

# Load Balancing for Multiple Traffic Matrices Using SDN Hybrid Routing

Junjie Zhang\*, Kang Xi\*, Min Luo<sup>†</sup>, H. Jonathan Chao\*

New York University\* Huawei Technologies<sup>†</sup>  
 junjie.zhang@nyu.edu\*, kang.xi@nyu.edu\*, min.ch.luo@huawei.com<sup>†</sup>, chao@nyu.edu\*

**Abstract**—Classical traffic engineering (TE) methods calculate the optimal routing based on a single traffic matrix. However, they are unable to handle unexpected traffic changes. Thus, it is of interest to find a good routing configuration to accommodate multiple possible traffic scenarios. There are two major approaches to achieve load balancing for multiple traffic matrices: destination-based routing and explicit routing. It has been shown that explicit routing performs better than destination-based routing for multiple traffic matrices. However, explicit routing has high complexity and requires large Ternary Content Addressable Memory (TCAM) in the routers. Thus, it is power hungry and unscalable.

This paper presents an approach called hybrid routing to achieve load balancing for multiple traffic matrices with low complexity and good scalability. Our basic idea is to complement destination-based routing with a small number of explicit routing forwarding entries to take advantage of both two routing approaches. Hybrid routing greatly reduces the number of forwarding entries compared with pure explicit routing. This has great value for practice in that the scheme requires very small TCAM to implement. Hybrid routing is very suitable for implementation using SDN. A heuristic algorithm is developed to obtain the near-optimal hybrid routing configuration. Extensive evaluation demonstrates the effectiveness of hybrid routing. The results show that hybrid routing achieves near-optimal load balancing compared with pure explicit routing. In particular, hybrid routing saves at least 84.6% TCAM resources in all practical networks used in our evaluation.

**Index Terms**—Multiple Traffic Matrices, Hybrid Routing, Load Balancing, TCAM Saving .

## I. INTRODUCTION

Intra-domain traffic engineering (TE) has been widely adopted in today's Internet Service Provider (ISP) networks. TE configures the parameters of the routing system to control how traffic is routed across the network to optimize network performance and utilize network resources efficiently. Given the highly competitive nature of Internet ISP market and the high cost of network resource [1], TE has become an indispensable tool for ISPs. Classical TE methods calculate the optimal routing off-line based on a single predicted or previously measured traffic matrix [2]–[5]. They perform well when the real traffic is relative stable and similar to the predicted traffic matrix. However, such schemes are unable to handle unexpected traffic changes. The optimized routing may become no longer suitable due to traffic fluctuations. As a result, it may cause network links and routers to be overloaded, and thus reduces network throughput. Thus, it is critical to find a good routing configuration to accommodate changing traffic.

A feasible approach to model changing traffic is to represent the traffic with multiple traffic matrices. Considering multiple traffic matrices has great importance in practice, since multiple traffic matrices reflect potential traffic fluctuation and cover worst-case load. There are two major approaches to achieve load balancing for multiple traffic matrices: destination-based routing and explicit routing [6]–[8]. Destination-based routing is the typical, most common type of routing, where routers make forwarding decisions based on the destination addresses specified in packet headers. Thus, each router forwards packets destined to the same destination in the same way regardless of the source addresses. Explicit routing offers fine-grained traffic distribution control. It supports flexible routing for each individual flow (a flow can be flexibly defined, e.g., source and destination address, 5-tuple header fields). It has been shown that explicit routing performs better than destination-based routing when conducting routing optimization for multiple traffic matrices [7]. Figure 1 shows a simple example to explain the above. Given a simple topology with two traffic matrices (see Fig. 1(a)), the worst case maximum link load achieved by destination-based routing is 3 units (see Fig. 1(b)). In contrast, explicit routing distributes traffic flexibly, and thus the worst case maximum link load can be decreased to 2 units using the routing configuration shown in Fig. 1(c). However, with explicit routing, routers have to maintain at worst  $O(N^2)$  entries for a network with  $N$  hosts. In Software-Defined Networking (SDN) networks, explicit routing<sup>1</sup> relies on TCAM to maintain line rate lookup. However, due to high cost-to-density ratio (US\$350 for a 1M-bit chip) and high power consumption (about 15 Watt / 1 Mbit) of TCAM [9], explicit routing may suffer from scalability issues and consume too much power. Thus, it is of interest to design a new routing approach that is able to achieve good performance for multiple traffic matrices with low TCAM resource requirement.

In this paper, we propose a low-complexity approach called hybrid routing. Our basic idea is to complement destination-based routing with a small number of explicit routing forwarding entries to achieve load balancing for multiple traffic matrices. With a hybrid routing configuration, traffic belonging to the majority of node pairs is routed by the destination-based routing and traffic belonging to a few selected key node

<sup>1</sup>Explicit routing can also be supported by Multiprotocol Label Switching (MPLS) without TCAM issues, but it is beyond the scope of this paper.

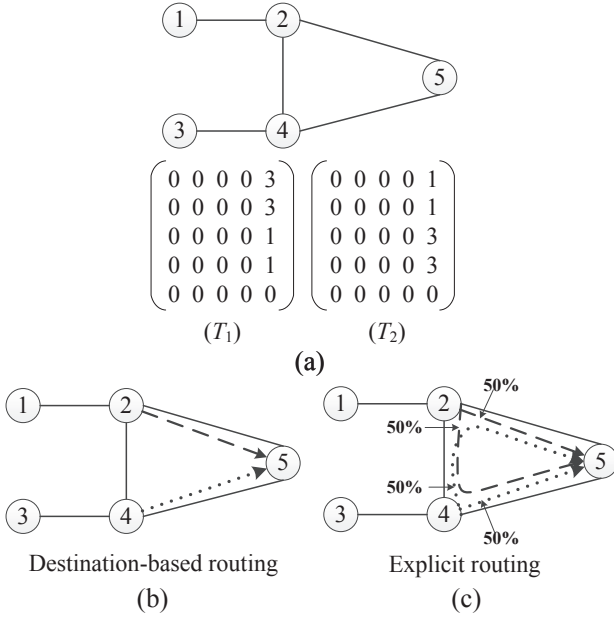


Fig. 1. Explicit routing vs Destination-based routing

pairs is routed by the complementary explicit routing. Hybrid routing stores the destination-based routing forwarding entries in Static Random-Access Memory (SRAM) and thus saves significant TCAM resources, since the majority of forwarding entries of a hybrid routing are destination-based routing forwarding entries (see Section V-B2). The contribution of this paper is summarized below.

- 1) We propose a low-complexity hybrid routing framework that achieves good load balancing for multiple traffic matrices and requires low TCAM resources.
- 2) Given the high complexity of hybrid routing optimization for multiple traffic matrices, we design a method to obtain the near-optimal hybrid routing solution.

Extensive evaluation demonstrates the effectiveness of hybrid routing. We analyze the performance gain of hybrid routing and show that hybrid routing achieves near-optimal load balancing compared with optimal pure explicit routing. In particular, it applies explicit routing only for a very small fraction of ingress/egress node pairs (e.g., 15%). Evaluation in all practical networks shows that hybrid routing saves at least 84.6% TCAM resources compared with pure explicit routing.

The remainder of this paper is organized as follows. Section II presents the system description. Section III presents the problem description. Section IV describes the hybrid routing algorithm design. Section V evaluates the performance of hybrid routing. Section VI concludes the paper.

## II. SYSTEM DESCRIPTION

The hybrid routing algorithm is implemented in a central controller. The controller translates the hybrid routing solution (the destination-based routing configuration and the set of complementary explicit routing configuration) to forwarding

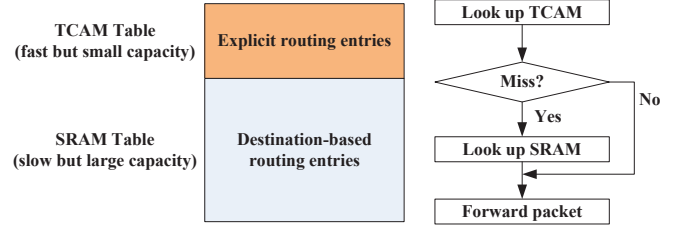


Fig. 2. Possible implementation of hybrid forwarding table

entries and installs them in the routers. In case of multiple matches, explicit routing forwarding entries take strict precedence over destination based routing forwarding entries.

### A. Hybrid Forwarding Table

In our implementation, a router's forwarding table composes of a small TCAM table and a large SRAM table. The TCAM table is allocated to store explicit routing forwarding entries that can process "selected key node pairs". The SRAM table is used to store the destination-based routing forwarding entries. This saves significant TCAM resources, since the majority of forwarding entries for hybrid routing is destination-based routing entries (see Section V-B2).

Figure 2 shows a possible implementation of forwarding table. When doing a packet matching, a router first looks up entries in the TCAM table. Once an entry is matched, the action specified by the corresponding forwarding entry would be applied directly to the packet (e.g., forward to a port). If no entry is matched in the TCAM table, the router keeps searching entries in the SRAM table and operates corresponding actions to the packet.

### B. Traffic Splitting

Hybrid routing may use multiple paths in both destination-based routing and explicit routing. A router may need to split the incoming traffic to multiple output ports, based on the pre-computed split ratios. In practice, we prefer to let a TCP (or UDP) flow follow a single path to avoid packet mis-order. An approximate to the traffic splitting is to hash the 5-tuple packet header and then allocate flows to one of the output port based on the hash results and the ratios (refer to RFC 2992 [10] and the standard hashing technique [11]). In backbone networks, each link (e.g. 10G) carries a large number of TCP/UDP flows and each individual flow is relatively small compared to the link capacity. Thus, the above method performs closely to the theoretical results of hybrid routing [11].

## III. PROBLEM DESCRIPTION

The objective of hybrid routing is to find an optimal routing combination of destination-based routing and explicit routing to accommodate multiple traffic matrices. Thus, there are three challenging issues. First, *how to select the small set of key node pairs to apply explicit routing*. Second, *how to obtain the explicit routing configuration for the selected key node pairs*. Third, *how to obtain the destination-based routing*

configuration for the remaining node pairs. Unfortunately, the optimal solutions of the above problems cannot be obtained in polynomial time, because the proposed hybrid routing optimization problem is NP-complete. (destination-based routing optimization problem with multiple traffic matrices is NP-complete [7], which is a sub-problem of the proposed hybrid routing optimization problem.)

Given the complexity of hybrid routing optimization, we design a low-complexity method to obtain the near-optimal solution in four steps. First of all, we obtain the *destination-based routing configuration*. Given the high complexity of destination-based routing problem with multiple traffic matrices, we calculate the destination-based routing according to a basic traffic matrix derived from all given traffic matrices (the multiple given traffic matrices for hybrid routing can be selected from historical measured traffic matrices [12], or simply include all previously seen traffic matrices). With this underlying routing, we are able to obtain the preliminary routing and congestion information. (see Section IV-C); Second step, we perform traffic categorization to select a *small set of key node pairs* based on the previous calculated destination-based routing and the given traffic matrices. Rebalancing the traffic belonging to these key node pairs using explicit routing would bring great potential benefits (see Section IV-D1); Third step, we formulate an optimization model to calculate the *explicit routing configuration for the selected key node pairs*, such that the combination of the explicit routing and the destination-based routing achieves near-optimal load balancing for multiple traffic matrices. (see Section IV-D2); Lastly, we configure routers' forwarding tables in terms of the hybrid routing solution.

#### IV. HYBRID ROUTING ALGORITHM DESIGN

##### A. Notations

$G(V, E)$	network with nodes $V$ and directed edges $E$ ( $ V  = N,  E  = M$ )
$T_t$	one of the given traffic matrices $\{T\}$ ( $t = 1, 2, \dots, L,  \{T\}  = L$ )
$c_{i,j}$	the capacity of link $\langle i, j \rangle$ ( $\langle i, j \rangle \in E$ )
$h_i^{s,d}$	the traffic demand from source $s$ to destination $d$ in traffic matrix $T_t$ ( $s, d \in V, s \neq d, t = 1, 2, \dots, L$ )
$\alpha_{i,j}^{s,d}$	the percentage of traffic demand from source $s$ to destination $d$ routed on link $\langle i, j \rangle$ ( $s, d \in V, s \neq d, \langle i, j \rangle \in E$ )
$\beta_{i,j}^d$	the split ratio at node $i$ to node $j$ for the traffic destined to node $d$ ( $d \in V, \langle i, j \rangle \in E$ )

$\{\alpha_{i,j}^{s,d}\}$  are the explicit routing ratios which specify the routes along which packets are forwarded for each source/destination node pair.  $\{\beta_{i,j}^d\}$  are the destination-based routing fractions which denote the packet forwarding routes for each destination at each router. Note that  $\{\beta_{i,j}^d\}$  are different from  $\{\alpha_{i,j}^{s,d}\}$  in that they are irrelevant to the source addresses. We use the combination of  $\{\alpha_{i,j}^{s,d}\}$  and  $\{\beta_{i,j}^d\}$  to represent a hybrid routing solution.

##### B. Definitions

The goal of the hybrid routing algorithm is to find a good hybrid routing configuration that achieves load balancing for multiple traffic matrices. A common TE objective [6] is to minimize

$$\max_{\langle i,j \rangle \in E, t=1,2,\dots,L} (l_{i,j}^t / c_{i,j}), \quad (1)$$

where  $l_{i,j}$  is the traffic load on link  $\langle i, j \rangle$  and  $c_{i,j}$  is its link capacity. (1) refer to the worst case maximum link utilization for multiple traffic matrices. Minimizing (1) is the goal of all the candidate solutions in the following section.

##### C. Calculating Destination-Based Routing

Since destination-based routing optimization problem with multiple traffic matrices is NP-complete [7], we calculate the destination-based routing according to a derived basic traffic matrix rather than the multiple given traffic matrices. There are several options to derive the basic traffic matrix from given traffic matrices to represent the worst-case traffic load.

(1) Choosing the maximum demand volume traffic matrix as the basic traffic matrix, then each demand in  $T_{basic}$  is given by

$$h_{basic}^{s,d} = \max_{t=1,2,\dots,L} h_t^{s,d}. \quad (2)$$

(2) Choosing the average demand volume traffic matrix as the basic traffic matrix, then each demand in  $T_{basic}$  is given by

$$h_{basic}^{s,d} = \sum_{t=1,2,\dots,L} (h_t^{s,d} / L). \quad (3)$$

(3) Choosing the minimum demand volume traffic matrix as the basic traffic matrix, then each demand in  $T_{basic}$  is given by

$$h_{basic}^{s,d} = \min_{t=1,2,\dots,L} h_t^{s,d}. \quad (4)$$

Next, we obtain the destination-based routing configuration in terms of the basic traffic matrix. Let  $y_{i,j}^d$  denote the traffic destined to  $d$  routed on link  $\langle i, j \rangle$ , let  $\bar{U}$  denote the maximum link utilization ( $\max_{\langle i,j \rangle \in E} (l_{i,j}^{basic} / c_{i,j})$ ) in the network, where  $l_{i,j}^{basic}$  is the traffic load on link  $\langle i, j \rangle$  when the basic traffic matrix  $T_{basic}$  is routed. The destination-based routing problem can be described as follows. Given a network  $G(V, E)$  with a traffic matrix  $T_{basic}$ , find the best destination-based traffic allocation  $\{y_{i,j}^d\}$ , so that  $\bar{U}$  is minimized. The destination-based routing problem can be formulated as an optimization problem as follows.

$$\text{minimize } \bar{U} + \varepsilon \cdot \sum_{\langle i,j \rangle \in E} \sum_{d \in V} y_{i,j}^d \quad (5a)$$

subject to

$$\sum_{d \in V} y_{i,j}^d \leq c_{i,j} \cdot \bar{U} \quad (5b)$$

$$\sum_{k:(k,i) \in E} y_{k,i}^d - \sum_{k:(i,k) \in E} y_{i,k}^d = \begin{cases} \sum_{s \in V, s \neq d} h_{basic}^{s,d} & \text{if } i = d \\ -h_{basic}^{i,d} & \text{otherwise} \end{cases} \quad (5c)$$

$$y_{i,j}^d \geq 0 \quad (5d)$$

For all the above, we have  $\forall s, d, i, j \in V, s \neq d, \forall \langle i, j \rangle \in E$ .

$\varepsilon \cdot \sum_{\langle i, j \rangle \in E} \sum_{d \in V} y_{i,j}^d$  in (5a) ensures that the routing is loop-free ( $\varepsilon$  is a sufficiently small positive number which ensures that the minimization of  $\bar{U}$  takes higher priority.). (5b) is the link capacity utilization constraint. (5c) is the flow conservation constraint. There are  $M$  constraints in (5b),  $N^2$  constraints in (5c) and  $MN$  constraints in (5d). The number of variables is  $MN + 1 = O(MN)$ .

Using linear programming (LP) solvers (like CPLEX [13]), we can obtain the optimal destination-based traffic allocation  $\{y_{i,j}^d\}$ . The corresponding destination-based routing fractions  $\{\beta_{i,j}^d\}$  can be derived from  $\{y_{i,j}^d\}$  by the follow equations.

$$\beta_{i,j}^d = \begin{cases} \frac{y_{i,j}^d}{\sum_{k:\langle i,k \rangle \in E} y_{i,k}^d} & \text{if } \sum_{k:\langle i,k \rangle \in E} y_{i,k}^d \neq 0 \\ 0 & \text{if } \sum_{k:\langle i,k \rangle \in E} y_{i,k}^d = 0 \end{cases}, \quad \forall d \in V, \forall \langle i, j \rangle \in E. \quad (6)$$

According to the fractions, we can compute the destination-based routing forwarding entries for the routers.  $\beta_{i,j}^d$  indicates the available next hop and the corresponding hash weight.

#### D. Hybrid Routing Optimization

In this section, we describe how to select a small set of key node pairs and rebalance the traffic belonging to these selected key node pairs using explicit routing, such that the combination of the explicit routing and the destination-based routing (calculated in the previous section) achieves near-optimal load balancing for multiple traffic matrices.

1) **Selecting Key Node Pairs:** In this paragraph, we describe how to select a set of key node pairs to apply explicit routing. Note that the traffic belonging to the selected key node pairs would be routed by explicit routing and the rest traffic would be routed by the destination-based routing calculated in the Section IV-C. Thus, the main purpose of key node pairs selection is to choose a set of node pairs that will maximize the potential benefit of hybrid routing optimization. Exhaustive search among all the node pairs to identify the optimal selection incurs considerable complexity and thus is impractical. Instead, we recursively select the most suitable node pair on the most congested links. Let  $D$  denote the entire set of node pairs,  $D_e$  denote the set of node pairs using explicit routing and  $D_d$  denote the set of node pairs using destination-based routing. The procedure for key node pairs selection is shown as below.

##### Procedure: Key Node Pairs Selection

- 1) Assume we want to choose  $K$  node pairs for explicit routing. Initially,  $D_e = \emptyset$  and  $D_d = D$ .  $\eta = 0$ .
- 2) For each given traffic matrix, route the traffic demands belonging to the node pairs  $\langle s, d \rangle \in D_d$  by the destination-based routing. Denote the worst case maximum link utilization as  $U_{dest.}^\eta$ .
- 3) Choose a link  $\langle i, j \rangle$  with the worst case link utilization  $U_{dest.}^\eta$ .

- 4) For each node pair  $\langle s, d \rangle$  contributes traffic on the chosen congested link  $\langle i, j \rangle$ , temporarily exclude this node pair  $\langle s, d \rangle$  and calculate the corresponding  $U_{dest.}^{\eta+1}$ , choose the node pair  $\langle s, d \rangle$  with minimum  $U_{dest.}^{\eta+1}$  as a key node pair.
- 5) Add the chosen node pair  $\langle s, d \rangle$  into  $D_e$ , and exclude it from  $D_d$ .  $\eta = \eta + 1$ ;
- 6) If  $\eta = K$ , exit. Otherwise go back to step 2.

$U_{dest.}^0$  stands for the worst case maximum link utilization when the traffic demands of the entire set of node pairs are routed by the destination-based routing.  $U_{dest.}^K$  stands for the worst case maximum link utilization when the traffic demands of the remaining node pairs  $\langle s, d \rangle \in D_d$  are routed by the destination-based routing ( $K$  key node pairs have been selected for explicit routing and added into  $D_e$ ). Let  $U_{hybrid}$  denote the worst case maximum link utilization when the traffic demands of the node pairs  $\langle s, d \rangle \in D_d$  are routed by the destination-based routing and the traffic demands of the selected key node pairs  $\langle s, d \rangle \in D_e$  are routed by explicit routing. The range of optimization using explicit routing is

$$U_{dest.}^K \leq U_{hybrid} \leq U_{dest.}^0. \quad (7)$$

What we do in selection procedure is to minimize  $U_{dest.}^K$ , such that the potential effect of explicit routing optimization can be maximized. Obviously, larger  $K$  would introduce more potential performance gain. When  $K = N(N-1)$  ( $N$  is the number of total nodes), the routing becomes pure explicit routing.

In our test using a variety of practical networks, we find that setting  $K = 15\% * N(N-1)$  (15% of the total node pairs) is good enough for hybrid routing to achieve near-optimal performance compared with optimal pure explicit routing (see Section V).

2) **Calculating Explicit Routing For Selected Key Node Pairs:** We conduct explicit routing optimization only for the selected key node pairs  $\langle s, d \rangle \in D_e$ . The remaining node pairs  $\langle s, d \rangle \in D_d$  that routed by the destination-based routing would contribute background load on each link  $\bar{l}_{i,j}^t$  (Since we are dealing with multiple traffic matrices,  $\bar{l}_{i,j}^t$  may have different volume in each individual traffic matrix). This load would never be affected by the explicit routing.

The problem of explicit routing for selected key node pairs can be described as follow. Given a network  $G(V, E)$  with multiple given traffic matrices  $T_t$  ( $t = 1, 2, \dots, L$ ) and the background link load  $\{\bar{l}_{i,j}^t\}$ , our objective is to obtain the optimal explicit routing ratios  $\{\alpha_{i,j}^{s,d}\}$  for the node pairs  $\langle s, d \rangle \in D_e$ , so that  $U_{hybrid}$  is minimized.

To search all possible under-utilized paths for selected key node pairs, we formulate the explicit routing problem as an optimization as follows.

$$\text{minimize } U_{hybrid} + \varepsilon \cdot \sum_{\langle i, j \rangle \in E} \sum_{s, d: \langle s, d \rangle \in D_e} \alpha_{i,j}^{s,d} \quad (8a)$$

subject to

$$l_{i,j}^t = \sum_{s, d: \langle s, d \rangle \in D_e} \alpha_{i,j}^{s,d} \cdot h_t^{s,d} + \bar{l}_{i,j}^t \quad (8b)$$

$$l_{i,j}^t \leq c_{i,j} \cdot U_{hybrid} \quad (8c)$$

$$\sum_{k:(k,i) \in E} \alpha_{k,i}^{s,d} - \sum_{k:(i,k) \in E} \alpha_{i,k}^{s,d} = \begin{cases} -1 & \text{if } i = s \\ 1 & \text{if } i = d \\ 0 & \text{otherwise} \end{cases} \quad (8d)$$

$$0 \leq \alpha_{i,j}^{s,d} \leq 1 \quad (8e)$$

For all the above, we have  $\forall s, d: \langle s, d \rangle \in D_e, \forall i, j \in V, \forall \langle i, j \rangle \in E, t = 1, 2, \dots, L$ .

$\varepsilon \cdot \sum_{\langle i,j \rangle \in E} \sum_{s,d: \langle s,d \rangle \in D_e} \alpha_{i,j}^{s,d}$  in (8a) is needed because otherwise the optimal solution may include unnecessarily long paths as long as they avoid the most congested link. (8b) indicates the traffic load on link  $\langle i, j \rangle$  contributed by the traffic demands routed by the complementary explicit routing and the traffic demands routed by the destination-based routing in each traffic matrix  $T_t$ . (8c) is the link capacity utilization constraint. (8d) is the flow conservation constraint for the selected key node pairs.

Using the CPLEX solver, we can obtain the optimal explicit routing ratios for selected key node pairs. According to the ratios, we can compute the explicit routing forwarding entries for the routers.

## V. PERFORMANCE EVALUATION

In this section, we evaluate the performance of hybrid routing.

### A. Evaluation Setup

We use different network topologies in our evaluation, including Abilene and four ISP networks collected by ROCKETFUEL [14]. The sizes of the topologies are listed in Table I. For Abilene, the router-level topology (i.e., link connectivity, weights, capacities) and measured traffic matrices are available at [15]. Abilene's traffic matrices are measured every 5 minutes. We choose total 288 traffic matrices on June 24th 2004 as the set of given traffic matrices for Abilene network. For each ISP network provided by ROCKETFUEL, the link costs are given and the link capacities are defined as the inverse of the link costs (Since link capacities are unavailable from ROCKETFUEL topologies, we deduce them from link costs based on the default link cost setting of Cisco routers, i.e., the link costs are inversely proportional to the link capacities [16]. This approach is commonly adopted in literature, such as [8], [17].). Since measured traffic matrices are unavailable for ISP networks provided by ROCKETFUEL, we generate 50 synthetic traffic matrices for each network using gravity model [18].

Our objective function is to minimize the worst case maximum link utilization for multiple traffic matrices. For comparison, we also evaluate the optimal pure explicit routing for multiple traffic matrices. It is obtained by solving (8a), when  $K = N(N-1)$ . Note that (8a) becomes pure explicit routing problem<sup>2</sup> when  $K = N(N-1)$ .

<sup>2</sup>when  $K = N(N-1)$ ,  $D_e = D$  and  $\forall \langle i, j \rangle = 0$ .

TABLE I  
ISP NETWORKS USED IN EVALUATION

Topology	# of nodes	# of directed links
Abilene	12	30
Exodus 3967	22	74
Abovenet 6461	22	84
EBONE 1755	23	76
Sprintlink 1239	44	166

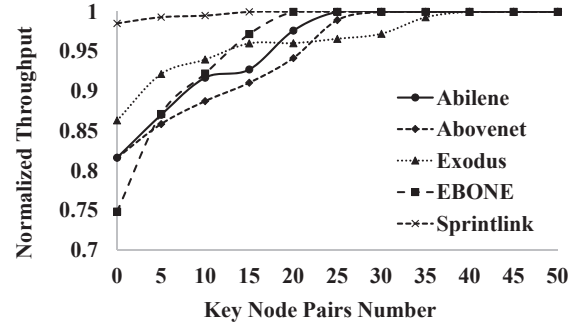


Fig. 3. Performance of hybrid routing with increasing number of key node pairs

For clarity, we use normalized throughput to demonstrate the performance gain of the proposed hybrid routing. The normalized throughput of hybrid routing is defined as  $NT = \frac{U_{optimal\_explicit}}{U_{hybrid}}$ , where  $U_{optimal\_explicit}$  is the worst case maximum link utilization achieved by the optimal pure explicit routing and  $U_{hybrid}$  is the worst case maximum link utilization achieved by the proposed hybrid routing.  $NT = 1$  indicates that the proposed hybrid routing performs as good as the optimal pure explicit routing. A lower ratio indicates that the performance of the proposed hybrid routing is farther away from that of the optimal pure explicit routing.

We conduct all the experiments on a computer with a i7 quad-core 2.0Ghz processor and 8 GB memory. The version of CPLEX we used is 12.5.1. For the largest network we tested (i.e., Sprintlink network), solving (5a) takes about 10 seconds, and solving (8a) takes about 45 seconds when  $K = 15\% * N(N-1)$ . The entire hybrid routing calculation time for Sprintlink network is less than 2 minutes.

### B. Evaluation

In this section, we evaluate hybrid routing framework and demonstrate the effectiveness of hybrid routing.

1) **Performance:** We introduce three options to derive the basic traffic matrix in Section IV-C and then calculate the destination-based routing based on the basic traffic matrix. In our simulations we choose first option (see (2)). Compared to the second and third options, it allows better load balancing because demands of node pairs with high traffic variability throughout the day are represented by the peak traffic.

Figure 3 shows the normalized throughput of the proposed hybrid routing with increasing key node pairs number  $K$ .

TABLE II  
COMPARISON OF FORWARDING ENTRIES

Topology	# of entries for optimal pure explicit routing	# of entries for hybrid routing (explicit routing configuration)	# of entries for hybrid routing (destination-based routing configuration)
Abilene	358	45	132
Exodus	1471	181	462
Abovenet	1137	130	462
EBONE	1485	228	506
Sprintlink	6287	852	1892

The initial value when  $K = 0$  corresponds to the destination-based routing calculated in Section IV-C based on the derived basic traffic matrix. When traffic is routed by that destination-based routing, Sprintlink has very high throughput, while the other four have considerable room for further improvement. In particular, the normalized throughput of EBONE is only 74.9%. This is because single traffic matrix can not precisely reflect traffic fluctuations and not all big flows are concurrent as we assumed. Thus, we need to introduce complementary explicit routing to rebalance traffic from congested links to underutilized links. The sharp increase in the normalized throughput shown in Fig. 3 indicates that our hybrid routing algorithm can achieve near-optimal performance compared with optimal pure explicit routing, even when applying explicit routing to a few selected key node pairs. For the subsequent experiments, we set  $K = 15\% * N(N - 1)$  for each network.

2) **Forwarding Entry Savings:** With 15% node pairs using explicit routing, hybrid routing introduces only a few explicit routing forwarding entries compared with optimal pure explicit routing. Table II shows the required forwarding entries installed in all routers for pure explicit routing and hybrid routing. Since destination-based routing entries are stored in SRAM, the proposed hybrid routing scheme greatly reduces the required forwarding entries by at least 84.6% and thus saves a great deal of TCAM resources in the routers compared with the optimal pure explicit routing, especially in large networks (e.g., Sprintlink).

## VI. CONCLUSION

We propose a low-complexity hybrid routing scheme to achieve near-optimal load balancing for multiple traffic matrices. The main idea is to complement destination-based routing with a small number of explicit routing forwarding entries. As a result, the load balancing performance is greatly improved while the extra complexity in terms of explicit forwarding entries is kept very low. This has great value for practice in that the scheme requires very small TCAM to implement. Hybrid routing is very suitable for implementation using SDN. We present the framework of hybrid routing and explain the details of the algorithm design. The performance of hybrid routing is verified in several practical networks using simulation. The results show that hybrid routing achieve near-optimal load balancing with only a very small fraction of node pairs using explicit routing. In particular, in all practical networks used in

our evaluation, hybrid routing can save at least 84.6% TCAM resources, compared with pure explicit routing.

## REFERENCES

- [1] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, "Overview and Principles of Internet Traffic Engineering," *IETF RFC 3272*, May 2002.
- [2] A. Sridharan, R. Guerin, and C. Diot, "Achieving near-optimal traffic engineering solutions for current OSPF/IS-IS networks," *Networking, IEEE/ACM Transactions on*, vol. 13, pp. 234 – 247, April 2005.
- [3] B. Fortz and M. Thorup, "Optimizing OSPF/IS-IS weights in a changing world," *Selected Areas in Communications, IEEE Journal on*, vol. 20, pp. 756 –767, May 2002.
- [4] J. Zhang, K. Xi, L. Zhang, and H. Chao, "Optimizing network performance using weighted multipath routing," in *Computer Communications and Networks (ICCCN), 2012 21st International Conference on*, pp. 1 –7, 30 2012-aug. 2 2012.
- [5] Y. Wang and Z. Wang, "Explicit routing algorithms for internet traffic engineering," in *Computer Communications and Networks, 1999. Proceedings. Eight International Conference on*, pp. 582 –588, 1999.
- [6] C. Jian and L. Chin-Tau, "Optimal link weights for IP-Based networks supporting Hose-Model VPNs," *Networking, IEEE/ACM Transactions on*, vol. 17, pp. 778 –788, June 2009.
- [7] C. Zhang, Y. Liu, W. Gong, J. Kurose, R. Moll, and D. Towsley, "On optimal routing with multiple traffic matrices," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 1, pp. 607–618 vol. 1, 2005.
- [8] M. Kodialam, T. V. Lakshman, J. B. Orlin, and S. Sengupta, "Oblivious routing of highly variable traffic in service overlays and IP backbones," *IEEE/ACM Trans. Netw.*, vol. 17, pp. 459–472, April 2009.
- [9] K. Kannan and S. Banerjee, "Compact team: Flow entry compaction in team for power aware sdn," in *Distributed Computing and Networking*, pp. 439–444, Springer, 2013.
- [10] C. Hopps, "Analysis of an equal-cost multi-path algorithm," *IETF RFC 2992*, November 2000.
- [11] Z. Cao, Z. Wang, and E. Zegura, "Performance of hashing-based schemes for internet load balancing," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 1, pp. 332 –341 vol.1, 2000.
- [12] Y. Zhang and Z. Ge, "Finding critical traffic matrices," in *Dependable Systems and Networks, 2005. DSN 2005. Proceedings. International Conference on*, pp. 188–197, 2005.
- [13] CPLEX. ILOG CPLEX Division [Online]. Available: <http://www.cplex.com>.
- [14] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson, "Measuring isp topologies with rocketfuel," *IEEE/ACM Trans. Netw.*, vol. 12, pp. 2–16, Feb. 2004.
- [15] "Yin Zhang's Abilene TM." [Online]. Available: <http://www.cs.utexas.edu/~yzhang/research/AbileneTM/>.
- [16] Configuring OSPF. Cisco Systems Product Documentation [Online]. Available: <http://www.cisco.com/univercd/home/home.htm>.
- [17] J. Zhang, K. Xi, M. Luo, and H. Chao, "Dynamic hybrid routing: Achieve load balancing for changing traffic demands," in *Quality of Service (IWQoS), 2014 22nd International Workshop on*, May 2014.
- [18] M. Roughan, "First order characterization of internet traffic matrices," in *In proceedings of the 55th session of the International Statistical Institute*, (Sydney, Australia), Apr. 2005.