

# A Traffic Engineering-Aware Shortest-Path Routing Algorithm in IP Networks<sup>\*</sup>

Youngseok Lee<sup>1</sup> and Biswanath Mukherjee<sup>2</sup>

<sup>1</sup> Department of Computer Science and Engineering,  
Chungnam National University, Daejeon, 305-764, Korea  
`yslee@cs.cnu.ac.kr`

<sup>2</sup> Department of Computer Science,  
University of California, Davis, CA 95616, USA  
`mukherje@cs.ucdavis.edu`

**Abstract.** Single shortest-path routing is known to perform poorly for Internet traffic engineering (TE) where the typical optimization objective is to minimize the maximum link load. Splitting traffic uniformly over equal-cost multiple shortest paths in OSPF and IS-IS does not always minimize the maximum link load when multiple paths are not carefully selected for the global traffic demand matrix. However, a *TE-aware shortest path* among all the equal-cost multiple shortest paths between each ingress-egress pair can be selected such that the maximum link load is significantly reduced. IP routers can use the TE-aware shortest path without any change to existing routing protocols and without any serious configuration overhead. While calculating TE-aware shortest paths, the destination-based forwarding constraint at a node should be satisfied, because an IP router will forward a packet to the next-hop towards the destination by looking up the destination prefix. In this paper, we present a mathematical problem formulation for finding a set of TE-aware shortest paths for the given network as an integer linear program (ILP), and we propose a simple heuristic for solving large instances of the problem. The proposed algorithm is evaluated through simulations in IP networks.

**Keywords:** Shortest-path routing, Traffic engineering, IP, ILP, Simulations, Optimization.

## 1 Introduction

### 1.1 Traffic Engineering

In recent years, the pervasive usage of Internet services has significantly increased the volume of Internet traffic, which has overtaken that of voice traffic.

---

<sup>\*</sup> This work was supported in parts by a Korean Science & Engineering Foundation (KOSEF) Postdoctoral Fellowship Program and by the US National Science Foundation (NSF) Grant No. ANI-02-07864. This work was performed when Dr. Youngseok Lee was a visiting scholar at University of California, Davis.

Due to the popularity of World Wide Web (WWW), audio and video streaming applications, and bandwidth-intensive services such as peer-to-peer applications, the network will become much more congested in spite of increased capacity unless appropriate techniques are designed. Aside from the rapid increase in traffic demand, congestion also happens even in the over-provisioned IP backbone network when traffic is detoured because of link failures. Traffic engineering (TE) is an effective solution to control the network congestion and optimize network performance. As stated in [1,2], traffic engineering encompasses the application of technology and scientific principles to the measurement, modeling, characterization, and control of Internet traffic. The purpose of traffic engineering is to facilitate efficient and reliable network operations while simultaneously optimizing network-resource utilization and traffic performance. The typical optimization objective of traffic engineering is to minimize the maximum link load in the network which represents the overall congestion in the network [3].

Traffic engineering in Internet Protocol (IP) networks is usually done by routing. The Interior Gateway Protocol (IGP) routing protocol in IP networks is based on the shortest-path routing algorithm where the shortest path is computed using link weights. An IP router maintains a routing table consisting of a next-hop (shortest path) and a destination routing prefix. An incoming packet is forwarded to the next-hop after the variable-sized destination IP prefix of the packet is looked up in the routing table, which is called the destination-based forwarding rule. Although the shortest-path routing algorithm is practical and easy to implement, it potentially limits the throughput of the network because only a single path between a source and a destination is used. Theoretically, optimal routing [3] can supply bifurcation of any source-destination traffic at strategic points with alternate routes. Optimal routing is based on the multi-commodity-flow problem formulation, and is also known as minimum-delay routing [4], because it can minimize the average end-to-end delay. Optimal routing can be realized through a form of multipath routing. In the shortest-path algorithm used by Internet routing protocols, multiple shortest paths can be found when the destination can be reached via multiple paths with the same cost from the source. Multi-path routing is employed by some Internet routing protocols in order to increase the total network utilization and the end-to-end throughput. IGPs such as Open Shortest Path First (OSPF) [5] and Intermediate System-Intermediate System (IS-IS) [6] explicitly support the option of multipath routing, which is known as Equal-Cost Multi-Path (ECMP) routing, when there are multiple paths with the same cost. Some router implementations allow equal-cost multipath with Routing Information Protocol (RIP) and other routing protocols. Optimized Multi-Path (OMP) [7] which improves the performance beyond that of ECMP has been proposed for OSPF, IS-IS, and Multi-Protocol Label Switching (MPLS) [8] networks in the Internet Engineering Task Force (IETF). Although ECMP is widely used in OSPF and IS-IS for splitting traffic equally to multiple paths, it does not always minimize the maximum link load if multiple paths are not carefully selected for the global traffic demand matrix.

On the other hand, a recent network technology has been developed, called MPLS, where the path between ingress and egress routers is established by

signaling protocols such as Resource ReSerVation Protocol (RSVP) [9] and a fixed-sized label is attached to IP packets by the ingress router. MPLS provides the functions of fast switching and explicit path setup to IP networks. Intermediate Label-Switched Routers (LSRs) forward packets by looking up not the variable IP prefix but the fixed-sized MPLS label. Thus, traffic engineering in MPLS networks is more efficient than in traditional IP networks due to the explicit Labeled-Switched Path (LSP). However, since MPLS plays the role of the control plane for IP routing, the maintenance cost may increase because of imposing an additional layer to the IP network.

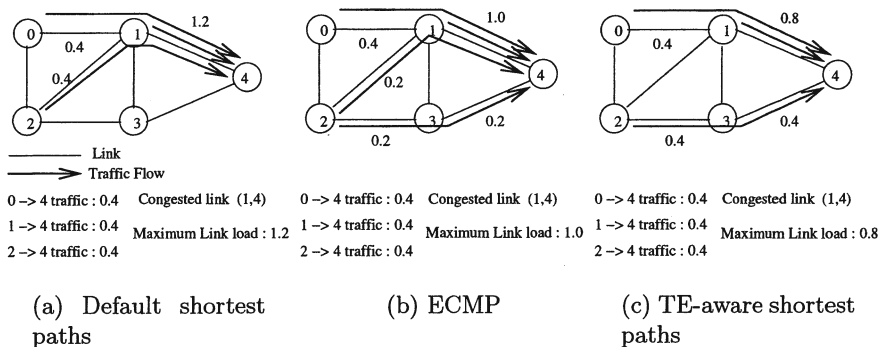


Fig. 1. An example of TE-aware shortest paths.

## 1.2 TE-Aware Shortest-Path Routing

It is well known that single shortest-path routing performs poorly for achieving the traffic-engineering goal. Usually, it is assumed that, when equal-cost multiple shortest paths are available for the routing prefix, IP routers will randomly select one of the possible candidates for the routing prefix without considering the link load. When the network topology and the traffic demand matrix are given, however, a *TE-aware shortest path* can be selected among all possible shortest paths for each node pair such that the maximum link load is significantly reduced even when a single shortest path is used for routing. TE-aware shortest-path routing is simple and easily realized in current networks. It also mitigates the complexity of maintaining multipath routing, in which, for example, the number of multiple paths should be carefully configured at each router for global optimization. In addition, when traffic is distributed per routing prefix to avoid packet reordering, load balancing may not be achieved due to different traffic load per routing prefix. Although TE-aware shortest-path routing uses a single path, it is shown by experiments that it achieves a near-optimal solution as does multipath routing in a typical network topology with a typical traffic demand matrix. It is likely that TE-aware shortest paths will be more viable as

the number of equal-cost shortest paths and the number of destination routing prefixes become large. However, it is shown by experiments that, even under the case of a small number of equal-cost multiple paths, TE-aware shortest paths are useful in reducing the maximum link load. Since it was shown in [10] that small routing prefixes which take a large fraction of traffic volume are enough for achieving the traffic-engineering goal, we assume that the destination routing prefixes are the egress node addresses for simplicity. Hence, when IP routers use TE-aware shortest paths instead of randomly chosen ones, the traffic-engineering goal will be easily achieved without any change to existing routing protocols and without any serious configuration overhead.

Figure 1 shows an example of the different behavior patterns of default shortest-path routing<sup>1</sup>, ECMP, and TE-aware shortest-path routing for a given set of traffic demands. In Fig. 1(a), default shortest-path routing, where node 2 selects node 1 as the next-hop for traffic from source node 2 to destination node 4, results in the high maximum link load of 1.2. If splitting traffic evenly across two paths between nodes 2 and 4 is allowed (Fig. 1(b)), the maximum link load is reduced, but it is not the optimal solution. When the alternate shortest path (2-3-4) is used rather than 2-1-4 from node 2 to node 4 in Fig. 1(c), the maximum link load can be minimized. Therefore, the TE-aware shortest path between source node 2 and destination node 4 is 2-3-4. In this example, optimal shortest-path routing which allows traffic bifurcation is same as TE-aware shortest-path routing. In this paper, we present the TE-aware shortest-path routing problem as an ILP for the optimal solution, and we propose a heuristic algorithm to solve the problem in polynomial time.

## 2 Related Work

For the MPLS network, in [11], the traffic bifurcation problem is formulated in linear program (LP) and heuristics for the non-bifurcating problem are proposed. Although the LP problem formulated in [11] minimizes the maximum of link utilization, its solution does not consider total network resources. Therefore, the routing loop or unnecessary long paths may be generated for the LP solution. Wang, et al., have included the total network resources as the optimization objective in [12] to prevent the unnecessary long paths. In [13], the dynamic routing algorithm for MPLS networks is proposed where the path for each request is selected to prevent the interference among paths for the future demands. It considers only single-path routing for simplicity. Reference [14] proposes a traffic-partitioning mechanism for differentiation in MPLS networks, where an expedited forwarding (EF) traffic demand is divided into multiple LSPs to minimize the average end-to-end delay. However, it needs recalculation of load-splitting ratios whenever new requests arrive or the average rate of requests changes.

On the other hand, Internet traffic-engineering schemes without the MPLS feature have been recently studied. In [12], it is shown that the optimal routing

<sup>1</sup> In this paper, it is assumed that, in default shortest-path routing, a next-hop is randomly chosen for the destination node if multiple next-hops exist.

problem can be transformed into the shortest-path routing problem by choosing appropriate link metrics. Traffic engineering in IP networks can be achieved by changing link weights appropriately [15]. Although link-weight-based traffic engineering is simple, it may cause slow convergence time because of flooding of Link State Advertisement (LSA) messages throughout the network and recalculating all the shortest paths. During the convergence period, unstable network status such as load fluctuation and routing loops may occur. In general, it is not easy for a network operator to determine which link metric should be modified and to predict its impact on the network performance. A recent study [10] showed that traffic bifurcation over equal-cost multiple shortest paths can achieve the near optimal solution when the subset of next-hops are intelligently chosen for each routing prefix.

### 3 Problem Statement for TE-Aware Shortest-Path Routing

In order to calculate the optimal TE-aware shortest paths, we formulate and solve the ILP formulation in two steps: the first step calculates the weights of the shortest paths by minimizing the sum of the integer link weights; and the second step minimizes the maximum link load among the possible sets of shortest paths which satisfy the minimum weights of the shortest paths found in the first step. The reasons behind choosing this two-step ILP formulation are as follows: the first reason is that two dependent objectives need to be minimized; and the second reason is that, when one ILP formulation with the combination of two objectives is used, as in [12], an inappropriately-selected coefficient for combining the two objectives sometimes does not find the lower bound for one objective. For the calculation of TE-aware shortest paths, we assume a centralized TE server which has all the information on the network topology, the traffic demand matrix, the link-load measurement data, and all the routing tables of the network.

The following constants regarding the network parameters are given for the problem statement.

- $G = (N, E)$  : a graph consisting of a set of nodes,  $N$ , and a set of links,  $E$ .
- $\Lambda = \{\Lambda_{sd}\}$  : traffic demand matrix.
- $C_{ij}$  : capacity of each link  $(i, j)$ .
- $w(i, j)$  : weight function of link  $(i, j)$  as an integer value, which is set to 1 for each  $w(i, j)$  in this investigation, but could be generalized, if necessary.

Given the above constants, we need to determine the following variables.

- $L_{max}$  : the maximum link load ( $\geq 0$ ).
- $X_{ij}^{sd}$  : the binary integer variable for flow assignment of traffic demand from source  $s$  to destination  $d$  on the link  $(i, j)$ .
- $Y_{ij}^d$  : the binary integer variable for aggregated flow assignment of traffic demand to destination  $d$  on the link  $(i, j)$ , which means the next-hop of destination  $d$  at node  $i$  is node  $j$ .



The final objective of the problem is to minimize the maximum link load,  $L_{max}$ , while all paths are the shortest ones and the destination-based forwarding constraint is preserved at each node. The problem formulation of the first step is as follows.

$$\text{Minimize} \quad \sum_{i,j,s,d} X_{ij}^{sd} \cdot w(i,j) \quad (1)$$

$$\sum_{j:(i,j) \in E} X_{ij}^{sd} - \sum_{j:(j,i) \in E} X_{ji}^{sd} = \begin{cases} 1, i = s, \forall i, s, d \\ 0, i \neq s, d, \forall i, s, d \\ -1, i = d, \forall i, s, d \end{cases} \quad (2)$$

$$\sum_{s,d} \Lambda_{sd} \cdot X_{ij}^{sd} \leq L_{max} \cdot C_{ij}, \forall i, j \quad (3)$$

$$X_{ij}^{sd} \leq Y_{ij}^d, \forall i, j, s, d \quad (4)$$

$$\sum_j Y_{ij}^d \leq 1, \forall i, d \quad (5)$$

$$\text{Int} \quad X_{ij}^{sd}, Y_{ij}^d \in \{0, 1\} \quad (6)$$

The objective of the first-step in the formulation, i.e., Eqn. (1), is to minimize the sum of the weights on links, which guarantees that all the traffic are assigned to the shortest path. Constraint (2) represents the flow-conservation rules for source, intermediate nodes, and destination. Constraint (3) explains that traffic assignment on each link will be constrained by the link capacity multiplied by  $L_{max}$  which indicates the maximum link load. Constraints (4) and (5) state that, if the traffic destined to  $d$  uses next-hop  $j$  at node  $i$  ( $Y_{ij}^d$ ), then all the traffic towards the destination  $d$  at node  $i$  should have the same next-hop  $j$  regardless of their sources (using the destination-based forwarding rule).

After finding the sum of weights of the shortest paths in the first step, which will be given as a constant value denoted by  $H_{total}$  to the second-step formulation, we turn to the second step in order to minimize the maximum link load,  $L_{max}$ , by using the following objective (7) and adding the constraint (8) to the first-step formulation.

$$\text{Minimize} \quad L_{max} \quad (7)$$

$$\sum_{i,j,s,d} X_{ij}^{sd} \cdot w(i,j) \leq H_{total} \quad (8)$$

The optimization objective in Eqn. (7) is to minimize the maximum link load,  $L_{max}$ . Constraint (8) states that all the paths for traffic assignment will be the shortest paths.

As the above problem formulation contains integer variables, its computation complexity is not scalable as the network grows. The problem without the destination forwarding constraint has been shown to be *NP-hard* [11]. Even though the number of equal-cost multiple paths between source and destination,  $|P_{sd}|$ , is known, the size of the search space is exponential ( $\prod_{i=1}^{|N|} \prod_{j=1, j \neq i}^{|N|} |P_{ij}|$ ). Therefore, we propose a simple heuristic algorithm for solving large instances of this problem in the next section.

#### 4 Heuristic Algorithm for TE-Aware Shortest-Path Routing

The proposed algorithm is based on the information of all the available equal-cost multiple shortest paths. Given the shortest-path information, we try to find a set of shortest paths which minimizes the maximum link load while the destination-based forwarding rule is satisfied at each node.

[Step 1 ] Calculate ECMPs for all  $(s, d)$  pairs.

[Step 2 ] Initialize a random set of single shortest paths satisfying the destination-based forwarding rule.

[Step 3 ] Find the set of flows,  $\{F_i\}$ , over the most congested link,  $l_c$ , with the maximum link load,  $L_{max}$ .

[Step 4 ] For each candidate flow,  $F_i$ , select an alternate path, and determine “induced flows” which also have to change their paths to meet the destination-based forwarding rule.

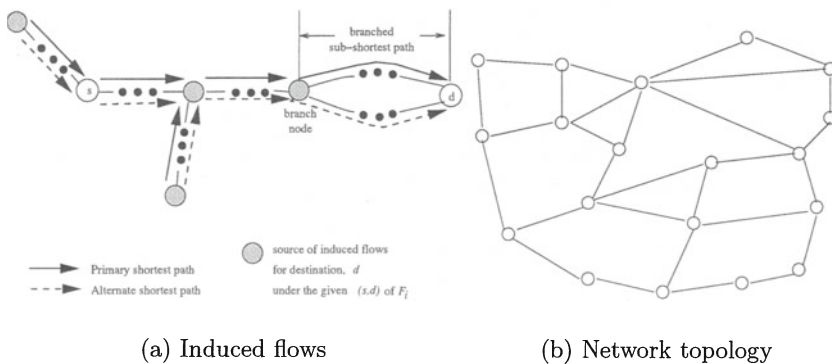
[Step 5 ] If swapping paths of  $F_i$  and induced flows determined in [Step 4] reduces  $L_{max}$ , exchange the current shortest path with the alternate one.

[Step 6 ] Mark  $F_i$  as examined.

[Step 7 ] If the new congested link is different from  $l_c$ , go to [Step 3].

[Step 8 ] If flows to be examined remain in  $\{F_i\}$ , go to [Step 4]; otherwise, terminate.

In order to satisfy the destination-based forwarding rule, [Step 4] enumerates the “induced flows” affected by  $F_i$  instead of all flows to be examined for fast computation. The induced flows for  $F_i$  are ones that share the “branched-sub-shortest path” of  $F_i$  as shown in Fig. 2(a). Thus, the shortest paths of the induced flows will also be changed when the primary shortest path for the flow,  $F_i$ , is swapped with the alternate one.



**Fig. 2.** (a) “Induced flows” affected by  $F_i$  flow from node  $s$  to node  $d$  and (b) sample network topology.

In [Step 4], various ways of deciding a candidate flow,  $F_i$ , to be swapped are possible such as random, maximum flow-preferred, or minimum flow-preferred selections. In addition, when the alternate shortest path for the candidate flow is examined, different selection criteria such as random, maximum residual capacity, and minimum residual capacity may be possible. However, these variations do not yield much difference on the performance, because any selection method may generate the induced flows to be swapped so that the difference between selection methods will be compensated. Therefore, randomly selecting a candidate flow and an alternate path is used for the experiments.

The complexity of the heuristic algorithm is  $O(|N|^3 + |E||N|(|N| - 1)^2)$ . The computation complexity of the initialization step for finding all the equal-cost shortest paths using linear *Dijkstra* algorithm is  $O(|N|^3)$ , and for each flow ( $O(|N|(|N| - 1))$ ) on the congested link ( $O(|E|)$ ), the induced flows for the same destination ( $O(|N| - 1)$ ) will be examined in the worst case.

## 5 Performance Evaluation

### 5.1 Case I: Sample Network

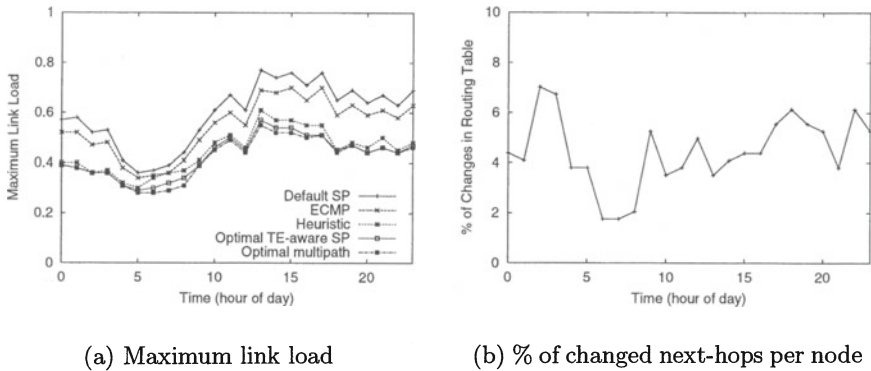
For our simulation experiments, we use a simple network topology (Fig. 2(b)) with 19 nodes and 62 directed links, and we use 24 sample traffic matrices for one day [16] where a traffic model is derived based on the observations on several backbones' link loads. Backbone traffic is the aggregation of several end systems' traffic, and the aggregation process filters out the short-term variations. On the other hand, long-term variations (on a scale of hours) remain and repeat their pattern in one-day periods. Hence, we believe that it is reasonable to use the representative sampled traffic matrices for the simulation of the periodic TE operations. To obtain a realistic model, 24 representative link traffic rates are sampled from real networks over a 24-hour period. These traffic rates are used to generate 24 traffic samples between any node pair in the network. We use a scaling factor in the range [0.2, 1.2] to create discrepancies in the volume of traffic flowing between different node pairs (values on the main diagonal remain zero). It is assumed that the capacity of each link is 310 Mbps (2 OC-3's).

**Dynamic TE-Aware Shortest-Path Routing.** In the dynamic TE-aware shortest-path routing algorithm, the routing table may be changed if the better shortest path is known. When the TE process is performed every one hour for one-day traffic matrix, we compare the heuristic for TE-aware shortest-path routing with default shortest-path routing, ECMP<sup>2</sup>, optimal TE-aware shortest-path routing, and optimal multipath routing for each sampled traffic matrix. Figure 3(a) shows that the proposed heuristic reduces the maximum link load by 25% when compared with default shortest-path routing, whereas the maximum link load of ECMP is less than that of shortest-path routing by 8.5%. When compared with the results of optimal TE-aware shortest-path routing (solved by

<sup>2</sup> The number of equal-cost shortest paths is set to 2, because its result was found to be the best among all the possible cases in this experiment.



CPLEX<sup>3</sup> [17]), the maximum link load of the heuristic has increased only by 4.9% on average. It is interesting to note that, even when compared with the results of optimal multipath routing which are obtained by LP after the integer variables are relaxed to be real and the destination-based forwarding rule is removed, the maximum link load of the heuristic increases only by 6.6%.



**Fig. 3.** Maximum link load and the percentage of changed next-hops per node in the sample network for one-day traffic demand matrix.

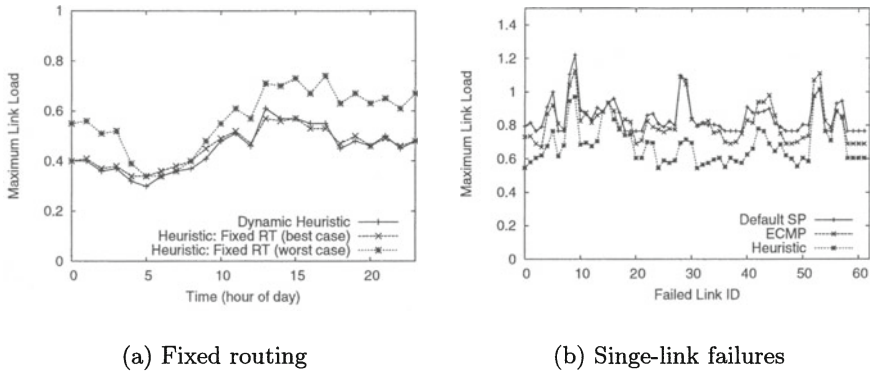
If the TE-aware shortest paths calculated by the TE server are to be utilized by IP routers, the next-hops for the destination may be changed, if they are different from those of the default shortest paths. Swapping the next-hops for the destination can be completed by configuring the new next-hop and deleting the old next-hop sequentially, while ECMP with two paths is enabled temporarily to prevent traffic disruption. Therefore, during the transition period, packets are distributed to two paths due to ECMP. The number of changed next-hops is considered as the configuration overhead of using the TE-aware shortest paths.

In Fig. 3(b), the percentage of changed next-hops over all the possible routing tables at each node ( $= \frac{\text{Next-hop Changes}}{N*(N-1)} * 100$ ) is shown. In this network topology which provides 1.56 equal-cost multiple paths on average, the percentage of next-hop changes over all the possible routing tables at each node is about 4.5%, and nine nodes experience a routing-table change. On average, two changes of next-hops per node are enough to reduce the maximum link load in this experiment.

**Static TE-Aware Shortest-Path Routing.** In order to avoid the configuration overhead and traffic disruption for every TE period, the static TE-aware shortest path routing algorithm uses the pre-configured routing table. We ran the simulation after fixing the routing table for the representative traffic demand matrix among 24 sample traffic matrices for one day [16], and the corresponding

<sup>3</sup> CPLEX 6.5.2 was run on a Linux 2.4.2-2 machine using a Pentium PC running at 1 GHz with 512 MB RAM.

results are shown in Fig. 4(a). In the best case (the first sample among 24 traffic demand tables), the maximum link load does not increase much on average (by 1.7%), compared with dynamic TE-aware shortest-path routing. However, in the worst case (the 7th traffic demand table), it increases on average by 26.5%, which is greater than that of ECMP, but still less than that of default shortest-path routing. Therefore, when the traffic demand matrix is known, the fixed routing table for the best performance case may be used to reduce the maximum link load without any configuration overhead.



**Fig. 4.** (a) Maximum link load of the heuristic with fixed routing table for one-day traffic demand matrix and (b) maximum link load under single-link failures.

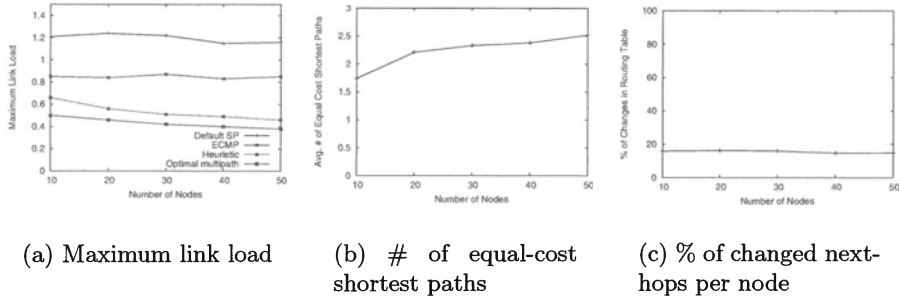
**Single-Link Failures.** Noting that single-link failures are the dominant form of failures in the network, the effects on the performance of every link failure are shown in Fig. 4(b). For a single-link failure, the maximum link load over 24-hour traffic demand is plotted. From the simulation results, it is observed that this network needs to be over-provisioned with at least 1.3 times of the current capacity to protect traffic against the worst case failure, namely link 9 failure, if default shortest path routing is used. In some cases, the maximum link load of ECMP is greater than that of default shortest-path routing because ECMP splits traffic equally without considering the congested link globally. Under the critical link 9 failure, the maximum link load of the heuristic is still less than that of default shortest-path routing by 20%.

## 5.2 Case II: Random Networks

Random network topologies and traffic demand matrices are generated to compare default single shortest-path routing, ECMP, optimal multipath routing, and our proposed heuristic<sup>4</sup>. The network topology consist of nodes with the size of

<sup>4</sup> Optimal TE-aware shortest-path routing was omitted because a solution to the ILP for a large network cannot be achieved within reasonable time.

{10, 20, 30, 40, 50} and links with the size of {30, 60, 90, 120, 150}, assuming that a node has three interfaces on average. The nodes are randomly distributed on the Euclidean plane, and the nodes are randomly connected until the connected graph has the given number of links. The capacity of each link is uniformly distributed over the range of [155, 310] Mbps, and the traffic demand is uniformly distributed over [0, 10] Mbps. We carried out 30 runs of experiments for each random graph with the given nodes and links.



**Fig. 5.** Maximum link load, average number of equal-cost shortest paths, and the percentage of changed next-hops per node in random networks.

In Fig. 5(a), it is seen that the proposed heuristic performs better than shortest-path routing and ECMP. When shortest-path routing is used, the maximum link load becomes greater than 1, which means that traffic is lost because of severe congestion. Although the ECMP way of utilizing up to two paths reduces the maximum link load by 30%, the maximum link load is still high (about 0.85). The proposed heuristic maintains the maximum link load at the level of 0.54. Even when compared with optimal multipath routing with the capability of splitting traffic unequally whose result was obtained by CPLEX, its marginal difference is about 20% on average. The performance of our proposed heuristic gets better as the number of equal-cost multiple paths becomes large (Fig. 5(b)).

The difference between next-hops of TE-aware shortest-path routing and those of default shortest-path routing is shown in Fig. 5(c) which describes the percentage of the next-hop changes over all the possible routing tables at each node. Overall, less than 20% of the routing-table entries have been changed in this experiment.

## 6 Conclusion

In this paper, we presented the mathematical problem formulation for TE-aware shortest-path routing as an ILP. We proposed a heuristic algorithm which can minimize the maximum link load without any serious modification overhead to the Internet routing protocols. TE-aware shortest paths found by the proposed

heuristic (from among several available shortest paths between any node pair) were shown to be useful for the global optimization goal of traffic engineering. Traffic engineering can be performed at IP routers by changing the next-hop for the destination with the optimally calculated TE-aware shortest paths. Using our simulation experiments, it was shown that the maximum link load can be minimized when IP routers can change the next-hops for the TE-aware shortest path. Even though the next-hops for the TE-aware shortest paths are fixed for the known traffic demand matrix and the network topology, it was also observed that the maximum link load can be lower than those of default shortest-path routing and ECMP. For the case of single-link failures, TE-aware shortest-path routing was found to perform better than default shortest-path routing and ECMP.

## References

1. D. Awduche, A. Chiu, A. Elwalid and X. Xiao, "Overview and Principles of Internet Traffic Engineering," IETF RFC3272, 2002.
2. D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus, "Requirements for Traffic Engineering Over MPLS," IETF RFC2702, 1999.
3. D. Bertsekas, and R. Gallager, *Data Networks*, Prentice Hall, 1992.
4. S. Vutukury and J. J. Garcia-Luna-Aceves, "A Traffic Engineering Approach based on Minimum Delay Routing," Proc., IEEE ICCCN, 2000.
5. J. Moy, "OSPF Version 2," IETF RFC2328, 1998.
6. R. Callon, "Use of OSI IS-IS for Routing in TCP/IP and Dual Environments," IETF RFC1195, 1990.
7. Optimized Multipath, <http://www.fictitious.org/omp>
8. E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture," IETF RFC3031, 2001.
9. D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels," IETF RFC 3209, 2001.
10. A. Sridharan, R. Guerin and C. Diot, "Achieving Near-Optimal Traffic Engineering Solutions for Current OSPF/IS-IS Networks," Proc., IEEE INFOCOM, 2003.
11. Y. Wang, and Z. Wang, "Explicit Routing Algorithms for Internet Traffic Engineering," Proc., IEEE ICCCN, 1999.
12. Z. Wang, Y. Wang, and L. Zhang, "Internet Traffic Engineering without Full Mesh Overlaying," Proc., IEEE INFOCOM, 2001.
13. M. Kodialam and T. V. Lakshman, "Minimum Interference Routing with Applications to MPLS Traffic Engineering," Proc., IEEE INFOCOM, 2000.
14. E. Dinan, D. O. Awduche, and B. Jabbari, "Analytical Framework for Dynamic Traffic Partitioning in MPLS Networks," Proc., IEEE ICC, 2000.
15. B. Fortz and M. Thorup, "Internet Traffic Engineering by Optimizing OSPF Weights," Proc., IEEE INFOCOM, 2000.
16. A. Gencata and B. Mukherjee, "Virtual-Topology Adaptation for WDM Mesh Networks Under Dynamic Traffic," Proc., IEEE INFOCOM, 2002.
17. CPLEX, <http://www.ilog.com>