Internet Traffic Engineering without Full Mesh Overlaying

Yufei Wang, Zheng Wang, Leah Zhang

Abstract—**The overlay approach has been widely used by many service providers for traffic engineering in large Internet backbones. In the overlay approach, logical connections are set up between edge nodes to form a full mesh virtual network on top of the physical topology. IP routing is then run over the virtual network. Traffic engineering objectives are achieved through carefully routing logical connections over the physical links. Although the overlay approach has been implemented in many operational networks, it has a number of well-known scaling issues. This paper proposes a new approach, which we call the integrated approach, to achieve traffic engineering without full-mesh overlaying. In the integrated approach, IP routing runs natively over the physical topology rather than over the virtual network. Traffic engineering objectives are realized by setting appropriate link metrics in IP routing protocols. We first illustrate our approach with a simple network, then present a formal analysis of the integrated approach and a method for deriving the appropriate link weights. Our analysis shows that for any given set of optimal routes of the overlay approach with respect to a set of traffic demands, the integrated approach can achieve exactly the same result by reproducing them as shortest paths. We further extend the result to a more generic one: for any arbitrary set of routes, as long as they are not loopy, they can be converted to shortest-paths with respect to some set of positive link weights. A theoretical insight of our result is that the optimal routing (with respect to any objective function) is always shortest path routing with respect to some appropriate positive link weights.**

Keywords—**Traffic engineering, performance optimization, IP routing, Linear Programming.**

I. INTRODUCTION

IP routing typically uses shortest-path computation with some simple metrics such as hop-count or delay. Although the simplicity of this approach allows IP routing to scale to very large networks, it does not make the best use of network resources [1]. In large Internet backbones, service providers typically have to explicitly manage the traffic flows in order to optimize the use of network resources. This process is often referred to as traffic engineering. Common objectives of traffic engineering include balancing traffic distribution across the network and avoiding congestion hot spots.

Currently, most large Internet backbones employ the so-called *overlay* approach for traffic engineering [1]. With this approach, service providers establish logical connections between the edge nodes of a backbone, and then overlay these logical connections onto the physical topology. The logical connections are set up as ATM or Frame Relay PVCs. The emerging Multi-Protocol Label Switching (MPLS) standard provides another method for setting up such logical connections. Service providers can control the distribution of traffic over physical topology through carefully routing these logical connections over physical links. For example, suppose that a service provider has a backbone network connecting all major cities in the US. The service provider sets up a logical connection between New York City and San Francisco for all traffic between the two cities. This logical connection can be mapped to a set of physical links going through Chicago or through Houston depending on which route achieves better traffic distribution. IP routing runs over the logical connection. Any logical connection appears to be one hop away from IP perspective. The mapping of the logical connections to physical links is therefore transparent to the IP layer.

These logical connections between edge nodes essentially form a full-mesh virtual network atop of the physical topology. In our previous work, we have established that the overlay approach can achieve optimal traffic distribution in a network that supports multi-link loadsharing [2]. The optimal mapping between the logical connections and the physical links can be computed using a linear programming formulation.

While the overlay approach has been widely implemented in current Internet backbones, it does have some scalability limitation. First, it suffers the so-called "N-square" problem. To establish full meshed logical connections between N edge nodes, each node has to set up logical connections to (N-1) other nodes. Thus, N*(N-1) logical connections have to be established for the full-mesh virtual network. As the size of the backbone network increases, the number of logical connections to be established will rise drastically, adding considerable management complexity and messaging overheads. Second, when IP routing runs over such a fully meshed virtual network, each edge node has to establish routing peering with (N-1) other nodes. This poses a significant problem to current IP routers as most of them can not support a large number of peers. Note that multiple logical connections may go over the same physical link. Thus, the breakdown of a single physical link may cause multiple logical connections to fail, and this will exaggerate the routing update load.

Let us look at an example. Suppose that a backbone has 20 Points of Presence (PoPs) and each PoP has 10 edge nodes connecting to customers. To establish logical connections between all edge nodes, we need to set up 39,800 logical connections. The number can go up very quickly as the network grows in size.

In this paper, we consider a new approach that accomplishes traffic engineering objectives without full mesh overlaying. Instead of overlaying IP routing over the logical virtual network, the new approach runs shortest-path IP routing natively over the physical topology, as it is the case in most enterprise networks. Traffic engineering objectives such as balanced traffic distribution are achieved through manipulating link metrics for IP routing protocols such as OSPF [3]. We refer this approach as the *integrated* approach since IP routing is running over the physical topology rather than the full-mesh virtual network.

Similar approaches have been tried by some service providers in the past. When a link is experiencing congestion, service providers typically increase the weight for that link in the hope that traffic will be moved away from it. These experiments,

---

Yufei Wang and Zheng Wang are with Bell Labs, Lucent Technologies and Leah Zhang is with AT&T Labs.

however, were done based on simple heuristics. The lack of a systematic strategy and comprehensive studies of link weight change impact has prevented it from being widely adopted in operational backbone networks.

Another related work on achieving better traffic distribution without full-mesh overlaying is to use equal-cost load balancing in the OSPF routing protocol [4]. In [5], [6], the performance of the equal cost load balancing approach and the implementation issues on traffic splitting in the context of next hop forwarding are studied. The effectiveness of this approach largely relies on how many equal cost shortest paths exist between each source and destination pair. In a related effort, OSPF link weights and equal traffic loadsharing are combined to improve performance [7]. The analysis shows that the link weight optimization problem under equal loadsharing is NP-hard. In addition, a local search heuristic algorithm is proposed which achieves a performance quite close to the optimal routing only on a specific example.

In this paper, we present a formal analysis of the integrated approach, and propose a systematic method for deriving the link metrics that convert a set of optimal routes for traffic demands to shortest-paths with respect to the link weights. Our theoretic results show that the integrated approach can achieve exactly the same result as the overlay approach. For any given traffic demands, it is possible to select a set of link weights such that the shortest paths based on the selected link weights produce the same traffic distribution as that of the overlay approach with the assumption that traffic between the same source-destination pair can be split across multiple equal cost shortest paths, if exist, according to specified proportions.

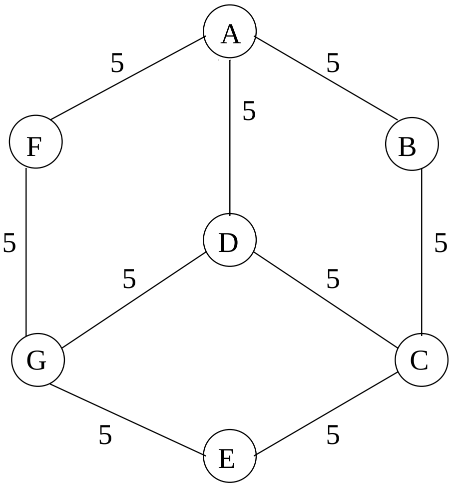
This integrated approach has a number of advantages. First, it retains the simplicity of IP routing and requires little changes to the basic Internet architecture. Once the link weights are calculated and set, the shortest-path routing protocol such as OSPF can calculate the paths in the normal way, and packets are forwarded along the shortest paths. In contrast, the overlay approach requires the setup of a full-mesh logical network based on traffic matrix and running of IP routing on top of the virtual network. Second, it eliminates the "N-square" problem all together and reduces messaging overheads in setting up logical connections.

The paper is organized as follows. Section 2 provides an overview of the integrated approach. Section 3 introduces the linear programming notation and terminology, and presents the formulation for the shortest path problem. Section 4 shows that the optimal routes for traffic demands can always be reproduced as shortest-paths with respect to appropriate link weights. Section 5 further extends the results to arbitrary routing, as long as non-pathological. Section 6 defines pathological routing and characterizes routing loops when a given set of paths can not be reproduced as shortest paths. Section 7 concludes the paper with some observations and remarks.

II. OVERVIEW OF INTEGRATED APPROACH

Before we present the theoretic results, let us first illustrate with a simple example how the integrated approach works.

Figure 1 shows a simple network topology, link capacities, and traffic demands. Each link has a capacity of 5 units and each demand needs bandwidth of 4 units. Although link capacities, link weights and traffic demands are unidirectional in IP networks, we assume they are bidirectional here for simplicity.



A->B: 4
A->F: 4
B->F: 4
A->E: 4

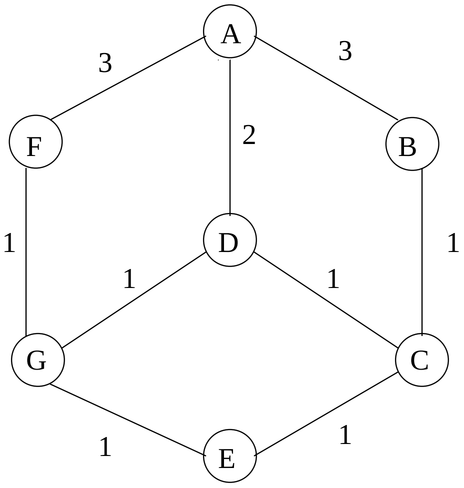
Fig. 1. Topology, Capacity and Traffic Demands

To meet the traffic engineering objectives, we need to place the demands over the links in a way that the traffic distribution is balanced and there is no congestion or hot spot in the network. The optimal routes can be calculated using a linear programming formulation [2].

Since the network here is rather small, this process of traffic engineering can be done manually. The optimal routes for achieving balanced traffic distribution is as follows. Demand A to B uses path AB, and demand A to F uses AF. Demand B to F has two paths. Half of the demand goes over $BCDGF$ and the other half over $BCEGF$. Demand A to E also has two paths. Half traverses path $ADCE$ and the other half traverses $ADGE$. This optimal routing results in a 4 unit load on every link - the traffic distribution is balanced and the link utilization is 80% uniformly for the entire network. It is easy to see that some other routing arrangement may produce unbalanced traffic distribution and cause hot spots in the network.

To implement the optimal routes with the overlay approach is quite straight forward. We simply set up six logical connections: AB, AF, $BCDGF$, $BCEGF$, $ADCE$ and $ADGE$, and run IP routing over them. $BCDGF$ and $BCEGF$ appear as equal-cost paths, so routing protocols such as OSPF will perform loadsharing over them. This will implement the optimal routes and balance traffic distribution.

With the integrated approach, there is no need to establish the logical connections. We simply calculate and set the appropriate link weights on the links, and the shortest-path routing will calculate the paths by itself. Figure 2 shows the link weights under which the shortest paths match the optimal routes exactly. Demand A to B will follow shortest-path AB and demand A to F follow shortest-path AF. For demand B to F and demand A to E, there are two equal-cost shortest paths for each of them.



A->B: 4 A-B
A->F: 4 A-F
B->F: 2 B-C-D-G-F
B->F: 2 B-C-E-G-F
A->E: 2 A-D-G-E
A->E: 2 A-D-C-E

Fig. 2. Optimal link weights and optimal routes

Both demands are distributed between their respective two equal cost shortest paths. This shows that in this example, setting appropriate link weights can achieve the same optimal routes as the overlay approach does. In the following sections we will show that the result is not a coincidence for this small example. The conclusion holds in general.

III. BASIC ASSUMPTIONS

In this section we first discuss the basic assumptions. We then introduce some mathematical notation, and present a linear programming formulation for the shortest path problem.

In the above example, we consider balanced traffic distribution as the overall objective for traffic engineering. For some other applications, the optimization objectives may be different, such as minimum congestion or maximum throughput. We do not restrict ourselves to any specific objectives. Our results are generic and can be applied to all of these objectives.

We model the IP network as a set of nodes connected by links with fixed bandwidth capacities. We assume that the point to point traffic demands between nodes are given, each with a fixed bandwidth requirement. Let digraph $G = (V, E)$ represent the IP network, where V is the set of nodes and E is the set of links. Please note that the links and their capacities are directional, i.e. link (i, j) is considered different from link (j, i), each with its own capacity. Let K be the set of point to point demands. For each $k \in K$, let d_k, s_k, t_k be the demand bandwidth (size), the source node, and the destination node respectively.

As a preparation step, we need to borrow some basic results from linear programming. The details can be found in [8], [9]. Since shortest path is the most elementary routing scheme and is used extensively as a building block in our analysis, we would like to first study the shortest path problem in the context of linear programming. For this purpose, let $\{w_{ij} : (i,j) \in E\}$ be a given set of link weights. We need to define the following variables. For each link $(i,j) \in E$ and for each demand $k \in K$, let X_{ij}^k represent the percentage of demand k which flows across link (i,j). Then the shortest path problem can be formulated as

Shortest Path Formulation (P-SP)

$$\begin{array}{lll} \min \ \sum_{k \in K} \sum_{(i,j) \in E} w_{ij} X_{ij}^k & & (1) \\ s.t. & & \\ \sum_{j:(i,j)\in E} X_{ij}^k - \sum_{j:(j,i)\in E} X_{ji}^k = 0, & \begin{array}{l} k \in K \\ i \neq s_k, t_k \end{array} & (2) \\ \sum_{j:(i,j)\in E} X_{ij}^k - \sum_{j:(j,i)\in E} X_{ji}^k = 1, & \begin{array}{l} k \in K \\ i = s_k \end{array} & (3) \\ 0 \leq X_{ij}^k \leq 1 & & \end{array}$$

The objective function (1) is to minimize the total weight of links used. Constraint (2) and (3) are flow conservation constraints. Equation (2) says that the traffic flowing into a node must equal the traffic flowing out of the node for any node other than the source node and the destination node for each demand. Equation (3) says that the net flow out of the source node is 1, which is the total required bandwidth after scaled by d_k. The last constraint restricts all the variables to be non-negative real numbers between 0 and 1. Please note that the flow conservation constraint at the destination node is not included because it's redundant.

We next introduce the concept of linear programming duality [9]. Each linear programming problem (P) is associated by another linear programming problem (D). (P) is called the primal and (D) is called the dual. (P) and (D) are mathematically equivalent in the sense that they have the same optimal objective value, and the optimal solution of (D) can be derived from the optimal solution of (P) and vice versa by the relationship of complementary slackness. The variables in (D) can be interpreted as shadow prices for the (resource) constraints in (P) [9].

The dual of the shortest path formulation (P-SP) is given as

Dual Shortest Path Formulation (D-SP)

$$\begin{array}{lll} \max \ \sum_{k \in K} U_{t_k}^k & & (4) \\ s.t. & & \\ U_j^k - U_i^k \leq w_{ij}, & k \in K, (i,j) \in E & (5) \\ U_{s_k}^k = 0, & k \in K. & \end{array}$$

Note that in these two formulations different demands do not interact. So each one of them can be decomposed to a number of subproblems, one for the shortest path of each demand. We prefer these combined formulations only for ease of presentation. Now for the shortest path problem the duality relationship can be interpreted in the following way. Let $\{\bar{X}_{ij}^k\}$ be the optimal solution of (P-SP). The values of $\{\bar{X}_{ij}^k\}$ determine the shortest path for each demand. If $\{\bar{X}_{ij}^k\}$ take values of only 0 and 1, a unique shortest path is determined for each demand. Such an integral solution is guaranteed to exist [8]. In the case that the problem (P-SP) has multiple optimal solutions, $\{\bar{X}_{ij}^k\}$ may take fractional values in some optimal solution. In this case there exist multiple shortest paths for a demand (equal cost multiple paths), and when we refer to a path, we mean any one of these paths. Let $\{\bar{U}_i^k\}$ be the optimal solution of the dual (D-SP). The value of $\bar{U}_i^k$ can be viewed as the distance from the source s_k to node i based on the shortest paths for demand k determined in (P-SP). In particular, $\bar{U}_{t_k}^k$ is the total length of the shortest path from s_k to t_k.

By applying the complementary slackness relationship of linear programming duality theory to the primal (P-SP) dual (D-SP) pair, we have

Lemma 1 (complementary slackness) If $\bar{X}_{ij}^k > 0$, then $\bar{U}_j^k - \bar{U}_i^k = w_{ij}$.

If P^k is a path determined by $\{\bar{X}_{ij}^k : (i,j) \in E\}$, then for every link $(i,j) \in P^k$, $\bar{X}_{ij}^k > 0$, which in turn implies that $\bar{U}_j^k - \bar{U}_i^k = w_{ij}$.

Theorem 1: Let P be a path from s_k to t_k. If for every link $(i,j) \in P$, $\bar{U}_j^k - \bar{U}_i^k = w_{ij}$, then P is a shortest path with respect to link weights $\{w_{ij}\}$.

Proof: Let $P = p_0, p_1, \ldots, p_{l-1}, p_l$ where $p_j, 0 < j < l$ are all the nodes on the path in order with $p_0 = s_k$ and $p_l = t_k$. Then we have

$$\bar{U}_{p_j}^k - \bar{U}_{p_{j-1}}^k = w_{p_{j-1}p_j}$$

for $0 \leq j \leq l$. If we sum over all these equations, we get

$$\bar{U}_{t_k}^k = \bar{U}_{t_k}^k - \bar{U}_{s_k}^k = \sum_{j=1}^{l} w_{p_{j-1}p_j}$$

Note that $\sum_{j=1}^{l} w_{p_{j-1}p_j}$ is the length of path P. Let $Q = q_0, q_1, \ldots, q_{m-1}, q_m$ be any other path between s_k and t_k. Then by constraint (7),

$$\bar{U}_{q_j}^k - \bar{U}_{q_{j-1}}^k \leq w_{q_{j-1}q_j}$$

for $0 \leq j \leq m$. Similarly add all the equations, we get

$$\bar{U}_{t_k}^k \leq \sum_{j=1}^{m} w_{q_{j-1}q_j}$$

which implies that $\sum_{j=1}^{l} w_{p_{j-1}p_j} \leq \sum_{j=1}^{m} w_{q_{j-1}q_j}$. That is, the length of path Q is at least as large as the length of path P. Therefore, P is a shortest path. ■

Lemma 1 and theorem 1 together say that every path determined by $\{\bar{X}_{ij}^k\}$ is a shortest path.

IV. THE SHORTEST PATH PROPERTY OF OPTIMAL ROUTING

Before we present the generic result, we first consider a special case where we show a set of optimal routes that achieve balanced traffic distribution can be translated into shortest-paths with respect to a set of positive link weights. Mathematically, the objective of balanced traffic distribution can be described as minimizing the maximum of link utilization.

We first introduce some notation and reinterpret some existing notation (without causing any confusion). Let α be the maximal link utilization cross the entire network, c_{ij} be the capacity of link (i,j). Let X_{ij}^k represent the percentage of the bandwidth of demand k routed over link (i,j). Then the linear programming formulation for this specific traffic engineering problem is given as

Linear Programming Formulation (LPF)

$$\begin{array}{lll}
\min\ \alpha + r\sum_{k\in K}\sum_{(i,j)\in E} X_{ij}^k & & (6)\\
s.t. & & \\
\sum_{j:(i,j)\in E} X_{ij}^k - \sum_{j:(j,i)\in E} X_{ji}^k = 0, & \begin{array}{l} k \in K \\ i \neq s_k, t_k \end{array} & (7)\\
\sum_{j:(i,j)\in E} X_{ij}^k - \sum_{j:(j,i)\in E} X_{ji}^k = 1, & \begin{array}{l} k \in K \\ i = s_k \end{array} & (8)\\
\sum_{k\in K} d_k X_{ij}^k \leq c_{ij}\alpha, & (i,j) \in E & (9)\\
0 \leq X_{ij}^k \leq 1 & &
\end{array}$$

The objective function (6) is to minimize the maximum of link utilization. Constraint (7) and (8) are flow conservation constraints. Equation (7) says that the traffic flowing into a node has to equal the traffic flowing out of the node for any node other than the source node and the destination node for each demand. Equation (8) says that the net flow out of the source node is 1, which is the total required bandwidth after scaled by d_k. Constraint (9) is the link capacity utilization constraint. It says that the total amount of bandwidth consumed by all the demands routed on a link should not exceed the maximum utilization rate times the total capacity of the link. The last constraint restricts the X_{ij}^k variables to be between 0 and 1. Please note that we added the second term $r\sum_{(i,j)\in E}\sum_{k\in K} X_{ij}^k$ to the objective function. This will ensure that the optimization not only minimizes α, but also all the X_{ij}^k variables. This is needed because otherwise the objective function is the utilization of only the most congested link and the optimal solution may include unnecessarily long paths as long as they can avoid the bottleneck link. The scalor r is a very small positive number which is introduced to ensure that the minimization of α takes higher priority.

We can solve the above LPF problem with the classic Simplex method [9]. The optimal solution of (LPF) gives a route or a set of routes (splitting) for each demand. In case that a demand has to be splitting, it also gives the proportions according to which the traffic between the source and the destination nodes should be distributed across the multiple routes. Obviously, these optimal routes can be implemented with overlay approach as logical connections. Now we show that how these optimal routes can also be reproduced as shortest paths by setting appropriate link weights.

Recall that the optimal routes are derived from the optimal solution of the linear programming problem (LFP). We need to rely on the linear programming duality theory introduced previously to show their shortest path property. The dual of (LPF) is given as

Dual formulation (DPF)

$$\begin{array}{lll}
\max & \sum_{k\in K} d_k U_{t_k}^k & \\
s.t. & U_j^k - U_i^k \leq W_{ij} + r, & k \in K, (i,j) \in E\\
& \sum_{(i,j)\in E} c_{ij} W_{ij} = 1, & \\
& W_{ij} \geq 0, U_{s_k}^k = 0 &
\end{array}$$

Let $\{\bar{X}_{ij}^k\}$ and $\bar{\alpha}$ be the optimal solution of (LPF) and $\{\bar{U}_i^k\}$ and $\{\bar{W}_{ij}\}$ be the optimal solution of the dual (DPF). Then complementary slackness relationship for the linear programming

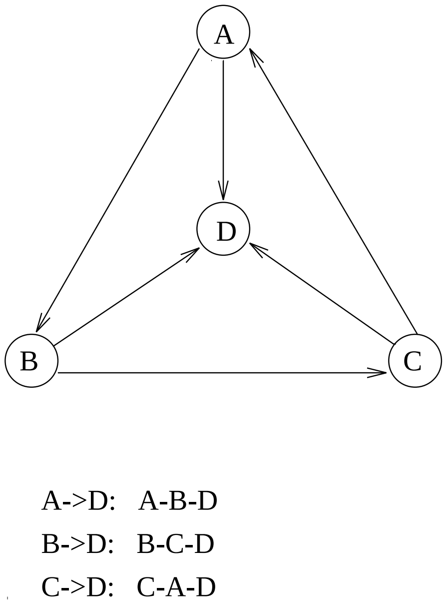


Fig. 3. Loopy Paths

pair (LPF)-(DPF) can be stated as follows:

Lemma 2 (complementary slackness) If $\bar{X}_{ij}^k > 0$ then $\bar{U}_j^k - \bar{U}_i^k = \bar{W}_{ij} + r$.

Similarly, the dual variables here can be interpreted as follows. If $\{\bar{W}_{ij}\}$ are viewed as constants, then the value of $\bar{U}_i^k$ is the shortest path distance from the source s_k to node i with respect to link weight $\{\bar{W}_{ij}\} + r$. In particular, $\bar{U}_{t_k}^k$ is the total length of the shortest path. Let $\bar{w}_{ij} = \{\bar{W}_{ij}\} + r$. Then $\bar{w}_{ij} > 0$.

Theorem 2: Let P be any path from s_k to t_k determined by the (LPF) optimal solution $\{\bar{X}_{ij}^k\}$. Then P is a shortest path with respect to link weights $\{\bar{w}_{ij}\}$.

Proof: Since P is a path for demand k determined by the (LPF) optimal solution, for every link (i, j) on P, we have $\bar{X}_{ij}^k > 0$. This implies from lemma 2 that $\bar{U}_j^k - \bar{U}_i^k = \bar{w}_{ij}$. By theorem 1, P is a shortest path from s_k to t_k with respect to link weights $\{\bar{w}_{ij}\}$. ■

In summary, we have shown that for the given objective, the optimal routes can be reproduced as shortest paths based on certain positive link weights, and the link weights can be derived through the optimal dual solution.

V. GENERALIZATION TO ARBITRARY ROUTING

Now we would like to extend the result of the previous section to a generic one. We show that any arbitrary set of routes, as long as it is not loopy (which will be defined later in this section), can be converted to shortest paths with respect to a set of positive link weights. Furthermore, we prove that for any loopy set of routes, there always exists a better set of routes that consumes less bandwidth, and this better set of routes can be converted to shortest paths.

First, we would like to show that not all routes can be reproduced as shortest paths with respect to a set of positive link weights. We use the routing problem illustrated by Figure 3 as a counter example. In this example, there are three point to point traffic demands, going from A to D, B to D, and C to D respectively. The chosen routes for these demands are given in the Figure 3. All link capacity and traffic demands are of 1 unit.

Lemma 3: It is not possible to convert the routes in Figure 3 to shortest paths with respect to any positive link weights.

Proof: Suppose otherwise. Let's choose node D as the common destination and define function $f(X)$ for each node X as the distance of the shortest path from X to D. Since all links have positive weights, the function is well-defined and has the following property: if Y is an intermediate node on the shortest path from X to D, then $f(X) > f(Y)$. Since B is on the shortest path from A to D, we have $f(A) > f(B)$. Similarly, the other two routes imply that $f(B) > f(C)$ and $f(C) > f(A)$. These three inequalities together imply that $f(A) > f(A)$, a contradiction. ■

One may observe that the routes for traffic demands in this example are "badly" chosen. Because once path $A - B - D$ is chosen for demand A to D, demand B to D has to use path $B - C - D$ since link $B - D$ has no capacity left. The same goes to demand C to D. However, if we choose path $A - D$ for demand A to D, demand B to D and demand C to D can use path $B - D$ and path $C - D$ respectively. This way we would have a better set of routes that uses less bandwidth because the triangle loop ABC is saved. This new set of routes can be easily converted to shortest paths with respect to a set of positive link weights. In fact, we can assign weight 1 to all links in this case.

As we will show later, this observation again holds true in general cases. That is, for any set of routes that can not be reproduced as shortest-paths with respect to a set of positive link weights, there always exists a better set of routes that can be reproduced as shortest-paths.

We first introduce the following definitions. A set of routes is called *shortest-path-reproducible* if there exists a set of positive link weights based on which all the routes in the set are shortest paths. Otherwise it is called *non-shortest-path-reproducible*. A set of routes $R1$ is called *loopy* if there exists another set of routes $R2$ such that on every link $R2$ uses bandwidth less than or equal to $R1$ and strictly less than on at least one link.

In the next section we will show that loopy routes indeed contain some kind of loops. For any practical purposes, $R2$ is certainly a better set of routes than $R1$ irrespective of what traffic engineering objectives we use since $R2$ uses less bandwidth uniformly.

We now show that for any given set of routes, it is either shortest-path-reproducible or loopy. Let $\{P^k : k \in K\}$ be any given set of routes, which can be a set of arbitrary routes. In order to identify whether $\{P^k : k \in K\}$ is loopy, let's consider the following problem of route improvement optimization: can we find a better set of routes which uses minimum total bandwidth and which consumes no more bandwidth than $\{P^k : k \in K\}$ on every single link? For each $(i, j) \in E$, let c_{ij} be the total bandwidth consumption by $\{P^k : k \in K\}$ on link (i, j), i.e.

$$c_{ij} = \sum_{k \in K:(i,j) \in P^k} d_k.$$

Then the optimization problem can be specifically formulated as the following linear programming problem:

Improvement Formulation (P-IM)

$$\min \sum_{k\in K}\sum_{(i,j)\in E} d_k X_{ij}^k \tag{11}$$

$s.t.$

$$\sum_{j:(i,j)\in E} X_{ij}^k - \sum_{j:(j,i)\in E} X_{ji}^k = 0, \qquad \begin{array}{l} k\in K \\ i\neq s_k, t_k \end{array} \tag{12}$$

$$\sum_{j:(i,j)\in E} X_{ij}^k - \sum_{j:(j,i)\in E} X_{ji}^k = 1, \qquad \begin{array}{l} k\in K \\ i = s_k \end{array} \tag{13}$$

$$\sum_{k\in K} d_k X_{ij}^k \leq c_{ij}, \qquad (i,j)\in E \tag{14}$$

$$0 \leq X_{ij}^k \leq 1$$

The objective function (11) is to minimize the total consumed bandwidth over all the links. Constraint (12) and (13) are flow conservation constraints as explained in formulation (P-SP). Constraint (14) says that the new paths cannot use more bandwidth than the old paths on any link. The last constraint restricts all the variables to be non-negative numbers between 0 and 1.

Intuitively speaking, if the optimal solution of (P-IM) results in a lower value of the objective function (11) than that of $\{P^k : k\in K\}$, then the optimal solution is a better routing because it uses no more bandwidth than $\{P^k : k\in K\}$ on every link, and uses less bandwidth over all. This implies that $\{P^k : k\in K\}$ is loopy.

The dual of (P-IM) is given as follows:

Dual formulation(D-IM)

$$\max \sum_{k\in K} d_k U_{t_k}^k - \sum_{(i,j)\in E} c_{ij} W_{ij} \tag{15}$$

$s.t.$

$$U_j^k - U_i^k \leq 1 + W_{ij}, \qquad \begin{array}{l} k\in K \\ (i,j)\in E \end{array} \tag{16}$$

$$W_{ij} \geq 0, U_{s_k}^k = 0 \tag{17}$$

Let $\{\bar{U}_i^k\}$ and $\{\bar{W}_{ij}\}$ be an optimal solution of the dual (D-IM). We make the following observation. If $\{\bar{W}_{ij}\}$ are viewed as constants, then $\{\bar{U}_i^k\}$ must be the optimal solution of the dual shortest path problem (D-SP) with link weights $\{w_{ij}\}$ being replaced by $\{\bar{W}_{ij}+1\}$. Let $\{\bar{X}_{ij}^k\}$ be any optimal solution of (P-IM) (there may be more than one optimal solutions).

By the complementary slackness relationship of primal (P-IM) dual (D-IM) pair, we have

Lemma 4 (complementary slackness) If $\bar{X}_{ij}^k > 0$ then $\bar{U}_j^k - \bar{U}_i^k = \bar{W}_{ij} + 1$.

Let $w_{ij} = \bar{W}_{ij} + 1$ for every $(i,j)\in E$. Note that $w_{ij} > 0$ for all $(i,j)\in E$ because $\bar{W}_{ij} \geq 0$.

Lemma 5: If P is a path from s_k to t_k defined by $\{\bar{X}_{ij}^k : (i,j)\in E\}$, then P is a shortest path with respect to link weights $\{w_{ij}\}$.

Proof: For every link $(i,j)\in P$, by definition, $\bar{X}_{ij}^k > 0$. From lemma 4, we have $\bar{U}_j^k - \bar{U}_i^k = w_{ij}$. By theorem 1, P is a shortest path with respect to the positive link weights $\{w_{ij}\}$. ■

Now we are ready to state the main result.

Theorem 3: Either $\{P^k : k\in K\}$ is a set of shortest paths with respect a set of positive link weights or it is loopy.

Proof: Let's define $\{\hat{X}_{ij}^k : (i,j)\in E\}$ as the feasible solution of (P-IM) representing $\{P^k : k\in K\}$, i.e. $\hat{X}_{ij}^k = 1$ if $(i,j)\in P^k$ and 0 otherwise. If $\{\hat{X}_{ij}^k\}$ is an optimal solution of (P-IM) (doesn't have to be identical to $\{\bar{X}_{ij}^k\}$), then by lemma 5 $\{P^k : k\in K\}$ are shortest paths with respect to positive link weights $\{w_{ij}\}$. If $\{\hat{X}_{ij}^k\}$ is not an optimal solution, we want to show that it is loopy. Let's $\{Q^k : k\in K\}$ be the paths determined by $\{\bar{X}_{ij}^k\}$. Since $\{\bar{X}_{ij}^k\}$ is an optimal solution, $\{Q^k : k\in K\}$ is a set of shortest paths with respect to weights $\{w_{ij}\}$. Since $\{\hat{X}_{ij}^k\}$ is not optimal and the (P-IM) objective function value at $\{\hat{X}_{ij}^k\}$ is $\sum_{(i,j)\in E} c_{ij}$, we have $\sum_{(i,j)\in E} c_{ij} > \sum_{(i,j)\in E}\sum_{k\in K} d_k \bar{X}_{ij}^k$. Constraint (14) guarantees that $\{Q^k : k\in K\}$ uses less or equal bandwidth than $\{P^k : k\in K\}$ on every link. If $\{Q^k : k\in K\}$ and $\{P^k : k\in K\}$ use the same amount of bandwidth on all the links, we can imply that $\sum_{(i,j)\in E} c_{ij} = \sum_{(i,j)\in E}\sum_{k\in K} d_k \bar{X}_{ij}^k$, a contradiction. Therefore $\{Q^k : k\in K\}$ consumes strictly less bandwidth than $\{P^k : k\in K\}$ on at least one link. ■

The proof of Theorem 3 is a constructive one. If the given set of paths is shortest-path-reproducible, it gives the link weights to reproduce them as shortest-paths. If the paths are loopy, it gives a better set of routes which is shortest-path-reproducible. Theorem 3 implies that non-shortest-path-reproducible routes are loopy. We will show that the reverse is also true in the next section.

VI. CHARACTERISTICS OF LOOPY PATHS

We now consider the case that $\{P^k : k\in K\}$ is loopy, and present some characteristics of loopy paths. According to Theorem 3, we can find a set of improved paths which is shortest-path-reproducible. The improvement comes in the form of capacity saving on at least one link. What we want to show now is that the capacity savings must happen on a cycle of links. Since a minimum of two links is needed to form a cycle, the capacity savings have to happen on at least two links. The links with capacity savings are called *reduction links*.

Let $\{\hat{X}_{ij}^k\}$ represent the old path $\{P^k : k\in K\}$, i.e. $\hat{X}_{ij}^k = 1$ if $(i,j)\in P^k$ and 0 otherwise. Let $\{\bar{X}_{ij}^k\}$ represent the new (improved) paths. Since both $\{\hat{X}_{ij}^k\}$ and $\{\bar{X}_{ij}^k\}$ are feasible solutions of (P-IM), both of them satisfy constraints (12), (13) and (14). Actually (14) becomes equalities for $\{\hat{X}_{ij}^k\}$: $\sum_{k\in K} d_k \hat{X}_{ij}^k = c_{ij}$, for each $(i,j)\in E$. If we subtract constraints (12), (13) and (14) for $\{\hat{X}_{ij}^k\}$ by constraints (12), (13) and (14) for $\{\bar{X}_{ij}^k\}$ respectively, we get:

$$\sum_{(i,j)\in E}\sum_{k\in K} d_k(\hat{X}_{ij}^k - \bar{X}_{ij}^k) > 0 \tag{18}$$

$$\begin{array}{ll} \sum_{j:(i,j)\in E}(\hat{X}_{ij}^k - \bar{X}_{ij}^k) - & k\in K \\ \sum_{j:(j,i)\in E}(\hat{X}_{ji}^k - \bar{X}_{ji}^k) = 0, & i\in V \end{array} \tag{19}$$

$$\sum_{k\in K} d_k(\hat{X}_{ij}^k - \bar{X}_{ij}^k) \geq 0, \qquad (i,j)\in E \tag{20}$$

If we let $Z_{ij} = \sum_{k\in K} d_k(\hat{X}_{ij}^k - \bar{X}_{ij}^k)$, then $\{Z_{ij} \geq 0\}$ represent precisely link capacity savings between the old routes and the new routes. Substitute $\{Z_{ij}\}$ in, multiply (19) by d_k and add it over all $k\in K$, we arrive at the following system:

$$\sum_{(i,j)\in E} Z_{ij} > 0 \tag{21}$$

$$\sum_{j:(i,j)\in E} Z_{ij} - \sum_{j:(j,i)\in E} Z_{ji} = 0, \qquad i\in V \tag{22}$$

$$Z_{ij} \geq 0, \qquad (i,j)\in E \tag{23}$$

Constraints (22) and (23) define non-negative flow circulations, while constraint (21) says that the total flow must be positive. This implies that there must exist non-zero $\{Z_{ij}\}$ variables which define positive flow circulations. Each such circulation is a cycle of reduction links.

In summary, we have proved

Theorem 4: The improved set of routes result in capacity savings which must happen in the form of cycles of links. On each such cycle, all the links have identical amount of capacity savings.

We already know that non-shortest-path-reproducible routes must be loopy. Now we want to show that the two concepts are actually identical.

Theorem 5: A set of routes is loopy if and only if it is non-shortest-path-reproducible.

Proof: We only need to show the "only if" part. Let $\{P^k : k \in K\}$ be a set of routes which is loopy. Associated with $\{P^k : k \in K\}$, let $\{s_k, t_k, d_k, c_{ij}\}$ be the same as defined previously. Suppose $\{P^k : k \in K\}$ is shortest-path-reproducible. Then there exists a set of positive link weights $\{w_{ij}\}$ under which all the paths in $\{P^k : k \in K\}$ are shortest paths. Let's imagine a network cost structure as follows: The total network cost is the sum of individual link costs. The cost of each link is linearly proportional to the bandwidth consumed. If we view w_{ij} as the unit bandwidth cost on link (i, j), then each shortest path P^k is the least cost path from source s_k to destination t_k. Since the cost function is linearly additive, all the paths $\{P^k : k \in K\}$ collectively form the mathematical optimal solution. So the minimum value of the total cost is $\sum_{(i,j)\in E} w_{ij} c_{ij}$. On the other hand, since $\{P^k : k \in K\}$ is loopy, there exists an improved set of routes $\{Q^k : k \in K\}$ which uses no more bandwidth than $\{P^k : k \in K\}$ and strictly less on at least one link. Let $b_{ij} = \sum_{k\in K:(i,j)\in Q^k} d_k$. Then $b_{ij} \leq c_{ij}$ for every $(i, j) \in E$, and $b_{kl} < c_{kl}$ for some $(k, l) \in E$. The total cost of $\{Q^k : k \in K\}$ is

$$\sum_{(i,j)\in E} w_{ij} b_{ij} = w_{kl} b_{kl} + \sum_{(i,j)!=(k,l)} w_{ij} b_{ij}$$

$$\leq w_{kl} b_{kl} + \sum_{(i,j)!=(k,l)} w_{ij} c_{ij}$$

$$< w_{kl} c_{kl} + \sum_{(i,j)!=(k,l)} w_{ij} c_{ij} = \sum_{(i,j)\in E} w_{ij} c_{ij}$$

where the strict inequality is because $w_{kl} > 0$ and $b_{kl} < c_{kl}$. This contradicts the fact that $\sum_{(i,j)\in E} w_{ij} c_{ij}$ is the minimum value possible. ■

VII. CONCLUSION

In this paper, we proposed a new approach for achieving traffic engineering in the backbones. Instead of relying on the mapping of logical connections to physical links to manage traffic flows in the network, we run IP routing natively over the physical topology, and control the distribution of traffic flows through setting appropriate link weights for shortest-path routing. We have shown that, for any given set of optimal routes, we can always convert them to shortest-paths with respect to a set of positive link weights. And the link weights can be calculated by solving the dual of a linear programming formulation. The result can be extended to a more generic one where any arbitrary set of routes, as long as it is not loopy, can be reproduced as shortest-paths with respect to a set of positive link weights. For any loopy routes, we have shown that there exists a better set of routes that use less bandwidth and that can be reproduced as shortest-paths with respect to a set of positive link weights.

We will conclude the paper with the following remarks.

1. The results discussed in this paper are the properties of routing only, independent of individual demand sizes $\{d_k\}$. We included them in the formulations only for the completeness of modeling. The results are valid for any values of $\{d_k\}$.
2. Traffic splitting across multiple equal cost shortest paths between the same source-destination pair is an important assumption. The notation P^k and Q^k representing paths for demand k is not a precise one, because k can have multiple paths. We used them only for ease of presentation. The results stay the same even if P^k and Q^k represent multiple paths.
3. The properties discussed in this paper are attributes of a set of routes, not individual routes. In particular, a cycle in loopy routing is formed collectively by multiple routes, not by any individual route.
4. The distance relation method used in the proof of lemma 3 only applies to that specific example and other simple cases. A general approach for proving a set of routes non-shortest-path-reproducible should be the one used by theorem 5.
5. A theoretical insight behind all the results is that an optimal solution for any traffic engineering problem can always be converted to a set of shortest-paths with respect to some link weights. The key is the flow conservation constraint (12) and (13) which produce the dual constraint (16). Flow conservation is almost a universal constraint.

ACKNOWLEDGMENTS

The authors would like to thank Bharat Doshi, P. Harshavardhana, Murali Kodialam, Wing Lau and Steve Strickland for their insightful comments.

REFERENCES

[1] D. Awduche, J. Malcolm, M. O'Dell, and J. McManus, "Requirements for traffic engineering over mpls," Internet draft, IETF, April 1998, draft-awduche-mpls-traffic-eng-00.txt.

[2] Yufei Wang and Zheng Wang, "Explciit routing algorithms for internet traffic engineering," *Proceedings of ICCCN'99*, Sept 1999.

[3] J. Moy, "Ospf version 2," Internet rfc, IETF, May 1998, Technical Report RFC 2328.

[4] C. Villamizar, "Ospf optimized multipath," Internet draft, IETF, March 1998, draft-ietf-ospf-omp-00.txt.

[5] C. Hopps, "Analysis of an equal-cost multi-path algorithm," Internet draft, IETF, April 1999, draft-hopps-ecmp-algo-analysis-03.txt.

[6] D. Thaler and C. Hopps, "Multipath issues in unicast and multicast," Internet draft, IETF, April 1999, draft-thaler-multipath-03.txt.

[7] B. Fortz and M. Thorup, "Internet traffic engineering by optimizing ospf weights," *Proceedings of INFOCOM'2000*, March 2000.

[8] G. Nemhauser and L. Wolsey, *Integer and Combinatorial Optimization*, WILEY, 1988.

[9] V. Chvatal, *Linear Programming*, W H Freeman and Company, 1983.