Reduced Flow Routing: Leveraging Residual Capacity to Reduce Blocking in GMPLS Networks

Xiaolan J. Zhang, Sun-il Kim, *Student Member, IEEE*, Steven S. Lumetta, *Member, IEEE*
ECE and CS Departments, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign
Email: {xzhang29, sunilkim, lumetta}@uiuc.edu

***Abstract*— Traffic engineering has been extensively studied to maximize network resource utilization while minimizing call blocking [1]. As the the demand for high data rate services over wide area networks (WAN) continue to grow [2], and as traffic patterns become subject to more frequent changes, resource utilization and the ability to guarantee quality-of-service become more important. However, accurate, predictive traffic models are difficult to construct [3], implying that the routing mechanism will need to adapt to the difference between the anticipated and the observed loads. Online routing based on network residual capacity plays an important role in such settings, since optimal offline solutions require knowledge of future traffic, rendering long-term optimization nearly impossible. shortest path first (SPF) based routing is fast and is currently the most widely-used online algorithm for optical networks. Many variants of SPF, like CSPF (Constrained SPF), have been proposed to further reduce blocking and network congestion.**

This paper focuses on the problem of online open routing for connection-oriented optical networks. As part of the major contributions of this paper, we propose a novel online, link-state based routing algorithm called reduced flow routing (RFR), an oracular optimization model, and an efficient network provisioning algorithm. The RFR algorithm uses a fast analysis of each potential route's impact on future requests to select amongst the available choices. The strategy of RFR leverages information about network topology and residual capacity to reduce blocking by rejecting 20% less connection requests relative to other online routing algorithms with little additional cost. The RFR algorithm can be integrated readily into current and future Generalized Multi-Protocol Label Switching (GMPLS) networks as well as many other relevant networks.

***Keywords* Traffic engineering, online routing, dynamic, SPF, RFR.**

I. INTRODUCTION

Internet communication has grown rapidly for many years. The current backbone networks that support data traffic face rapidly changing traffic patterns due to the emergence of new data-hungry services (such as online video gaming, seasonal sports event broadcasting, etc.), which brings steep changes in traffic over a certain period of time. At the same time, the traffic patterns of both old and new services can also change over time [2]. Because many networks are designed based on certain traffic projections and require considerable cost to upgrade physically, changing in traffic pattern lays an increasing burden on the backbone networks. It is essential to be able to address this issue as these networks also emerge as a key component in infrastructures for metropolitan areas. The physical immobility of network infrastructures and the unpredictability in traffic patterns require future network platform that are capable of supporting heterogeneous traffic models and capable of providing low service blocking rates while minimizing cost of expensive network resources.

In this paper, we study commonly used online routing algorithms and propose an oracular optimization scheme. We also introduce a new routing algorithm that can improve blocking rates and link utilization for underutilized links in optical backbones. In particular, we focus on Generalized Multiprotocol Label Switching (GMPLS) [4] networks. The GMPLS network control platform extends traditional MPLS [5] to support routing through various physical infrastructures, including synchronous optical networks (SONET/SDH) and optically transparent networks, aiming at integrating multiple network carriers into a single control structure. As an extension to MPLS, GMPLS also provides signaling for explicit routing for connection-oriented optical networks. RSVP-TE [6] protocol provides resource reservation or non-reservation routing through label switched routers (LSRs), including establishing label switched paths (LSPs), preemption, and loop detection functions. According to [7], GMPLS networks employ dynamic routing and signaling mechanism so that LSPs can be established and terminated on demand, providing a platform necessary for dynamic routing.

The difficulty of predicting changes in traffic demands and the need for rapid and robust response to traffic pattern changes in WAN have generated great interest in online routing algorithms at all levels of networking as a means to prevent network congestion. Inspired by the idea and performance comparison of variants of the shortest path algorithm, such as, widest-shortest path (WSP) [8], and general routing algorithms, such as, the minimal interference routing (MIR) [9], we studied the problem of selection amongst multiple shortest paths and propose a new shortest path selection method, called reduced flow routing (RFR). RFR provides improvements over previous results by reducing blocking with little additional cost. We compare the results of the different algorithms on several realistic network topologies using dynamic non-uniform time-varying traffic loads with Poisson arrivals and hold times. We give an ILP formulation that allows us to calculate the lower bound for the oracular algorithm (i.e., optimal online,

---

The material presented in this paper is based in part upon work supported by National Science Foundation grants ANI 01-21662 ITR and ACI 99-84492 CAREER. We also wish to thank Intel for their generous donations. This work was made possible with support from the Information Trust Institute of the University of Illinois at Urbana-Champaign and the Hewlett-Packard Company through its Adaptive Enterprise Grid Program. The content of the information does not necessarily reflect the position or the policy of these organizations.

using future information) on blocking rate, and compare it with our simulation results. Our results shows that RFR outperforms the other online routing algorithms in terms of blocking by a significant amount. The average delay of RFR is lower compared to other flow based general routing schemes. No switching overhead for path setup and tear-down is added to the simple SPF scheme. We also propose a network capacity provisioning algorithm which balances the network resource to eliminate topological bottlenecks and provide realistic and insightful comparison for the routing algorithms.

The remainder of the paper is organized as follows. In the next section, we provide a discussion of background work and several different routing algorithms. In Sec. III, we provide a short survey of commonly used routing algorithms. We then present the motivation and construction of the new RFR algorithm in Sec. IV. Our ILP formulation that calculates optimal blocking rates for an online algorithm is provided in Sec. V. In Sec. VI, we discuss balancing network capacity, an essential preprocessing step to create a realistic network environment for testing, and present simulation results. Finally we present our conclusion in Sec. VII.

II. BACKGROUND

The basic goal of traffic engineering is to optimize resource usage, balance network load, and reduce network congestion. Routing algorithms aimed at balancing these objectives have been studied extensively for many years. These algorithms can be divided into two classes, online and offline. Offline algorithms collect all routing requests and compute a set of routes that minimizes, for example, network resource usage. Mixed integer linear programming (ILP) is a common tool to solve the network optimization problem with resource constraints. However, ILP is NP-hard, and all known algorithms grow exponentially to the size of the problem. Offline routing for large networks over a long period is thus infeasible in practice.

Offline routing algorithms are also infeasible when future information is simply unavailable, as is the case when network requests arrive dynamically. This type of scenario is addressed by online routing algorithms, which route one request (or a small group of requests) at a time based on the current state of the network. Shortest path first (SPF) routing has been widely adopted for online routing because it is fast and responsive to the changes in network traffic demands.

However, due to the shortcomings of dimensioning techniques, utilizing online routing may saturate critical network resources quickly, leading to high overall blocking rate while most of the network is relatively under-utilized. Many variants of SPF with additional resource constraints (CSPF) have been developed to avoid this issue [1]. The basic method used is CSPF with trunk reservation (CSPF-TR). To ensure that high-priority traffic can be accommodated, the CSPF-TR algorithm pre-reserves some resources during normal routing. The widest-shortest path (WSP) [8] [10] and shortest-widest path (SWP) [11] algorithms choose paths that balance the number of hops and available (i.e., residual) capacity in the network. WSP selects the SPF path with maximum residual capacity on the bottleneck link. In [10], a variant of WSP is proposed with hop extensions. SWP chooses the shortest one among those paths with maximum residual capacity. Minimal interference routing (MIR) [9] is a flow-based algorithm that attempts to select the path that is least likely to interfere with future requests from other node pairs. The MIR work considers only a fraction of the MPLS network nodes as potential ingress and egress nodes, assuming that intermediate nodes (LSRs) are less likely to receive new connections than edge nodes (LERs). A more detailed discussion appears in Sec. III.

Combinations of online and offline approaches have been suggested in [7] and [12]. The adaptive design based routing (Adaptive DBR) scheme [7] takes advantage of expected network traffic demands to select and provision candidate routes for each pair optimally at the offline stage. The online stage of DBR then assigns either a pre-computed route or dynamically re-route using CSPF-TR and current network state. This method utilizes the currently monitored load to compensate for the variations from the static traffic demand estimates used in offline optimization. Profile-based routing (PBR) [12] uses the expected traffic demands to perform offline computation of the threshold of trunk reservation for each class of connections. SPF is then used at the online stage. A more detailed discussion on classification of traffics, network resource reservation, preemption and re-routing can be found in [1] [13] [14]. Reserved network resources can be occupied by low-priority traffic and subsequently preempted by the arrival of high-priority requests. Re-routing may also occur when a network is congested. Resource reservation and offline routing requires foreknowledge of traffic patterns and rerouting agreements. Streaming data connections used for future applications such as remote surgery and live, remote classroom applications may not be able to tolerate such rerouting. Similarly, some data formats (e.g., those used on optically transparent networks) may not even be known to the network infrastructure, making rerouting difficult. To avoid interference and constant rerouting overheads, rerouting is usually performed periodically, and the problem of selecting routes effectively remains between rerouting intervals.

III. SURVEY ON ROUTING ALGORITHMRITHMS

In general, the goal of residual capacity constrained online routing algorithm is to choose an available path upon the arrival of a connection request based on the current network capacity state. A connection request is initiated by a source-destination node pair. Without loss of generality, we consider bidirectional connections in this paper. For abbreviation, node pair or request pair refers to a pair of nodes which initiate a connection request. A few routing algorithms are introduced in this section, including both SPF based routing and general flow based routing. They are used to compare with RFR algorithm in this paper. The randomized SPF uses Dijkstra's algorithm to find all available shortest paths on the current residual network and picks one shortest path randomly. The WSP algorithm selects a path with the widest bottleneck residual capacity among all shortest paths. The WSP+1 is an one hop relaxed WSP variant that chooses the widest path among all paths within shortest hops plus one. Further

TABLE I

NETWORK DEFINITIONS.

$G(N, E)$	Undirected graph with node set N and edge set E.
C_G	Residual link capacity of graph G.
N_G	Set of nodes of graph G.
E_G	Set of links of graph G.
$[n, m]$	Link connecting nodes n and m.
C_l	Residual capacity of link $l \in E(G)$.
w_l	Weight of link $l \in E(G)$.
(s, d)	Connection request pair from source node s to destination node d. $(s, d) \in R$.
R_G	Set of all request pairs on G. $R \subseteq N \times N$.
R_i	Requested capacity of a node request pair $i \in R$.
$a - b \cdots - z$	Path through nodes $a, b, \ldots, z$.
SRG_i^G	Set of all links involving in all the SPF paths of the pair i on the topology of G. No capacity information is considered.
SPF_i^G	Set of SPF paths on topology for pair i. No capacity information is considered.
$SPF_i(C_G)$	Set of SPF paths meeting the capacity requirement of the request pair i on G with residual capacity C_G.
$P_i(C_G)$	Set of all available paths meeting the capacity requirement of the request pair i on G with residual capacity C_G.
len_p	Length of path p.
$f_i(C_G)$	Maximum flow of pair i on graph G.
$U_i(G)$	Set of links forming the minimum cut of pair i on graph G.
λ_i	Poisson arrival rate per request pair i.
μ_i	Poisson departure rate per request pair i. The average holding time is $1/\mu_i$.

tie are broken arbitrarily since the choice would have little impact on the capacity distribution of future residual networks. On the non-SPF-based side, the SWP algorithm chooses the shortest hop path among all paths with the widest residual bottleneck capacity. Minimal interference routing (MIR) uses the flow information to compute network weights for routing. We define the network and traffic parameters in Table I. When clear from context, we omit graph subscripts (e.g., G) for brevity. The SPF reduced graph (SRG) concept is described in Sec. IV.

The routing algorithm of WSP and SWP are straightforward.

```
procedure WSP($(s, d)$)
    for all $p \in SPF_{s,d}(C)$ do
        $width_p \leftarrow \min_{l \in p} C_l$.
    end for
    $path \leftarrow \arg\max_p width_p$.
end procedure
procedure SWP($(s, d)$)
    for all $p \in P_{s,d}(C)$ do
        $width_p \leftarrow \min_{l \in p} C_l$.
    end for
    $PATHSET \cup \arg\max_p width_p$.
    $path \leftarrow \arg\min_{p \in PATHSET} len_p$.
end procedure
```

MIR [9] employs a flow view to analyze the residual capacity and considers all paths, including SPFs. In the setting of MPLS networks where label edge router (LER) nodes generate requests and label switch router (LSR) nodes only routes, they specify the set of connection requesting pairs as all ingress-egress node pairs. They define the interference of a path with respect to a request pair as the reduction in maximum flow between the request pair due to the allocation of that path. The computation of complete MIR route is NP-hard. Therefore, an online version of MIR was proposed to approximate the optimal solution while reducing the computation complexity. Here is the online algorithm to routing pair (s, d) with current network information C_G. R is the set of pre-defined ingress-egress node pairs. No other pairs $i \in R^c$ initiate connection requests. α_i is some predetermined weight for pair i. MIR first compute the min-cut for all other ingress-egress pairs excluding the current one. Then, compute the link weight by counting the number of min-cut of all other pairs where this link involves. Finally, MIR chooses the path with minimum total weight. The major draw back is the computational time for each request as we show in Sec. VI-B.

```
procedure MIR($(s, d)$)
    for all $i \in R$ do
        Compute $U_i$.
    end for
    Compute accumulated link weight:
    for all $l \in E$ do
        $w_l = \sum_{i \neq (s,d): l \in U_i} \alpha_i$
    end for
    $path \leftarrow \arg\min_p \sum_{l \in p} w_l$
end procedure
```

IV. REDUCED FLOW ROUTING (RFR)

A. Multiple Shortest Paths

Among many different routing algorithms, SPF-based routing algorithms generally perform better than non-SPF algorithms under same traffic distribution among all possible source-destination request pairs. Some representative comparisons are shown in Sec. VI-B along with the RFR results. Given the superior performance and timing benefit of choosing amongst shortest paths, we were motivated to better understand the selection amongst multiple shortest paths and design a routing algorithm based on the trade-offs.

The shortest path between any pair of nodes in a typical network topology is often not unique, even when network capacity constraints are considered. Intuitively, since most topologies are at least two-link-connected so as to be robust to link failures, any node pair sits on some cycle in the network. If the cycle is of even length, multiple paths exist. To illustrate how this property holds under dynamic loading of networks, we counted the number of dynamic shortest paths (i.e., paths based on residual capacity) found in routing 25,000 connection requests selected uniformly at random on networks with capacity of 120 channels on each link. Table II summarizes the results; majority of the requests for all but one of the networks had multiple shortest paths.

The simplest and the most common approach to dealing with this multiplicity is to break ties randomly. Although such an approach is fast, it also increases the chance of blocking future requests. By using slightly more complicated tie-breaking algorithms that consider information about the network topology and the traffic load, we can make better decisions and reduce the expected blocking rate. As an exam-

TABLE II

AVERAGE NUMBER OF DYNAMIC SHORTEST PATHS (SPF TIES) PER REQUEST.

Network	# ties per req
NJ_LATA	1.7920
NATIONAL	2.0713
COST 239	1.7971
ARPANET	1.4839

ple, consider the problem of routing a connection from node 1 to node 6 on the network of Figure 2.

Assume that each link has sufficient capacity to route the current request. The two shortest paths are 1-5-6 and 1-2-6. Choosing either of these two paths may interfere with future requests. For example, if we select path 1-5-6, link [1, 5] is on one of the SPF paths of six other node pairs: (0, 5), (0, 6), (1, 5), (2, 5), (3, 5), and (4, 5). Similarly, link [5, 6] may interfere with future requests from five pairs: (0, 6), (2, 5), (3, 5), (4, 5), and (5, 6). Thus, a total of 11 pairs (some counted twice in this simple analysis) may be blocked later if route 1-4-5 is taken. However, links [1, 2] and [2, 6] are part of the SPF paths of 14 other pairs. Assuming uniform traffic for all node pairs, we can then argue that path 1-2-6 is more likely to interfere with future requests. Intuitively, a more careful selection of shortest paths among ties may reduce overall blocking. In the next section, we explore this problem and describe our algorithm to address the issue using information on network topology and residual capacity.

B. Leveraging Residual Capacity

Network residual capacity is the most commonly used information for online algorithms in selecting a path. WSP provides one simple selection method based on the bottleneck residual capacity information. Our novel reduced flow routing (RFR) is a flow-based method, providing fast analysis of the residual capacity and reduces blocking. First we compute the SPF reduced graph (SRG) for each request pairs on a given network topology. SRG is a subgraph consisting of all topologically shortest paths for one node pair. Figure 1 is an example of SRG for pair (1, 10) in NJ_LATA with residual capacity indicated on each link. SRG is shown on the right side.

RFR computes the maximum flow on the SRG with current residual capacity C_{NJ_LATA} for each request pair other than the current connection request. The max-flow on SRG $f_{1,10}(C_{SRG_{1,10}^{NJ\_LATA}})$ for pair (1, 10) is 5. The reason for computing the flow on SRG rather than the entire graph is that, first, the computation time is greatly reduced and, second, considering too many links will increase the sharing of links amongst different pairs and cancel out the essential information indicating future network capacity demands, eventually resulting in more blocking. Therefore, we think of SRG in general as a critical part of the graph for a particular pair. Any occupation on the critical part of the graph by routes of other connections would interfere with the future routing requests issued by this pair. Such interference is instantiated by the reduction of max-flow caused by the routing pair. As we have found a few SPFs for the routing request pair, we select juts one SPF, temporary take away the request capacity along the path. Then the network capacity becomes $\widetilde{C}$. Compute the maximum flow for each other pair on the SRG with modified capacity information again. Then, we compute the difference between two flows for each pair and normalize it by the original flow. In the example, Figure 3 shows two SPF choices for request pair (2, 7), path 2-0-7 or 2-4-7. If the requested capacity $R_{2,7} = 1$, the routing through path 2-4-7 will result in a reduction of 1 capacity on link [2, 4] leading to a reduction of $f_{1,10}(\widetilde{C}_{SRG_{1,10}^{NJ\_LATA}})$ by 1. The normalized reduction is 1/5, but path 2-0-7 reduces the capacity on link [0, 7] which is not a critical link that affects the original max-flow of (1, 10). In view of pair (1, 10), path 2-0-7 is the preferable route for request (2, 7).

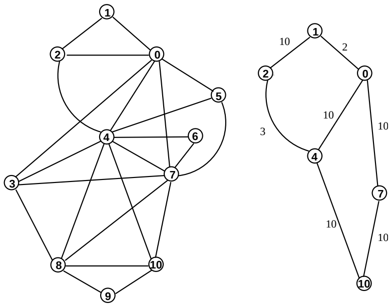

Fig. 1. NJ_LATA and SRG of connection (1, 10).

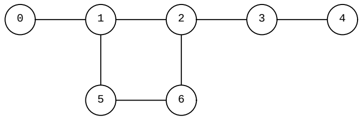

Fig. 2. Sample graph for SPF ties.

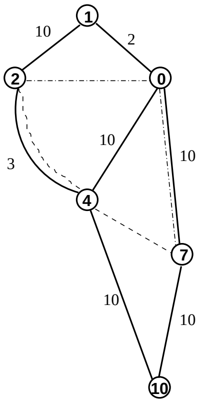

Fig. 3. Impact of routing connection (2, 7) on the flow of the SRG of connection (1, 10).

Considering all other pairs, the total cost for picking up one SPF will be the summation of all other pairs by their normalized flow. For each shortest path, we compute the cost and finally choose the one with the minimal cost. Essentially, we are choosing the path which results in minimum interference on the critical subgraph of the other pairs. The detailed algorithm is presented as follows. Suppose we route pair (s, d) on graph G with capacity C. The set of request pairs will be the set of all possible pairs $R = N \times N$.

```
procedure INIT SRGS($G$)▷ The initialization step is made only once after the network topology is designed.
    for all $i \in R$ do
        Compute $SRG_i^G$.
    end for
end procedure
procedure RFR($s, d$)
    for all $p \in SPF_{s,d}(C)$ do
        for all $i \neq (s, d) \in R$ do
            $a_i \leftarrow f_i(C_{SRG_i})$
            for all $l \in p$ do
                $\widetilde{C}_l = C_l - R_{s,d}$
            end for
            $b_i \leftarrow f_i(\widetilde{C}_{SRG_i})$
            $g_i \leftarrow \frac{a_i - b_i}{a_i}$
```

```
        end for
        $cost_p \leftarrow \sum_i g_i$
    end for
    $path \leftarrow \arg\min_{p \in SPF_{s,d}(C)} cost_p$
end procedure
```

If the traffic pattern, like the arrival rate of each pairs, is well-known and stable, the computation of cost in the normalized flow could be weighted with each arrival rate. However, we tested both weighted version (RFRw) and unweighted version, and little difference was found between these two schemes. The unweighted algorithm, presented in Sec.VI-B, outperforms other online algorithms and, sometimes, outperforms the RFRw as well. Therefore, our scheme does not necessarily require an anticipated traffic model, which is an essential property for the success of other online algorithms.

V. ORACULAR OPTIMIZATION OF DYNAMIC TRAFFIC

In order to evaluate our results relative to the optimal answer, we utilized an oracular optimization for dynamic traffic. As mentioned earlier, long-term optimization of dynamic traffic demands is neither feasible nor meaningful as a point of comparison. Lack of future knowledge renders precomputed results suboptimal. Instead, we consider the use of an oracle for short sequences of requests starting from a steady-state snapshot of the network. In particular, given information about a number of future requests as well as the current state of the network and the termination times for active connections, the oracle selects routes for future requests in a way that minimizes the number of those requests that cannot be routed. This oracular result thus puts a lower bound on the blocking rate achieved by any algorithm operating in the window considered. The window size is selected so as to make the problem of finding the oracular optimum tractable.

We now describe the oracle optimization problem in more detail and give an ILP formulation that allows us to solve it. Optimization begins at a specific simulation time after the network has reached a steady state. In addition to information about the current state of the network and hold time information for all active connections, the optimizer receives a number of future requests with arrival and departure information (this is the "oracle" part; no real algorithm can have such information). Figure 4 illustrates an example of six oracle requests and active connection information that should be recorded. (The routes used by each connection active at the start of the optimization window must also be recorded, of course, but are not readily shown by a timeline.)

Let E be the set of graph links, and let R be the set of oracle request pairs including the arrival and departure timestamps. Let t be a discrete time line ordered by the arrival time of requests. Therefore a set of requests sorted by arrival time can be indexed by t. Let r^t be the request that arrives at time t, and let d_{r^t} be the scheduled departure time for request r^t. Let $C_{e,t}$ be the residual capacity on link e at time t, at which point any capacity freed by the termination of active connections before time t has been considered in the network resource constraints. Let P_{r^t} be the set of all available paths for request r based on the network's residual capacity at time t. Let q_{r^t} be the requested capacity of pair r^t. Let Q_e represent the set of paths that traverses edge $e \in E$. Denote by y_{r^t} a binary variable representing the acceptance of request r^t. A solution with $r^t = 0$ implies that r^t is not taken, while a value of 1 means taken. The variable $x_{r^t,p}$ is a positive integer representing the occupancy of path $p \in P_{r^t}$. Again, 0 means not taken, while any other number represents the number of wavelength channels assigned to that path. The ILP formulation is stated as follows.

$$\text{Maximize } \sum_{r^t} y_{r^t} q_{r^t}$$

$$\forall r^t \in R, \sum_{p \in P_{r^t}} x_{r^t,p} \geq y_{r^t} q_{r^t} \tag{1}$$

$$\forall r^t \in R, \forall e \in E,$$

$$\sum_{p \in Q_e \cap P_{r^t}} x_{r^t,p} + \sum_{r^\tau : \tau < t, d_{r^\tau} \geq t} \sum_{p \in Q_e \cap P_{r^\tau}} x_{r^\tau,p} \leq C_{e,t} \tag{2}$$

$$x_{r^t,p} \in Z^+ \tag{3}$$

$$y_{r^t} \in \{0,1\} \tag{4}$$

Equation (1) ensures that for each accepted request (right side), enough paths have been allocated to handle the request (left side). Equation (2) constrains the channels utilized at any point t in time and at any edge e to the capacity available in the network. The right side of the equation is simply the capacity available in the link e at time t, considering both the initial capacity of the link and the capacity dedicated to connections that were active at the start of the optimization period and have not yet terminated. The terms on the left side include the channels to be provided for request t as well as the channels provided for other connections allocated before t and terminating after t. The last two equations constrain the values of the integer variables that specify how requests are accepted and routed. As a general form, the ILP solution may assign multiples to one request which maximizes the utilization of network resources. However, we do not consider request capacity more than one in this paper thus no splitting of capacity would occur on multiple available paths. Equivalently in the ILP formulation, the x variables are binary and qs are uniformly one.

VI. PERFORMANCE ANALYSIS

In this section, we present simulation results, which show that RFR achieves the lowest blocking rate for several representative network topologies. Because RFR is applicable to any complex routing schemes based on SPF variants, without loss of generality, we made the following simplification in our simulations. We assume that the optical layer of GMPLS networks is connection-oriented and do not consider preemption of resources or prioritization amongst connection requests, thus once a connection is set up, it persists for its entire hold time. We also assume an optically opaque transport network, with either optoelectronic (OEO) conversion or full wavelength conversion capabilities at every node. We assume the availability of signaling for connection setup

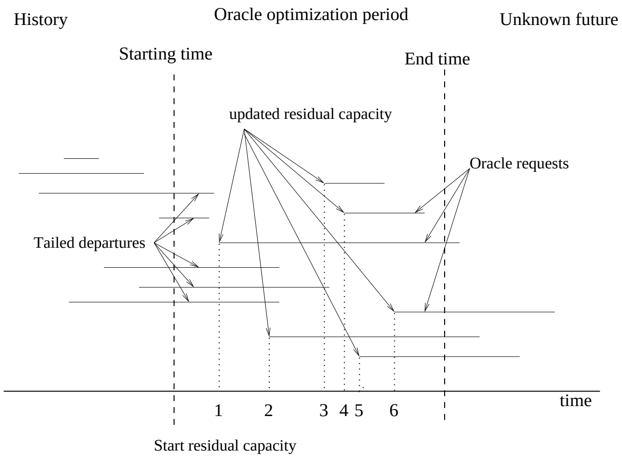

Fig. 4. Illustration of oracle requests

and termination as well as distribution of residual capacity information. We further assume that request arrival rates and hold times for connections allow sufficient time for signaling, such that all new connections are routed based on up-to-date information about residual capacity for all links in the network. Because traffic bursty on WAN are usually suppressed by huge amount of data on the network, Poisson processes are used to model connection requests and durations (arrivals and departures). The arrival rate for each connection pair is chosen uniformly from 1 to 10. The departure rate stays the same for all connections. Note that our focus is on the change of traffic patterns in terms of Poisson rate rather than transient traffic fluctuation. The real-world networks used for testing are NJ LATA, COST 239, NATIONAL and ARPANET. When not otherwise stated, network references in this section refer to NJ LATA, the local access and transport area network for the state of New Jersey.

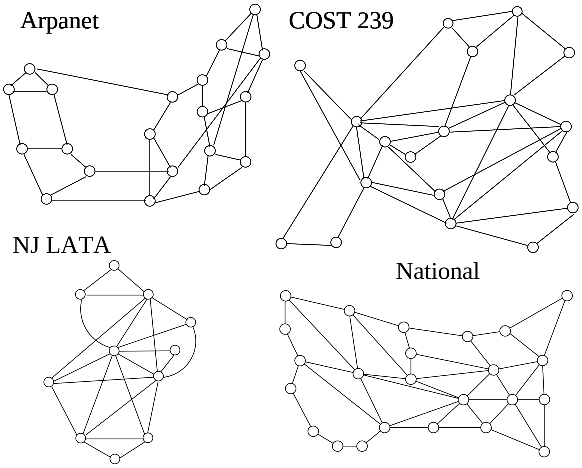

Fig. 5. Network topologies.

A. Network Capacity Provisioning

We provision network capacity before doing the blocking rate test. The reason for doing this is that some networks may present a topological bottleneck that gets saturated much faster than other part of the network, contributing a higher portion of blocking caused by those connection requests that have to go through the bottleneck. Figure 6 and Figure 8 demonstrates an example bottleneck in NJ LATA. We perform blocking tests on a uniform static traffic model. The network link capacity is uniformly assigned with 120 wavelengths and each connection requests one wavelength capacity.

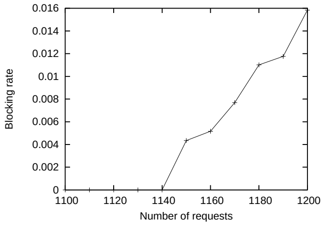

Fig. 6. NJ_LATA rate 1 static traffic on unbalanced network.

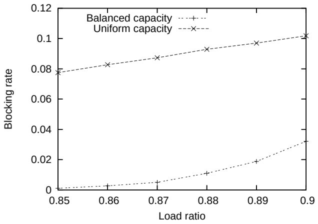

Fig. 7. Comparison of blocking on balanced and unbalanced network.

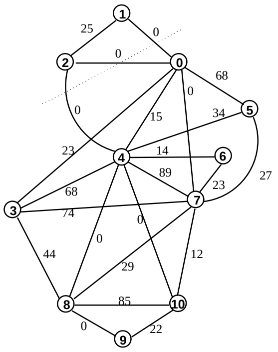

Fig. 8. A cut in NJ_LATA on the arrival of 1140 requests.

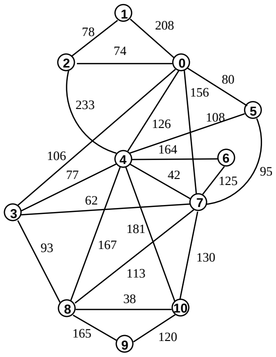

Fig. 9. NJ_LATA network with balanced capacity.

Figure 6 shows a sharp increase in blocking rate at the point of 1140 Poisson arrivals. Figure 8 is a snapshot of the current network residual capacity on the arrival of 1140 requests. It shows that link [1, 0] [2, 0] and [2, 4] with 0 residual capacity form a cut blocking future requests from or to nodes 1 and 2. However, the rest of the network still has ample capacity. In term of network design, it would be very inefficient to have an early cut in the topology given the available total resources. The cut also produces an undesirable factor in performance tests because the result would be largely affected by only a small part of the network rather than the whole. Given the same total number of wavelengths on a network, significant reduction in blocking can be achieved by using even with a very rough anticipated traffic model over a balanced network. A comparison between balanced and unbalanced result is clearly shown in Figure 7. Therefore, network capacity balancing based on provisioned traffic becomes an important preprocessing step for both practical and theoretical purposes.

The dynamic traffic load is determined by the following formula. $\mathsf{E}(\cdot)$ is the expectation operator.

$$load = \frac{\mathsf{E}(\lambda)\mathsf{E}(R)\sum_{i\in R} len_{p\in SPF_i}}{\mathsf{E}(\mu)\sum_{l\in E} C_l} \qquad (5)$$

The average link capacity represents the network resource budget. The average arrival rates are configurable based on traffic projection. In the simulation, the holding time is proportional to $load$ for a given network topology and projected traffic. Approximately, a network is considered to be full when $load = 1$. The capacity balancing algorithm used to assign wavelength Cap_l for each link l is described as follows. The network topology as well as $len_{p \in SPF_i}, E, R$ are known.

```
1: Compute $\mathsf{E}(\mu)$ by initial traffic information and Equation 5.
2: Virtually set infinite available capacity on each link $C_l \leftarrow \infty \ \forall l \in E$
3: for Each arrival request $i$ until the system reaches steady state do
4:     Route $i$ by shortest path first algorithm (SPF).
5: end for
6: $\widetilde{C}_l$ is the actually used network capacity of each link.
7: Repeat from Line 1 to get the distribution of $\widetilde{C}_l$ and therefore mean $\mathsf{E}_l(\widetilde{C}_l)$ and deviation $\sigma_l$.
8: Use mean capacity $Cap_l \leftarrow \mathsf{E}_l(\widetilde{C}_l) \ \forall l \in E$.
```

Figure 9 shows the capacity provisioned for NJ LATA network with an average of 120 wavelength per link. The traffic model is dynamic uniform for all arrivals. Links [1, 0] and [2, 4] which formed a cut very early are now assigned a higher capacity budget than other links such as [8, 10], achieving a more uniform congestion. Notice that the total capacity on balanced network is 2741 compared to the unbalanced network capacity of 2760 (23 links × 120). Fewer capacity in practice also means fewer lasers needed to drive the network. Figure 7 shows a comparison between balanced and unbalanced networks in terms of blocking for SPF for another 5000 rate 1 requests at the steady states. The balanced network has a 7% reduction in blocking rate.

Although the dimensioning based on traffic projections may not always be accurate, majority of real world networks are designed with some traffic projections beforehand. In simulations, a provisioned network actually eliminates extreme bottlenecks and provides a more practical environment to compare different routing algorithms.

One interesting question here is whether the variety of anticipated loads will affect the actual performance even when the ratio of the actual load to the anticipated load stays the same. Figure 10-15 present the blocking rates for SPF and WSP routing algorithms with various anticipated loads during network provisioning. The horizontal axis shows the ratio between actual and anticipated load. The blocking ratios from both figures show the same trend between SPF and WSP under the same anticipated load. Therefore, one network provisioned on a fixed load could be good enough to provide a universally reliable result for routing experiments. For simplicity, we balance the network capacity with $load = 1$ and the actual loads are automatically normalized to be the ratio to the anticipated load.

B. Simulation Results

The capacity of all the sample networks are balanced by the same dynamic arrival rate as the real traffic. The anticipated average link capacity is 120 wavelengths. Each request uniformly demands 1 unit capacity. The provisioned load is 1. Each point on Figure 16 shows the average blocking probability of SPF, WSP and RFR on NJ LATA sampled on the next 5000 request arrivals. Steady states behavior is shown after 10000 dynamic arrivals. Since it is a large dynamic system, some amount of fluctuations are shown at the steady states. However, all routing algorithms present similar trends of fluctuations. In our experiment, the blocking rate is measured by sampling 5000 arrivals after running 20000 warm up requests before the sampling. Each pair has the same arrival rate under the uniform traffic load. Each pair holds a randomly pre-assigned arrival rate uniformly ranging from 1 to 10. The average hold time for each pair is determined by the network load. Each experiment will repeat 100 times with different arrivals to obtain the average blocking rate. It is desirable to have more test samples to average. However the sample size that we choose in this paper is adequate to provide representative results for the system performance aspects that we intend to study. We pay special attention to the blocking probability ranging from 0.5% to 8% because a higher blocking rate indicates an overloading of the network. It also indicates that more than one cut has been formed, which makes it impossible to route a significant portion of requests. Since our goal is to find a better routing algorithm to prevent congestion, any comparison under heavy congestion is undesirable.

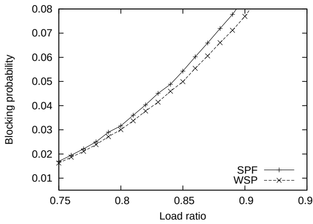

Fig. 10. Prj load 0.5.

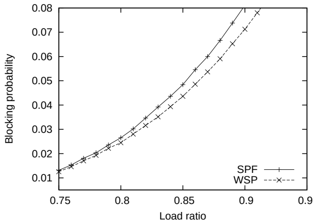

Fig. 11. Prj load 0.6.

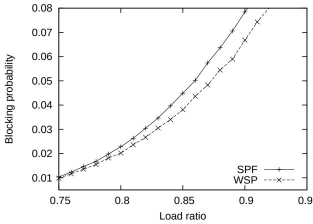

Fig. 12. Prj load 0.7.

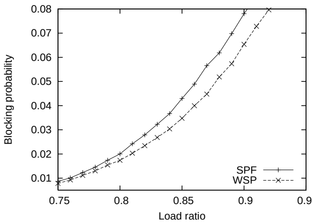

Fig. 13. Prj load 0.8.

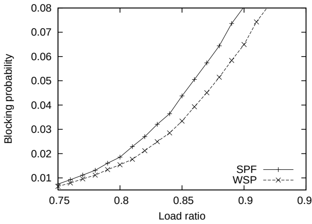

Fig. 14. Prj load 0.9.

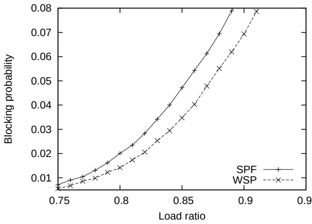

Fig. 15. Prj load 1.0.

Figure 17-20 show the comparison of blocking probability among online routing algorithms. The horizontal axis represents the actual load that has been normalized by the projected

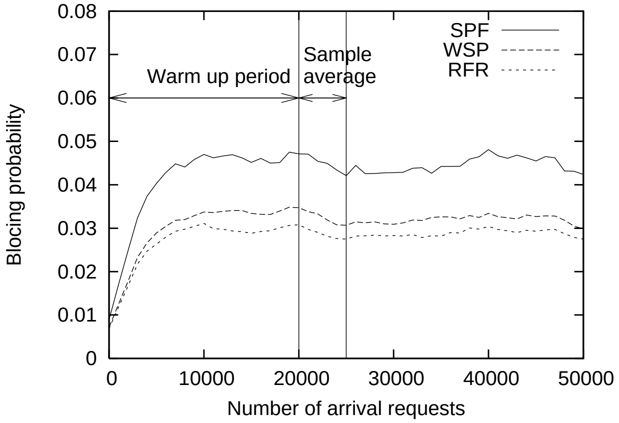

Fig. 16. Steady state performance on NJ LATA.

load. All node pairs in the network are considered as ingress-egress pairs for MIR testing. The α parameter is set to be the arrival rate for each pair. SPF uses random tie breaking strategy. WSP chooses the widest bottleneck SPF path. SWP chooses the shortest path amongst all widest paths. Results show that SWP can not compete with SPF under the random traffic arrival model. RFR demonstrates the lowest blocking. The traffic rate weighted version, RFRw, shows little difference from the non-weighted version, RFR, which shows that RFR performs well even without having a priori knowledge of traffic patterns. WSP has marginally higher blocking rate compared to RFR when the load is low and increases faster as the network load increases. WSP+1 is a loosed WSP version which picks the widest path among all paths with lengths up to shortest length plus one. SWP and WSP+1 do not perform as well as other algorithms. MIR presents low blocking when the load is low. However, the blocking rate increases faster compared to many other algorithms as the load increases. MIR also presents a higher blocking rate when every pair in the network issues connection requests compared to when arrival requests are generated by a specific set of predetermined ingress-egress pairs [9]. Generally, the algorithms that pick up shortest paths perform better than others, such as SWP, WSP+1 and MIR. The big difference between WSP and WSP+1 strongly motivated the design of online algorithms in choosing amongst shortest paths. RFR has less than 2% blocking compared to WSP and 4% less compared to SPF when the network is fully loaded, and is clearly the algorithm in preventing network congestion with the slowest growing slope at the beginning.

Figure 17-20 illustrate the actual network utilization by offered load. The actual utilization is computed by the sum of average used capacity on each link, divided by the total available capacity. The actual utilization for all algorithms is larger than the offered load because the offered load is computed by topological shortest paths length while real routing scenarios can utilize longer paths when available. The results show that RFR uses the least amount of total capacity compared to other algorithms and provides the best resource balancing for future requests.

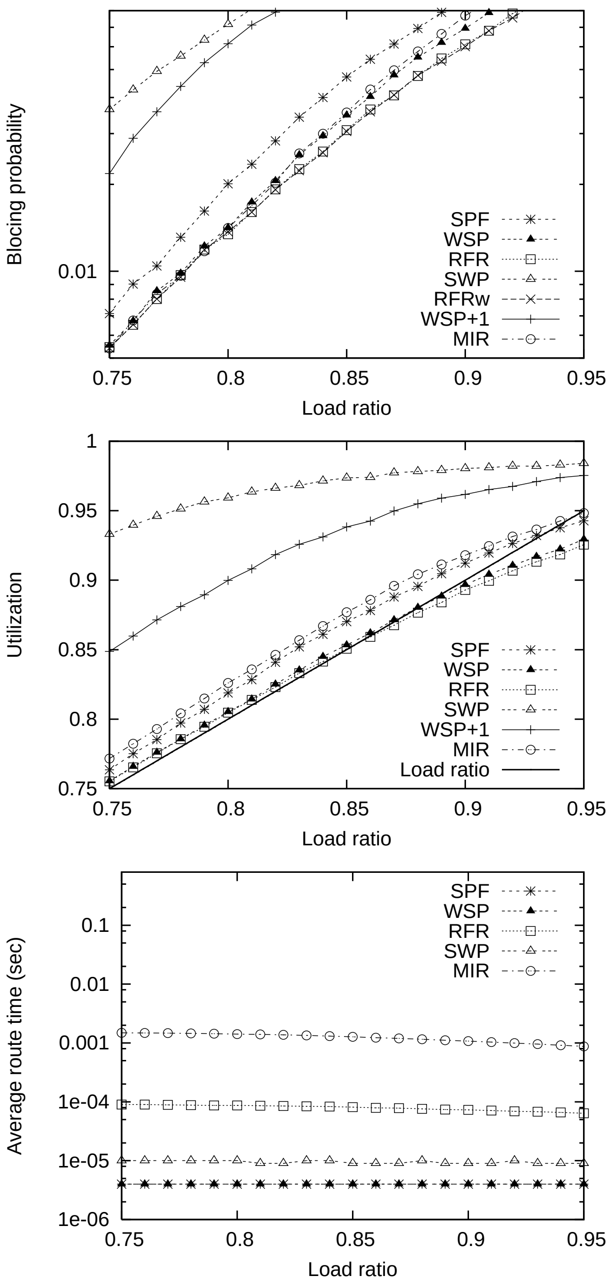

Fig. 17. Comparison of blocking probability, utilization and time on NJ LATA.

We now show the results from comparing routing times between SPF, WSP, RFR, SWP and MIR, which are representative of common routing algorithms. We should note that the quality of timing depends on the real inter-arrival times in the network. If requests arrive in the order of minutes, all 5 algorithms will be good enough (less than 1 sec to route one request) to meet the deadline. Figure 17-20 shows the timing in unit of logarithmic scale second. In smaller networks, such as NJ LATA, the routing times for SPF and WSP are mostly independent to the traffic load. While in larger networks, such as NATIONAL, noticeable increase in routing time is shown because SPF-based algorithms take more time to search for a longer available path, especially when heavily loaded. SWP is affected least by the change in network size. RFR is 10

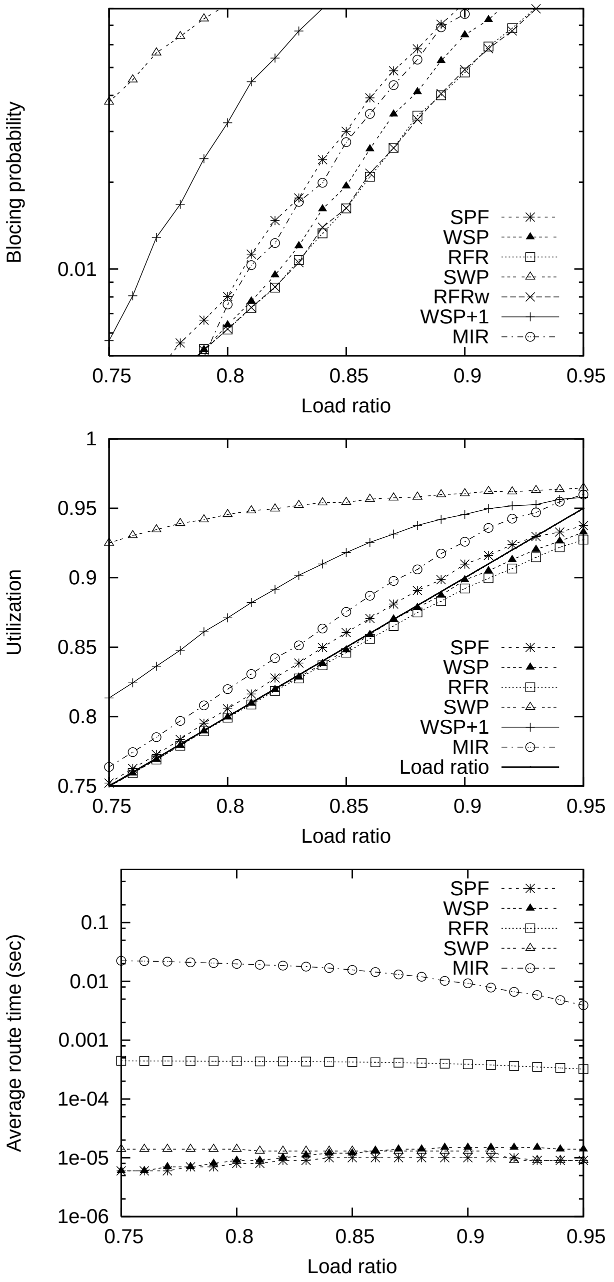

Fig. 18. Comparison of blocking probability, utilization and time on COST 239.

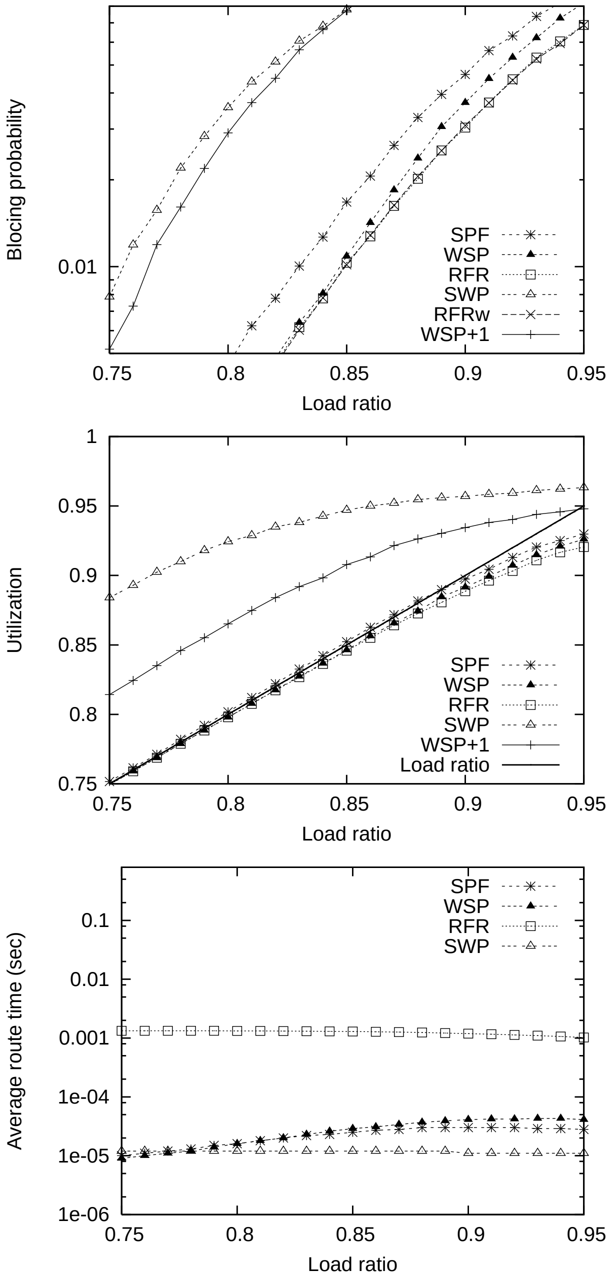

Fig. 19. Comparison of blocking probability, utilization and time on NATIONAL.

times better than MIR. Both RFR and SWP decreases as the load increases because their routing time are mainly spent on choosing among available paths, the number of which decreases when the network becomes congested.

Finally, we compare RFR with the oracular optimal result (OPT). Due to the limitation of computational resources for larger networks, we reduce the per link capacity to 48. Dynamic arrival rate ranges from 1 to 10. Each request asks for one unit capacity. No splitting of path can occur during the optimal routing. We perform our experiment on the NJ LATA network. We use SPF to run the first 10,000 arrivals on the network to reach and obtain a steady state. Then, the next 200 requests are routed by SPF, WSP, RFR and OPT respectively. Each data is an average of 1000 experiments. In this experiment, starting with the same initial network state, RFR is the closest online algorithm to OPT. As the network load increases, the gap between RFR and OPT increases, showing that oracular knowledge becomes increasingly important to reduce blocking in congested network states. Without the aid of oracular knowledge, RFR provides the lowest blocking rate.

VII. CONCLUSION

In this paper, we proposed a reduced flow online routing algorithm that provide the lowest blocking and network resource usage compared to commonly used online algorithms. Our algorithm uses current network state information and does

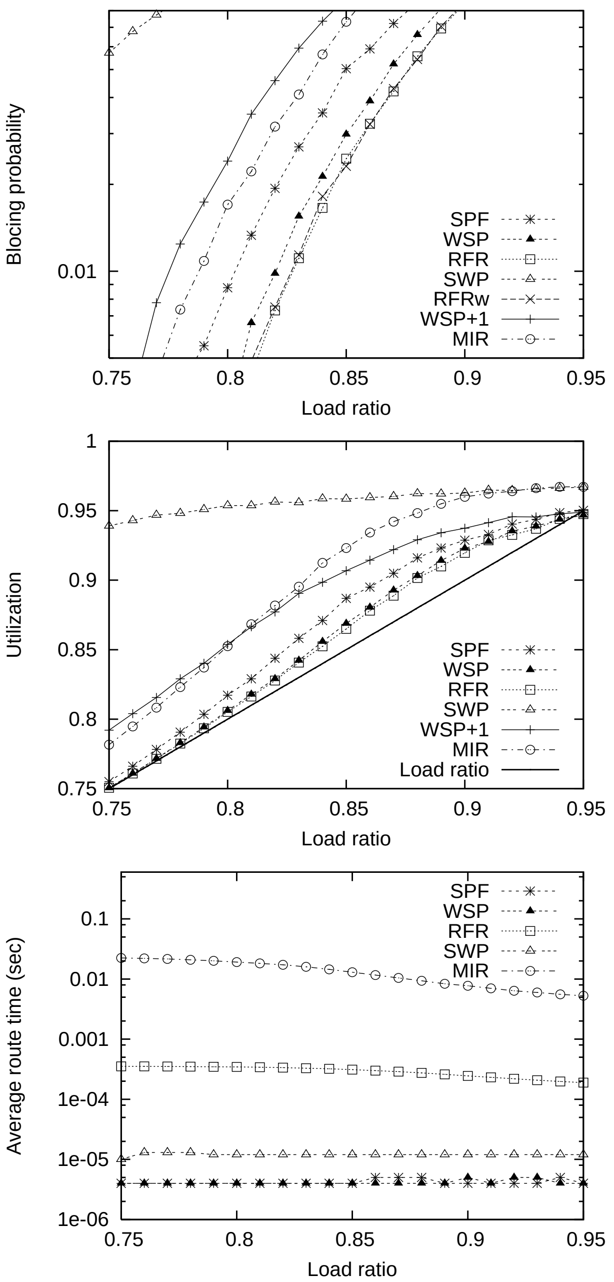

Fig. 20. Comparison of blocking probability, utilization and time on ARPANET.

not require anticipated network traffic or knowledge of future traffic. The routing overhead is the same as simple SPF-based algorithms. We also proposed an oracular optimization model, showing that the long term optimization of network becomes harder without a priori knowledge of future traffic changes. We also demonstrated that the carefully choosing between multiple SPFs is important in achieving lower blocking rate. Following realistic network models, the importance of network capacity provisioning based on anticipated network traffic is shown. More realistic test environment can be provided by pre-provisioning to eliminate "early cuts" in the network. In the provisioned network environment, under dynamic random traffic model, the blocking probability of RFR/RFRw has been tested against other online routing algorithms (SPF, WSP, WSP+1, SWP and MIR) with various loads on four real network topologies. The results show that RFR has the lowest blocking rate and is most resistant to the changes in traffic load. RFR also shows timing advantage to other flow based algorithms.

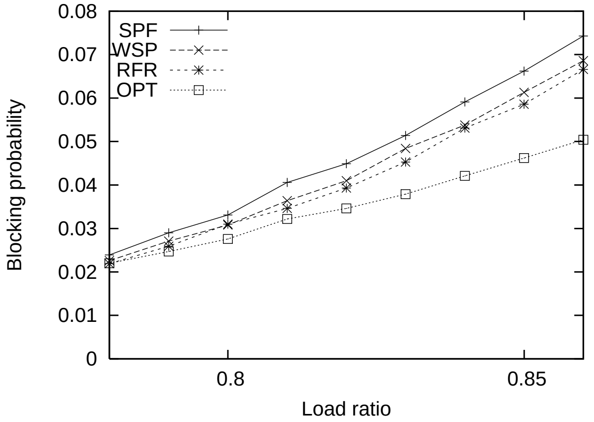

Fig. 21. Comparison of blocking rate on NJ LATA with OPT.

REFERENCES

[1] A. S. B. Szviatovszky and A. Juttner, "Minimizing re-routing in mpls networks with preemption-aware constraint-based routing," *Computer Communications*, vol. 25, pp. 1076–1083, 2002.

[2] S. McCreary and K. Claffy, "Trends in wide area IP traffic patterns - A view from ames Internet exchange," *Proceedings of the 13th ITC Specialist Seminar on Internet Traffic Measurement and Modelling, Monterey, CA*, 2000.

[3] D. Clark, W. Lehr, and I. Liu, "Provisioning for bursty internet traffic: Implications for industry structure," in *Internet Service Quality Economics*, L. McKnight and J. Wroclawski, Eds. MIT Press, 2001.

[4] L. Berger, "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description," RFC 3471, Jan. 2003.

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

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

[7] A. Elwalid, D. Mitra, and I. Widjaja, "Routing and protection in GMPLS networks: From shortest paths to optimized designs," *IEEE Journal on Lightwave Technology*, vol. 21, no. 11, pp. 2828–2838, 2003.

[8] A. O. Roch Guerin and D. Williams, "QoS routing mechanisms and ospf extensions," in *2nd Global Internet Miniconference*, 1997.

[9] M. S. Kodialam and T. V. Lakshman, "Minimum interference routing with applications to MPLS traffic engineering," in *INFOCOM*, 2000, pp. 884–893.

[10] A. Shaikh, J. Rexford, and K. G. Shin, "Load-sensitive routing of long-lived ip flows," in *SIGCOMM '99*.

[11] Z. Wang and J. Crowcroft, "Quality-of-service routing for supporting multimedia applications," *IEEE Journal of Selected Areas in Communications*, vol. 14, no. 7, pp. 1228–1234, 1996.

[12] S. Suri, M. Waldvogel, and P. R. Warkhede, "Profile-based routing: A new framework for MPLS traffic engineering," in *Quality of future Internet Services*, ser. Lecture Notes in Computer Science, F. Boavida, Ed., no. 2156. Berlin: Springer Verlag, Sept. 2001, pp. 138–157, an earlier version is available as Washington University Computer Science Technical Report WUCS-00-21, July 2000.

[13] P. Iovanna, R. Sabella, and M. Settembre, "A traffic engineering system for multilayer networks based on the GMPLS paradigm," *Network, IEEE*, vol. 17, no. 2, p. 28.

[14] P. Iovanna, R. Sabella, and P. D'Aprile, "GMPLS traffic engineering system to dynamically manage several classes of services: performance analysis," in *Optical Network Design and Modeling, IEEE Conference on*, p. 483.