

# Efficient and Robust Congestion Estimation for Dynamic WDM Networks

Xiaolan J. Zhang<sup>1</sup>, Sun-il Kim<sup>2</sup>, Steven S. Lumetta<sup>1</sup>

<sup>1</sup>Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign  
email: {xzhang29,lumetta}@illinois.edu

<sup>2</sup>Department of Computer Science, University of Alabama at Huntsville  
email: sunilkim@cs.uah.edu

## Abstract

Emerging applications require high-bandwidth network connectivity on demand, exacerbating the already bursty nature of data traffic and motivating the development of new dynamic management techniques for the wavelength-based optical backbones underlying the Internet.

This paper focuses on the problem of routing new connection requests in such an environment, introducing efficient algorithms as well as models that help to provide a deeper understanding of the algorithms' benefits. We first introduce a threshold-based Congestion Aware Routing (CAR) algorithm that leverages quick congestion estimation and admission control techniques to dynamically shift between low- and high-load strategies. Using the link load information from a prospective path, the congestion estimator responds effectively to localized congestions. By adapting to observed congestion levels, this algorithm provides reductions in blocking that are robust to variations in offered load and traffic. We also develop a model that illuminates the choice of threshold used by our algorithm to determine the level of congestion, and build improved routing algorithms (CAR-C/M) with robust congestion estimation techniques. Simulation results show that our algorithms, at a low cost, not only reduce call blocking over a wide range of loads, but provide better admission probability for long connections. We also compare several routing algorithms under the influence of network scaling and misdimensioning. Our algorithms are simpler in implementation and threshold parameter selection compared with previous admission control algorithms, sustaining good performance under various network conditions.

## I. INTRODUCTION

The growth in network applications and systems that require high data-rate communication and flexible reservation motivates the design of dynamic optical network infrastructures with the ability to efficiently establish connections on demand [1]–[3]. The emerging Generalized Multi-protocol Label Switching (GMPLS) networks already provide signaling capability for dynamic lightpath reservation at the WDM layer. As WDM backbone networks continue to evolve toward a more dynamic model using reconfigurable Add/Drop Multiplexers (ROADMs) and large port count MicroElectroMechanical system switches (MEMs), wavelength routing schemes that can better utilize network resources become more attractive.

Adaptive shortest path routing (ASPF) selects a shortest hop available path on the current wavelength residual network. The benefit of choosing paths exclusively amongst shortest available paths in a dynamic routing environment was discussed in [4]. Assigning a longer path when a shorter path is available increases the opportunity cost for the network to accept future arrivals. However, even in a network routed only by ASPF, the shortest available paths discovered for an end-to-end node pair at run time become longer when the network is congested. In Section III, we show that limiting the number of hops for accepted connections effectively reduces overall blocking probability when the network load is high. However, it has a negative effect when the load is low. Determining when to apply hop constraints during dynamic routing is challenging due to the difficulty of estimating the current network load. First, accurately estimating the overall network load may take a while. Second, temporary hot spots in the network created by short term traffic fluctuations can render the overall network load useless for some end-to-end pairs.

This work was made possible with the generous donations from Intel and the 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.

TABLE I  
NETWORK NOTATION.

$N$	Set of nodes.
$E$	Set of links.
$C_e$	Total capacity of link $e \in E$ .
$A_e$	Current available capacity of link $e \in E$ .
$U_e$	Current used capacity of link $e \in E$ . $U_e = C_e - A_e$ .
$R$	Set of all end-to-end request pairs. $R \subseteq N \times N$ .
$\kappa$	Mean requested capacity per pair.
$\lambda$	Mean arrival rate per pair.
$\mu$	Mean departure rate per pair. The expected holding time is $1/\mu$ .
$TSL_r$	Topological shortest path length for pair $r$ .
$TSP_r$	Topological shortest paths set for pair $r$ .
$SPG_r$	Topological shortest path graph for pair $r$ . $SPG_r = \cup_{p \in TSP_r} p$ .
$T\{\lambda_r, \kappa_r, \mu_r\}$	Traffic matrix with per pair arrival rate $\lambda_r$ , capacity $\kappa_r$ and departure rate $\mu_r$ .
$\eta$	Congestion threshold.
$\alpha$	Misdimensioning factor.

Third, network capacity (e.g. installed/active optical transceivers and fiber channels) in an optical network changes infrequently relative to the speed at which the network load scales due to variations by time of day, unexpected events (some video websites gain overnight popularity), etc.

In this paper, we approach the problem by adaptively limiting the length of paths that are selected by online routing algorithms. We propose a group of Congestion Aware Routing (CAR) algorithms that uses threshold based admission control techniques. Our contribution over previous threshold based work is an algorithm that has a simple implementation and robust threshold that maintains good performance regardless of load variations, network topologies, and network scaling. We begin presenting congestion estimation algorithms with fast implementation that outperform pure routing algorithms. We then formulate a model of opportunity cost to understand threshold selection for various networks. We show that utilizing additional links to route a connection at a congested locality penalizes more future connection requests. The model further helps to build a family of improved congestion awareness routing algorithms. Amongst all algorithms that we study, CAR-C and CAR-M have the most robust thresholds, but CAR-C is much simpler and faster. We also study the impact of network misdimensioning on different routing algorithms. The performance of CAR-M is more robust to misdimensioned network than CAR-C.

This paper is organized as follows. Section II reviews network dimensioning techniques and previous work. Section III shows some preliminary studies and motivations. Section IV presents our threshold based routing and congestion estimation algorithm (CAR) and its variants with different local congestion estimator techniques. Section V then presents an analytical model to predict the best threshold. Inspired by our analysis, we designed modified CAR routing algorithms, which we present in Section VI. Comparison of the overall blocking and timing metrics with other algorithms on different network topologies, scaling and misdimensioning are shown in Section VII. The comparison of CAR-M to previous work, SDR and COL, appears in Section VIII. A brief discussion of the general applicability of our algorithms to practical networks is in Section IX. Finally, Section X concludes the paper.

## II. BACKGROUND

In this section, we briefly review the concept of network dimensioning that provides a load-normalized environment for dynamic routing to ensure that the meaning of the offered network load is clearly defined. Then we discuss some previous work.

Before addressing the details, we first define some terminology and notation. *Topological Shortest Length (TSL)* is the minimum number of hops for a connection to route in an empty network. It may be less than the length of an available shortest path found on a residual network, which varies as link capacity and network load change.  $TSL+n$  means that a path length is equal to its TSL plus  $n$  extra hops. Our notation is summarized in Table I.

### A. Network Dimensioning

The problem of dimensioning dynamically wavelength routed networks has been addressed in [5]. A network in which resources have been dimensioned for the expected traffic pattern can accept more connections than a

```

1:  $\forall e \in E, C_e \leftarrow 0$ 
2: for each request pairs  $r \in R$  do
3:   for each topological shortest paths  $p \in TSP_r$  do
4:     for each links  $e \in p$  do
5:        $C_e \leftarrow C_e + \frac{\lambda_r \kappa_r}{\mu_r |TSP_r|}$ 
6:     end for
7:   end for
8: end for
9: Rounds all link capacities  $C_e$ s to the nearest integers.

```

**Algorithm 1:** Basic dimensioning with Poisson independent traffic matrix  $T\{\lambda_r, \kappa_r, \mu_r\}$ .

misdimensioned network using the same total capacity. Also, studying a network that has not been dimensioned for expected traffic artificially obscures differences in the impact of routing algorithms. Therefore all networks in this paper are dimensioned by a basic dimensioning algorithm.

We dimension the networks using a basic dimensioning technique that allocates wavelength resources according to network topologies and traffic patterns. Algorithm 1 is the dimensioning algorithm for independent Poisson traffic. If more than one projected traffic matrix are used, Algorithm 1 is run repeatedly for each of them and the averaged link capacity is used. The initial wavelength capacity on each link is dimensioned at 100% projected load. A fraction of the projected load is then used to model the actual offered load on a network. If the offered load increases, the network becomes congested. The *projected load* of a network is defined in Equation 1.

$$\text{projected load} = \frac{\lambda \kappa \sum_{r \in R} TSL_r}{\mu \sum_{e \in E} C_e} \quad (1)$$

The dimensioning algorithm dimensions capacities for each end-to-end connection request pair uniformly over all topological shortest paths for the pair. At routing time, the connections that stay within their topological shortest path graphs (union of all links in the pair’s topological shortest paths) use their own dimensioned resources and introduce minimal interference with other connection pairs. However, if the shortest available path found for a request uses links outside of its topological shortest path graph, the connection is trying to use the resource of other pairs. If the network load is low, we may want to allow the connection as other resource is probably not used often. If the network load is high, we may reject this connection since it blocks the chance of other connections with higher probability. The rest of the paper is to find the best online admission control strategy with into account the opportunity cost of wavelength capacities.

## B. Related Work

As early as the era of ATM networks, Krishnan *et al.* proposed admission control mechanisms (called SDR) for dynamic circuit routed networks [6]. Later, Gawlick *et al.* [7] proposed another admission control mechanism called competitive online routing (COL). The performance of these algorithms were compared by Zhang *et al.* [8]. However, optimizing the parameters for these algorithms is challenging. In parallel with admission control algorithms, least loaded routing (LLD) was studied extensively in the context of ATM broadband networks [9], [10], WDM networks [4], [11]–[16] and general broadband networks [17]–[20]. The goal is to balance network resource utilization through path selection, and to reduce blocking probability. The set of candidate paths can be pre-selected or dynamically discovered. Pre-selected paths allow a simplified signaling mechanism for path setup. On the other hand, dynamically discovering paths offer flexibility and tolerance to traffic fluctuations. The basic idea of online dynamic routing is to pick a path that induces minimum opportunity cost. Many routing algorithms proposed actually calculate the “opportunity cost” implicitly based on the residual capacity. Späth [11] discussed dimensioning, routing and traffic models for dynamic wavelength routed networks. The paper proposed an adaptive path length limit strategy that limits the length of adaptive paths within a multiplier of the topological shortest path length. Their results motivate the investigation of constraining path hops adaptively to traffic load changes. Many LLD algorithms select a path with maximum residual capacity on bottleneck links. Some others, like the fixed-paths least-congestion (FPLC) algorithm for wavelength continuity networks [12], use the sum of total number

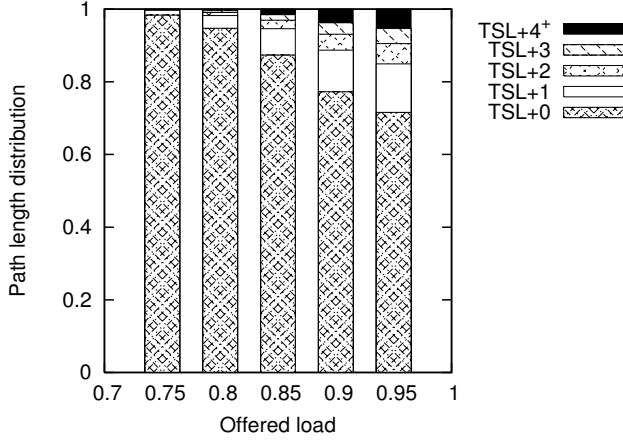


Fig. 1. Distribution of accepted path lengths on ARPANET.

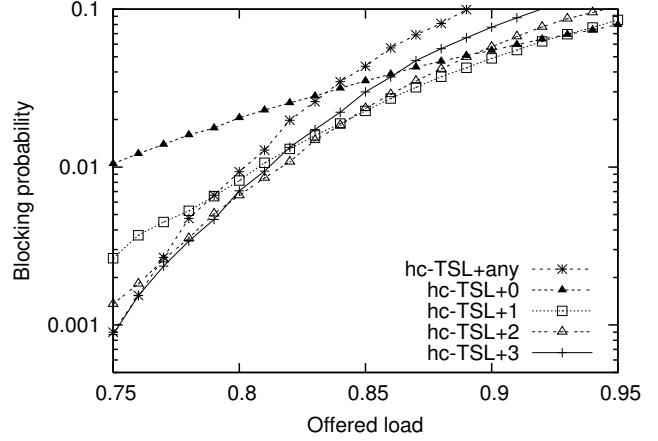


Fig. 2. ASPF with different hop constraints on ARPANET.

of free capacity along the path. Analytical models [12], [19], [20] have been proposed to approximate the blocking probability with LLD-like state-dependent routing for pre-selected adaptive routing. Instead of looking at the absolute residual bandwidth, Zhou and Mouftah [14] proposed a convex function that takes fraction of wavelength utilization to compute the link congestion cost. Wen and Sivalingam [13] proposed a Least Resistance Weight (LRW) routing that uses the available capacity normalized by the maximum link capacity in the network. Mewanou and Pierre [15] proposed another congestion estimation using the inverse of available link capacity. Applying congestion estimation in the context of dimensioned network and building mathematic models to explicitly calculate the opportunity cost are the key differences between this paper and previous studies.

### III. MOTIVATION

The performance of ASPF is not ideal when networks are congested. Network congestion increases along with the increase in the offered load, causing congestion on many links. Admitting connections with excessive path lengths can cause an increase in blocking for future requests. To understand the impact of accepting these connections, a mathematical model is built (Section V) to calculate the expectation of increased blocking on one or two additional links, under various traffic loads. In this section, we illustrate the relationship between path lengths and network loads for dynamic routed connections.

Our simulation methodology is explained in Section VII. ARPANET is used as an example, but other topologies show the same trend. Figure 3(a) shows the dimensioned capacity of ARPANET using Algorithm 1 with 100 random traffic matrices. Each number is one unit of capacity (or one wavelength for simplicity). Figure 1 presents the distribution of extra hops to the TSL of each connection request among accepted connections routed by ASPF at different offered network loads. When the network is lightly loaded (75%), over 98% of the accepted paths use TSL. At 95% offered load, the percentage of TSL paths drops to 70. TSL is defined on individual connections, so two connections with different path lengths may be grouped into the same  $TSL+n$ .

Figure 2 motivates constraining hop lengths for connections depending on the traffic load (or the level of congestion).  $hc-TSL+n$  denotes a hop-constrained routing in which a connection request is admitted if the available shortest path length is less or equal to its  $TSL+n$ . We see that constraining hop lengths reduces blocking in highly loaded networks, but hurts performance when the load is low. Figure 2 compares the blocking probabilities of different hop constraint policies applied on ASPF. The reason for using  $TSL+n$  as a hop limit is to prevent connection pairs with long TSL from being blocked much more frequently.  $hc-TSL+n$  is a similar approach to the adaptive length limit strategy mentioned in [11]. Figure 2 shows that constraining path length reduces network congestion when load is high. However, routing without constraint allows more aggressive use of the network and provides the opportunity to admit more connections when load is low. The two extremes are shown at  $hc-TSL+any$  and  $hc-TSL+0$ . Although it has good low-load performance,  $hc-TCL+3$  provides over 10% blocking at a higher load. Similarly,  $hc-TCL+1$  is attractive at high loads but presents two times more blocking than the best achievable result at low loads. We want to find an adaptive hop constraint routing policy that determines hop limits dynamically

<p>1: <b>for each</b> available shortest path <math>p</math> found for a connection generated by request pair <math>r</math> by ASPF routing <b>do</b></p> <p>2:     Compute the sum of congestion ratio</p> <p style="text-align: center;"> <math display="block">\gamma_p = \frac{1}{ p } \sum_{e \in p} \frac{U_e}{C_e} \quad (2)</math> </p> <p>3: <b>end for</b></p> <p>4: Path <math>q \leftarrow p</math> with <math>\min \gamma_p</math>.</p> <p>5: <b>if</b> <math> q  = TSL_r</math> <b>then</b></p> <p>6:     Accept the connection.</p> <p>7: <b>end if</b></p> <p>8: The congestion estimator for path <math>q</math> is <math>\Gamma_q = \gamma_q</math></p> <p>9:</p> <p>10: <b>if</b> <math>\Gamma_q &gt; \eta</math> <b>then</b></p> <p>11:     Reject the connection.</p> <p>12: <b>else</b></p> <p>13:     Accept the connection.</p> <p>14: <b>end if</b></p>	<p style="text-align: right;"><math>\triangleright</math> Path selection metric varies across routing algorithms</p> <p style="text-align: right;"><math>\triangleright</math> Optimality condition (min or max)</p> <p style="text-align: right;"><math>\triangleright</math> Always accept TSL paths</p> <p style="text-align: right;"><math>\triangleright</math> Different congestion estimator can be used in combining with routing algorithms.</p> <p style="text-align: right;"><math>\triangleright</math> Threshold comparison (greater or smaller depending on optimality condition)</p>
---	---

**Algorithm 2:** Congestion Aware Routing (CAR).

based on the current network state, and potentially achieves lower blocking compared to the best non-adaptive routing algorithms across all loads.

#### IV. LOCAL CONGESTION ESTIMATION AND ROUTING

In this section, we introduce a routing algorithm with threshold-based congestion estimation and admission control. It quickly evaluates local congestion along a dynamically routed path and decides whether this connection should be admitted based on congestion along the path and whether or not the path is longer than the TSL for the request. The technique presented can be applied to other dynamic routing algorithms to reduce blocking probability when they are used on dimensioned networks. In addition, we introduce several approaches that examine the network around a prospective path for congestion estimation. All of our approaches present similar performance, but CAR is the simplest, so we focus on CAR for the rest of the paper.

##### A. Congestion Aware Routing (CAR)

Algorithm 2 shows the threshold-based congestion estimation and routing algorithm (CAR) executed for each connection  $r$ . The degree of congestion on the path of  $\gamma_p$  is computed using Equation 2. The determination of threshold  $\eta$  is discussed in Section V. CAR selects a least loaded path amongst all available shortest paths (other routing algorithms can also be used for this step). As shown in [4], selecting a longer available path in a dynamic network reduces admission opportunities for future connections, the algorithm considers only shortest available paths on the residual network. If an available path is found, with the same length as the TSL of the request node pair, we automatically accept it because it does not use any extra resources (links along the paths with high TSL have been planned with more resources when the network is dimensioned, so the routing algorithm should not bias against them). If not, we accept the request only if the congestion estimation value  $\Gamma_q$  is smaller than a predetermined threshold  $\eta$ . In other words, we accept a request either by no hop constraint or maximum hop constraint of TSL+0, depending on the estimated level of congestion. When a network starts to use longer paths to accept new connections due to traffic load increase, the opportunity cost of routing path increases. More links become unavailable as a result of excess use of resource. To suppress the spread of congestion, when links are highly loaded in a neighborhood, no path of extra hops should be admitted. In addition, we tried to estimate congestion by looking at different local subnets, but found that the path alone already equally good results.

##### B. Congestion Estimation on Other Local Subnets

Congestion estimation for the chosen path  $q$  in CAR algorithm is straightforward. At line 8 of Algorithm 2, the estimator  $\Gamma_q$  is equal to  $\gamma_q$ , which averages the load on the path. In this section, we study congestion estimators

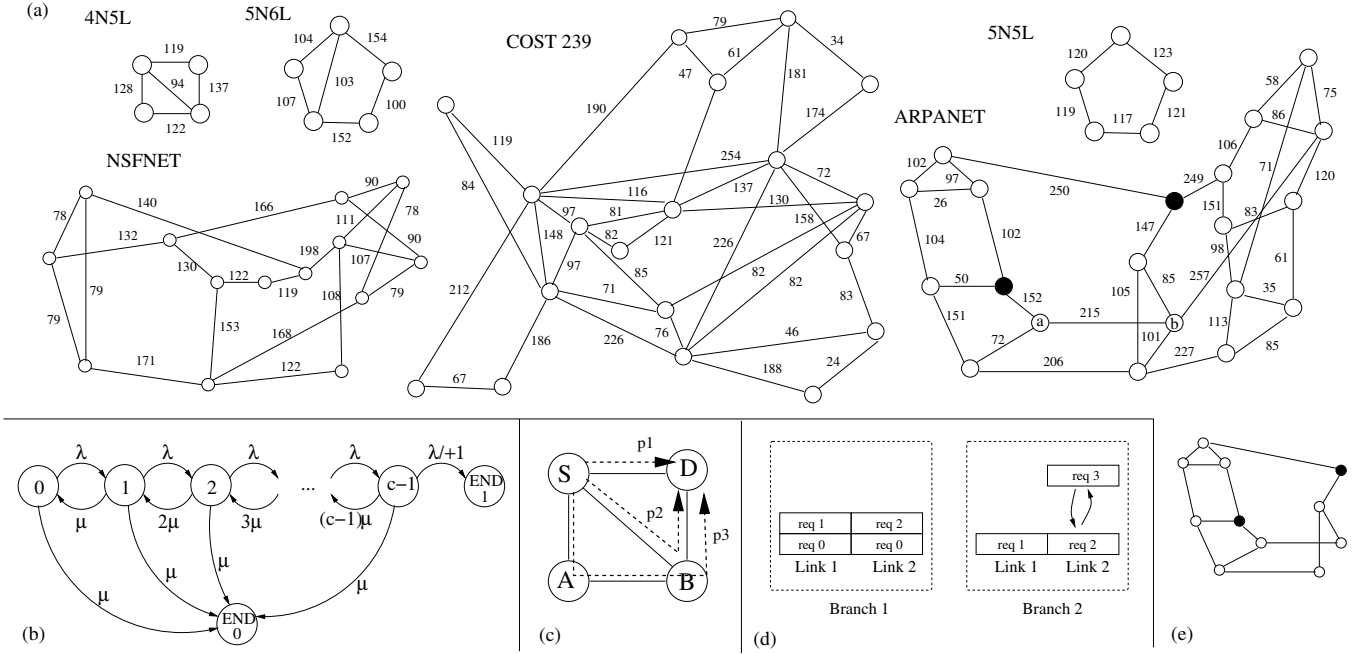


Fig. 3. (a) Network topologies with dimensioned link total capacity. (b) The Markov chain for the one-link model. (c) An illustration of TSL+0 (p1), TSL+1 (p2) and TSL+2 (p3) paths for (S, D). (d) An illustration of unbounded expected difference in blocking for the two-link model. (e) Subgraph SPG2 for the solid pair on ARPANET.

TABLE II  
LLE2/M AND FBE2/M ALGORITHMS.

	line 8
CAR	$\Gamma_q = \gamma_q$
LLE2	$\Gamma_q = \frac{1}{ G } \sum_{e \in G} \frac{U_e}{C_e}, G = SPG2_r \quad (3)$
LLEM	$\Gamma_q = \frac{1}{ G } \sum_{e \in G} \frac{U_e}{C_e}, G = SPGm_r^q \quad (4)$
FBE2	$\Gamma_q = 1 - \frac{f(G, A)}{f(G, C)}, G = SPG2_r \quad (5)$
FBEM	$\Gamma_q = 1 - \frac{f(G, A)}{f(G, C)}, G = SPGm_r^q \quad (6)$

based on a subgraph local to the path. Selection of the subgraph is based on the idea of the Shortest Path Reduced Graph (SPG) that is introduced by [4]. The shortest path reduced graph for a connection pair,  $SPG_r$ , is the union of edges of all topological shortest paths of  $r$ . Four routing algorithms are proposed: LLE2, FBE2, LLEM and FBEM. Each algorithm uses either a Link Load Estimator (LLE) or a Flow Based Estimator (FBE) on subgraph  $SPG2_r$  or  $SPGm_r^q$  for the connection pair  $r$  and the chosen path  $q$ . The notations are explained in the paragraph immediately follows. These algorithms use the same routing algorithm as CAR and the difference only occurs on line 8. The changes on line 8 for LLE2/M and FBE2/M algorithms are summarized in Table II.

The subgraph  $SPG2_r$  is defined as the topological shortest path reduced graph with two-hop extension for connection pair  $r$ . In other words, it is a union of all topological paths of length less or equal to TSL+2 between an

end-to-end node pair. The definition of  $SPG2$  is an extension of the definition of shortest path reduced graph (SPG) that is a union of TSL paths. Figure 3(e) shows an example  $SPG2$  for one node pair (marked solid) on ARPANET. Figure 1 shows that the lengths of over 90% paths taken are within TSL+2 when the acceptance rate of the network is more than 90%. The neighborhood congestion information of the chosen path is included in  $SPG2$  with high probability. Similarly,  $SPGm_r^q$  is the shortest path reduced graph of all path of length less or equal to TSL+ $m_r$ , where  $m_r^q = |q| - TSL_r$ . Unlike  $SPG2_r$ ,  $SPGm_r^q$  is variable to the chosen path  $q$ . It captures the growth of available shortest path length in congested environment.

Congestion estimator LLE (like in Equations 3 and 4) averages the load ratio of all links in the subnet, which is a methodology close to the one used in CAR. The flow based estimator FBE (like in Equations 5 and 6) computes the ratio of flows on the residual subgraph and on the empty subgraph.  $f(G, A)$  is the maximum flow computed on graph  $G$  with current residual capacity  $A$ .  $f(G, C)$  is the flow on an empty network, where  $C$  denotes original dimensioned capacity.

Combining the subgraph selection techniques and the congestion estimation algorithms, LLE2 (Equation 3) denotes LLE on selected subnet  $G = SPG2_i$ , LLEM (Equation 4) denotes LLE with selected subnet  $G = SPGm_i^q$ . FBE2/M (Equation 5-6) are defined in the same fashion. The reader may already noticed that CAR is a special case of LLE with the path  $G = q$ . Our simulation result in Section VII shows that the routing algorithms presented in this section equally effective as the performance of CAR because the average load of a path is unlikely to be much lower (or higher) than the neighborhood graph. Since CAR is the simplest compared to LLE/FBE methods, we consider only CAR in the rest of the paper.

## V. THRESHOLD MODELING AND PREDICTION

To find the right threshold value is essentially to compute the opportunity cost to route a connection. If the resource used to accept a connection causes blocking of more than one connection in future, it would be better off rejecting the connection right away. Otherwise, we should accept the connection. The threshold is the condition of residual network when there is exactly one more connection blocking caused by admission of the current connection.

### A. One-Link Model

We first build an one-link model to calculate the opportunity cost. Our simplified one-link model uses a random link in a network with the assumption that all links have independent traffic. Let the maximum capacity for a link be  $c$ . The aggregated arrival rate to this link is  $\lambda$ , and the departure rate for each connection is  $\mu$ . Let the offered network load be  $l$ . When the network has been dimensioned according to Section II-A and each arrival requests one unit capacity, the load follows Equation 7. We define  $t = 0$  to be the decision time when an arrival requests a unit capacity on the link as an extra hop to its TSL. All future arrivals are assumed to be accepted if there is link capacity available. We ignore future arrivals that use this link as an extra hop.

$$\frac{\lambda}{\mu} = c \cdot l \quad (7)$$

$$\int_0^t P_{c,i}(s) \lambda ds \quad (8)$$

If the link is loaded with  $i$  wavelengths at  $t = 0$ , where  $i < c$ , the expected number of future blocking in an arbitrary time interval  $(0, t)$  can be formulated by Equation 8, where  $P_{c,i}(s)$  is the probability that this link is full at time  $s$  given total capacity  $c$  and starting capacity  $i$ . As time interval increases,  $t \rightarrow \infty$ , the expected blocking is nondecreasing. Therefore, Equation 8 is not bounded.

Regarding the decision on whether to accept or reject the request at time 0, we want to see the difference in future blocking if we choose to use this link (accept the request) or not. Given our two choices, we then have two different initial link states. If we decide to take the extra request, the initial capacity state is  $i + 1$ . If we do not, the initial state is still  $i$ . The expected difference in blocking of two choices is thereby  $\int_0^t P_{c,i+1}(s) \lambda ds - \int_0^t P_{c,i}(s) \lambda ds$ . Since we assume that future arrivals are always accepted if there is free capacity, the difference of probability,  $P_{c,i+1}(s) - P_{c,i}(s)$ , decreases exponentially. Therefore, the difference of the blocking will converge as  $t \rightarrow \infty$  to a finite number, which is denoted as  $d_i^c$  in Equation 9.

$$d_i^c = \lim_{t \rightarrow \infty} \left( \int_0^t P_{c,i+1}(s) \lambda ds - \int_0^t P_{c,i}(s) \lambda ds \right) \quad (9)$$

By viewing the problem slightly differently, we show that  $d_i^c$  ranges from 0 to 1 for any starting state  $i < c$  and capacity  $c$ . The initial capacity is  $i$  for the case where the extra-hop arrival is rejected at time 0, and the initial capacity for the accepting case is  $i + 1$ . Let the link state (wavelengths in use) at time  $s$  be  $L_i(s)$  with initial state  $i$ .  $L_{i+1}(s)$  be the link state for the case where the arrival is accepted. The sequences of further arrivals are the same for both decision branches and they will be admitted as long as there is unused capacity. Therefore, the number of wavelengths in use for the decision branch  $L_{i+1}$  always exceeds the branch of  $L_i$  by 1 until the extra link request leaves  $L_{i+1}$  or  $L_i$  is full. The time when  $L_i$  gets full is exactly the same time when  $L_{i+1}$  has to reject the first future arrival. Two branches then have equal capacity onwards. Otherwise, the extra-hop arrival in  $L_{i+1}$  leaves before  $L_i$  becomes full. The expected future blocking for two branches are equal after they reach equal capacity, as all connections have equal exponential holding time distributions. We use the term END to describe the state when the two branches reach equal capacity. If an END is caused by the case where  $L_{i+1}$  rejected a request (equivalently  $L_i$  gets full), the difference of blocking is 1. Such an END is denoted as *END1*. If an END is caused by the departure of the extra-hop arrival before  $L_{i+1}$  gets the first reject, there is 0 difference in blocking. The END is then denoted *END0*. Equation 9 is the probability that the system ends with END1 given the total capacity and initial state. The probability equals one if the link starts with fully loaded, i.e.,  $i + 1 = c$ . Therefore, the expected blocking difference is a binary random variable with expected value  $d_i^c$ .

Computing  $d_i^c$  directly from  $P_{c,i}$  is difficult because  $P_{c,i}$  is unbounded. However, we can use a continuous Markov chain to compute the expectation. The transition of states is shown in Figure 3(b). The number in each state is the current number of used link capacity for the case of  $L_i$  ( $L_{i+1} = L_i + 1$  always). END1 state is equivalent to the state where  $L_i$  is full (also  $L_{i+1}$  rejects the first request). As the initial link capacity  $i$  ranges from 0 to  $c - 1$ , the system can start at any state except the END states. Again, all connections request one unit capacity, the arrival rate is  $\lambda$ , and the departure rate on the link is  $L_i \mu$  as each connection departs at rate  $\mu$ . From the  $L_{i+1}$  state, the extra-hop arrival at  $t = 0$  may also depart at rate  $\mu$ . If that happens, the system immediately enters END0. Since  $\lambda > 0$  and  $\mu > 0$ , the system will eventually reach one of the two sink states, END1 or END0.  $d_i^c$  from Equation 9 can be computed by finding the probability that the system ends at END1 when starting at state  $i$ . Trivially,  $d_{END1}^c = 1$  and  $d_{END0}^c = 0$ . The probability of the ending state is computed recursively by equation set 10-12. Basically it balances the in/out probability flows at each state. The detailed steps of solving these equations are not shown. Basically, one can derive the general form of  $d_i^c$  after a few substitutions starting from  $d_2^c$  and then the form can be proved by mathematical induction.

$$d_0^c = d_1^c \frac{\lambda}{\lambda + \mu} \quad (10)$$

$$d_{c-1}^c = \frac{\lambda}{\lambda + c\mu} + d_{c-2}^c \frac{(c-1)\mu}{\lambda + c\mu} \quad (11)$$

For  $0 < i < c - 1$ ,

$$d_i^c = d_{i+1}^c \frac{\lambda}{\lambda + (i+1)\mu} + d_{i-1}^c \frac{i\mu}{\lambda + (i+1)\mu} \quad (12)$$

Solving these equations by mathematical induction, we get the general form for  $d_i^c$ .

$$d_i^c = \frac{\sum_{k=0}^i \frac{i!}{(i-k)!} \left(\frac{\mu}{\lambda}\right)^k}{\sum_{k=0}^c \frac{c!}{(c-k)!} \left(\frac{\mu}{\lambda}\right)^k} \quad (13)$$

Combined with Equation 7 to cancel  $\lambda/\mu$ , we have

$$d_i^{c,l} = \frac{\sum_{k=0}^i \frac{i!}{(i-k)! (cl)^k}}{\sum_{k=0}^c \frac{c!}{(c-k)! (cl)^k}} \quad (14)$$

Equation 14 shows that the expected difference in future blocking on a link depends on total capacity  $c$ , initial capacity  $i$  and traffic load  $l$ . Figure 4 shows how  $d_i^{c,l}$  varies with  $l$  and  $i/c$  when  $c = 100$ .

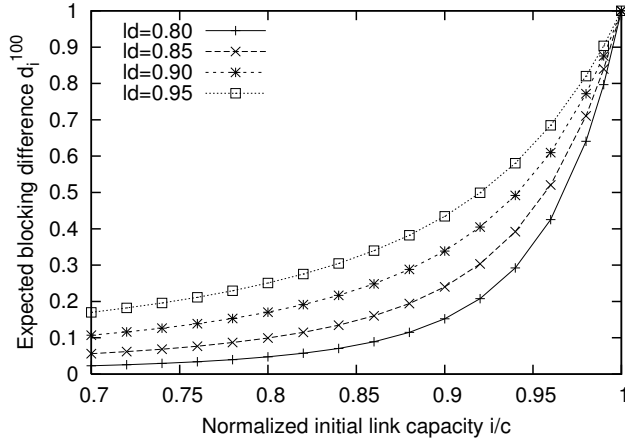


Fig. 4. The one-link model with various loads at  $c = 100$ .

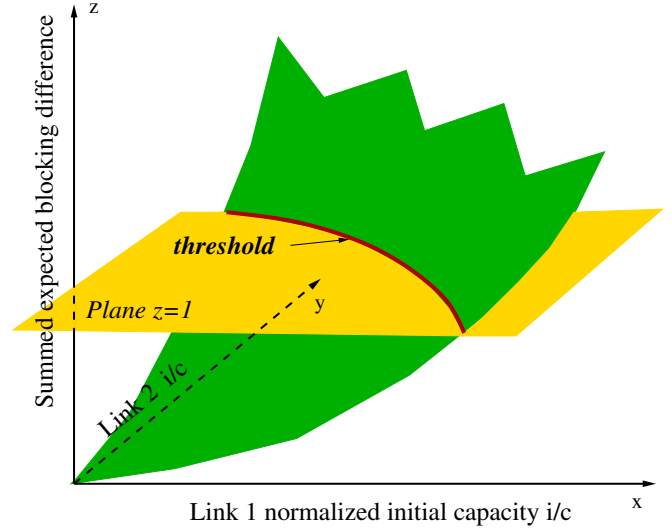


Fig. 5. Illustration of threshold prediction using the two-link independent model.

### B. Two-Link Independent Model

The one-link model is not useful because the expected blocking difference is always less than one. We need to consider two links and find the condition where the expected blocking difference can be greater than one. When a TSL path is not available, an online routing algorithm may need two or more additional links to detour around unavailable links. Figure 3(c) shows an example. Path p1 is the TSL+0 path for the connection S to D. p2 is a TSL+1 path when the direct link S-D is full. p3 is a TSL+2 path that uses two additional links to detour around unavailable links S-D and S-B. In this section, we develop a model of two extra links with independent traffic assumption. The next section discusses the case in which joint traffic is presented across two links. However, the independent traffic model reveals important information that allows us to predict the optimal CAR threshold.

In the two-link case, the expected blocking difference is not bounded by one. Figure 3(d) illustrates an unbounded situation. Suppose there are two links each with capacity  $c = 2$ . Req 0 is the initial extra two-link request. Later arrivals, req 1 and req 2, used the remaining capacity. Future arrivals, such as req 3, arriving on link 2 before other connections leave, can increase the number of blocking differences between the req 0 accepted and rejected cases to an arbitrary number. However, the probability that infinite number of an arrival request during the finite holding-time of the current connections is sufficiently small that expected value remains finite.

Assuming that each link behaves independently, the expected blocking difference is equivalent to the sum of  $d_i^{c,l}$  for each one-link mode. We can construct a three-dimensional plot with the  $x$ - and  $y$ -axis representing the normalized initial capacity of each link. Figure 5 illustrates the construction. The  $z$ -axis is the expected difference in blocking computed by adding the results of two one-link models with the corresponding traffic load and link capacity. If the sum is greater than 1, accepting the connection at  $t = 0$  is expected to cause the link to reject more than one future connection. The threshold curve is shown in Figure 5 on the  $z = 1$  plane. In this case, the initial request should be rejected. Equation 15 defines  $\zeta_l$ , the contour curve of the 3D graph at load  $l$  on plane  $z = 1$ . Figure 6 shows  $\zeta_l$  (with legends “mod  $ld=l$ ”) at various loads; these curves show the ideal threshold at load  $l$ . The lower the load, the higher the threshold can be to allow aggressive admission. The threshold is lower at higher loads because fewer resources are available for routing excessive long paths. At routing time, based on the current residual graph, we know the operating point on the plane but not which  $\zeta_l$  curve should be used, since we may not know the offered load  $l$ . Therefore, the threshold used at routing time must work for all loads.

$$\zeta_l = \{(x, y) | d_{xc_1}^{c_1,l} + d_{yc_2}^{c_2,l} = 1, 0 < x < 1, 0 < y < 1\} \quad (15)$$

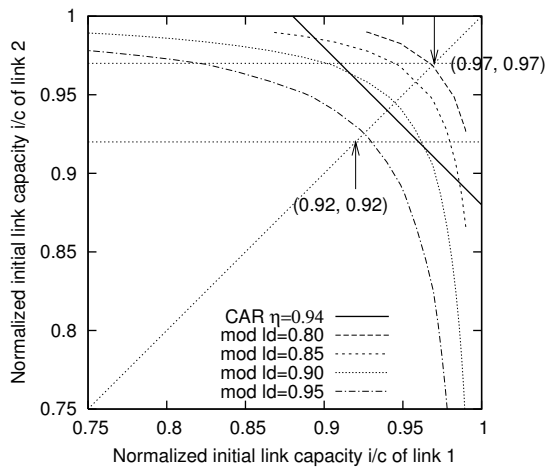


Fig. 6. Two links with independent traffic ( $dp = 0$ ).  $c_1 = c_2 = 100$ .

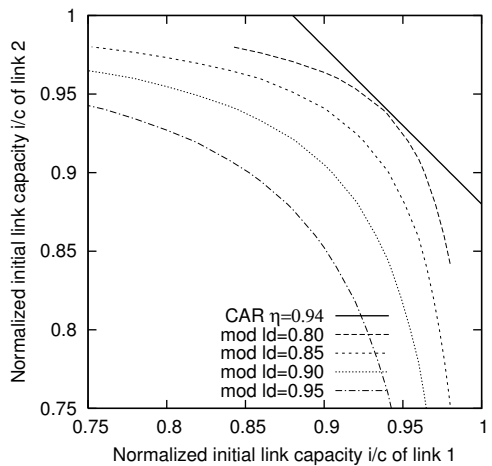


Fig. 7. Two links with independent traffic ( $dp = 0$ ).  $c_1 = c_2 = 50$ .

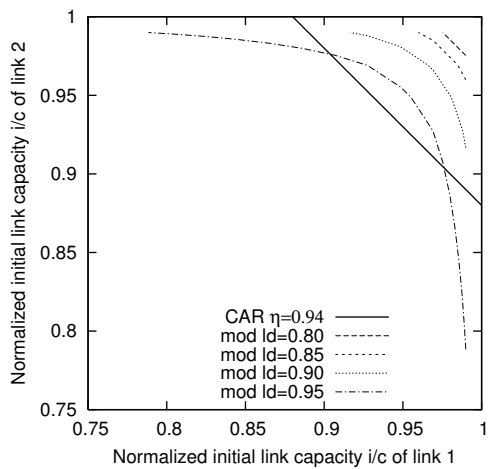


Fig. 8. Two links with independent traffic ( $dp = 0$ ).  $c_1 = c_2 = 200$ .

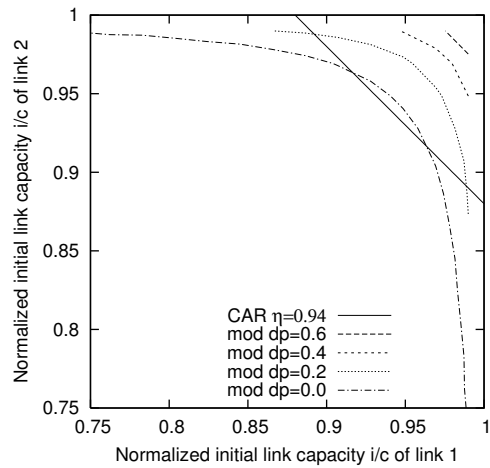


Fig. 9. Two links with various dependent traffic.  $c_1 = c_2 = 100, ld = 0.9$ .

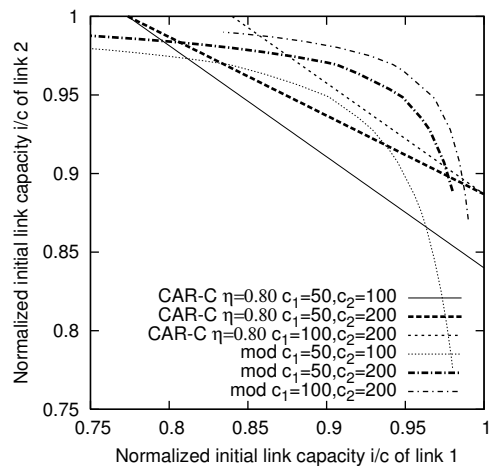


Fig. 10. Two links with independent traffic and different capacity.  $ld = 0.9$ .

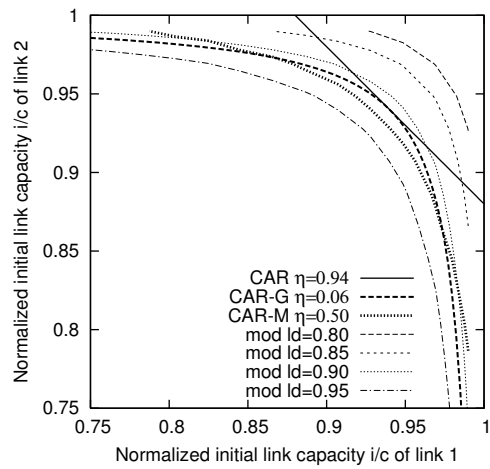


Fig. 11. Two links with independent traffic ( $dp = 0$ ) and various routing algorithms.  $c_1 = c_2 = 100$ .

### C. Two-Link Dependent Model

In general there are three types of traffic on two links: traffic using only link 1, traffic using only link 2, and traffic using both links. A two dimensional Markov process suffices to represent the new ending situations, but does not terminate until accept/reject cases are loaded with the same amount of two-link traffic and the same amount of one-link traffic on each link. For example, if link 1 of the accept case carries 50 one-link connections and 20 two-link connections, link 1 of the initial rejection case should also have 20 two-link connections and 50 one-link connections. Other combinations that occupy the same total resources may diverge due to subsequent events. Unfortunately the complexity of the 2-D Markov chain grows quickly: the Markov chain of two links of capacity 2 contains 42 states and 203 transitions. We use simulation here to solve the problem. Figure 9 shows that  $z = 1$  contour line of the simulation results for various traffic dependency at offered load 0.9. Traffic dependency  $dp$  is the fraction of two-link traffic amongst all arrivals. When  $dp = 1$ , all connections use both links, the model reduces to a one-link model. When  $dp = 0$ , only the initial request uses both links, and the model is equivalent to the two-link independent model. The only joint event is the departure of the initial request. The result shows that the threshold drifts upwards in  $x$  and  $y$  when traffic dependency increases, becoming a point at (1, 1) for  $dp = 1$ .

In a real dynamic mesh network, the traffic dependency among two links is difficult to track. The flow pattern on the links not only depends on the original dimensioned traffic but also on the usage from other connections as alternative routes. We see that the threshold computed under independent assumption could be under estimated in a network with high traffic correlation. However, the threshold estimated by the independent model is good enough on typical topologies because long chains of nodes (which might create dependent traffic) are rare in backbone networks.

### D. Thresholds Estimation

It is worth noting that each ideal threshold curve in the 2D graphs is optimal for a fixed and homogeneous global traffic load  $l$ . In practice, however, we do not know the exact offered traffic load on the part of network at the time when we route a path through it. Basically, the congestion metric provides an estimation of the local load and compares it with the threshold for average load.

We can identify the congestion threshold from the 2D contour line graph. Assume that the residual capacities on the two links are the same. In this case, we can draw a line  $x = y$  across the plane in Figure 6. Then, the contour of load 0.80 intersects the line around (0.97, 0.97), and the contour of load 0.95 intersects around (0.92, 0.92). Contours for loads less than 0.80 are hardly visible on the plane: when the network is too lightly loaded, admission control is rarely needed. Therefore, the threshold should be a value in the range of [0.92, 0.97] if the network is operating around offered load 0.75 to 0.95. The average threshold is about 0.94. Simulation results suggest a threshold 0.94 as the optimal value for CAR. Assuming homogeneous load on all links in the path, CAR with threshold 0.94 can be represented as a line  $\frac{1}{2}(x + y) = 0.94$  in Figure 6. As shown in Figures 7-8, the model predicts that the threshold should increase when the network scales. Simulation results shown later verify the prediction of such movement.

The result of the one-link model is useful in studying the two-link case with different capacity on each link. Figure 10 shows an example with link independent traffic load 0.9 and link capacity  $c_1 = 50$  and  $c_2 = 100$ . The threshold becomes lower when moving close to the axis of link with smaller total capacity. The same trend is observed with other traffic loads too. In the next section, we explain other routing algorithms that better approximate these types of curves.

## VI. IMPROVED ROUTING AND CONGESTION ESTIMATION

The two-link model was useful discovering better routing algorithms. Figures 6-8 show the ideal curves a routing algorithm should achieve. Our original CAR algorithm is only a linear approximation to the curve. Naturally, we want to see if other routing algorithms can be found by fitting the curves. Three algorithms are presented: CAR-G, CAR-C, CAR-M.

All improved algorithms share most of the steps with Algorithm 2 except line 2 (computation of  $\gamma_p$ ), line 4 (picking minimum or maximum  $\gamma_p$ ) and line 9 (comparison criteria). The changes are summarized in Table III.

**CAR-G** uses a geometric mean of the link availability ratio to compute  $\gamma_p$  by Equation 16 in line 2 of Algorithm 2. Since it uses available capacity ratio, the threshold  $\eta^G$  instead is a maximum threshold. CAR-G with optimized

TABLE III  
CAR-X ALGORITHMS.

	line 2	line 4	line 9
CAR	$\gamma_p = \frac{1}{ p } \sum_{e \in p} \frac{U_e}{C_e}$	$\min \gamma_p$	$\Gamma_q > \eta$
CAR-G	$\gamma_p = \left( \prod_{e \in p} \frac{A_e}{C_e} \right)^{1/ p }$ (16)	$\max \gamma_p$	$\Gamma_q < \eta^G$
CAR-C	$\gamma_p = \frac{1}{ p } \sum_{e \in p} \frac{A_e}{\sqrt{C_e}}$ (17)	$\max \gamma_p$	$\Gamma_q < \eta^C$
CAR-M	$\gamma_p = \frac{1}{ p } \sum_{e \in p} d_{U_e}^{C_e, \frac{U_e}{C_e}}$ (18)	$\min \gamma_p$	$\Gamma_q > \eta^M$

TABLE IV  
PERFORMANCE COMPARISON OF ALL ALGORITHMS.

	CAR	CAR-G	CAR-C	CAR-M	SDR-orig [6]	SDR	COL-orig [7]	COL
Lowest blocking (comparable)	yes	yes	yes	yes	no	no	yes	yes
Threshold robust across major topologies	yes	yes	yes	yes	N/A	no	no	yes
Threshold robust to capacity scaling	no	no	yes	yes	N/A	no	no	no
Routing time relative to ASPF	$\approx 1.5x$	$\approx 1.5x$	$\approx 1.5x$	$\approx 1.5x$	$> 30x$	$> 30x$	$\approx 2x$	$\approx 2x$

simulation threshold 0.06 translates to the equation  $((1-x)(1-y))^{1/2} = 0.06$  and the curve exactly cuts the line of CAR threshold 0.94 in Figure 11. The curve of CAR-G shows the best fitting shape of the ideal threshold curves.

**CAR-C** uses Equation 17, which normalizes available capacity by the statistical deviation of the total capacity. Remember that the total capacity is determined by 100% traffic load. Square root of the total capacity approximates the expected standard deviation of the resource demands from the total traffic. If the available capacity is greater than the deviation of total capacity, the link can be considered empty. Otherwise, the transient traffic demand likely exceeds the average. Figure 10 shows the equation  $\frac{1}{2}(\sqrt{c_1}(1-x) + \sqrt{c_2}(1-y)) = 0.8$  for CAR-C with optimized simulation threshold  $\eta^C = 0.8$  with link capacity  $c_1$  and  $c_2$ . CAR-C lines are able to automatically adjust to the different of total capacities on each link to match the trend of the ideal curves from the model. CAR-C's threshold is much less sensitive than CAR, as the denominator (square root of the total capacity) in each term is smaller.

**CAR-M** is directly inspired by the two-link independent model. Equation 18 computes the average of the current expected blocking difference  $d_i^{c,l}$  on each link with estimated load  $l = U_e/C_e$ , capacity  $c = C_e$ , and initial capacity  $i = U_e$ . The optimized simulation threshold  $\eta^M = 0.5$  gives the curve  $\{(x, y) | \frac{1}{2}(d_{xc_1}^{c_1, x} + d_{yc_2}^{c_2, y}) = 0.5, 0 < x < 1, 0 < y < 1\}$  shown in Figure 11. Since the computation of  $\gamma_p$  takes the total capacity into account, the threshold remains the same as link capacity scales.

## VII. SIMULATION RESULTS

In this section, we present simulation results of the threshold-based congestion awareness routing algorithms on different networks. CAR-x routing algorithms use little computation, improve call blocking across a wide range of traffic loads, and provide blocking probabilities that are less dependent on path length than those of previous algorithms. CAR-x algorithm thresholds can be identified analytically, and can thus be robust to changes in topology, capacity, and other factors. A comparison of CAR-x algorithms is summarized in Table IV. Among them, CAR-M has the most robust threshold with little additional computing complexity.

### A. Simulation Setup

Our simulation uses a simplified model so that the important features of our algorithms can be shown in a straightforward manner. However, our assumptions do not limit the practical applicability of the routing algorithms in a more complex environment, where these assumptions may not hold. First, we use a non-preemptive connection model. Once a connection is set up, it persists for the entire hold time. We work on an OEO network (or equivalently, an optically transparent network with full wavelength conversion at each node) with a centralized controller. We further assume that inter-arrival and hold times for connections allow sufficient time for signaling, such that all new connections are routed based on up-to-date residual capacity information. All nodes in the core are edge nodes that generate bi-directional connection requests. The arrival rate for each connection request is chosen uniformly from 1 to 10 per unit time for each traffic matrix evaluated. Since the load in Poisson model can be varied using only the arrival rate (Equation 1), we use the same departure rate and request capacity (one unit) for all connections. The initial network capacity is dimensioned at 100% projected load using the basic dimensioning algorithm (Section II-A). Figure 3(a) shows the allocation of capacity on all networks dimensioned with average link capacity 120. In our experiment, the blocking probability is measured by sampling 5000 arrivals after the network is in steady-state (usually after 20,000 warm up requests). Each data point is an average of multiple runs of randomly chosen traffic matrices. The 95% confidence interval is within  $\pm 5\%$  of the data value. We pay special attention to the blocking probability ranging from 0.1% to 10% because higher blocking rates indicate an overloading of the network, which is impractical.

### B. Threshold Study on Dimensioned Small Networks

To formally prove that an optimal threshold works for all backbone topologies is difficult, as the analytical approach is NP-hard due to inter-path dependence. We attempt to approximate these dependencies by studying some small topologies that are similar in structure to the geographic elements from which most backbone networks are composed. If the traffic interdependencies on large networks have patterns similar to those of small topologies, the optimal threshold values for these small topologies will also serve well on the larger networks.

We start with topologies 4N5L and 5N6L from Figure 3(a). These two topologies have a strong connection to bigger networks. For example, the connections on 4N5L and 5N6L mostly utilize one or two additional links if one link is fully loaded. The extra length of dynamic shortest paths in ARPANET in Figure 1 follows a similar distribution. Alternative paths in COST also have the same pattern as shown in the small networks (using a straddle link or a one or two hops longer neighboring path). Figures 12-13 show that CAR 0.94 threshold works well on dimensioned 4N5L and 5N6L networks, although the impact of varying the threshold is minor, as the choice of longer paths is limited on small networks.

However, the optimal threshold is lower on ring topologies. Figure 14 shows that the optimal threshold is 0.85 on ring topology 5N5L. The difference occurs because most connections on the loop only have one TSL path. If the TSL path is not available, a connection tries to use the opposite ring direction, which lies outside of the SPG originally dimensioned for this request pair to use. This path is typically much longer than the TSL. In this case, the threshold should be lowered to limit admission more strictly. Fortunately, such ring topologies are practically avoided in mesh backbones due to reliability and performance issues (traffic overflow paths or backup paths can become too long to be effective). On well-designed mesh networks that include more straddle links, 0.94 is a robust threshold for CAR (more results in the next section). Further, the result of 5N5L can be useful to adjust the ideal threshold on a particular topology that contains big loops. As ARPANET and NSF contain more ring-like structures than COST, their optimal simulation thresholds tend to move down at high loads (Figures 15 and 17), but 0.94 is still the optimal threshold across all loads.

### C. Performance on Dimensioned Backbone Mesh Networks

The analytically computed best threshold works on major backbone mesh networks dimensioned by Algorithm 1. Figures 15-17 show that the optimal simulation threshold for CAR: i.e., the threshold that provides the best improvement in blocking at all loads, falls within a small range around 0.94 on different topologies. The results of ASPF hc-TSL+x routing algorithms (same from Figure 2) are shown for comparison. Only the best ASPFs (forming the lowest bound in blocking) on each network are plotted. In this paper, ARPANET, COST and NSFnet

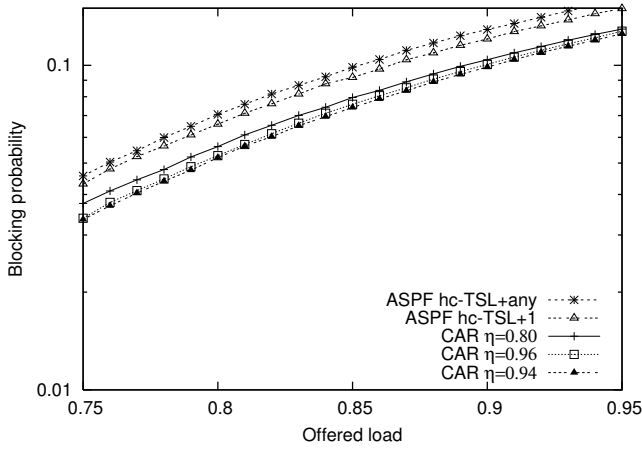


Fig. 12. CAR with different thresholds on 4N5L compared with the best ASPFs.

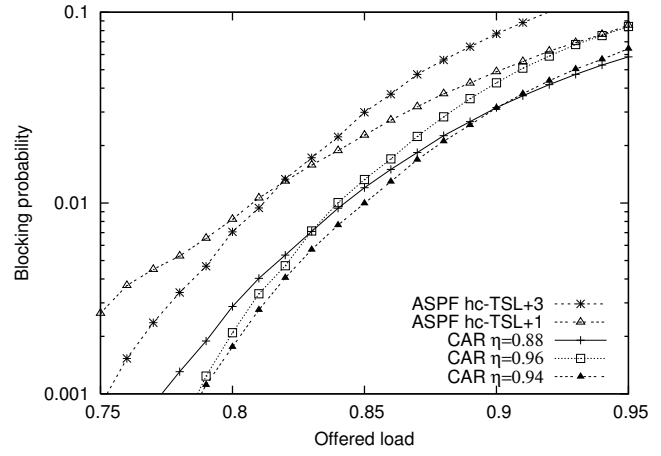


Fig. 15. CAR with different thresholds on ARPANET compared with the best ASPFs.

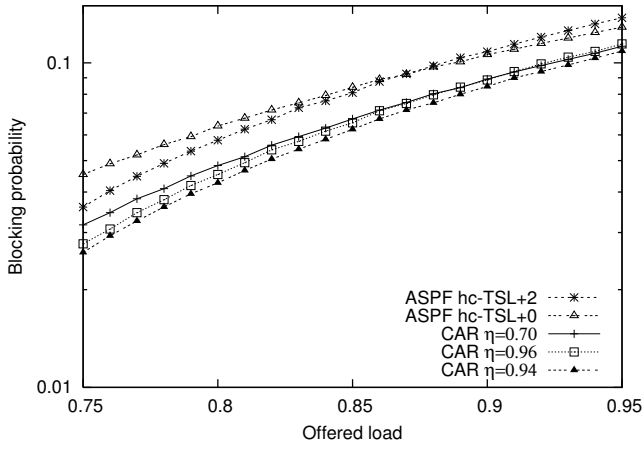


Fig. 13. CAR with different thresholds on 5N6L compared with the best ASPFs.

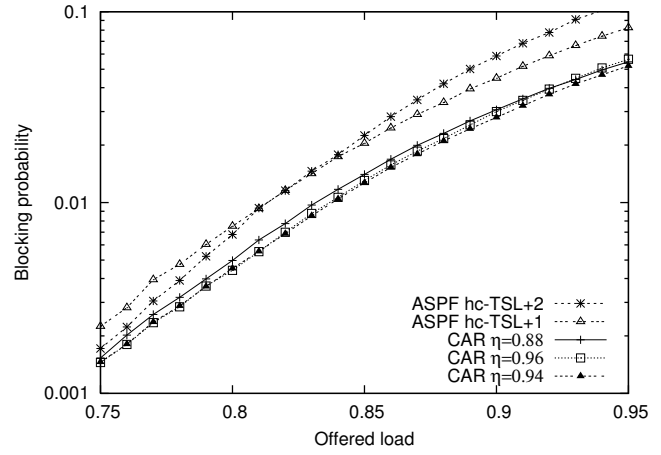


Fig. 16. CAR with different thresholds on COST compared with the best ASPFs.

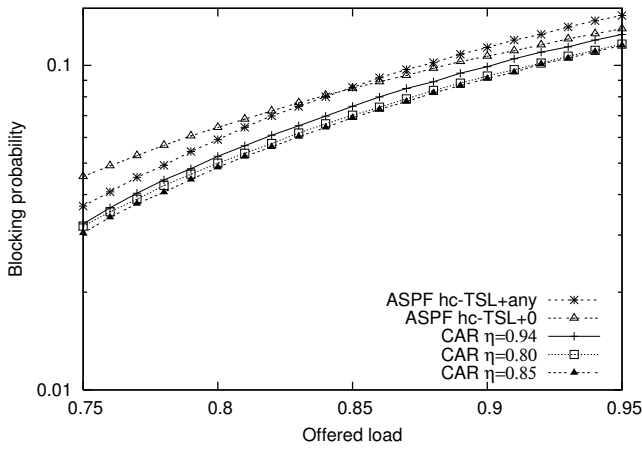


Fig. 14. CAR with different thresholds on 5N5L compared with the best ASPFs.

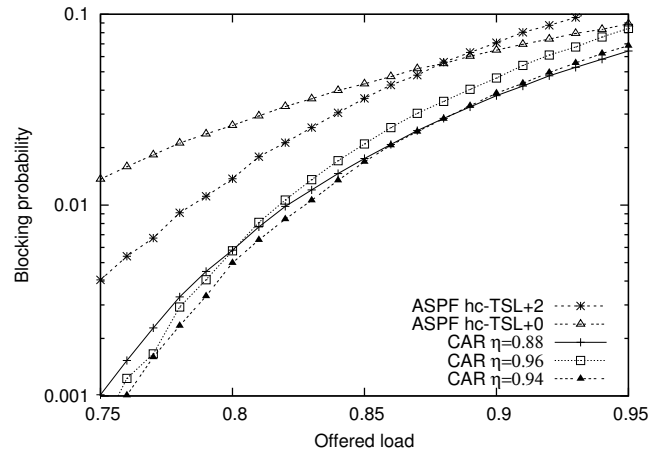


Fig. 17. CAR with different thresholds on NSF compared with the best ASPFs.

are used to illustrate our results, but similar results were also obtained on other topologies (NJ LATA, National). The thresholds of other CAR-x routing algorithms are also stable on major backbones.

Figure 18 compares CAR with one of the best flow-based routing algorithms, Reduced Flow Routing (RFR) [4]. The result shows that CAR reduces blocking by over 60% of RFR at high load (>90%) and is comparable at low load. The figure also demonstrates the applicability of our congestion estimation and admission control techniques to other existing routing algorithms (such as ASPF, LLD, etc). RFR-CAR is an algorithm that uses RFR to pick a path  $q$  (instead of the steps in line 1–4 of Algorithm 2), then uses the rest of the CAR procedure for admission control. We find that RFR-CAR further improves RFR and reaches the same performance as CAR. Later, we show that CAR is more than an order of magnitude faster than RFR-CAR.

CAR’s admission control algorithm reduces blocking for connections of all different hops. Figure 19 shows the blocking probability at offered load 0.85 per topological shortest path length group (connection requests with the same TSL) for ASPF, RFR, RFR-CAR, CAR and CAR-M. The TSLs of ARPANET ranges from 1 to 6. CAR reduces blocking for each TSL group improved by more than 1/3 in RFR-CAR. The blocking probability for shorter connections is higher than for longer connections in CAR and CAR-M, as shorter connections are more likely to be rejected when the local area around the route is congested, while longer connections may still be accepted if only a small part of the route crosses congested areas.

Figure 20 compares CAR-M with other congestion estimation algorithms using different local subnets. The results of LLE2/M and FBE2/M are shown at their best thresholds. All our algorithms show the same result, and demonstrate that evaluating the load on the picked path is usually sufficient to estimate the congestion level accurately for the network neighborhood. The flow based algorithms have a higher ideal threshold than the link load based one because maximum flow only counts the flow from available links in the subgraph, while average link load sums the load all links (available and unavailable). With the same network state, the  $\Gamma_q$  computed by FBE tends to be higher than LLE.

All CAR-x algorithms perform similarly at their best thresholds in Figures 21-23, showing that threshold selection can be performed independent of topology. Figures 24-25 show stability of thresholds in CAR-C and CAR-M upon capacity scaling. ARPANET is dimensioned at average link capacity 60 and 600 respectively. The best threshold for CAR drifts down as the capacity reduces. This phenomenon was predicted by our model in Figures 7-8. The threshold for CAR-G also drifts with the capacity change for the same reason. However, CAR-C and CAR-M have the ability to automatically adjust the curve at different total link capacity and thus the same thresholds are robust to the scaling of network capacity.

Finally, Figure 26 compares the routing time per connection request. CAR-x algorithms are more than 10 times faster than RFR, one of the fastest flow-based routing. The results are obtained from a Linux desktop with a 3.20GHz P4 CPU. CAR (CAR-G/C are similar), CAR-M, and FBE2 (LLE2, LLEM and FBEM are similar) are within 10 microseconds of ASPF on the same network. CAR-M is implemented with a fast lookup table having all  $d_i^{c,l}$  value precomputed for each link.

#### D. Performance on Misdimensioned Networks

In this section, we briefly study the performance of routing algorithms on networks that are not well dimensioned. We show that CAR-G and CAR-M are more robust to network misdimensioning than CAR and CAR-C.

Network misdimensioning can happen in practice for many reasons. One possible cause is that the network provider decides to upgrade the capacity for a couple of links but not the others. Some well-known link load based algorithms are robust to this situation, such as Widest Shortest path Routing (WSP), which selects a path with maximum available capacity on the most congested link. In this case, the upgraded link is unlikely to affect the decision process since it is rarely the most congested link. However, some other link load based algorithms could especially suffer from such upgrading because they will try to route more traffic through the upgraded link as a result of its having more available capacity, while the surrounding unupgraded links are not able to support the increased amount of traffic. To illuminate such effects, we create an misdimensioned network by doubling the total capacity of link (a, b) on ARPANET (Figure 3) and without changing capacity in the rest of the network. All algorithms are compared in Figure 27. The results of WSP and CAR-M on original dimensioned ARPANET are also shown for comparison. WSP, CAR-G, and CAR-M are found to be robust because the results from both networks are the same. But CAR and CAR-C have increased blocking for about 16% at high loads. CAR and

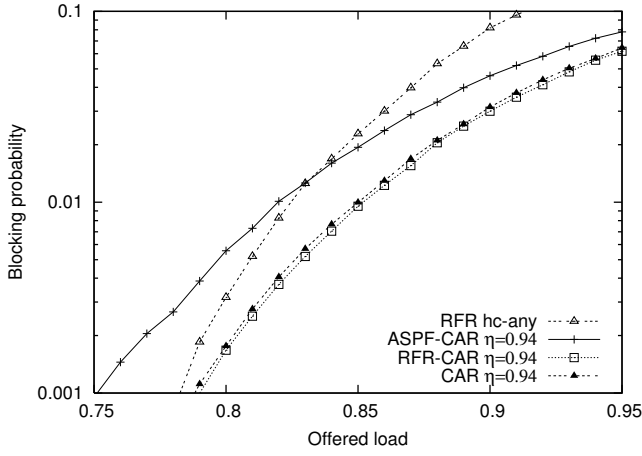


Fig. 18. CAR with the best thresholds on ARPANET compared with other algorithms.

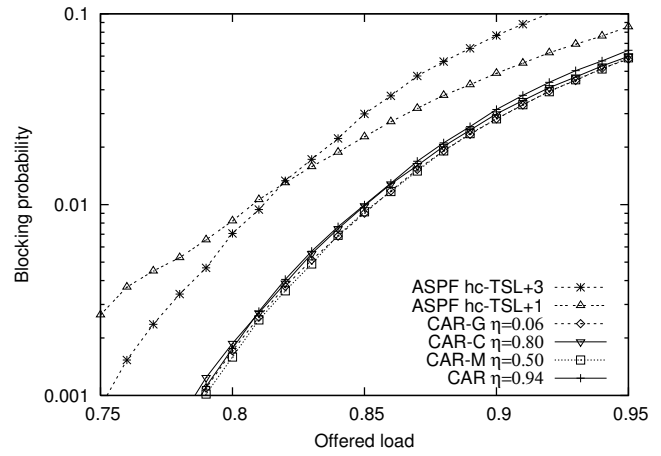


Fig. 21. CAR-G/C/M with the best thresholds on ARPANET compared with the best ASPFs.

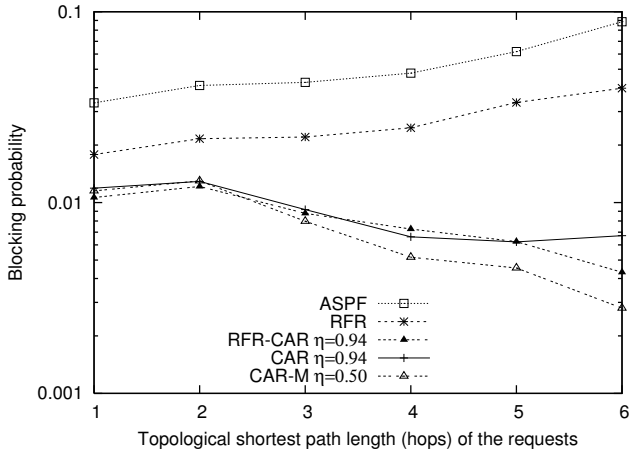


Fig. 19. Blocking probability in group of the TSL of connections on ARPANET at offered load 0.85.

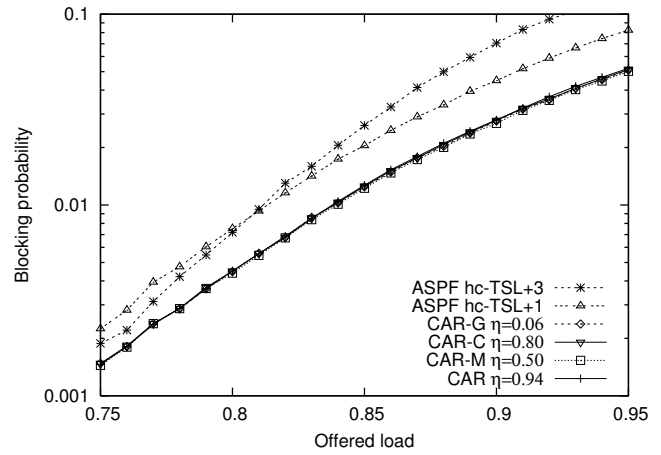


Fig. 22. CAR-G/C/M with the best thresholds on COST compared with the best ASPFs.

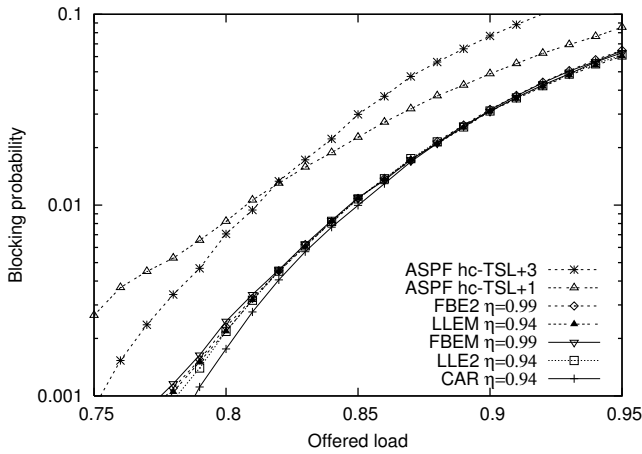


Fig. 20. LLE/FBE/CAR with the best thresholds on ARPANET compared with the best ASPFs.

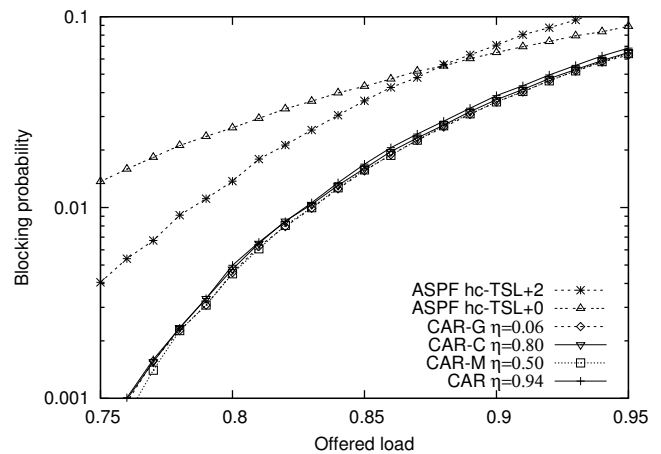


Fig. 23. CAR-G/C/M with the best thresholds on NSF compared with the best ASPFs.

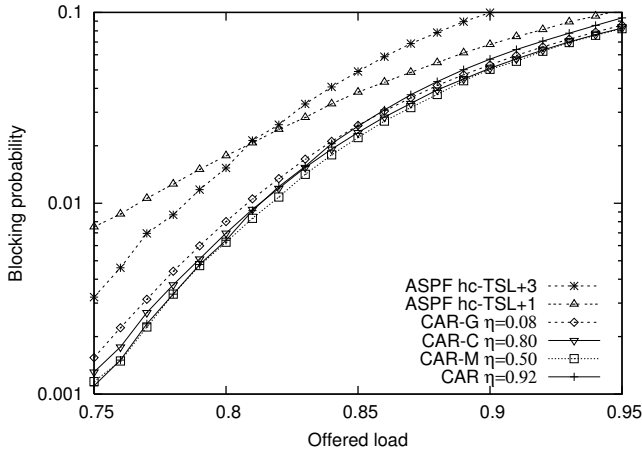


Fig. 24. CAR-G/C/M with the best thresholds on ARPANET with average link capacity 60 compared with the best ASPFs.

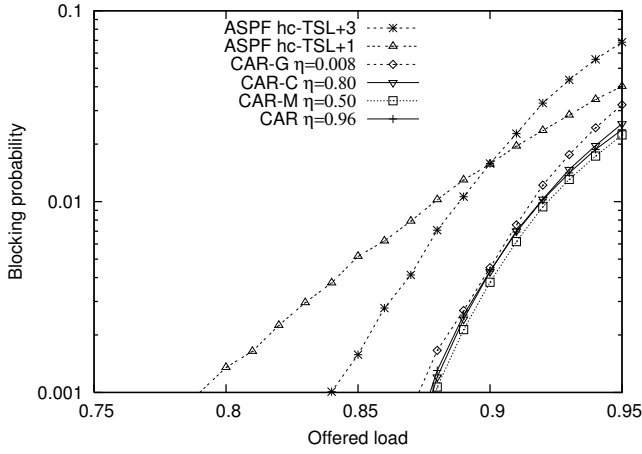


Fig. 25. CAR-G/C/M with the best thresholds on ARPANET with average link capacity 600 compared with the best ASPFs.

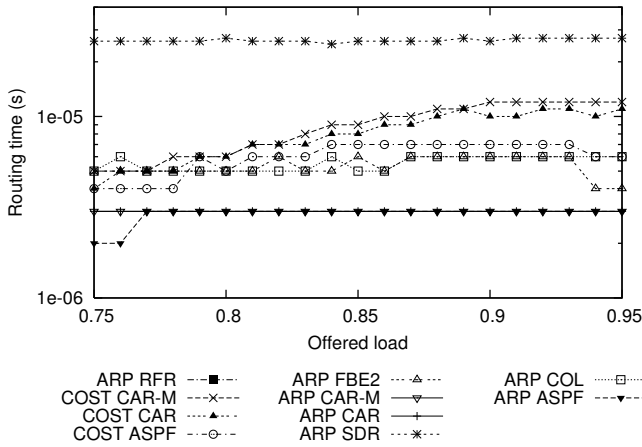


Fig. 26. Comparison of routing time per connection request. RFR on ARPANET requires at least  $200\mu s$  over the range of loads shown.

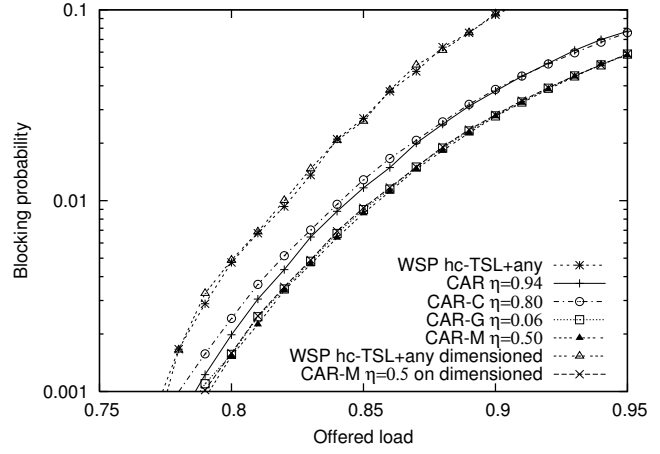


Fig. 27. CAR-G/C/M on misdimensioned ARPANET with total capacity of link (a,b) in Figure 3 doubled.

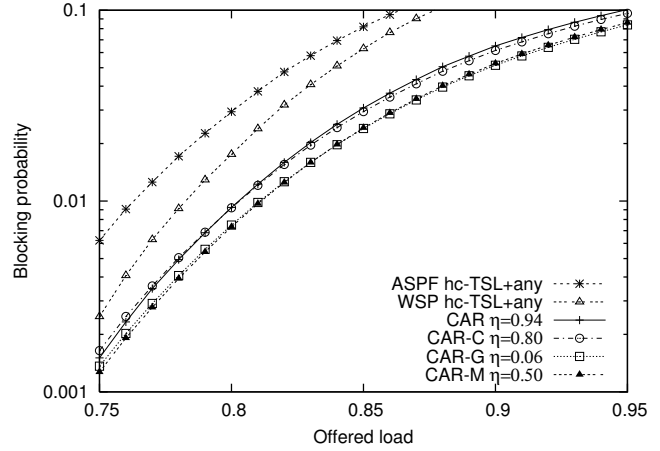


Fig. 28. CAR-G/C/M on misdimensioned ARPANET.  $\alpha = 0.2$ .

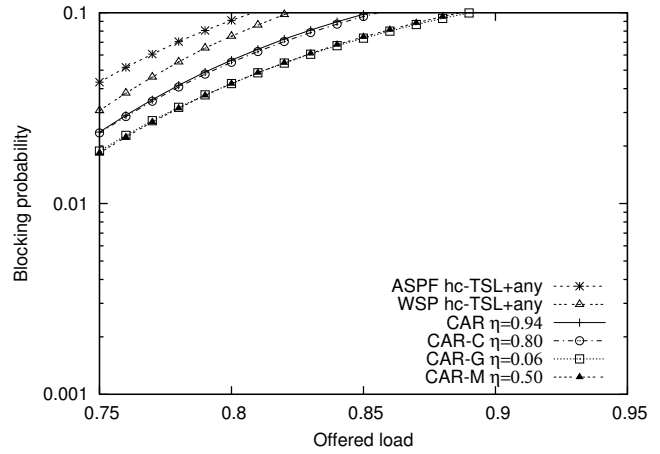
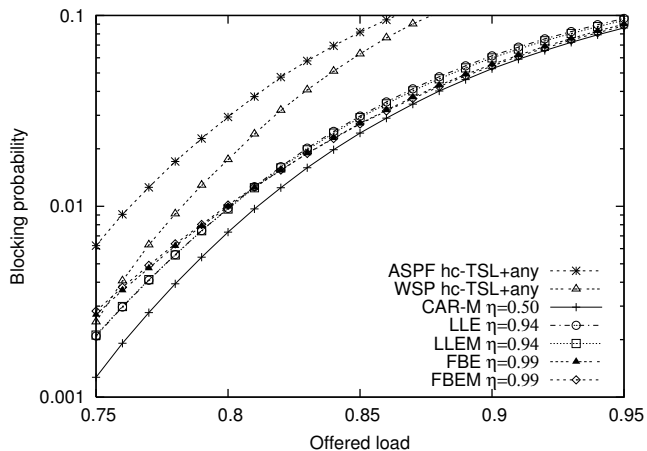
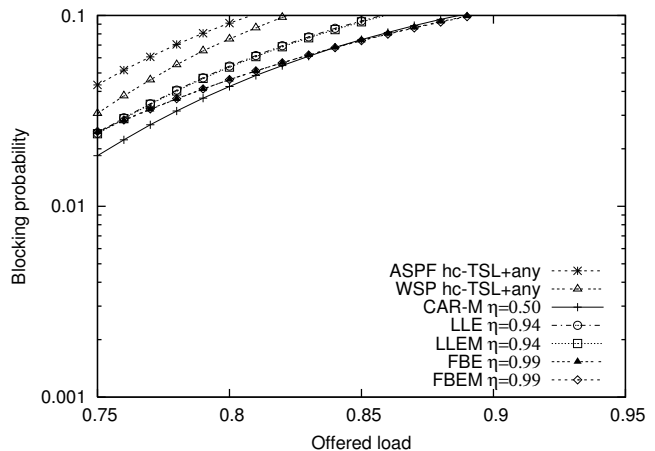


Fig. 29. CAR-G/C/M on misdimensioned ARPANET.  $\alpha = 0.4$ .

Fig. 30. LLE/FBE/CAR on misdimensioned ARPANET.  $\alpha = 0.2$ .Fig. 31. LLE/FBE/CAR on misdimensioned ARPANET.  $\alpha = 0.4$ .

CAR-C are not robust because both of them use a linear sum of link load ratio for all the links in the path. If one of the links presents an exceptionally low load (e.g., the link with doubled capacity), the estimated congestion level is greatly reduced due to this single link. In contrast, CAR-G does not suffer from this defect because the factors of highly loaded links (with a lower  $A_e/C_e$  ratio) play more heavily in a geometric sum than lightly loaded links. For CAR-M, links of higher load have more effect on the sum of  $d_i^{c,l}$ 's because the one-link model (Figure 4) shows that the probability is exponential with respect to the traffic load and increases sharply after 90% traffic load.

Figure 28-29 further shows that CAR-G and CAR-M also perform better than other algorithms in general misdimensioned networks. In this experiment, each link of ARPANET is dimensioned randomly in a uniform range  $[\max(\lfloor C_e(1 - \alpha) \rfloor, 0), \lceil C_e(1 + \alpha) \rceil]$ , where  $C_e$  is the original dimensioned capacity shown in Figure 3 and the real number  $\alpha \in [0, 1]$  denotes *misdimensioning factor*. The network is more likely misdimensioned as  $ub$  increases. We average the results of many randomly misdimensioned network for each  $\alpha$  so the 95% confidence interval is within  $\pm 5\%$  of the data value. Since the network is misdimensioned in a much larger scale than the previous setup, the average blocking is always worse than routing on the dimensioned network. However, CAR-G and CAR-M still maintain a lower blocking probability than their counterparts. Figure 30-31 shows that CAR-M outperforms flow based algorithms on misdimensioned networks.

## VIII. COMPARISON TO PREVIOUS WORK

In this section, we compare our solutions to two previous routing and congestion control approaches. State-dependent routing (SDR) [6] approaches the routing problem by deriving an optimal solution using Markov decision process. Since computation of the optimal solution is intractable, a link cost model is proposed to approximate the optimal solution. The cost of each link is  $cost = \frac{B(C_e, l_e)}{B(U_e, l_e)}$  where  $B(n, l) = \frac{l^n}{n!} / \sum_{k=0}^n \frac{l^k}{k!}$  is the Erlang-B blocking formula and  $l_e$  is the link load estimation using SDRAdapt algorithm in [8]. The SDRAdapt requires to selection of two parameters: scan interval  $\delta$  and sample number  $\Delta$ . Their paper picked  $\delta = 0.5$  and  $\Delta = 30$ . But we found the result was sensitive to these parameters and obtained better results with  $\delta = 0.05$ ,  $\Delta = 50$ . The difference is due to lack of normalization of time scales between papers. Later, Gawlick *et al.* [7] proposed a competitive online routing and admission control algorithm (COL). The original version of the algorithm requires the knowledge of the holding time of each connection. So they proposed a practical version that each link cost is computed by  $\mu \frac{A_e}{C_e}$ . However the determination of  $\mu$  is nontrivial and the result is sensitive to  $\mu$ , as shown by a later study [8]. The authors did not give any effective method to find the optimal  $\mu$ .

In this paper, we compare both the original SDR and COL algorithms (denoted as SDR-orig and COL-orig, respectively) as well as modified versions (SDR and COL) that always accept connection requests that can be routed using a TSL path. The original SDR only uses a fixed threshold 1. The modified SDR tries to optimize the threshold to improve the blocking probability. Table V summarizes the changes to the CAR algorithm to implement SDR-orig, SDR, COL-orig, and COL. A comparison of blocking probability with CAR-M for the dimensioned ARPANET is shown in Figure 32. The optimized  $\mu$  for COL-orig is  $10^{20}$  and for COL is  $10^{12}$  on ARPANET.

TABLE V  
SDR AND COL ALGORITHMS.

	line 2	line 4	line 6	line 9
SDR-orig [6]	$\gamma_p = \sum_{e \in p} \frac{B(C_e, l_e)}{B(U_e, l_e)} \quad (19)$	$\min \gamma_p$	(remove line 6)	$\Gamma_q < \eta = 1$
SDR	$\gamma_p = \sum_{e \in p} \frac{B(C_e, l_e)}{B(U_e, l_e)} \quad (20)$	$\min \gamma_p$	Accept the connection	$\Gamma_q < \eta$
COL-orig [7]	$\gamma_p = \sum_{e \in p} \mu^{-\frac{A_e}{C_e}} \quad (21)$	$\min \gamma_p$	(remove line 6)	$\Gamma_q < \eta = 1$
COL	$\gamma_p = \sum_{e \in p} \mu^{-\frac{A_e}{C_e}} \quad (22)$	$\min \gamma_p$	Accept the connection	$\Gamma_q < \eta = 1$

SDR-orig shows higher blocking than the rest. The comparison of blocking of each TSL group, in Figure 33, shows that the original versions favor short connections. Our modified version improves the admission chance of long connections. The difference between single-hop blocking probability and that of any other length in the original SDR represents a deliberate choice to favor these connections by always accepting any single-link path. This choice may be appropriate in a network dominated by telephony traffic, which tends to have high geographic locality [21], but the impact of this choice on overall blocking (Figure 32) is not acceptable in a network dominated by other types of data traffic.

We compare the results of CAR-M, COL, and SDR (and original versions) on dimensioned N4L5, ARPANET and NSF in Figures 34, 32, and 35. SDR-orig always present a higher blocking probability at low loads. SDR's threshold is not robust across topologies: the original threshold, which is 1, can be improved on NSF (COST, too) and does not present competitive results to other algorithms on N4L5. CAR-M, COL-orig and COL show robust thresholds on different topologies. With different network size scaling in Figures 32, 36, and 37, CAR-M shows the most robust thresholds amongst the three. On networks with average link capacity 120, all algorithms perform the same. As we change the size of the network, the original thresholds of SDR/SDR-orig and COL/COL-orig are no longer competitive, as shown in Figures 36 and 37. Once we optimize the threshold for these cases, we find that the best thresholds drift down as the network scales. At their optimal thresholds, the performance of COL and SDR is the same as CAR-M. In practice, finding the parameters for SDR and COL is challenging. SDR has three parameters to optimize: the threshold, scan interval, and sample number. For COL, the best  $\mu$  value increases exponentially with the network capacity.

Robustness to network misdimensioning is also checked. We compare COL and SDR on dimensioned ARPANET with link (a,b) capacity doubled (Figure 38). All algorithms except SDR-orig perform equally. When we randomly vary link capacity by 20 percent (Figure 39), CAR-M and COL show an advantage over SDR at low loads. As the network becomes more misdimensioned (with 40% unbalanceness in Figure 40), SDR and COL show slightly better results than CAR-M. However the network is so misdimensioned that achieving decent performance is difficult for reasonable values of offered load.

Comparison of timing is shown in Figure 26. Both original and modified versions are similar in timing. COL is comparable to CAR. Routing time for SDR is much slower, even without accounting for the overhead of updating link load at 1/20 the interval of the average holding time.

## IX. PRACTICAL DISCUSSIONS

We briefly discuss the applicability of our proposed algorithms on current or future WDM networks. Our algorithms address the border routing and resource management problem, which can be combined with other

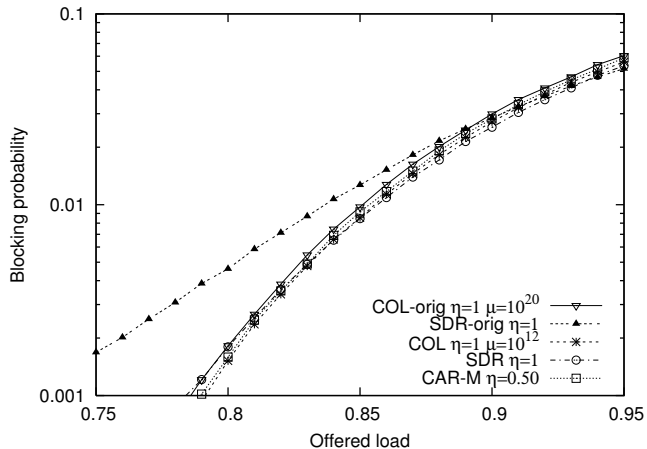


Fig. 32. Blocking of SDR/COL/CAR-M on dimensioned ARPANET w/ avg. link capacity 120. SDR-orig does not show comparable lowest blocking.

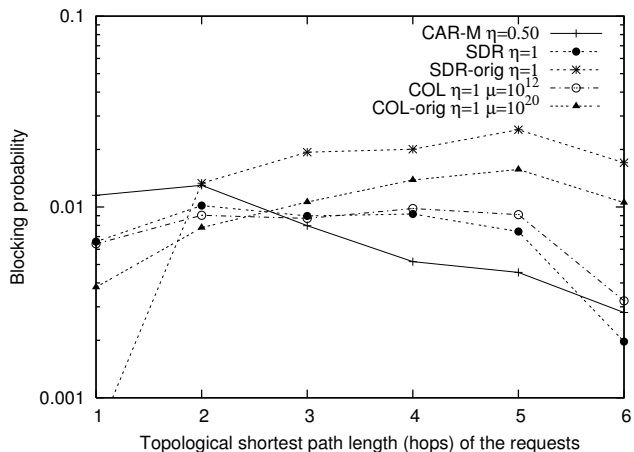


Fig. 33. Blocking probability in group of the TSL of connections on ARPANET at offered load 0.85.

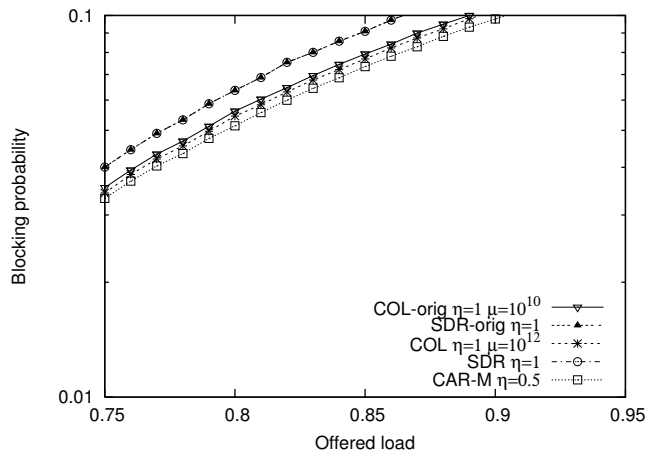


Fig. 34. Blocking of SDR/COL/CAR-M on dimensioned 4N5L w/ avg. link capacity 120. SDR does not show comparable lowest blocking at the best threshold. The best threshold for COL-orig is different from ARPANET.

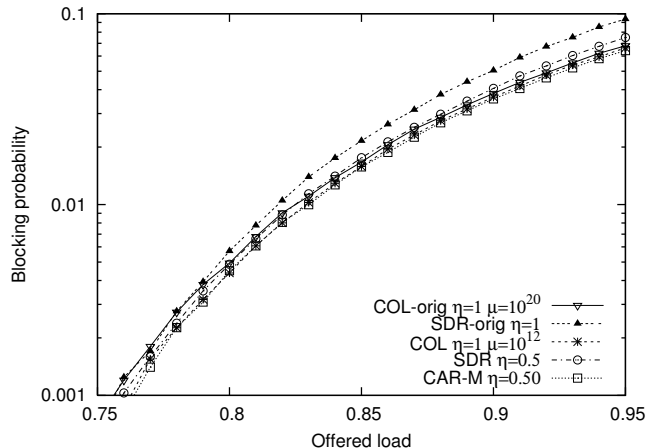


Fig. 35. Blocking of SDR/COL/CAR-M on dimensioned NSF w/ avg. link capacity 120. The best threshold for SDR is different from ARPANET.

techniques in practice. As carrier networks use technologies at different stages to build dynamic optical layers, the wavelength capacity may not be the only resource in the system. Node switching capability, regenerators and wavelength converters (in optically opaque/transparent networks, respectively) are all critical resources whose availability depends on the type of devices and technologies deployed in the network system. Given a real cost model and resource profile, our algorithms can be readily extended to a detailed load model including each edge/intermediate node. Wavelength assignment techniques for all-optical networks that have been addressed in other studies can also be applied with our techniques to improve the overall performance.

## X. CONCLUSION

We proposed efficient dynamic routing algorithms with a threshold-based hop constraint admission control mechanism based on an estimation of network congestion. The routing algorithm CAR achieves over 60% improvement in blocking relative to flow based routing algorithms on ARPANET at high network loads (>90%), and presents the lowest blocking at low loads. The timing cost of CAR is less than 10 percent of flow-based algorithms. We showed that our mechanism effectively slows down the spreading of local congestion to the rest of the network and increases the chance of admission for connections with long topological path length. To understand our algorithms and to predict the best threshold to use, we developed theoretical models to calculate the expected future impact of accepting connections requiring extra resources from the dimensioned network. The analytical results from the

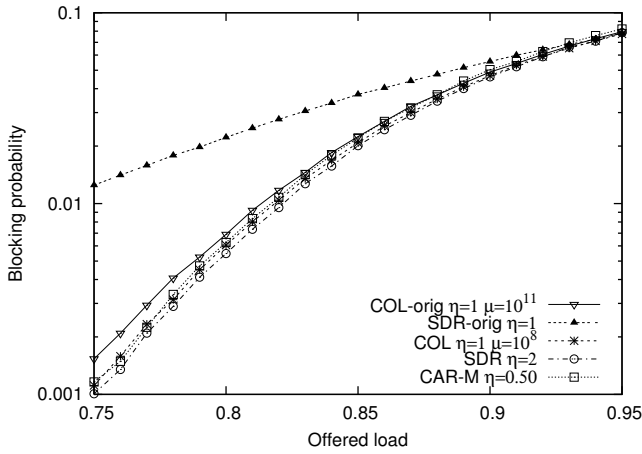


Fig. 36. Blocking of SDR/COL/CAR-M on dimensioned ARPANET w/ avg. link capacity 60. The best thresholds for SDR and COL are different than for avg. capacity 120.

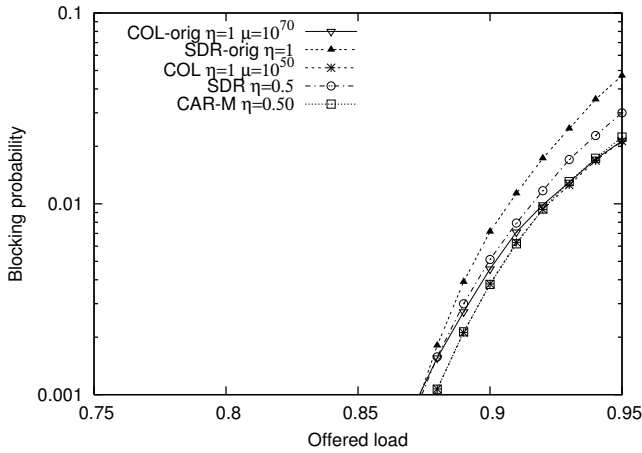


Fig. 37. Blocking of SDR/COL/CAR-M on dimensioned ARPANET w/ avg. link capacity 600. The best thresholds for SDR and COL are different than for avg. capacity 120.

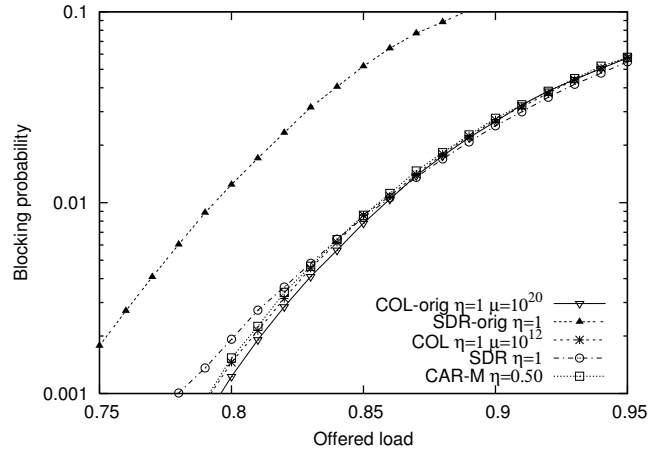


Fig. 38. SDR/COL/CAR-M on misdimensioned ARPANET with the total capacity of link (a,b) in Figure 3 doubled.

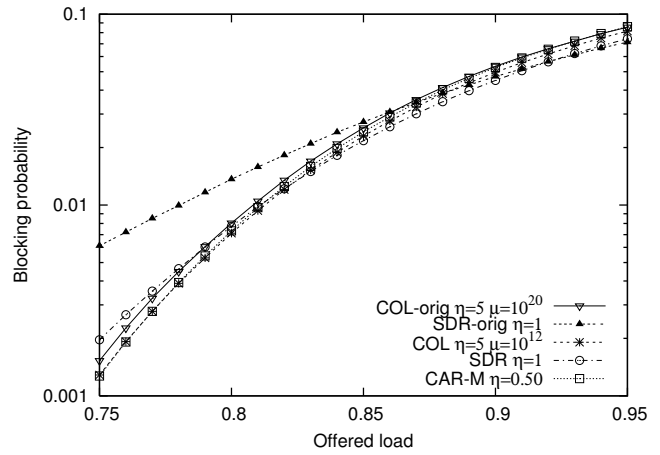


Fig. 39. SDR/COL/CAR-M on misdimensioned ARPANET with  $\alpha = 0.2$ .

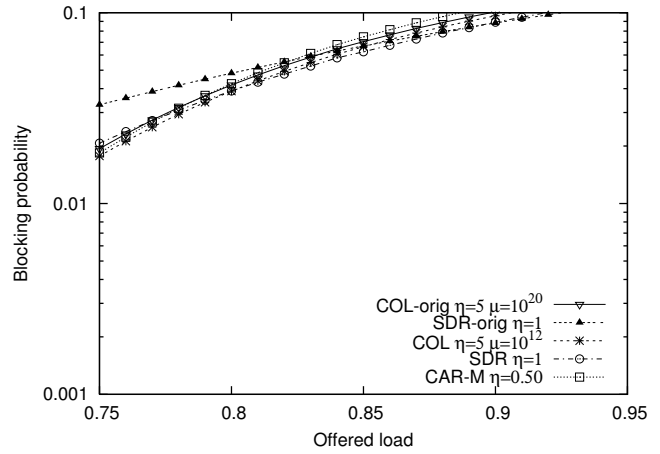


Fig. 40. SDR/COL/CAR-M on misdimensioned ARPANET with  $\alpha = 0.4$ .

model successfully predict the simulation results and allow us to discover several improved congestion estimation routing algorithms. Compared with other algorithms in previous studies, CAR-C and CAR-M algorithm thresholds can be identified analytically, and can thus be robust to changes in topology, capacity, and other factors.

## REFERENCES

- [1] A. Saleh and J. Simmons, "Evolution toward the next-generation core optical network," *Journal of Lightwave Technology*, vol. 24, no. 9, pp. 3303–3321, Sept. 2006.
- [2] C. Xin, "Blocking analysis of dynamic traffic grooming in mesh wdm optical networks," *IEEE/ACM Transactions on Networking*, vol. 15, no. 3, pp. 721–733, June 2007.
- [3] V. Shukla, D. Brown, C. J. Hunt, T. Mueller, and E. Varma, "Next generation optical network – enabling dynamic bandwidth services," in *OFC 2007 - Optical Fiber Communication Conference and Exhibit*, Anaheim, California, USA, 2007.
- [4] X. J. Zhang, S. Kim, and S. S. Lumetta, "Reduced flow routing: Leveraging residual capacity to reduce blocking in GMPLS networks," in *Fourth International Conference on Broadband Communications, Networks, and Systems (BroadNets'07)*, Raleigh, North Carolina, USA, Sep. 2007.
- [5] X. J. Zhang, S. Kim, and S. S. Lumetta, "Resource dimensioning in wdm networks under state-based routing schemes," in *Fourth International Conference on Broadband Communications, Networks, and Systems (BroadNets'07)*, Raleigh, North Carolina, USA, Sep. 2007.
- [6] K. R. Krishnan and T. J. Ott, "State-dependent routing for telephone traffic: Theory and results," *25th IEEE Conference on Decision and Control*, vol. 25, pp. 2124–2128, Dec. 1986.
- [7] R. Gawlick, A. Kamath, S. Plotkin, and K. G. Ramakrishnan, "Routing and admission control in general topology networks," Stanford University, Stanford, CA, USA, Tech. Rep. TR-95-1548, 1995.
- [8] L. Zhang, M. Andrews, W. Aiello, S. Bhatt, and K. Krishnan, "A performance comparison of competitive on-line routing and state-dependent routing," *Global Telecommunications Conference, 1997. GLOBECOM '97., IEEE*, vol. 3, pp. 1813–1819 vol.3, Nov 1997.
- [9] H.-W. Chu and D. H. K. Tsang, "Modified least loaded routing in virtual path based atm networks," *Telecommunication Systems*, vol. 7, no. 1, pp. 45–57, Jun. 1997.
- [10] D. Medhi and I. Sukiman, "Admission control and dynamic routing schemes for wide-area broadband networks: Their interaction and network performance," in *Proc. of IFIP/IEEE Conf. on Broadband Communications*, Montreal, Canada, Apr. 1996, pp. 99–110.
- [11] J. Späth, "Dynamic routing and resource allocation in wdm transport networks," *Computer Networks*, vol. 32, pp. 519–538, May 1999.
- [12] L. Li and A. Somani, "Dynamic wavelength routing techniques and their performance analyses," in *Optical WDM Networks: Principles and Practice*, K. M. Sivalingam and S. Subramaniam, Eds., Oct. 2002, pp. 247–272.
- [13] B. Wen and K. Sivalingam, "Routing, wavelength and time-slot assignment in time division multiplexed wavelength-routed optical wdm networks," *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, pp. 1442–1450 vol.3, 2002.
- [14] B. Zhou and H. T. Mouftah, "Adaptive least loaded routing for multi-fiber wdm networks using approximate congestion information," in *IEEE International Conference on Communications, 2002. ICC'02*, vol. 5, 2002, pp. 2725–2749.
- [15] R. Mewanou and S. Pierre, "Dynamic routing algorithms in all-optical networks," *Electrical and Computer Engineering, 2003. IEEE CCECE 2003. Canadian Conference on*, vol. 2, pp. 773–776 vol.2, May 2003.
- [16] X. Tian, X. Qi, Q. Ma, and X. Zhang, "Study on a distributed wavelength routing algorithm in wdm optical transport networks," *Photonic Network Communications*, vol. 11, no. 3, pp. 271–276, May 2006.
- [17] A. Shaikh, J. Rexford, and K. G. Shin, "Load-sensitive routing of long-lived IP flows," in *SIGCOMM '99: Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for computer communication*. New York, NY, USA: ACM Press, 1999, pp. 215–226.
- [18] B. S. A. Szentesi; and A. Juttner, "Minimizing re-routing in MPLS networks with preemption-aware constraint-based routing," *Computer Communications*, vol. 25, pp. 1076–1083, 2002.
- [19] Y. Lee and J. M. Tien, "Static and dynamic approaches to modeling end-to-end routing in circuit-switched networks," *IEEE/ACM Transactions on Networking*, vol. 10, no. 5, pp. 693–705, 2002.
- [20] M. Liu and J. Baras, "Fixed point approximation for multirate multihop loss networks with state-dependent routing," *IEEE/ACM Transactions on Networking*, vol. 12, no. 2, pp. 361–374, Apr. 2004.
- [21] A. Dwivedi and R. Wagner, "Traffic model for usa long-distance optical network," vol. 1, 2000, pp. 156–158 vol.1.