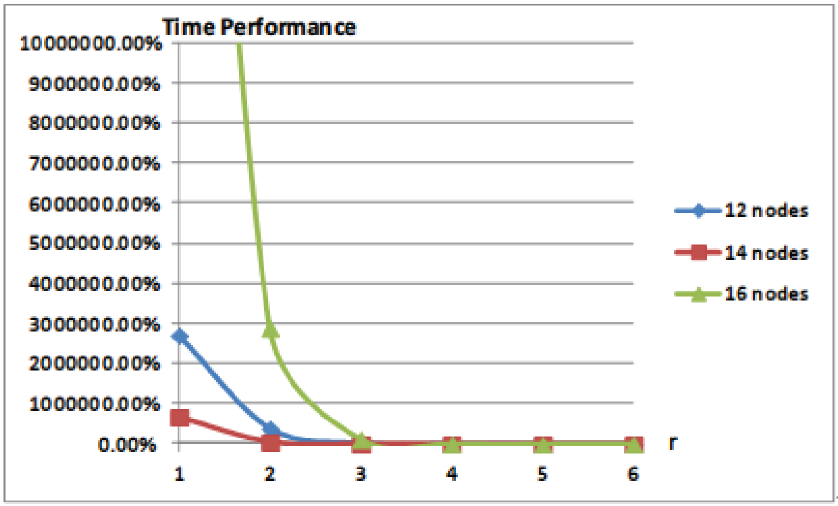

Figure 4: Time Performance Comparison.

In fig.4, time performance of 12, 14 and 16 nodes network are compared and plot versus "r" value. We can see that the branch exchange with low "r" value give the very high time performance because the less cables chosen to be exchange, the greatly less number of cases which was defined by (15) then process will be done much faster.

5. Conclusion

In this research, we apply the principle of branch exchange by adapting the permutation technique for exchange the links' weights in IP network and analyze the routing and time performance compare to the interior point method. We use the simulation program to generates the channels' bandwidth and the data traffic and then get the "r" value which is the number that only r links with the highest utilization can be exchanged by branch exchange method. After that, branch exchange start to find the best solution which has the lowest total routing cost and record the processing time.

The results of the experiment show that the higher number of exchangeable links (r), the less network delay (routing cost) in the network. This means that we can get more efficient weights' set with high processing time. When compared to the interior point methods, it appears to be less routing performance but this method can process faster. But if the number of lines (r), is high, it consumes long process time and may be worse than the interior method.

Therefore, we conclude that the most effective method for determining the links' weight in IP network is the interior point algorithms for Linear Programming. But this method takes a long time to process then it is not suitable for use with large IP network. We proposed the modified branch exchange method to determine the links weights of the IP network. By the way, this method allow you to adjust between the network performance and the processing time which imply that it can be used on a large network by choosing small number of exchangeable links' weight, and it still has the slightly less performance than the linear programming method.

References

[11] Aaron Kershenbaum, 1993. Telecommunications Network Design Algorithms, McGraw-Hill, New York, NY..

[12] Bernard Fortz and Mikkel Thorup, Internet traffic engineering by optimizing OSPF weights, Proceeding of IEEE INFOCOM, 2, Mar. 2000, 519–528.

[13] Bernard Fortz, Jennifer Rexford, and Mikkel. Thorup, 2002, Traffic engineering with traditional IP routing protocols, IEEE Communications Magazine, 40, 118–124.
[14] Walid Ben-Ameur, Eric Gourdin, Bernard Liau, Nicholas Michel, Designing Internet Networks, Proceeding of DRCN2000, Munich, April 2000, 56-61.
[15] Andrew Makhorin. GLPK (GNU Linear Programming Kit), 2000, http://www.gnu.org/software/glpk/
[16] Edelmiro Míguez, José Cidrás, Eloy Díaz-Dorado, and José Luis García-Dornelas, 2002, An improved branch-exchange algorithm for large-scale distribution network planning, IEEE Transactions on Power Systems, 17(4), 931-936.
[17] N. Karmarkar , M. Resende and K. Ramakrishan, 1991, An interior point algorithm to solve computationally difficult set covering problems, Mathematical Programming, 52, 597 -618.