On Selection of Paths for Multipath Routing

Srihari Nelakuditi and Zhi-Li Zhang

Deparment of Computer Science & Engineering
University of Minnesota, Minneapolis, MN55455
`{srihari,zhzhang}@cs.umn.edu`

Abstract. Multipath routing schemes distribute traffic among multiple paths instead of routing all the traffic along a single path. Two key questions that arise in multipath routing are *how many paths are needed* and *how to select these paths*. Clearly, the number and the quality of the paths selected dictate the performance of a multipath routing scheme. We address these issues in the context of the proportional routing paradigm where the traffic is proportioned among a few "good" paths instead of routing it all along the "best" path. We propose a hybrid approach that uses both *globally* exchanged link state metrics — to identify a set of good paths, and *locally* collected path state metrics — for proportioning traffic among the selected paths. We compare the performance of our approach with that of global optimal proportioning and show that the proposed approach yields near-optimal performance using only a few paths. We also demonstrate that the proposed scheme yields much higher throughput with much smaller overhead compared to other schemes based on link state updates.

1 Introduction

It has been shown [21] that *shortest path* routing can lead to unbalanced traffic distribution — links on frequently used shortest paths become increasingly congested, while other links are underloaded. The *multipath* routing is proposed as an alternative to single shortest path routing to distribute load and alleviate congestion in the network. In multipath routing, traffic bound to a destination is split across multiple paths to that destination. In other words, multipath routing uses multiple "good" paths instead of a single "best" path for routing. Two key questions that arise in multipath routing are how many paths are needed and how to find these paths. Clearly, the number and the quality of the paths selected dictate the performance of a multipath routing scheme. There are several reasons why it is desirable to minimize the number of paths used for routing. First, there is a significant overhead associated with establishing, maintaining and tearing down of paths. Second, the complexity of the scheme that distributes traffic among multiple paths increases considerably as the number of paths increases. Third, there could be a limit on the number of explicitly routed paths such as label switched paths in MPLS [16] that can be setup between a pair of nodes. Therefore it is desirable to use *as few paths as possible* while at the same time *minimize the congestion* in the network.

For judicious selection of paths, some knowledge regarding the (global) network state is crucial. This knowledge about resource availability (referred to as *QoS state*) at

L. Wolf, D. Hutchison, and R. Steinmetz (Eds.): IWQoS 2001, LNCS 2092, pp. 170–184, 2001.

network nodes, for example, can be obtained through (periodic) information exchange among routers in a network. Because network resource availability changes with each flow arrival and departure, maintaining *accurate* view of network QoS state requires *frequent* information exchanges among the network nodes and introduces both communication and processing overheads. However, these updates would not cause significant burden on the network as long as their frequency is not more than what is needed to convey connectivity information in traditional routing protocols like OSPF [11]. The QoS state of each link could then be piggybacked along with the conventional link state updates. Hence it is important to devise multipath routing schemes that *work well even when the updates are infrequent*.

We propose such a scheme *widest disjoint paths* (wdp) that uses proportional routing — the traffic is proportioned among a few widest disjoint paths. It uses *infrequently* exchanged *global* information for selecting a few good paths based on their long term available bandwidths. It proportions traffic among the selected paths using *local* information to cushion the short term variations in their available bandwidths. This *hybrid* approach to multipath routing adapts at different time scales to the changing network conditions. The rest of the paper discusses what type of global information is exchanged and how it is used to select a few good paths. It also describes what information is collected locally and how traffic is proportioned adaptively.

1.1 Related Work

Several multipath routing schemes have been proposed for balancing the load across the network. The Equal Cost Multipath (ECMP) [11] and Optimized Multipath (OMP) [20,21] schemes perform packet level forwarding decisions. ECMP splits the traffic *equally* among multiple equal cost paths. However, these paths are determined statically and may not reflect the congestion state of the network. Furthermore, it is desirable to apportion the traffic according to the quality of each path. OMP is similar in spirit to our work. It also uses updates to gather link loading information, selects a set of best paths and distributes traffic among them. However, our scheme makes routing decisions at the flow level and consequently the objectives and procedures are different.

QoS routing schemes have been proposed [3,5,10,22] where flow level routing decisions are made based upon the knowledge of the resource availability at network nodes and the QoS requirements of flows. This knowledge is obtained through global link state information exchange among routers in a network. These schemes, which we refer to as global QoS routing schemes, construct a global view of the network QoS state by piecing together the information about each link, and perform path selection based solely on this global view. Examples of global QoS routing schemes are *widest shortest path* [5], *shortest widest path* [22], and *shortest distance path* [10]. While *wdp* also uses link state updates, the nature of information exchanged and the manner in which it is utilized is quite different from global QoS routing schemes. In Section 4, we demonstrate that *wdp* provides higher throughput with lower overhead than these schemes.

Another approach to path selection is to precompute maximally disjoint paths [19] and attempt them in some order. This is static and overly conservative. What matters is not the sharing itself but *the sharing of bottleneck links*, which change with network conditions. In our scheme we dynamically select paths such that they are disjoint *w.r.t* bottleneck links.

The rest of the paper is organized as follows. In Section 2, we introduce the proportional routing framework and describe a global optimal proportional routing procedure (*opr*) and a localized proportional routing scheme *equalizing blocking probability* (ebp). In both these cases, the candidate path set is *static* and large. In Section 2.4, we propose a hybrid approach to multipath routing that selects a few good paths *dynamically* using global information and proportions traffic among these paths using local information. Section 3 describes such a scheme *wdp* that selects widest disjoint paths and proportions traffic among them using *ebp*. The simulation results evaluating the performance of *wdp* are shown in Section 4. Section 5 concludes the paper.

2 Proportional Routing Framework

In this section, we first lay out the basic assumptions regarding the proportional routing framework we consider in this paper. We then present a global optimal proportional routing procedure (*opr*), where we assume that the traffic loads among all source-destination pairs are known. The *opr* procedure gives the least blocking probability that can be achieved by a proportional routing scheme. However, it is quite complex and time consuming. We use the performance of *opr* as a reference to evaluate the proposed scheme. We then describe a localized adaptive proportioning approach that uses only locally collected path state metrics and assigns proportions to paths based on their quality. The localized schemes are described in detail in [12,13], a brief summary of which is reproduced here. We then present our proposed hybrid approach to multipath routing that uses global information to select a few good paths and employs localized adaptive proportioning to proportion traffic among these paths.

2.1 Problem Setup

In all the QoS routing schemes considered in this paper we assume that source routing (also referred to as explicit routing) is used. More specifically, we assume that the network topology information is available to all source nodes (e.g., via the OSPF protocol), and one or multiple explicit-routed paths or label switched paths are set up *a priori* between each source and destination pair using, e.g., MPLS [16]. Flows arriving at a source to a destination are routed along one of the explicit-routed paths (hereafter referred to as the *candidate* paths between the source-destination pair). For simplicity, we assume that all flows have the same bandwidth requirement — one unit of bandwidth. When a flow is routed to a path where one or more of the constituent links have no bandwidth left, this flow will be blocked. The performance metric in our study will be the overall blocking probability experienced by flows. We assume that flows from a source to a destination arrive randomly with a Poisson distribution, and their holding time is exponentially distributed. Hence the offered traffic load between a source-destination pair can be measured as the product of the average flow arrival rate and holding time. Given the offered traffic load from a source to a destination, the task of proportional QoS routing is to determine how to distribute the load (i.e., route the flows) among the candidate paths between a source and a destination so as to minimize the overall blocking probability experienced by the flows.

2.2 Global Optimal Proportioning

The global optimal proportioning has been studied extensively in the literature (see [17] and references therein). Here it is assumed that each source node knows the complete topology information of the network (including the maximum capacity of each link) as well as the offered traffic load between every source-destination pair. With the global knowledge of the network topology and offered traffic loads, the *optimal* proportions, for distributing flows among the paths between each source-destination pair, can be computed as described below.

Consider an arbitrary network topology with N nodes and L links. For $l = 1, \ldots, L$, the maximum capacity of link l is $\hat{c}_l > 0$, which is assumed to be fixed and known. The links are unidirectional, i.e., carry traffic in one direction only. Let $\sigma = (s, d)$ denote a source-destination pair in the network. Let λ_σ denote the average arrival rate of flows arriving at source node s destined for node d. The average holding time of the flows is μ_σ. Recall that each flow is assumed to request one unit of bandwidth, and that the flow arrivals are Poisson, and flow holding times are exponentially distributed. Thus the offered load between the source-destination pair σ is $\nu_\sigma = \lambda_\sigma / \mu_\sigma$.

Let $\hat{R}_\sigma$ denote the set of *feasible* paths for routing flows between the pair σ. The global optimal proportioning problem can be formulated [6,7,9] as the problem of finding the optimal proportions $\{\alpha_r^*, r \in \hat{R}_\sigma\}$ where $\sum_{r \in \hat{R}_\sigma} \alpha_r^* = 1$, such that the overall flow blocking probability in the network is minimized. Or equivalently, finding the optimal proportions $\{\alpha_r^*, r \in \hat{R}_\sigma\}$ such that the total carried traffic in the network, $W = \sum_\sigma \sum_{r \in \hat{R}_\sigma} \alpha_r \nu_\sigma (1 - b_r)$ is maximized. Here b_r is the blocking probability on path r when a load of $\nu_r = \alpha_r \nu_\sigma$ is routed through r. Then the set of *candidate* paths R_σ are a subset of feasible paths $\hat{R}_\sigma$ with proportion larger than a negligible value ϵ, i.e., $R_\sigma = \{r : r \in \hat{R}_\sigma, \alpha_r^* > \epsilon\}$. This global optimal proportional routing problem is a constrained nonlinear optimization problem and can be solved using an iterative procedure based on the Sequential Quadratic Programming (SQP) method [4,15].

2.3 Localized Adaptive Proportioning

The optimal proportioning procedure described above requires global information about the offered load between each source-destination pair. It is also quite complex and thus time consuming. We have shown [12] that it is possible to obtain near-optimal proportions using simple localized strategies such as equalizing blocking probability *ebp* and equalizing blocking rate *ebr*. Let $\{r_1, r_2, \ldots, r_k\}$ be the set of k candidate paths between a source destination pair. The objective of the *ebp* strategy is to find a set of proportions $\{\alpha_{r_1}, \alpha_{r_2}, \ldots, \alpha_{r_k}\}$ such that flow blocking probabilities on all the paths are equalized, i.e., $b_{r_1} = b_{r_2} = \cdots = b_{r_k}$, where b_{r_i} is the flow blocking probability on path r_i. On the other hand, the objective of the *ebr* strategy is to equalize the flow blocking rates, i.e., $\alpha_{r_1} b_{r_1} = \alpha_{r_2} b_{r_2} = \cdots = \alpha_{r_k} b_{r_k}$. By employing these strategies a source node can adaptively route flows among multiple paths to a destination, in proportions that are commensurate with the *perceived* qualities of these paths. The perceived quality of a path between a source and a destination is inferred based on locally collected flow statistics: the offered load on the path and the resulting blocking probability of the flows routed along the path.

In this work, we use a simpler approximation to *ebp* that computes new proportions as follows. First, the current average blocking probability $\bar{b} = \sum_{i=1}^{k} \alpha_{r_i} b_{r_i}$ is computed. Then, the proportion of load onto a path r_i is decreased if its current blocking probability b_{r_i} is higher than the average $\bar{b}$ and increased if b_{r_i} is lower than $\bar{b}$. The magnitude of change is determined based on the relative distance of b_{r_i} from $\bar{b}$ and some configurable parameters to ensure that the change is gradual. The mean time between proportion computations is controlled by a configurable parameter θ. This period θ should be large enough to allow for a reasonable measurement of the quality of the candidate paths. The blocking performance of the candidate paths are observed for a period θ and at the end of the period the proportions are recomputed. A more detailed description of this procedure can be found in [14].

2.4 Hybrid Approach to Multipath Routing

The global proportioning procedure described above computes optimal proportions α_r^* for each path r given a feasible path set $\hat{R}_\sigma$ for each source-destination pair σ. Taking into account the overhead associated with setting up and maintaining the paths, it is desirable to minimize the number of candidate paths while minimizing the overall blocking probability. However achieving both the minimization objectives may not be practical. Note that the blocking probability minimization alone, for a fixed set of candidate paths, is a constrained nonlinear optimization problem and thus quite time consuming. Minimizing the number of candidate paths involves experimenting with different combinations of paths and the complexity grows exponentially as the size of the network increases. Hence it is not feasible to find an optimal solution that minimizes both the objectives. Considering that achieving the absolute minimal blocking is not very critical, it is worthwhile investigating heuristic schemes that *tradeoff slight increase in blocking for significant decrease in the number of candidate paths*.

The localized approach to proportional routing is simple and has several important advantages. However it has a limitation that routing is done based solely on the information collected locally. A network node under localized QoS routing approach can judge the quality of paths/links only by routing some traffic along them. It would have no knowledge about the state of the rest of the network. While the proportions for paths are adjusted to reflect the changing qualities of paths, the candidate path set itself remains static. To ensure that the localized scheme adapts to varying network conditions, many feasible paths have to be made candidates. It is not possible to preselect a few good candidate paths statically. Hence it is desirable to *supplement localized proportional routing* with a mechanism that *dynamically selects a few good candidate paths*.

We propose such a hybrid approach to proportional routing where locally collected path state metrics are supplemented with globally exchanged link state metrics. A set of few good candidate paths R_σ are maintained for each pair σ and this set is updated based on the global information. The traffic is proportioned among the candidate paths using local information. In the next section we describe a hybrid scheme *wdp* that selects widest disjoint paths and uses *ebp* strategy for proportioning traffic among them.

3 Widest Disjoint Paths

In this section, we present the candidate path selection procedure used in *wdp*. To help determine whether a path is good and whether to include it in the candidate path set, we define *width* of a path and introduce the notion of *width* of a *set of paths*. The candidate path set R_σ for a pair σ is changed only if it increases the width of the set R_σ or decreases the size of the set R_σ without reducing its width. The widths of paths are computed based on link state updates that carry *average residual bandwidth* information about each link. The traffic is then proportioned among the candidate paths using *ebp*.

A basic question that needs to be addressed by any path selection procedure is what is a "good" path. In general, a path can be categorized as good if its inclusion in the candidate path set decreases the overall blocking probability considerably. It is possible to judge the utility of a path by measuring the performance with and without using the path. However, it is not practical to conduct such inclusion-exclusion experiment for each feasible path. Moreover, each source has to independently perform such trials without being directly aware of the actions of other sources which are only indirectly reflected in the state of the links. Hence each source has to try out paths that are likely to decrease blocking and make such decisions with some local objective that leads the system towards a global optimum.

When identifying a set of candidate paths, another issue that requires attention is the sharing of links between paths. A set of paths that are good *individually* may not perform as well as expected *collectively*. This is due to the sharing of *bottleneck* links. When two candidate paths of a pair share a bottleneck link, it may be possible to remove one of the paths and shift all its load to the other path without increasing the blocking probability. Thus by ensuring that candidate paths of a pair do not share bottleneck links, we can reduce the number of candidate paths without increasing the blocking probability. A simple guideline to enforce this could be that the candidate paths of a pair be mutually disjoint, i.e., they do not share *any* links. This is overly restrictive, since even with shared links, some paths can cause reduction in blocking if those links are not congested. What matters is not the sharing itself but *the sharing of bottleneck links*. While the sharing of links among the paths is *static* information independent of traffic, identifying bottleneck links is *dynamic* since the congestion in the network depends on the offered traffic and routing patterns. Therefore it is essential that candidate paths be *mutually disjoint w.r.t bottleneck links*.

To judge the quality of a path, we define *width* of a path as the the residual bandwidth on its bottleneck link. Let $\hat{c}_l$ be the maximum capacity of link l and ν_l be the average load on it. The difference $c_l = \hat{c}_l - \nu_l$ is the average residual bandwidth on link l. Then the *width* w_r of a path r is given by $w_r = \min_{l \in r} c_l$. The larger its width is, the better the path is, and the higher its potential is to decrease blocking. Similarly we define *distance* [10] of a path r as $\sum_{l \in r} \frac{1}{c_l}$. The shorter the distance is, the better the path is. The widths and distances of paths can be computed given the residual bandwidth information about each link in the network. This information can be obtained through periodic link state updates. To discount short term fluctuations, the *average residual bandwidth* information is exchanged. Let τ be the update interval and u_l^t be the utilization of link l during the period $(t-\tau, t)$. Then the average residual bandwidth at time t, $c_l^t = (1-u_l^t)\hat{c}_l$. Hereafter without the superscript, c_l refers to the most recently updated value of the average residual bandwidth of link l.

To aid in path selection, we also introduce the notion of *width* for a *set of paths* R, which is computed as follows. We first pick the path r^* with the largest width w_{r^*}. If there are multiple such paths, we choose the one with the shortest distance d_{r^*}. We then decrease the residual bandwidth on all its links by an amount w_{r^*}. This effectively makes the residual bandwidth on its bottleneck link to be 0. We remove the path r^* from the set R and then select a path with the next largest width based on the just updated residual bandwidths. Note that this change in residual bandwidths of links is local and only for the purpose computing the width of R. This process is repeated till the set R becomes empty. The sum of all the widths of paths computed thus is defined as the *width of* R. Note that when two paths share a bottleneck link, the width of two paths together is same as the width of a single path. The width of a path set computed thus, essentially accounts for the sharing of links between paths. The narrowest path, i.e., the last path removed from the set R is referred to as NARROWEST(R).

Based on this notion of width of a path set, we propose a path selection procedure that *adds* a new candidate path only if its inclusion *increases the width*. It *deletes* an existing candidate path if its exclusion *does not decrease* the total width. In other words, each modification to the candidate path set either *improves the width* or *reduces the number* of candidate paths. The selection procedure is shown in Figure 1. First, the load contributed by each existing candidate path is deducted from the corresponding links (lines 2-4). After this adjustment, the residual bandwidth c_l on each link l reflects the load offered on l by all source destination pairs other than σ. Given these adjusted residual bandwidths, the candidate path set R_σ is modified as follows.

The benefit of inclusion of a feasible path r is determined based on the number of existing candidate paths (lines 6-8). If this number is below the specified limit η, the resulting width W_r is the width of $R_\sigma \cup r$. Otherwise, it is the width of $R_\sigma \cup r \backslash$ NARROWEST($R_\sigma \cup r$), i.e., the width after excluding the narrowest path among $R_\sigma \cup r$. Let W^+ be the largest width that can be obtained by adding a feasible path (line 9). This width W^+ is compared with width of the current set of candidate paths. A feasible path is made a candidate if its inclusion in set R_σ increases the width by a fraction ψ (line 10). Here $\psi > 0$ is a configurable parameter to ensure that each addition improves the width by a significant amount. It is possible that many feasible paths may cause the width to be increased to W^+. Among such paths, the path r^+ with the shortest distance is chosen for inclusion (lines 11-13). Let r^- be the narrowest path in the set $R_\sigma \cup r$ (line 14). The path r^- is replaced with r^+ if either the number of paths already reached the limit or the path r^- does not contribute to the width (lines 15-16). Otherwise the path r^+ is simply added to the set of candidate paths (lines 17-18). When no new path is added, an existing candidate path is deleted from the set if it does not change the width (lines 20-22). In all other cases, the candidate path set remains unaffected. It is obvious that this procedure always either increases the width or decreases the number of candidate paths.

```
PROCEDURE SELECT($\sigma$)
  For each path $r$ in $R_\sigma$
    For each link $l$ in $r$
      $c_l = c_l + (1 - b_r)\nu_r$
  If $|R_\sigma| < \eta$
    $W_r =$ WIDTH($R_\sigma \cup r$), $\forall r \in \hat{R}_\sigma \setminus R_\sigma$
  Else
    $W_r =$ WIDTH($R_\sigma \cup r \setminus$ NARROWEST($R_\sigma \cup r$)), $\forall r \in \hat{R}_\sigma \setminus R_\sigma$
  $W^+ = \max_{r \in \hat{R}_\sigma \setminus R_\sigma} W_r$
  If ($W^+ > (1 + \psi)$ WIDTH($R_\sigma$))
    $R^+ = \{r : r \in \hat{R}_\sigma \setminus R_\sigma, W_r = W^+\}$
    $d^+ = \min_{r \in R^+} d_r$
    $r^+ = \{r : r \in R^+, d_r = d^+\}$
    $r^- =$ NARROWEST($R_\sigma \cup r$)
    If ($|R_\sigma| = \eta$ or WIDTH($R_\sigma \cup r^+ \setminus r^-$) $= W^+$)
      $R_\sigma = R_\sigma \cup r^+ \setminus r^-$
    Else
      $R_\sigma = R_\sigma \cup r^+$
  Else
    $r^- =$ NARROWEST($R_\sigma$)
    If WIDTH($R_\sigma \setminus r^-$) $=$ WIDTH($R_\sigma$)
      $R_\sigma = R_\sigma \setminus r^-$
END PROCEDURE
```

Fig. 1. The Candidate Path Set Selection Procedure for Pair σ.

It should be noted that though *wdp* uses link state updates it does not suffer from the *synchronization* problem unlike global QoS routing schemes such as *wsp*. There are several reasons contributing to the stability of *wdp*: 1) The information exchanged about a link is its *average* not *instantaneous* residual bandwidth and hence less variable; 2) The traffic is proportioned among few "good" paths instead of loading the "best" path based on inaccurate information; 3) Each pair uses only a few candidate paths and makes only incremental changes to the candidate path set; 4) The new candidate paths are selected for a pair only after deducting the load contributed by the current candidate paths from their links. Due to such adjustment even with link state updates, the view of the network for each node would be different; 5) When network is in a stable state of convergence, the information carried in link state updates would not become outdated and consequently each node would have reasonably accurate view of the network. Essentially the nature of information exchanged and the manner in which it is utilized work in a mutually beneficial fashion and lead the system towards a stable optimal state.

4 Performance Analysis

In this section, we evaluate the performance of the proposed hybrid QoS routing scheme *wdp*. We start with the description of the simulation environment. First, we compare the performance *wdp* with the optimal scheme *opr* and show that *wdp* converges to near-optimal proportions. Furthermore, we demonstrate that the performance of *wdp* is relatively insensitive to the values chosen for the configurable parameters. We then contrast the performance of *wdp* with global QoS routing scheme *wsp* in terms of the overall blocking probability and routing overhead.

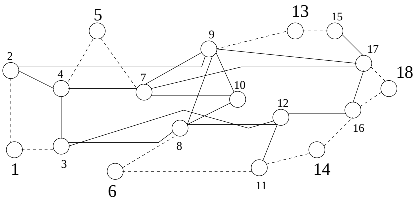


Fig. 2. The Topology Used for Performance Evaluation.

4.1 Simulation Environment

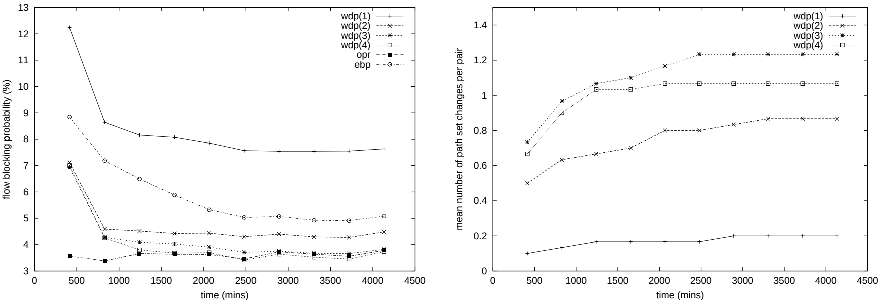
The Figure 4 shows the *isp* topology used in our study. This topology of an ISP backbone network is also used in [1,10]. For simplicity, all the links are assumed to be bidirectional and of equal capacity in each direction. There are two types of links: *solid* and *dotted*. All solid links have same capacity with C_1 units of bandwidth and similarly all the dotted links have C_2 units. The dotted links are the access links and for the purpose of our study their capacity is assumed to be higher than solid links. Otherwise, access links become the bottleneck limiting the impact of multipath routing and hence not an interesting case for our study. Flows arriving into the network are assumed to require one unit of bandwidth. Hence a link with capacity C can accommodate at most C flows simultaneously.

The flow dynamics of the network is modeled as follows (similar to the model used in [18]). The nodes labeled with bigger font are considered to be source (ingress) or destination (egress) nodes. Flows arrive at a source node according to a Poisson process with rate λ. The destination node of a flow is chosen randomly from the set of all nodes except the source node. The holding time of a flow is exponentially distributed with mean $1/\mu$. Following [18], the offered network load on *isp* is given by $\rho = \lambda N\bar{h}/\mu(L_1C_1 + L_2C_2)$, where N is the number of source nodes, L_1 and L_2 are the number of solid and dotted links respectively, and $\bar{h}$ is the mean number of hops per flow, averaged across all source-destination pairs. The parameters used in our simulations are $C_1 = 20$, $C_2 = 30$, $1/\mu = 1$ minute (here after written as just *m*). The topology specific parameters are $N = 6$, $L_1 = 36$, $L_2 = 24$, $\bar{h} = 3.27$. The average arrival rate at a source node λ is set depending upon the desired load ρ.

The parameters in the simulation are set as follows by default. Any change from these settings is explicitly mentioned wherever necessary. The values for configurable parameters in *wdp* are $\psi = 0.2$, $\tau = 30$ *m*, $\theta = 60$ *m*, $\xi = 180$ *m*. For each pair σ, all the paths between them whose length is at most one hop more than the minimum number of hops is included in the feasible path set $\hat{R}_\sigma$. The amount of offered load on the network ρ is set to 0.55. Each run simulates arrival of $1,000,000$ flows and the results corresponding to the later half the simulation are reported here.

4.2 Performance of *wdp*

In this section, we compare the performance of *wdp* and *opr* to show that *wdp* converges to near-optimal proportions using only a few paths for routing traffic. We also demonstrate that *wdp* is relatively insensitive to the settings for the configurable parameters.



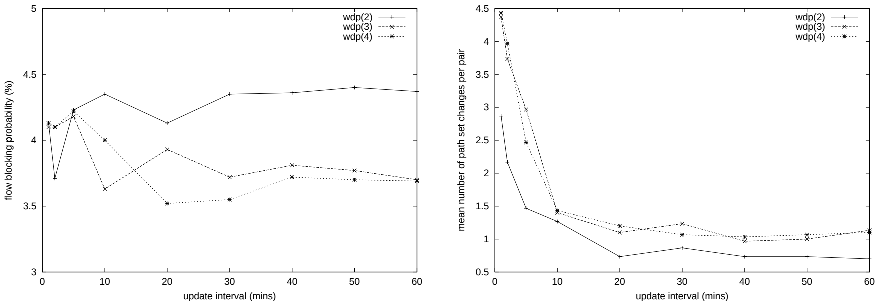
(a) Blocking Probability.

(b) Number of Changes to Candidate Paths.

Fig. 3. Convergence Process of *wdp*.

Convergence. Figure 3 illustrates the convergence process of *wdp*. The results are shown for different values of $\eta = 1 \cdots 4$. Figure 3(a) compares the performance of *wdp*, *opr* and *ebp*. The performance is measured in terms of the overall flow blocking probability, which is defined as the ratio of the total number of blocks to the total number of flow arrivals. The overall blocking probability is plotted as a function of time. In the case of *opr*, the algorithm is run offline to find the optimal proportions given the set of feasible paths and the offered load between each pair of nodes. The resulting proportions are then used in simulation for statically proportioning the traffic among the set of feasible paths. The *ebp* scheme refers to the localized scheme used in isolation for adaptively proportioning across all the feasible paths. As noted earlier all paths of length either minhop or minhop+1 are chosen as the set of feasible paths in our study.

There are several conclusions that can be drawn from Figure 3(a). First, the *wdp* scheme converges for all values of η. Given that the time between changes to candidate path sets, ξ, is 180 *m*, it reaches steady state within (on average) 5 path recomputations per pair. Second, there is a marked reduction in the blocking probability when the number of paths allowed, η, is changed from 1 to 2. It is evident that there is quite a significant gain in using multipath routing instead of single path routing. When the limit η is increased from 2 to 3 the improvement in blocking is somewhat less but significant. Note that in our topology there are at most two paths between a pair that do not share any links. But there could be more than two paths that are mutually disjoint w.r.t bottleneck links. The performance difference between η values of 2 and 3 is an indication that we only need to ensure that candidate paths do not share congested links. However using more than 3 paths per pair helps very little in decreasing the blocking probability. Third, the *ebp* scheme also converges, albeit slowly. Though it performs much better than *wdp* with single path, it is worse than *wdp* with $\eta = 2$. But when *ebp* is used in conjunction with path selection under *wdp* it converges quickly to lower blocking probability using only a few paths. Finally, using at most 3 paths per pair, the *wdp* scheme approaches the performance of optimal proportional routing scheme.



(a) Blocking Probability. (b) Number of Path set Changes.

Fig. 4. Sensitivity of *wdp* to Update Interval τ.

Figure 3(b) establishes the convergence of *wdp*. It shows the average number of changes to the candidate path set as a function of time. Here the change refers to either addition, deletion or replacement operation on the candidate path set R_σ of any pair σ. Note that the cumulative number of changes are plotted as a function of time and hence a plateau implies that there is no change to any of the path sets. It can be seen that the path sets change incrementally initially and after a while they stabilize. Thereafter each pair sticks to the set of chosen paths. It should be noted that starting with at most 3 minhop paths as candidates and making as few as 1.2 changes to the set of candidate paths, the *wdp* scheme achieves almost optimal performance.

We now compare the average number of paths used by a source-destination pair for routing. Note that in *wdp* scheme η only specifies the maximum allowed number of paths per pair. The actual number of paths selected for routing depends on their widths. The average number of paths used by *wdp* for η of 2 and 3 are 1.7 and 1.9 respectively. The number of paths used stays same even for higher values of η. The *ebp* scheme uses all the given feasible paths for routing. It can measure the quality of a path only by routing some traffic along that path. The average number of feasible paths chosen are 5.6. In case of *opr* we count only those paths that are assigned a proportion of at least 0.10 by the optimal offline algorithm. The average number of such paths under *opr* scheme are 2.4. These results support our claim that *ebp* based proportioning over widest disjoint paths performs almost like optimal proportioning scheme while using fewer paths.

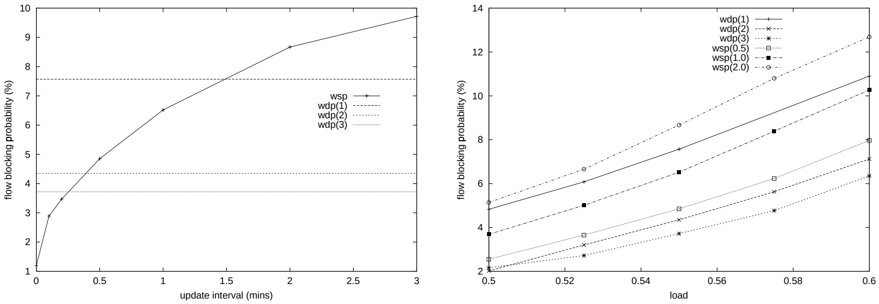
Sensitivity. The *wdp* scheme requires periodic updates to obtain global link state information and to perform path selection. To study the impact of update interval on the performance of *wdp*, we conducted several simulations with different update intervals ranging from 1 *m* to 60 *m*. The Figure 4(a) shows the flow blocking probability as a function of update interval. At smaller update intervals there is some variation in the blocking probability, but much less variation at larger update intervals. It is also clear that increasing the update interval does not cause any significant change in the blocking probability. To study the effect of update interval on the stability of *wdp*, we plotted the average number of path set changes as a function of update interval in Figure 4(b). It shows that the candidate path set of a pair changes often when the updates are frequent. When the update interval is small, the average residual bandwidths of links resemble their instantaneous values, thus highly varying. Due to such variations, paths may appear wider or narrower than they actually are, resulting in unnecessary changes to candidate paths. However, this does not have a significant impact on the blocking performance due to adaptive proportional routing among the selected paths. For the purpose of reducing overhead and increasing stability, we suggest that the update interval τ be reasonably large, while ensuring that it is much smaller than the path recomputation interval ξ. We have also varied other configurable parameters and found that *wdp* is relatively insensitive to the values chosen. For more details, refer to [14].

4.3 Comparison of *wsp* and *wdp*

We now compare the performance of hybrid QoS routing scheme *wdp* with a global QoS routing scheme *wsp*. The *wsp* is a well-studied scheme that selects the widest shortest path for each flow based on the global network view obtained through link state updates. The information carried in these updates is the residual bandwidth at the instant of the update. Note that *wdp* also employs link state updates but the information exchanged is average residual bandwidth over a period not its instantaneous value. We use *wsp* as a representative of global QoS routing schemes as it was shown to perform the best among similar schemes such as shortest widest path (*swp*), shortest distance path (*sdp*). In the following, we first compare the performance of *wdp* with *wsp* in terms of flow blocking probability and then the routing overhead.

Blocking Probability. Figure 5(a) shows the blocking probability as a function of update interval τ used in *wsp*. The τ for *wdp* is fixed at 30 *m*. The offered load on the network ρ was set to 0.55. It is clear that the performance of *wsp* degrades drastically as the update interval increases. The *wdp* scheme, using at most two paths per pair and infrequent updates with $\tau = 30$ *m*, blocks fewer flows than *wsp*, that uses many more paths and frequent updates with $\tau = 0.5$ *m*. The performance of *wdp* even with a single path is comparable to *wsp* with $\tau = 1.5$ *m*. Figure 5(b) displays the flow blocking probability as a function of offered network load ρ which is varied from 0.50 to 0.60. Once again, the τ for *wdp* is set to 30 *m* and the performance of *wsp* is plotted for 3 different settings of τ: 0.5, 1.0 and 2.0 *m*. It can be seen that across all loads the performance of *wdp* with $\eta = 2$ is better than *wsp* with $\tau = 0.5$. Similarly with just one path, *wdp* performs better than *wsp* with $\tau = 2.0$ and approaches the performance of $\tau = 1.0$ as the load increases. It is also worth noting that *wdp* with two paths rejects significantly fewer flows than with just one path, justifying the need for multipath routing.

It is interesting to observe that even with a single path and very infrequent updates *wdp* outperforms *wsp* with frequent updates. There are several factors contributing to the superior performance of *wdp*. First, it is the nature of information used to capture the link state. The information exchanged about a link is its *average* not *instantaneous* residual bandwidth and hence less variable. Second, before picking the widest disjoint



(a) Varying Update Interval.

(b) Varying Load.

Fig. 5. Performance Comparison of *wdp* and *wsp*.

paths, the residual bandwidth on all the links along the current candidate path are adjusted to account for the load offered on that path by this pair. Such a *local adjustment* to the global information makes the network state appear differently to each source. It is as if each source receives a customized update about the state of each link. The sources that are currently routing through a link perceive higher residual bandwidth on that link than other sources. This causes a source to continue using the same path to a destination unless it finds a much wider path. This in turn reduces the variation in link state and consequently the updated information does not get outdated too soon. In contrast, *wsp* exchanges highly varying instantaneous residual bandwidth information and all the sources have the same view of the network. This results in mass synchronization as every source prefers *good* links and avoids *bad* links. This in turn increases the variance in instantaneous residual bandwidth values and causes route oscillation[1]. The *wdp* scheme, on the other hand, by selecting paths using both local and global information and by employing *ebp* based adaptive proportioning delivers stable and robust performance.

Routing Overhead. Now we compare the amount of overhead incurred by *wdp* and *wsp*. This overhead can be categorized into per flow routing overhead and operational overhead. We discuss these two separately in the following.

The *wsp* scheme selects a path by first pruning the links with insufficient available bandwidth and then performing a variant of Dijkstra's algorithm on the resulting graph to find the shortest path with maximum bottleneck bandwidth. This takes at least $O(E \log N)$ time where N is the number of nodes and E is the total number of links in the network. Assuming precomputation of a set of paths R_σ to each destination, to avoid searching the whole graph for path selection, it still need to traverse all the links of these precomputed paths to identify the widest shortest path. This amounts to an

---

[1] Some remedial solutions were proposed in [1,2] to deal with the inaccuracy at a source node. However, the fundamental problem remains and the observations made in this paper still apply.

overhead of $O(L_\sigma)$, where L_σ is the total number of links in the set R_σ. On the other hand, in *wdp* one of the candidate paths is chosen in a weighted round robin fashion whose complexity is $O(\eta)$ which is much less than $O(L_\sigma)$ for *wsp*.

Now consider the operational overhead. Both schemes require link state updates to carry residual bandwidth information. However the frequency of updates needed for proper functioning of *wdp* is no more than what is used to carry connectivity information in traditional routing protocols such as OSPF. Therefore, the average residual bandwidth information required by *wdp* can be piggybacked along with the conventional link state updates. Hence, *wdp* does not cause any additional burden on the network. On the other hand, the *wsp* scheme requires frequent updates consuming both network bandwidth and processing power. Furthermore *wsp* uses too many paths. The *wdp* scheme uses only a few preset paths, thus avoiding per flow path setup. Only admission control decision need to be made by routers along the path. The other overheads incurred only by *wdp* are periodic proportion computation and candidate path computation. The proportion computation procedure is extremely simple and costs no more than $O(\eta)$. The candidate path computation amounts to finding η widest paths and hence its worst case time complexity is $O(\eta N^2)$. However, this cost is incurred only once every ξ period. Considering both the blocking performance and the routing cost, we proclaim that *wdp* yields much higher throughput with much lower overhead than *wsp*.

5 Conclusions

The performance of multipath routing hinges critically on the number and the quality of the selected paths. We addressed these issues in the context of the proportional routing paradigm, where the traffic is proportioned among a few good paths instead of routing it all along the best path. We proposed a hybrid approach that uses both global and local information for selecting a few good paths and for proportioning the traffic among the selected paths. We presented a *wdp* scheme that performs *ebp* based proportioning over widest disjoint paths. A set of widest paths that are disjoint *w.r.t* bottleneck links are chosen based on globally exchanged link state metrics. The *ebp* strategy is used for adaptively proportioning traffic among these paths based on locally collected path state metrics. We compared the performance of our *wdp* scheme with that of optimal proportional routing scheme *opr* and shown that the proposed scheme achieves almost optimal performance using much fewer paths. We also demonstrated that the proposed scheme yields much higher throughput with much smaller overhead compared to other link state update based schemes such as *wsp*.

References

1. G. Apostolopoulos, R. Guerin, S. Kamat, S. Tripathi, “Quality of Service Based Routing: A Performance Perspective”, ACM SIGCOMM, September 1998.
2. G. Apostolopoulos, R. Guerin, S. Kamat, S. Tripathi, “Improving QoS Routing Performance under Inaccurate Link State Information”, ITC’16, June 1999.

3. Shigang Chen, Klara Nahrstedt, “An Overview of Quality-of-Service Routing for the Next Generation High-Speed Networks: Problems and Solutions”, IEEE Network Magazine, Special Issue on Transmission and Distribution of Digital Video, Vol. 12, No. 6, November-December 1998, pp. 64-79.
4. T. Coleman and Y. Li, “An Interior, Trust Region Approach for Nonlinear Minimization Subject to Bounds,” in *Journal on Optimization, Vol. 6*, pp. 418–445, SIAM, 1996.
5. R. Guerin, S. Kamat, A. Orda, T. Przygienda, D. Williams, “QoS Routing Mechanisms and OSPF Extensions”, *Work in Progress*, Internet Draft, March 1997.
6. F.P. Kelly, “Routing in Circuit-Switched Networks: Optimization, Shadow Prices and Decentralization”, Advances in Applied Probability 20, 112-144, 1988.
7. F.P. Kelly, “Routing and capacity Allocation in Networks with Trunk Reservation”, Mathematics of Operations Research, 15:771-793, 1990.
8. F.P. Kelly, “Dynamic Routing in Stochastic Networks”, In *Stochastic Networks*, ed. F.P. Kellyand R.J. Williams, Springer-Verlag, 169-186, 1995.
9. F.P. Kelly, “Fixed Point Models of Loss Networks,” J. Austr. Math. Soc., Ser. B, 31, pp. 204-218, 1989.
10. Q. Ma, P. Steenkiste, “On Path Selection for Traffic with Bandwidth Guarantees”, IEEE ICNP, October 1997.
11. J. Moy, “OSPF Version 2”, Request For Comments 2328, Internet Engineering Task Force, April 1998.
12. S. Nelakuditi, Z-L. Zhang, R.P. Tsang, “Adaptive Proportional Routing: A Localized QoS Routing Approach”, IEEE INFOCOM’00, March 2000.
13. S. Nelakuditi, S. Varadarajan, and Z-L. Zhang, “On Localized Control in Quality-of-Service Routing”, October 2000. Submitted to IEEE Transactions on Automatic Control, Special Issue on Systems and Control Methods for Communication Networks.
14. S. Nelakuditi, and Z-L. Zhang, “On Selection of Paths for Multipath Routing”, Techincal Report, Department of Computer Science, University of Minnesota, April 2001.
15. M. Powell, “A Fast Algorithm for Nonlinear Constrained Optimization Calculations,” in *Numerical Analysis, ed. G.A. Watson, Lecture Notes in Mathematics, Vol 630,* Springer Verlag, 1978.
16. E. Rosen, A. Viswanathan, and R. Callon, “Multi-Protocol Label Switching Architecture”, *work in progress*, Internet Draft draft-ietf-mpls-arch-06.txt, August 1999.
17. K.W. Ross, “Multiservice Loss Models for Broadband Telecommunication Networks”, Springer-Verlag, 1995.
18. A. Shaikh, J. Rexford, K. Shin, “Evaluating the Overheads of Source-Directed Quality-of-Service Routing”, ICNP, October 1998.
19. N. Taft-Plotkin, B. Bellur, and R. Ogier, “Quality-of-Service Routing using Maximally Disjoint Paths”, IWQOS, June 1999.
20. C. Villamizar, “MPLS Optimized Multipath (MPLS-OMP)”, Internet Draft, February 1999.
21. C. Villamizar, “OSPF Optimized Multipath (OSPF-OMP)”, Internet Draft, February 1999.
22. Z. Wang, J. Crowcroft, “Quality-of-Service Routing for Supporting Multimedia Applications”, IEEE JSAC, September 1996