Link Weight Optimization for Routing Considering Link Utilization in Communication Networks

Seong-Lyong Gong, Sung-Yeon, Kim, Jang-Won Lee
Dept. of Electrical & Electronic Engineering,
Yonsei University, Seoul, Korea
Email: {slgong, ksy0725, jangwon}@yonsei.ac.kr

Sang Il Lee, Myung Kil Ahn
Agency for Defense Development (ADD),
Seoul, Korea
Email: {happyjoy, happymkahn}@add.re.kr

Abstract—In this paper, we study the problem that finds optimal link weights for routing in communication networks with a target utilization for each link. Since the problem is known as NP-hard, we take a heuristic approach based on the simulated annealing method. However, to make the algorithm to be more robust and converge fast, we modify the basic simulated annealing method. Numerical results show that our algorithm provides a better performance than the basic simulated annealing method.

I. INTRODUCTION

In commonly used packet routing protocols in communication networks, such as OSPF (Open Shortest Path First) in IP networks, each link in the network is assigned its weight. Based on weights of links in the network, routing protocols find the end-to-end path for each source-destination pair such that the sum of weights of links on the path is minimized, so called the shortest path. Hence, to find a ‘good’ path for each source and destination pair that satisfies some network constraints such as the quality of service (QoS) of each session and the target utilization of each link, it is important to assign the appropriate weight for each link in the network.

In this paper, we study a link weight optimization problem for routing considering the target utilization for each link in the network. Hence, by adjusting the weight of each link appropriately, we find the end-to-end path for each source and destination pair such that the utilization of each link in the network is below its target utilization. Since the utilization of a link directly affects the QoS of its sessions, such as delay and packet loss, it is important to maintain its utilization below a certain threshold.

There are several works for finding the optimal setting of link weights. Fortz and Thorup [1] proved that finding the optimal setting of link weights is NP-hard. Hence, they used a local search algorithm to find the approximated solution. Sqalli et al. [3] proposed a simulated annealing approach to solve the same problem.

In this paper, we will use a simulated annealing approach as in [3]. However, we will propose several new strategies based on the problem structure to improve the performance of the algorithm that results in a faster and more stable convergence.

II. PROBLEM FORMULATION

We consider a network consists of set N of nodes and set A of links. Each link $a \in A$ has a capacity C_a and

weight w_a for routing. We define $\bar{w} = (w_a)_{a \in A}$ as a link weight vector for the network. In addition, there are set Ω of pairs of source and destination nodes. The pair of source node s and destination node t is denoted as (s, t) . Each pair (s, t) has its traffic demand d_{st} that should be transmitted from node s to node t through a single path in the network. The routing is done with the shortest path algorithm with the link weight vector $\bar{w}$. We define $L_{(s,t)}(\bar{w})$ and $P_a(\bar{w})$ as set of links on the shortest path of source-destination pair (s, t) and set of source-destination pairs that has link a in their shortest path, respectively, given link weight vector $\bar{w}$. Then, the utilization of each link a given link weight vector $\bar{w}$ is defined as $U_a(\bar{w}) = \sum_{(s,t) \in P_a(\bar{w})} d_{st} / C_a$. Each link a has its target utilization γ_a and its cost function $\Phi_a(\cdot)$ that is defined as an increasing function of its normalized utilization $NU_a(\bar{w})$, where $NU_a(\bar{w}) = U_a(\bar{w}) / \gamma_a$. We assume that the cost function has the following property:

- If $NU_a(\bar{w}) < 1$, i.e., link utilization is less than its target, it increases slowly;
- If $NU_a(\bar{w}) > 1$, i.e., link utilization is greater than its target, it increases very fast.

Hence, the cost function penalizes the over-utilization of the link more severely. For example, the cost function can be the following form:

$$\frac{d\Phi_a(x)}{dx} = \begin{cases} 0.1 & , \text{ if } x \leq 1, \\ 100 & , \text{ otherwise} \end{cases} \quad (1)$$

Now the optimization problem is formulated as:

$$\text{minimize } \bar{w} \in W \Phi(\bar{w}) = \sum_{a \in A} \Phi_a(NU_a(\bar{w})),$$

where W is a set of all possible link weight vectors.

III. ALGORITHM

As mentioned before, in general, our problem is NP-hard. Hence, in this paper, we take a heuristic approach based on the simulated annealing method [2]. Due to the page limitation, we omit the details for the simulated annealing method and we focus only on describing how we modify the basic simulated annealing method to improve the performance of the algorithm.

A New probability function for accepting bad solutions
In the basic simulated annealing method, in each iteration, a

new candidate solution is generated randomly. If the candidate solution provides a lower cost than the current solution, it accepts the candidate solution as a new solution. However, to prevent from converging to the local optimal solution, even though the candidate solution provides a higher cost than the current one, it also accepts the candidate solution as a new solution with probability

$$\exp\left(-\frac{\Delta\Phi}{T}\right), \quad (2)$$

where $\Delta\Phi$ is the increment of cost and T is the searching range in the current iteration that exponentially decreases over iterations. We can adjust the tradeoff between the accuracy of the solution and the speed of convergence by controlling some parameters such as decreasing factor of T and initial value of T . However, in our trials, we found that it is not easy to find the good parameters that can be applied to diverse topologies. Hence, we propose a new probability function that substitutes (2) as

$$\exp\left(-\frac{\Delta\Phi/\Phi_{min}}{(T/T_0)^\kappa}\right), \quad (3)$$

where Φ_{min} is the minimum cost up to the current iteration, T_0 is the initial value of searching range, and κ is a positive constant. In our trials, we found that our function in (3) provides more robust results than the original one in (2) for diverse topologies due to the effect of the normalization in (3). In addition, we can easily control the tradeoff between accuracy of the solution and speed of convergence by simply adjusting κ . As we will show later, as κ increases, the speed of convergence also increases.

Utilization-aware weight update

In the basic simulated annealing method, a new candidate solution is generated randomly, i.e., randomly select a link and also randomly change its weight. However, in our problem, the objective is to maintain the utilization of each link below its target and we can evaluate the current utilization of each link. Hence, in our strategy, each link a calculates x_a based on its current utilization as

$$x_a = z_a \cdot \exp(-NU_a(\bar{w})),$$

where z_a is a random variable that is uniformly distributed between (0,1), and then, a link with the minimum x among all links is selected. Hence, a link with a higher normalized utilization has a higher probability to be selected. After selecting a link, if its normalized utilization is greater than 1, i.e., its utilization exceeds its target, its weight is increased randomly.

IV. NUMERICAL RESULTS AND CONCLUSION

We consider a network in Fig. 1 with 12 sessions. We use a function in (1) as a cost function for each link. We first show the convergence of our algorithm varying κ in Fig. 2(a) - 2(c). As these figures show, as κ increases, the algorithm converges faster. However, even though it is not clear in these figures, it does not mean that with a large κ , we can always obtain the optimal solution faster. In general, with a large κ , the

Fig. 1. Network topology.

Fig. 2. Numerical results.

probability that the algorithm converges to the local optimum would be large. Hence, in our algorithm, by adjusting only κ , we can control the tradeoff between accuracy and convergence speed of the algorithm.

In Fig. 2(d), we compare the performance of our link selection strategy with that of the basic random link selection strategy. In this figure, we calculate the deviation of the cost at each iteration from the minimum cost and provide its distribution for each scheme. As this figure shows, our utilization-aware link selection strategy provides a better deviation distribution than the random selection strategy, which implies that our strategy provides not only a faster convergence but also more accurate solutions even in transient periods than the random selection strategy.

REFERENCES

- [1] B. Fortz and M. Thorup, "Internet traffic engineering by optimizing ospf weights," in *IEEE INFOCOM*, 2000, pp. 519–528.
- [2] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May, 1983.
- [3] M. Sqalli, S. Sail, and M. Mohiuddin, "An enhanced estimator to multi-objective ospf weight setting problem," in *IEEE/IFIP NOMS*, 2006, pp. 240–247.