Routing and Protection in GMPLS Networks: From Shortest Paths to Optimized Designs*

A. Elwalid, D. Mitra, I. Saniee†, I. Widjaja
Bell Laboratories, Lucent Technologies
Murray Hill, NJ 07974, USA

Abstract— **Shortest path algorithms such as SPF and CSPF are widely used in online traffic engineering where connections need to be setup one at a time as connection requests arrive sequentially. We propose an approach, called design-based routing (DBR), whereby optimized paths computed offline are used to guide online path setups. Offline path computation in GMPLS networks does not pose a significant challenge since optical core or metro networks typically consist of a few dozen to 100s of nodes compared to 100s to 1000+ nodes in pure data networks. DBR takes advantage of available demand information based on customer prescriptions, traffic projections and historical measurements, to build an *approximate* traffic demand matrix for path optimization. By means of simulation, we perform comparative evaluations of opaque GMPLS networks under static and dynamic connections with different protection modes. The results indicate that DBR outperforms SPF and CSPF under a wide range of operating conditions, and is robust to inaccuracies in the estimation of the traffic demand matrix. We then construct routing schemes with resource management and online measurement. The simulation results indicate that resource management provides an effective way to mitigate greed inherent in CSPF, and online measurement provides an effective way to improve DBR performance when the traffic demand information used in the design of DBR paths is different from the actual traffic demand.**

Keywords—**Traffic engineering, online/offline, static/dynamic, optimization, protection, SPF, CSPF, DBR.**

I. Introduction

A. Background and Motivation

Service Providers are facing the challenge of designing their networks to support increasing customer interest in fast, reliable and quality-differentiated services. The main goals of the service provider are to optimize network resource usage and meet customer service level agreements. Multi-Protocol Label Switching (MPLS) has emerged as a potential enabling technology for traffic engineering (TE) in connection-oriented packet networks [1]. The signaling protocol (e.g., RSVP-TE) provides mechanisms for establishing label switched paths (LSPs) to facilitate explicit routing [2].

Stimulated by recent progress in optical networking, there has also been a growing interest in designing the control plane (i.e., routing and signaling) for the optical layer based on reusing and leveraging existing control-plane protocols. To this end, Generalized MultiProtocol Label Switching (GMPLS), which is an extension of MPLS, is emerging as the candidate control-plane solution for next-generation optical networking [3]. GMPLS requires the definition of an MPLS label to be generalized so that a label can also be encoded as a time slot, a wavelength, or a spatial identifier. By taking advantage of the new definition of a generalized label, it becomes apparently clear that MPLS can also be extended to control and configure a TDM crossconnect, a lambda crossconnect, or a fiber crossconnect.

TE systems in MPLS and GMPLS networks may be based on similar optimization methods, with the main differences being the granularity of traffic demands and link capacities, and the service constraints. Generally, a TE system for connection provisioning in a (G)MPLS-based network involves three components: (1) resource discovery and distribution that can be facilitated by a routing protocol capable of disseminating TE-related information, (2) routing or path design that determines the "optimal" paths/routes (working and possibly with protection paths) according to a given objective, and (3) path setup which is facilitated by a signaling protocol. The second component is typically implementation-dependent, thus allowing vendors to provide added value to their TE capabilities. This paper focuses on the second component.

TE system in a connection-oriented network can be generally classified as online or offline [4]. With an online TE system, connections requests are assumed to arrive one at a time. For each connection request, an online TE system typically computes each path independently without assuming knowledge of future requests. After the path is determined, the connection is usually setup by signaling. The total provisioning time from when a connection request arrives to when it is established may take tens to hundreds of milliseconds. Thus path determination in an online TE system must be relatively fast (on the order of milliseconds). With an offline TE system, typically the aggregate behavior of all connection requests are assumed exactly known a priori before the corresponding paths are computed. Using the topology information, link capacities and traffic demand matrix, a centralized server typically performs global optimization to determine the path for each connection request (see, for example, [5]). Once path design is completed, the connections are typically setup by a network management system. It is obvious that an offline system with global optimization can achieve considerable improvement in resource utilization over an online system. However, the total provisioning time with an offline system typically takes minutes or hours (days or even weeks in practice due manual intervention or various nontechnical issues).

Unlike the situation for the traditional transport networks that are provisioned by means of network management, (G)MPLS-based networks using dynamic routing and signaling allow connections to be established on-demand in response to client requests. Thus online TE in a (G)MPLS-based network is viewed by many as the preferred operational model, since new services requiring rapid provisioning can be readily provided. Online TE also appears attractive in a (G)MPLS network, since the path-

†Corresponding author. Contact iis@bell-labs.com.

*An earlier version of this paper with partial results was presented at the ITC Specialist Workshop in Wuerzburg, Germany, July 22-24, 2002.

design engine that is integrated at each node allows automation of the path setup process and minimizes configuration steps. In this paper, we consider an online TE system for a connection-oriented network such as an opaque (G)MPLS-based network whereby each connection is provisioned one at a time as a new connection request arrives.

B. Review of Online TE Approaches

The traditional datagram IP network that uses shortest path first (SPF) routing to determine a path to each destination can be principally applied to a connection-oriented network. The path computation is oblivious to the loading of each link, and hence may easily cause over-utilization of some links and under-utilization of others. SPF is also inflexible for multi-service networks since connections with different QoS's cannot be routed through different paths.

A common enhancement to SPF that is widely implemented by many vendors involves extensions to the IGP (e.g., OSPF or IS-IS) so that the available bandwidth together with other relevant constraints are also flooded throughout the network. Path design is performed by constraint shortest path first (CSPF) whose main idea is to first prune all links that do not meet some specified constraints or the bandwidth requirement of a new request. Then shortest path computation is run on the pruned topology. In practice, the edge weight W_e used in the shortest path computation is inversely proportional to the capacity; that is, $W_e = 1/C_e$ for link e.

Recently, an improved version of an online system called Minimum Interference Routing Algorithm (MIRA) was proposed [6]. Given a new request from nodes s_i to d_i with bandwidth b_i, the main idea of MIRA is to compute the maximum flows, the "critical" links and the weights for all source-destination pairs excluding the pair (s_i, d_i). Then all links with available bandwidth less than b_i is pruned, and shortest path routing is run on the pruned topology using the pre-computed weights. One drawback with MIRA is its high computation cost, which is undesirable in an online TE system where connection requests have to be established as quickly as possible. Moreover, lack of knowledge of loading information may result in poor performance for certain topologies.

C. Our Contributions

In this paper, we propose an online TE system where a connection request typically uses a pre-designed path that is already computed by an offline route computation server, called a **designed-based routing (DBR)** server, if the pre-designed path is deemed suitable. Otherwise, the connection request may use a path computed online (e.g., via CSPF). We study the problem of path design taking into account the traffic demand information between each pair of nodes. Unlike approaches used in the offline TE system where the aggregate traffic demand information is assumed to be known with a high degree of precision, we cannot make such an assumption with online TE, since connection requests arrive one at a time and there is no knowledge of future requests. Nevertheless, it is still reasonable to assume that some estimate of the traffic demand information is available based on network planning outputs, customer prescriptions, traffic projections, and historical measurements.

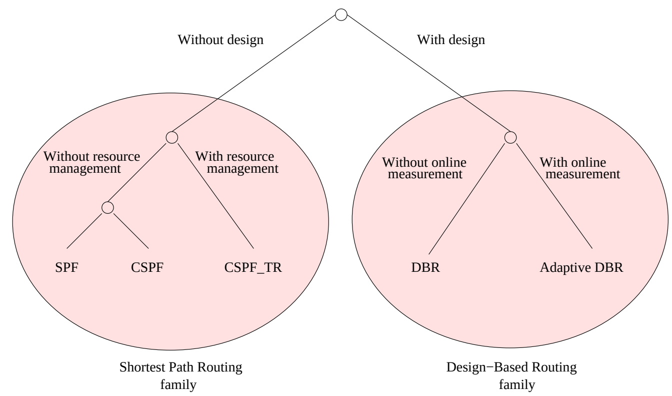


Fig. 1. Routing classification.

We investigate the impact of the accuracy of the traffic demand information on the performance of the DBR-based TE under a variety of conditions. In particular, we consider *static connections* where established connections stay permanently, and *dynamic connections* where connections arrive and depart at random. We also consider survivable networks with different protection modes: (1) no protection, (2) dedicated path protection, and (3) shared path protection. Our study reveals that even imprecise or approximate information on the traffic demand matrix can improve the resource requirements of the network compared to the case where such information is not used as in SPF/CSPF. We introduce resource management concepts to both SPF and CSPF, as well as to DBR. Specifically, the concept of trunk reservation, which is well-known in telephony, and its generalization, virtual partitioning, are used to enforce controlled resource sharing. Finally, we show that DBR with online measurement, called **Adaptive DBR**, can be used to provide further performance gain.

D. Outline of The Paper

In the next section, we describe an online path setup model that relies on offline path computation possibly assisted with online path computation. In Sec. III, we formulate the basic DBR optimization problems with accurate and inaccurate traffic demand information under static and dynamic connection models. In Sec. IV, we present simulation results under a variety of conditions, and compare DBR with SPF and CSPF. In Sec. V, we present routing techniques with resource management. Finally, we present our conclusions in Sec. VI.

II. From Shortest Path Routing to Design-Based Routing

In Fig. 1 we give a taxonomy of routing algorithms for traffic engineering, which are covered in this paper. This taxonomy is not intended to be exhaustive; however, it covers a broad array of schemes that make use of offline and online measurements, local and global optimizations, resource management schemes and signaling requirements. At the highest level the algorithms may be classified according to whether offline design optimization is used. The design-based routing family is distinguished by the use of an estimated traffic demand matrix and global path optimization. The shortest-path routing family involves certain

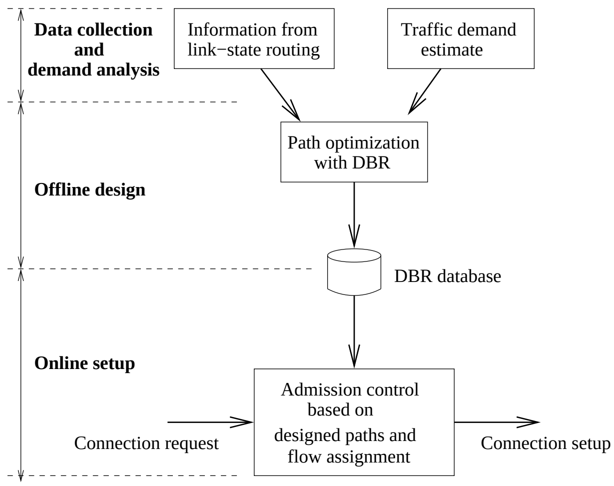


Fig. 2. Components for basic DBR framework.

shortest path computations, with or without resource management, for one source-destination pair at a time. The routing algorithms in the shortest path family include the standard SPF and CSPF algorithms. In this family, we also present an extension with resource management, called CSPF_TR, which is a novel CSPF algorithm with trunk reservation (discussed in Sec. V).

The routing algorithms in the design-based routing family may be further grouped based on their use of online measurements. In DBR (path design discussed in Sec. III and performance in Sec. IV) optimally precomputed paths based on approximate knowledge of the demand matrix are used in setting all source-destination paths online. In the hybrid scheme, called Adaptive DBR, online measurements are used for path selection and admission control. Adaptive DBR is discussed in Sec. V.

Fig. 2 shows the main components for the DBR framework. The path optimization component collects input information, such as the network topology, link capacities, traffic demand estimate, and other relevant information such as constraints. Based on this input information, the DBR optimization engine computes all possible paths offline and stores the results in a database for future use. The admission control component makes use of the optimization results stored in a database. In basic DBR, admission control is only applied on a DBR path. In adaptive DBR, admission control is applied on a DBR as well as alternate (e.g., CSPF) paths.

III. Path Design with DBR

A. *Definitions and Notations*

We consider a network topology represented by a graph $G(V, E)$ where V denotes the set of vertices (nodes) and E denotes the set of edges (links) and each edge $e \in E$ has a capacity of C_e.

The length may be a function of actual distance or other salient parameters used in the optimization criteria. Our interest in this paper is to focus on the optical transport network, where the edges and the connections are typically bi-directional. Demands, in terms of the number of connections, vary from one source-destination pair to another, but each connection is a lightpath of the same size (i.e., one lambda).

Let K denote the set of all node pairs in the network. For static connections, the traffic demand d_k is expressed in terms of the total number of connections that need to be routed for node pair $k \in K$. For dynamic connections, the traffic demand δ_k is the offered load in Erlangs for node pair $k \in K$. Let P_k denote the set of all paths for node pair k, and let P denote the set of all paths for all node pairs; that is, $P = \bigcup_k P_k$. The output of path design determines the amount of demand (flow) that needs to be assigned to each path. Let the quantity x_p represent the amount of demand in units of lambdas assigned to path $p \in P$. Finally, let Q_e denote the set of all paths that traverse edge e. The subsequent sections present DBR formulations for the case with no protection.

B. *DBR with Accurate Static Demand Information*

The DBR server determines all end-to-end paths concurrently and optimally using the network topology, edge capacities, traffic demand matrix and possibly other constraints. The path design objective is to minimize the total bandwidth-length product. Furthermore, the optimization problem is subject to meeting the designated end-to-end demands while not exceeding the edge capacities. If all connections cannot be established in the existing network due the constraints implied by the fixed edges capacities, the problem needs to be solved in two stages. In stage 1, the goal is to maximize the total amount of the carried demand. In stage 2, optimal routing is performed for the level of demand derived in stage 1. A related two-stage formulation is given in [5]. We formulate stage-1 as follows:

$\mathbf{P1}(\theta)$

$$\max_{x_p, \theta_p} \sum_k \theta_k d_k \tag{1}$$

subject to

$$\sum_{p \in P_k} x_p \geq \theta_k d_k, \quad \forall k \in K \tag{2}$$

$$\sum_{p \in Q_e} x_p \leq C_e, \quad \forall e \in E \tag{3}$$

$$x_p \geq 0, \quad \forall p \in P \tag{4}$$

$$0 \leq \theta_k \leq 1, \quad \forall k \in K \tag{5}$$

Having obtained the threshold θ_k^* that maximizes the number of connections that can be accepted, we then adjust the traffic demand for each node pair k as $d_k \leftarrow \theta_k^* d_k$. (We note that stage-1 optimization can be performed with a common demand multiplier for all node-pairs, i.e., $\theta_k = \theta, \forall k$, which will ensure a fair admission policy amongst all node-pair demands at the expense of reducing the overall carried traffic.) Note that the set of all paths for all node pairs have been computed and qualified at the outset. For example, for each node pair, the set of possible paths may consist of all paths with length less than the length of the shortest path plus some threshold value. In general, using very long paths for multicommodity flow problems has diminishing utility.

The stage-2 optimization problem is as follows:

$\mathbf{Q1}(\lambda * \mathbf{length})$

$$\min_{x_p,\lambda_e} \sum_{e\in E} \lambda_e \tag{6}$$

subject to

$$\sum_{p\in P_k} x_p \geq d_k, \quad \forall k \in K \tag{7}$$

$$\sum_{p\in Q_e} x_p \leq \lambda_e, \quad \forall e \in E \tag{8}$$

$$\lambda_e \leq C_e, \quad \forall e \in E \tag{9}$$

$$x_p \geq 0, \quad \forall p \in P \tag{10}$$

It is not hard to see that the total demand met in Q1 is the same as the objective value of problem P1, however, the output of Q1 provides the assignments of demands to paths for each node pair (i.e., x_p) with least cost packing. It is also readily seen that when the edge capacities are large, the effect of performing this optimization is simply to let each node pair use the shortest path. In this case, no concurrent optimization of all demands is necessary. Indeed, Q1 which is minimization of the total bandwidth-length product subject to (edge) capacity constraints can be thought of as the simplest non-trivial extension of the shortest path routing when there are capacity constraints on the edges.

The output of this ILP, or its rounded LP relaxation, gives the paths and their allocated flows for each node pair. Clearly, the total bandwidth-length product of any other scheme is at best equal to the optimum value obtained above for the prescribed demand matrix. In particular, this is true for any online TE system, where each connection request is processed one at a time at each arrival, such as in SPF and CSPF.

In SPF static edge weights are used for path computation, where the weight could be a function of distance, inverse of edge capacity ($1/C_e$), or other parameters. Arriving requests are continuously assigned to the computed paths although some edges may become saturated, resulting in blocking, while other edges may still be underutilized. To alleviate this lack of load balancing inherent in SPF, CSPF can use dynamic weights, such as the residual capacity (e.g., $1/(C_e - \lambda_e)$) that reflects the current availability of the edge resources. CSPF clearly dominates SPF in many circumstances, but its utility vanishes rapidly as the total demand in the network increases, a scenario that we discuss and characterize in Sec. IV-B. To summarize, if $v(policy)$ denotes the optimal objective (carried demand) of a routing policy, then

Observation 1: $v(DBR) \geq v(CSPF)$

C. Value of Traffic Demand Information

It may be argued that the reason for the lack of optimality of online path computation is purely due to the non-optimal arrival sequence of connection requests in time. That is, if one could examine all possible permutations of arrival sequences, the optimal solution will simply follow by applying SPF/CSPF to that specific sequence. To see why solutions based on the optimal arrival sequence cannot match the solution to Q1, consider the simple example shown in Fig. 3. Here, each edge has capacity 1 unit (lambda) and length, or cost, as indicated in the figure. There are two connection requests between A and F to be routed. Observe that for SPF and CSPF, the paths are (A, C, E, F) and one of (A, B, F) or (A, D, F). Thus the total cost for SPF or CSPF is 14. However, the optimal paths would take (A, B, E, F) and (A, C, D, F), resulting in the total cost of 6. Furthermore, notice that no permutation of the arrival sequences in SPF or CSPF will result in the optimal paths. We conclude that the concurrency of path assignments afforded by optimization, is a stronger factor for the best use of network resources than the order of arrivals or load balancing (CSPF). In other words, use of the traffic demand information and concurrent assignment of demands to routes enable better packing of the demands in the network in ways that even the best sequential solution, with or without load balancing, cannot match.

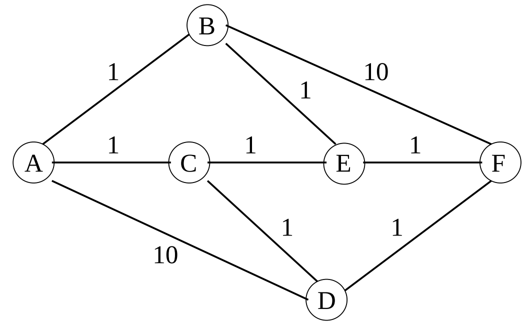


Fig. 3. Example topology.

This concurrency in the design of paths (as enabled by DBR) turns out to be an even more important factor for the effective use of network resources than the network load itself. This is true whether we interpret load in the static case with or without perfect information or in the dynamic case. We illustrate this point further in Sec. IV. First we need to define DBR in the context of uncertain traffic demand and dynamic connections.

D. DBR with Uncertain Static Demand Information

It is commonly argued that exact end-to-end demand information is almost never available in practice. Often, this argument is used erroneously to conclude that the class of purely online path computation schemes is the only suitable solution in practice, since this class does not require information about the end-to-end traffic demand matrix. We will show in Sec. IV that even demand information with considerable uncertainty can be utilized to give a better path design than can be computed by SPF or CSPF. To explore the uncertainty issue further, we assume that $\tilde{d}_k \in \mathcal{N}(\bar{d}_k, \sigma_k)$ and reformulate P1/Q1 when end-to-end demands are no longer fixed but are random variables. The direct translation of the stochastic variant of $P1$ using point estimates of the random variable $d_k, \forall k$ would not lead to the correct optimization model, however. This is because the *carried* load in a stochastic setting is the minimum of (i) the provisioned bandwidth and (ii) the offered load; in other words, $min\{\sum_{p\in P_k} x_p, \sum_k \theta_k \tilde{d}_k\}$. Condition set (2) with point estimates for $\bar{d}_k$ only ensure that the mean provisioned demand exceeds the offered load. We therefore substitute this condition with a more stringent requirement that the offered load is met with a high probability. Thus, stage 1 optimization in this setting becomes,

$\tilde{\mathbf{P}}\mathbf{1}(\theta)$

$$\max_{x_p,\theta_k} E\{\sum_k \theta_k d_k\} \tag{11}$$

subject to

$$Pr\{\sum_{p\in P_k} x_p \geq \theta_k d_k\} \geq 1-\epsilon_k, \quad \forall k \in K \tag{12}$$

$$\sum_{p\in Q_e} x_p \leq C_e, \quad \forall e \in E \tag{13}$$

$$x_p \geq 0, \quad \forall p \in P \tag{14}$$

$$0 \leq \theta_k \leq 1, \quad \forall k \in K \tag{15}$$

We note that we have replaced constraint set (2) with (12) which ensure, with arbitrarily high probability $1-\epsilon_k$, that the optimized fraction of the offered load is met. We note also that because of this constraint, the carried load, namely $min\{\sum_{p\in P_k} x_p, \sum_k \theta_k \tilde{d}_k\}$ is equal to $\sum_k \theta_k \bar{d}_k$ which is the objective maximized. Given that $d_k \sim N(\bar{d}_k, \sigma_k)\ \forall k$, we can simplify the constraint set (12) as

$$\sum_{p\in P_k} x_p \geq \theta_k(\bar{d}_k + \eta_k\sigma_k) \tag{16}$$

where

$$\eta_k = \sqrt{2}\,\mathrm{erf}^{-1}(1-2\epsilon_k) \tag{17}$$

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-y^2}dy \tag{18}$$

We observe that over a (long) series of the realizations of the random demands, the above procedure will outperform any online scheme with the same requirements. In particular:

Observation 2: $E\{v(DBR)\} \geq E\{v(CSPF)\}$

As in the deterministic formulation, better packing of the flows x_p to paths p can be obtained from the equivalent model to $Q1$ obtaining θ^* from $P2$:

$\mathbf{Q2}(\lambda * \mathbf{length})$

$$\min_{x_p,\lambda_p} \sum_{e\in E} \lambda_e l_e \tag{19}$$

subject to

$$\sum_{p\in P_k} x_p \geq \theta_k^*(\bar{d}_k + \eta_k\sigma_k) \quad \forall k \in K \tag{20}$$

$$\sum_{p\in Q_e} x_p \leq \lambda_e, \quad \forall e \in E \tag{21}$$

$$\lambda_e \leq C_e, \quad \forall e \in E \tag{22}$$

$$x_p \geq 0, \quad \forall p \in P \tag{23}$$

We note that $Q2$ is a deterministic optimization problem and is feasible due to the rescaling of demands derived from $P2$.

E. DBR with Dynamic Connections

When connections are dynamic, there is always a chance of blocking even with generous dimensioning of the paths. In this case, the DBR objective is to minimize network resource usage subject to meeting the traffic demand up to a given level of blocking probability. Variants of this problem have been considered in [7] where three approaches were proposed: complete partitioning, complete sharing and virtual partitioning. For simplicity and clarity of the presentation, we only discuss complete partitioning in this paper. However, DBR is naturally extensible to all three approaches.

Assume that connection requests between node pair k arrive according to a Poisson process with offered load δ_k Erlangs, and a logical path with capacity L_k $(=\sum_{p\in P_k} x_p)$ lambdas is assigned for node pair k. Then the end-to-end blocking probability for node pair k is given by the Erlang B formula

$$B(\delta_k, L_k) = \frac{\delta_k/L_k!}{\sum_{i=0}^{N_k}\delta_i/i!} \tag{24}$$

The optimization procedure for DBR with dynamic connections can be formulated as follows:

$\mathbf{Q3}(\lambda * \mathbf{length})$

$$min \sum_{e\in E}\lambda_e \tag{25}$$

subject to

$$B(\delta_k, L_k) \leq \beta, \quad \forall k \in K \tag{26}$$

$$\sum_{p\in P_k} x_p \geq L_k, \quad \forall k \in K \tag{27}$$

$$\sum_{p\in Q_e} x_p \leq \lambda_e, \quad \forall e \in E \tag{28}$$

$$\lambda_e \leq C_e, \quad \forall e \in E \tag{29}$$

$$x_p \geq 0, \quad \forall p \in P \tag{30}$$

Note that the non-linear constraint (26) can be eliminated by first solving for L_k. In this case, the optimization problem can be simplified by using the following procedure:

$\mathbf{Q4}(\lambda * \mathbf{length})$
1. Compute $L_k^* = \mathrm{argmin}_{L_k}\{B(\delta_k, L_k) \leq \beta\}, \forall k \in K$.
2. Solve Q1 with $d_k \leftarrow L_k^*, \forall k \in K$ (using pre-processing P1 if necessary).

The formulation of DBR with protection (dedicated or shared) follows similar approaches presented in the preceding discussions. For details of these formulations and the computations of link capacities for the case of accurate static demand, see [8].

IV. Performance Investigation

In this section, we assume an optical transport network that supports dynamic routing (e.g., OSPF-TE) to distribute link-state information and a signaling protocol (e.g., RSVP-TE) to setup and teardown connections on demands. This paper assumes an opaque optical network (extensions to a transparent or semi-transparent optical network will be discussed elsewhere).

One main objective is to compare DBR with other path design methods under different conditions. In particular, we adopt two online path design methods that are widely deployed, namely SPF and CSPF. SPF selects a path that has the least cost. The most common implementation of CSPF is similar to that using SPF, except that links that do not meet the bandwidth requirement or other constraints are first pruned. Then SPF is run on the pruned topology.

We consider two connection models: static and dynamic. In the static model, connection requests arrive one-by-one. Each request is tagged with the identifiers of the ingress node and the egress node, and the task is to setup the connection (or lightpath) between the ingress and egress nodes. If the connection is successfully routed, it will stay indefinitely (i.e., the holding time is infinite). If a connection request is blocked, we assume that the blocked request is lost forever. Note that the static model is similar to the Permanent Virtual Circuit (PVC) model in ATM networks.

In the dynamic model, connection requests arrive at random and the holding time of a connection is finite. A blocked connection request is assumed lost forever. The dynamic model may be more suited when clients of an optical network are networks that use bandwidth on demand, which can be used for purposes of traffic engineering in the client network. In such environments, connection establishments and tear-downs are initiated by client nodes on demand through Optical User-to-Network Interface (O-UNI). Note that the dynamic model is similar to the Switched Virtual Circuit (SVC) model in ATM networks.

Optical networks may be operated under different protection scenarios depending on service requirements. In this paper, we consider three path protection scenarios for all routing schemes used:

- no protection,
- 1+1 or 1:1 dedicated path protection, and
- 1:1 shared path protection.

A. Network Model

In our simulation experiment, we consider the topology of a generic U.S. optical network, as shown in Fig. 4. The network has 30 nodes and 38 links with each link assumed to be bidirectional. For static model, the input to the simulator is driven by the traffic demand matrix $D = [d_{ij}]$, for each $i, j \in V$ (note that the end-to-end demand is previously defined between a pair of nodes $k = (i, j)$). Note that the demand is symmetric; that is $d_{ij} = d_{ji}$. For dynamic model, $D = [\delta_{ij}]$. In the subsequent simulation results, we first consider the practical case where the network is initially designed so that the link capacities are optimized (e.g., via Q1). Later we consider the case where the link capacities may have arbitrary values (i.e., undesigned network with badly dimensioned links).

B. No Protection

This section investigates the case where connections are not protected. We first consider the static connection model where the traffic demand matrix is assumed to be accurate for DBR. The network is initially unloaded, and a total of $\sum_{i,j} d_{ij} = 3495$ connection requests are to be provisioned. To emulate the online operational model, we assume that connection requests arrive one by one in order.

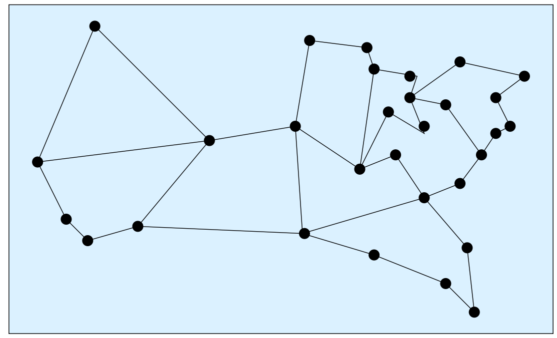


Fig. 4. Generic U.S. network topology.

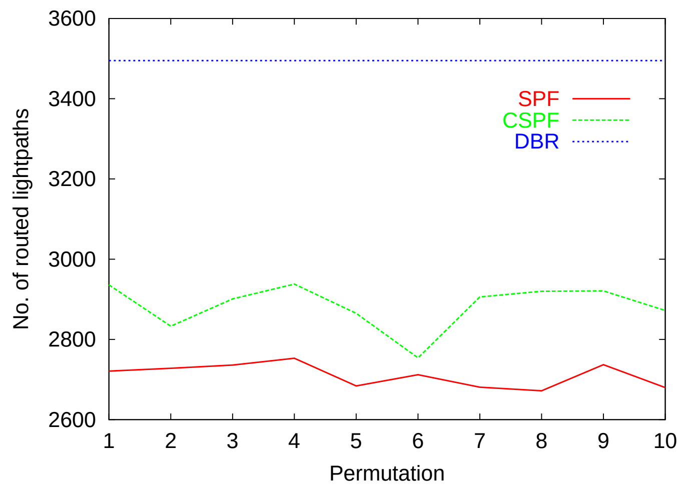


Fig. 5. Static case with perfect traffi c demand.

Each request is provisioned until all requests have been processed. The performance measure of interest here is the total number of connections (lightpaths) that are successfully established (routed). Because the order of requests determines the number of connections that can be successfully established, we perform a number of trials where each trail corresponds to a permutation of requests. Fig. 5 compares the performance of three path computation approaches (SPF, CSPF, and DBR) with the OSPF weight set to $W_e = 1/C_e$. In this example, DBR is able to route all connection requests. As expected, CSPF performs better than SPF. CSPF can route about 80% of all requests [1], while SPF can route about 75% of them.

Next, we consider the case where the traffic demand matrix used in DBR can be inaccurate. To model the inaccuracies, we independently perturb each original traffic demand, d_k, by a gaussian noise $\mathcal{N}(0, \sigma)$, properly truncated and rounded, so that the value of the perturbed traffic demand is a non-negative integer. The degree of inaccuracies in the traffic demand is determined by the coefficient of variation $c = \sigma_k/d_k$. The larger the value of the coefficient of variation, the worse the original traffic demand estimate is.

We compare the performance of DBR relative to another path

[1] We note that the performance of CSPF improves slightly (by about 3% more requests routed in our case) if the OSPF weight is $W_e = \frac{C_e}{C_e - \lambda_e}$.

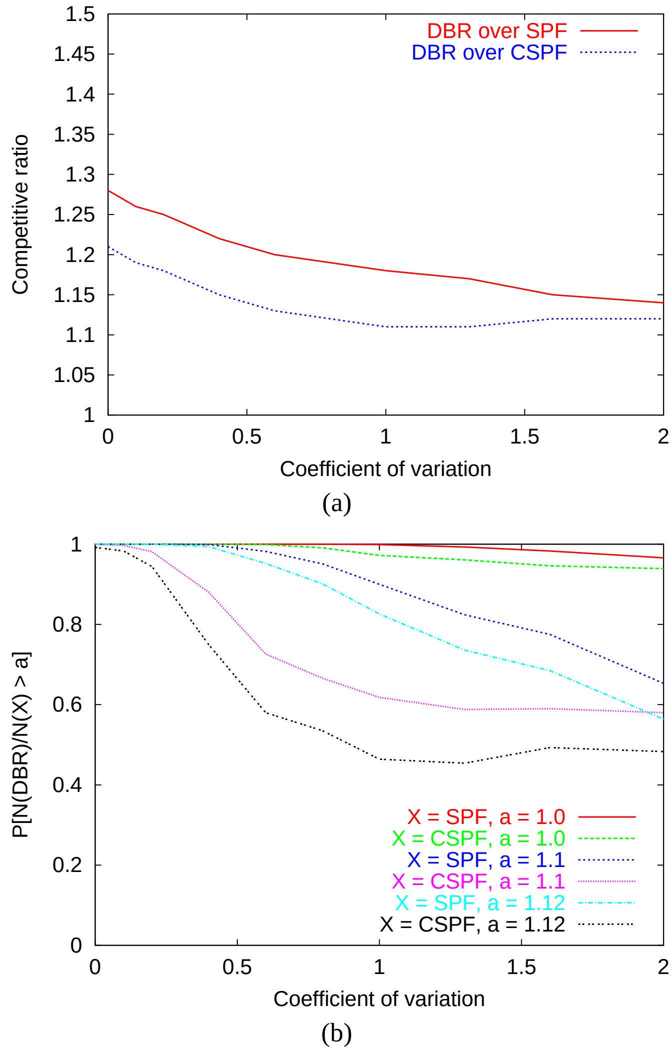


Fig. 6. Static case with inaccurate traffi c demand: (a) competitive ratio, (b) probability that DBR outperforms X for a random order of requests.

computation approach by a measure called *competitive ratio* adopted from [9]. Let $N(X)$ denote the number of connections that are successfully routed via approach X for a given order of requests. Then, the competitive ratio is

$$CR(DBR/X) = \frac{E[N(DBR)]}{E[N(X)]} \tag{31}$$

Fig. 6(a) plots the competitive ratio against the coefficient of variation for SPF and CSPF. Here, the expected value is obtained by averaging the results of 500 independent trials, where each trial corresponds to a particular order of requests. The point where $c = 0$ corresponds to the case where the traffic demand is perfectly known a priori. This ideal operating point reveals that DBR can route 20% more connections than CSPF, and about 25% more connections than SPF. As expected, the advantage of DBR diminishes as the traffic demand estimate becomes less accurate (i.e., c increases). However, notice that DBR still outperforms SPF and CSPF even when c becomes very large, indicating the robustness of DBR against errors in traffic demand estimates.

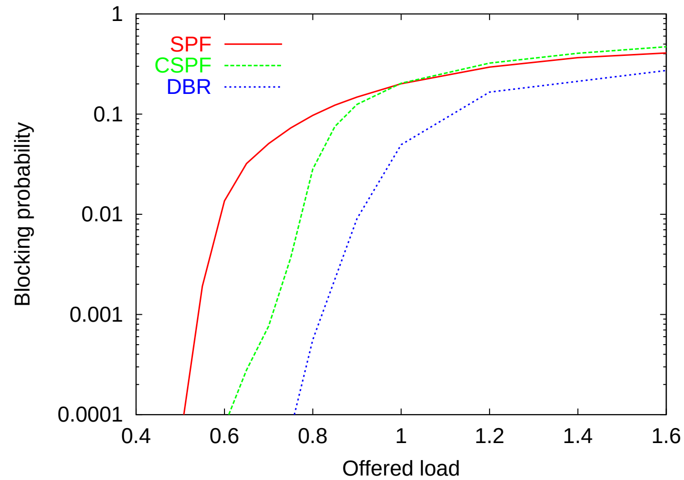


Fig. 7. Dynamic case.

Although competitive ratio gives information on the relative performance of DBR over SPF or CSPF on average, it does not describe the relative performance for a particular order. Another interesting measure is $P[N(DBR)/N(X) > a]$, which gives the probability that DBR routes $(1 - a)100\%$ more connections than SPF/CSPF for a random order of requests. Fig. 6(b) shows that the probability that DBR outperforms SPF or CSPF is close to 1. Even when DBR has to route more than 10% more connections than SPF or CSPF, the probability that is attainable is still greater than 0.8 when $c < 1$.

We now consider the dynamic model without protection. We assume that connection requests arrive according to a Poisson process, and that the holding time of a connection is exponentially distributed. Fig. 7 compares SPF, CSPF, and DBR with respect to the path blocking probability as the normalized total offered load is varied. As can be seen from the figure, DBR reduces the blocking probability by at least an order of magnitude less than others under moderate load. Observe an interesting case where SPF outperforms CSPF when the network becomes congested. The reason is that CSPF greedily chooses longer and longer paths in an attempt to satisfy the current requests without regard to any possible future requests when the load increases. In Sec. V we show how CSPF can be made less greedy by applying trunk reservation.

C. Dedicated Protection

In this section, we turn our attention to the scenario where the optical network employs dedicated path protection for each node pair. Both 1+1 and 1:1 path protections are applicable in this context. Path protection requires a path computation algorithm for two disjoint paths between each node pair. To this end, we adopt the disjoint-path computation algorithm due to [10] for SPF and CSPF. With dedicated protection, the protection path is allocated with the same amount of bandwidth as the corresponding working path.

The performance results with dedicated protection for static and dynamic models are displayed in Fig. 8. DBR again outperforms CSPF which in turn outperforms SPF. Looking into more detail, we note from comparing Fig. 8 with Fig. 6(a), that CSPF

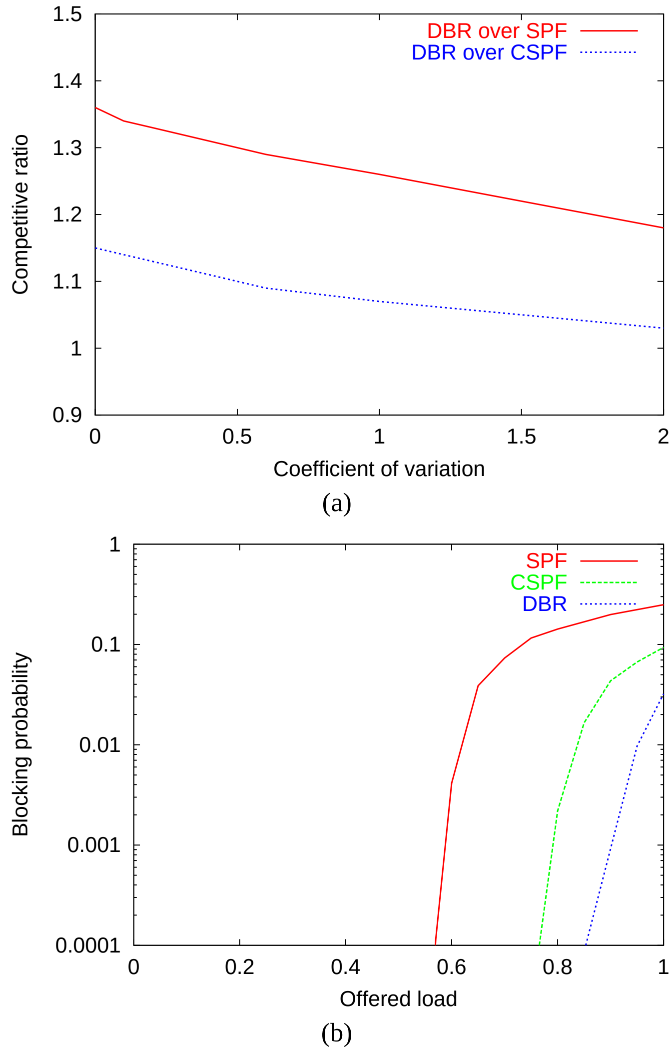


Fig. 8. Dedicated protection: (a) Static case with inaccurate traffic demand, (b) dynamic case.

continues to outperform SPF even in congested state under dedicated protection. This is attributed to the fact that CSPF is less able to choose long disjoint paths than long single paths. This is because CSPF has a richer set of two-disjoint paths, and can take advantage of this fact without incurring the penalty for being too greedy.

D. Shared Protection

This section deals with path protection with shared bandwidth allocation. Let W_{ij} denote the working bandwidth that uses both link i and link j. Let Q_{ij} denote the overflow to link i when link j fails. The protection bandwidth that is needed on link i for any single failure is given by $Q_i = min\{0, max_{j \in E}\{Q_{ij} - W_{ij}\}\}$, where E is the set of all links. When a new connection request arrives, the ingress node would first try to establish the working path R_w before the protection path R_p. The Connection Admission Control (CAC) performs the usual bandwidth allocation for the working path, except that it needs to update $W_{ij} \leftarrow W_{ij} + 1, j \in R_w$, on link i if the request is accepted. For the protection path, the CAC on link $i \in R_p$ performs the following:

if $Q_i > max_{j \in R_w}\{Q_{ij} - W_{ij}\}$
    $\hat{Q}i \leftarrow Q_i$
else
    $\hat{Q}_i \leftarrow Q_i + 1$
if $\hat{Q}i - Q_i \leq$ available lambdas,
    accept the new request
    $Q_{ij} \leftarrow Q_{ij} + 1, j \in R_w$
    $Q_i \leftarrow \hat{Q}_i$
else
    reject the new request

When a connection departs, the CAC on link i performs the following:

if the connection is a working path
    $W_{ij} \leftarrow W_{ij} - 1, j \in R_w$
else
    $Q_{ij} \leftarrow Q_{ij} - 1, j \in R_w$
    $Q_i \leftarrow min\{0, max_{j \in L}\{Q_{ij} - W_{ij}\}\}$

Note that the above algorithm can be easily modified if a new request needs an arbitrary amount of bandwidth.

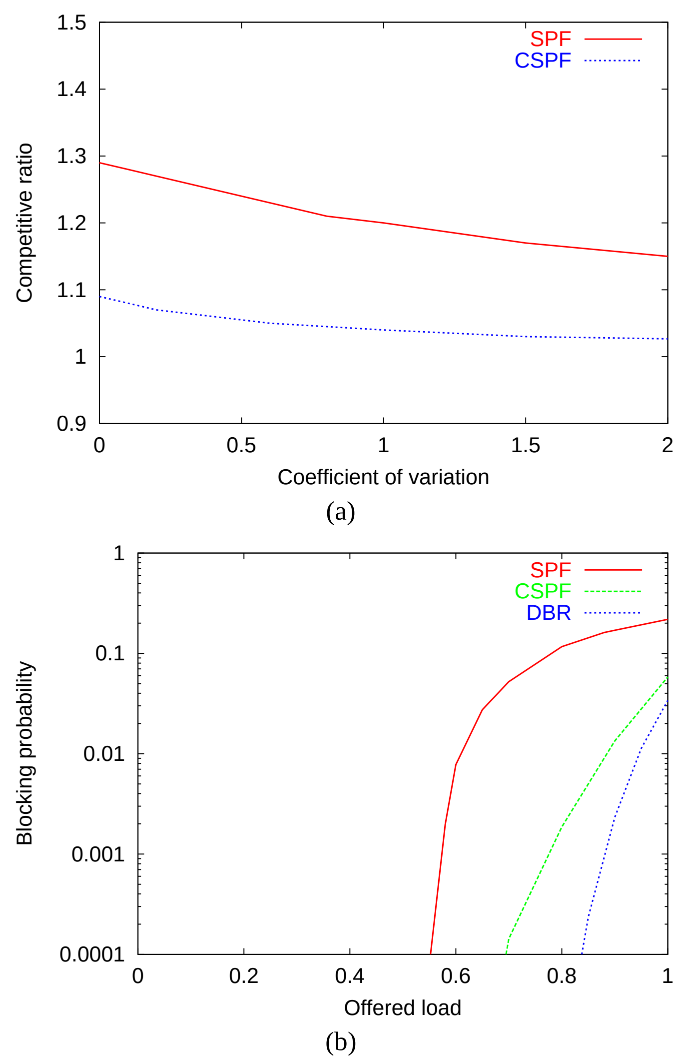


Fig. 9. Shared protection: (a) Static case with inaccurate traffic demand, (b) dynamic case.

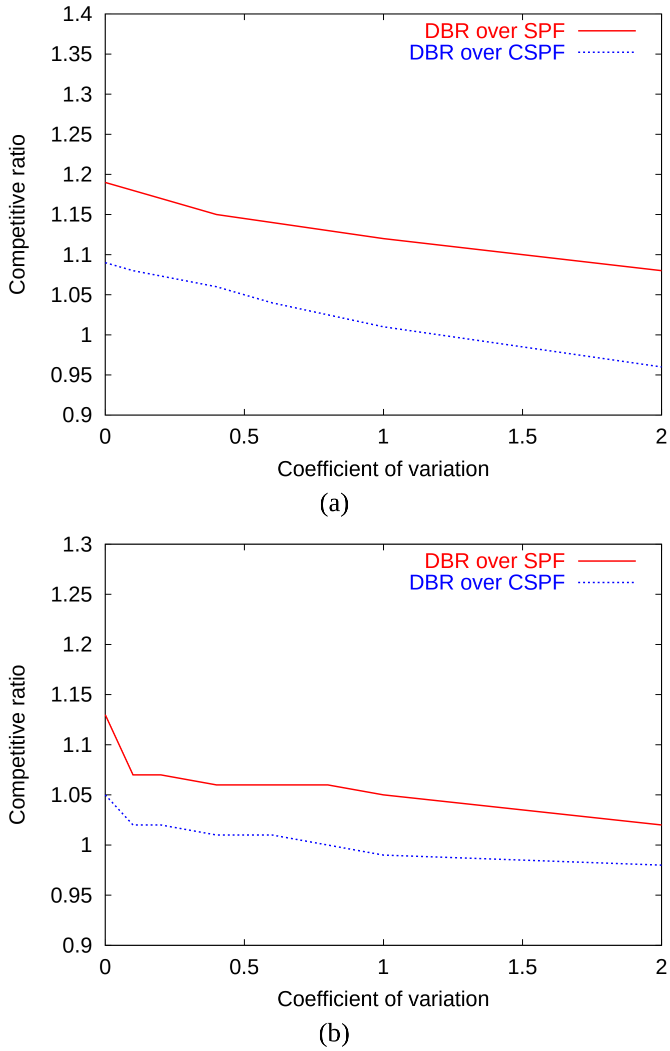


Fig. 10. Undesigned links: (a) Fixed capacity, (b) random capacity.

Fig. 9 shows the performance results with shared path protection for static and dynamic models. Again, we observe that DBR outperforms CSPF and SPF under a wide range of traffic conditions. Moreover, since DBR can take advantage of the shared bandwidth pools in the path optimization, the performance of DBR with shared protection relatively improves compared with dedicated protection.

E. DBR for Networks with Arbitrary Link Capacities

The preceding sections consider the typical case where the network is appropriately designed to minimize network cost given the anticipated demand. In particular, given fixed nodal locations, network design may involve optimization of link capacities.

This section uses the same network topology as before, but assumes that the link capacities have not been properly dimensioned. Given that the total anticipated demands are 3495, the average shortest path is about 4 hops and the network has 38 links, we expect each link capacity to be at least 400 ($\sim$ 3495*4/38) lambdas if the network is a uniform grid.

We consider two examples where link capacities have not been properly dimensioned: 1) *fixed capacity* where each link has a fixed capacity, and 2) *random capacity* where each link has a randomly assigned capacity. In the simulation, we use 600 lambdas for the fixed capacity, and a uniform random capacity ranging from 200 to 700 lambdas.

Fig. 10 shows the competitive ratio for both examples for the network without protection. The competitive ratio is computed by averaging as described before (in Sec.IV-B). With random capacity, the competitive ratio is obtained by further averaging over several independent network instances where each instance corresponds to a certain set of random link capacities. When the traffic demand information is accurate, observe that DBR outperforms SPF and CSPF even if the link capacities are arbitrary. However, the value of DBR diminishes more rapidly as the coefficient of variation increases compared to the case where the network is carefully dimensioned (see Fig.6). This is due to the fact that relatively large excess capacity (approximately 20 %) that is randomly scattered in the network can be further taken advantage of by CSPF. Moreover, since the excess capacity for the network with random link capacities tends to be more than that with fixed link capacities, DBR with random link capacities does not perform quite as well as DBR with fixed link capacities.

V. Routing with Resource Management

This section presents an enhancement to the basic shortest path routing and design-based routing through the application of trunk reservation for managing bandwidth resources during congestion. It is well known that trunk reservation may be applied to two-hop paths to reduce resource usage. Here the trunk reservation concept is generalized to an arbitrary network with CSPF or DBR routing.

We first describe a scheme called CSPF with trunk reservation (CSPF_TR). We say that a network is pruned with trunk reservation level r (i.e., $TR = r$) if any link with available bandwidth less than or equal to r is removed. Let $l_{min}(s,d)$ denote the minimum path length from source s to destination d computed through SPF. Then, the CSPF_TR scheme is described as follows:

```
CSPF_TR:
prune with TR=0
compute shortest path on the pruned network
if the resulting path lengh = l_min(s,d)
   setup the connection
else
   prune with TR=r
   compute shortest path on the pruned network
   setup the the connection if possible
```

Fig. 11 compares various shortest path routing schemes for a dynamic case with no protection and accurate traffic demand estimate. Here the trunk reservation level r is 2 lambdas. Note that trunk reservation prevents CSPF_TR from being greedy under heavy load. Under light load, there is a small penalty for doing trunk reservation. One possible enhancement is to disable trunk reservation under moderate load and to apply it when the load reaches a certain critical level.

We now present an enhancement to the basic DBR scheme, called adaptive DBR, which is based on [11]. Adaptive DBR is intended to deal with inaccuracies in traffic demand forecast.

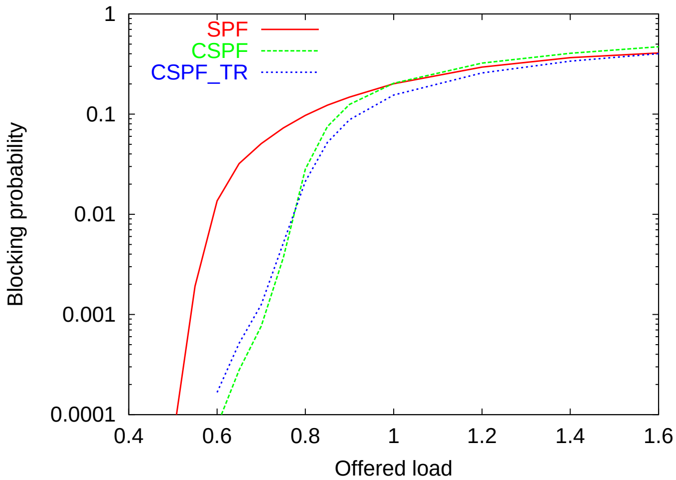


Fig. 11. Comparisons of shortest path routing.

The main idea of adaptive DBR is to use the current measured load for a given ingress-egress node pair as a surrogate to DBR. When the current measured load for a given node pair is lower than that used in the design of the DBR path, a connection request for the node pair will attempt to use the path given by DBR. On the other hand, when the current measured load is higher than that used in the design, a connection request can only use the path computed by DBR provided that the DBR path passes the trunk reservation test. In each case, a failed DBR path can fall back to CSPF with trunk reservation.

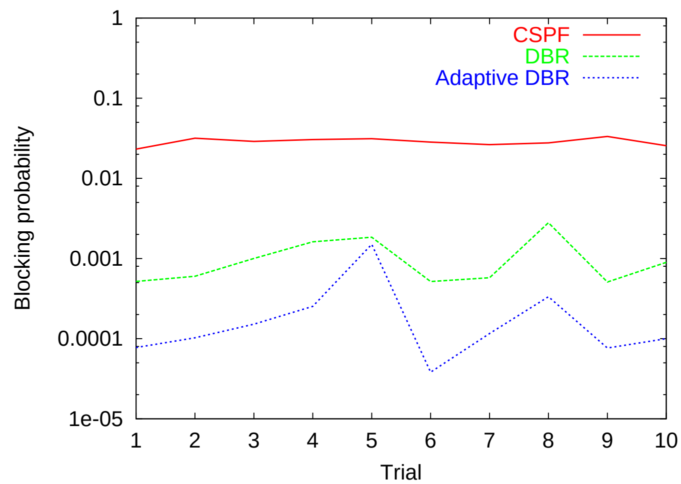
We note that adaptive DBR requires an online measurement component at each ingress node and an online path-selection component. For simplicity, we focus on the dynamic model with no protection. Let us first define the relevant notations. Let v_{sd} be the designed load between s and d that is used in the computation of DBR paths. Let $\hat{v}_{sd}$ be the current measured load between s and d.

Path selection for adaptive DBR is as follows:

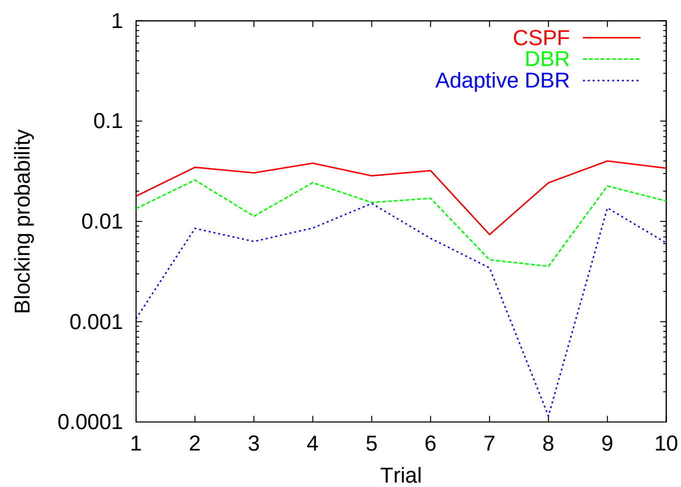
```
Adaptive DBR:
if $v_{sd} - \hat{v}_{sd}(t) > 0$
   prune with TR=0
   if a DBR path exists
      setup the DBR connection
   else
      prune with TR=r
      compute shortest path on the pruned network
      setup the connection if possible
else
   prune with TR=r
   if a DBR path exists
      setup the DBR connection
   else
      prune with TR=r
      compute shortest path on the pruned network
      setup the connection if possible
```

To investigate the effectiveness of adaptive DBR, we study the case when the estimate is inaccurate by perturbing the traffic demand matrix used in DBR computation. This is done by: (1) pick a pair of (s_1, d_1) and (s_2, d_2) at random and exchange their traffic demands; (2) repeat the process until a certain percentage M of traffic demands have been exchanged. The value of M indicates the degree of abnormality that the actual traffic demand differs from the reference traffic demand used in DBR.

In Fig. 12, we perform 10 trials of the path blocking performance, where each trial corresponds to a certain selection of pairs whose demands are exchanged. We assume that $\rho = 0.8$ and $r = 2$ in the simulation. The DBR paths was designed using $\rho = 1$. As can be seen from the figure, adaptive DBR provides additional improvement beyond that achieved by DBR, especially when the degree of abnormality increases.



(a)



(b)

Fig. 12. Adaptive DBR: (a) $M = 1\%$ and (b) $M = 10\%$.

VI. Conclusion

We have presented an approach, called design-based routing (DBR), whereby a path design computed offline is used to guide online path setups. The approach uses a DBR server which collects relevant data, computes paths offline, and responds to requests for routes. If up-to-date measured load for a given node pair is available, adaptive DBR provides further improvement via a hybrid offline-online path computation process. With typical operational networks that are carefully designed, DBR is superior to widely deployed online path computation approaches such as SPF and CSPF under static or dynamic connections with or without protection. In particular, in the network that we study, the following table summarizes the efficiency of DBR relative to SPF and CSPF for accurate traffic demand and static connections.

Scheme:	Normalized resource usage:
DBR with no protection	1.00
CSPF with no protection	1.21
SPF with no protection	1.28
DBR with shared protection	2.03
CSPF with shared protection	2.21
SPF with shared protection	2.62
DBR with dedicated protection	2.77
CSPF with dedicated protection	3.19
SPF with dedicated protection	3.77

For dynamic connections, the blocking probability of DBR typically results in two orders of magnitude reduction compared to the alternatives. For a given blocking probability less than 0.1, DBR typically sustains about 20%-30% more load. Our results indicate that load information, even in the presence of uncertainties and inaccuracies, can be utilized to improve the performance of an online path setup mechanism.

When the network is not properly designed (e.g., when the link capacities are random), the advantage of DBR rapidly diminishes with increase in uncertainty in the traffic demand information.

Our approach can be easily extended to an MPLS-based packet network where each connection request is associated with a certain bandwidth requirement. Another extension involves distributed DBR with minimal information whereby each ingress node computes its paths to all other egress nodes based on two types of information: the measured load between it and each egress node and the traffic demand estimate among other pairs.

Finally, we note that the entire methodology introduced in this paper can be naturally extended to transparent and semi- transparent GMPLS-based optical networks. In these networks, end-to-end connections are set up either with no optical-to- electrical conversion (transparent) or with minimal conversions (semi-transparent) along the path. In this context the DBR methodology consists of determining routes (with/without protection) to 1. optimize used bandwidth (as done in this paper) and 2. minimize use of shared resources, such as wavelength convertors. Related work which quantifies the benefits of this approach when there is no uncertainty in traffic demands may be found in [12],[13].

References

[1] E. Rosen, A. Viswanathan and R. Callon, "Multiprotocol Label Switching Architecture," RFC 3031, Jan. 2001

[2] D. O. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels," RFC 3209, Dec. 2001.

[3] E. Mannie et al., "Generalized Multi-Protocol Label Switching (GMPLS) Architecture," IETF Draft, Work in Progress, Mar. 2002.

[4] D. O. Awduche, A. Chiu, A. Elwalid, I. Widjaja and X.Xiao, "Overview and Principles of Internet Traffic Engineering," RFC 3272, May 2002.

[5] D. Mitra and K. G. Ramakrishnan, "A Case Study of Multiservice, Multipriority Traffic Engineering Design for Data Networks," in *Proc. Globecom'99*, pp. 1077-1083, Dec. 1999.

[6] M. Kodialam and T. V. Lakshman, "Minimum Interference Routing with Applications to MPLS Traffic Engineering," in *Proc. of IEEE Infocom*, 2000.

[7] D. Mitra and I. Ziedins, "Virtual Partitioning by Dynamic Priorities: Fair and Efficient Resource-Sharing by Several Services," in *Proc. 1996 International Zurich Seminar on Digital Communications*, Lecture Notes in Computer Science, Broadband Communications, B. Plattner (editor), Springer, 1996, pp. 173-185.

[8] R. D. Davis, K. Kumaran, G. Liu, and I. Saniee, "SPIDER: A Simple and Flexible Tool for Design and Provisioning of Protected Lightpaths in Optical Networks," *Bell Labs Technical Journal*, vol.6, no.1, Jun. 2001.

[9] S. Plotkin, "Competitive Routing of Virtual Circuits in ATM Networks," *IEEE Selected Areas in Communications*, vol.13, no.6, pp. 1128-1136, Aug. 1995.

[10] J. W. Suurballe, "Disjoint Paths in a Network," *Networks*, pp. 125-145, 1974.

[11] E. Bouillet, D. Mitra and K. G. Ramakrishnan, "The Structure and Management of Service Level Agreements in Networks," *IEEE JSAC* Vol. 20, No. 4, pp 691-699, May 2002.

[12] I. Widjaja, A. Elwalid, "Study of GMPLS Lightpath Setup over Lambda-Router Networks," *IEEE Proc. ICC 2002*, New York, 2002.

[13] R. Giles, K. Kumaran, D. Mitra, C. Nuzman, I. Saniee "Selective transparency in optical networks," *Proc. of ECOC2002*, Copenhagen, Oct 2002.