Evaluation of

A Load Profiling Approach to Routing Guaranteed Bandwidth Flows*

Ibrahim Matta†
College of Computer Science
Northeastern University
Boston, MA 02115
matta@ccs.neu.edu

Azer Bestavros
Computer Science Department
Boston University
Boston, MA 02215
best@cs.bu.edu

July 1997

Abstract

To support the diverse Quality of Service (QoS) requirements of real-time (*e.g.* audio/video) applications in integrated services networks, several routing algorithms that allow for the reservation of the needed bandwidth over a Virtual Circuit (VC) established on one of several candidate routes have been proposed. Traditionally, such routing is done using the least-loaded concept, and thus results in balancing the load across the set of candidate routes. In a recent study, we have established the inadequacy of this load balancing practice and proposed the use of *load profiling* as an alternative. Load profiling techniques allow the distribution of "available" bandwidth across a set of candidate routes to match the characteristics of incoming VC QoS requests.

In this paper we thoroughly characterize the performance of VC routing using load profiling and contrast it to routing using load balancing and load packing. We do so both analytically and via extensive simulations of multi-class traffic routing in Virtual Path (VP) based networks. Our findings confirm that for routing guaranteed bandwidth flows in VP networks, load balancing is not desirable as it results in VP bandwidth fragmentation, which adversely affects the likelihood of accepting new VC requests. This fragmentation is more pronounced when the granularity of VC requests is large. Typically, this occurs when a common VC is established to carry the *aggregate* traffic flow of many high-bandwidth real-time sources. For VP-based networks, our simulation results show that our load-profiling VC routing scheme performs better or as well as the traditional load-balancing VC routing in terms of revenue under both skewed and uniform workloads. Furthermore, load-profiling routing improves routing fairness by proactively increasing the chances of admitting high-bandwidth connections.

Keywords: Integrated services networks; virtual path based networks; admission control and routing of multi-class guaranteed flows; load balancing, packing, and profiling; real-time/on-line resource allocation; performance evaluation.

*This work was partially supported by research grant NSF CCR-9706685 and NU-RSDF 377090.

†Corresponding author. Tel: (617) 373-3694, fax: (617) 373-5121, e-mail: matta@ccs.neu.edu.

1 Introduction

Routing algorithms—allowing the selection of one out of many candidate source-to-destination paths for bandwidth reservation purposes—play a critical role in meeting the Quality of Service (QoS) requirements of real-time applications over high-speed integrated services networks, such as Asynchronous Transfer Mode (ATM) networks [29] and next generation Internet [6].

Routing Multi-Class Traffic under the VC Model

To support real-time QoS we adopt the *Virtual Circuit* (VC) model for resource reservation. Under this model, routing a connection (or VC) involves the selection of a path (or *route*) within the network from the source to the destination in such a way that the resources (*e.g.*, *bandwidth*) necessary to support the VC QoS requirements are set aside (or *reserved*) for use by the entity requesting the establishment of the VC. This entity might be an application or a router/switch. In the latter case, a router may request a VC to another router to carry the packets of a particular class of applications over a backbone network that connects internet service providers and implements IP switching [26] or similar schemes such as tag switching [11], IP/ATM [28], etc. Over the last few years, several routing protocols based on the VC model have been proposed (*e.g.*, [2, 27, 7]).

We consider a network that supports $S \geq 2$ classes of VCs. A VC of class s requires the reservation of a certain amount of bandwidth b_s that is enough to ensure a given QoS. This bandwidth can be thought of either as the peak transmission rate of the VC or its "effective bandwidth" [15, 10] which varies between the peak and average transmission rates. Without loss of generality, we assume that the bandwidths requested by different classes are distinct and that the classes are indexed in increasing order of their requested bandwidths, i.e., $b_1 < b_2 < \cdots < b_S$.

To support a class-s VC, the VC has to be setup on some path from the source to the destination; the QOS demand (b_s) is allocated on one of the candidate paths for the lifetime of the VC. The objective of the routing algorithm is to choose routes that result in high successful VC setup rate (or equivalently, high carried VC load) while maximizing the utilization of network resources (or equivalently, revenue).

Related Work

Traditionally, routing schemes have been based on the least-loaded concept (*e.g.*, [16, 9, 7, 18, 1, 3, 24]). According to this concept, a request is serviced by setting up the VC on the least utilized path selected from the set of candidate paths[1] between the source and destination, provided it can support the VC's bandwidth requirement. Thus, this scheme attempts to evenly distribute the load among the candidate routes. We call such scheme *Least Loaded Routing* (LLR).

As an alternative to the load-balancing philosophy of LLR techniques, VC *packing* techniques were proposed in [17]. The argument for VC packing is based on the observation that in order to maximize the utilization of available resources, a routing policy in a heterogeneous (multi-rate) environment should implement *packing* of narrowband VCs (having relatively small bandwidth requirement) on some paths in order to leave room on other paths for wideband VCs (having relatively large bandwidth requirement). Packing strategies achieve two desired properties: (1) They minimize the fragmentation of available bandwidth, resulting in an (2) improved fairness by increasing the chances of admittance for wideband VCs.

A routing scheme based on this packing concept was proposed in [17]. The scheme attempts to pack class-s VCs in order to reduce blocking only for the next higher class of VCs. In [22], we extended the scheme in order to reduce blocking for *all* higher classes. Both schemes are, however, based on pessimistic/deterministic analysis. They only account for the different bandwidth requirements of different classes, but not on their traffic intensities (demands). These traffic intensities may be known a priori (based on traffic forecasts) or dynamically estimated.

In a recently completed pilot study [5], we have established the inadequacy of load-balancing techniques and the impracticality of load-packing techniques. To that end, we proposed the use of *load profiling* as an alternative. Load profiling techniques allow the distribution of *available* bandwidth across a set of candidate routes to match the characteristics of incoming VC QoS requests.

In this paper, we investigate a load-profiling VC routing scheme based on the probabilistic selection of routes, where probabilities are chosen to match the distribution of traffic demand of different classes (i.e. the load profile) with the distribution of available resources on the candidate routes (i.e. resource availability profile). We call this scheme *Load Profiling Routing* (LPR). Alternately, a routing scheme that selects from the set of candidate routes the most utilized one is referred to as *Most Loaded Routing* (MLR). MLR is a simple scheme which attempts to achieve the same effect as packing-based schemes, and is asymptotically optimal (as will be shown in section 2). MLR performs particularly well when accurate feedback information about the available bandwidth on all candidate routes is available.

[1]To consume the least amount of resources, the set of candidate paths is typically chosen from the set of *shortest* paths.

Contributions

In this paper we thoroughly characterize the performance of VC routing using load profiling and contrast it to routing using load balancing and load packing. We do so both analytically and via extensive simulations.

Our findings confirm that for reservation-based protocols—which allow for the exclusive use of a preset fraction of a resource's bandwidth for an extended period of time—load balancing is not desirable as it results in resource fragmentation, which adversely affects the likelihood of accepting new reservations. This fragmentation is more pronounced when the granularity of VC requests is large. Typically, this occurs when a common VC is established to carry the *aggregate* traffic flow of many high-bandwidth real-time sources. These results support our preliminary investigation in [5] and indicate that LPR is a promising routing approach. LPR performs especially well in a distributed network environment, where a router's local view of global knowledge is often imprecise. In such environments, LPR is particularly appropriate because of its probabilistic selection of routes, which compensates for inaccuracy in the feedback information [25]. This stands in sharp contrast to MLR, which is susceptible to even minor inaccuracies in knowledge about reserved bandwidth on various routes.

For virtual path based networks, our simulation results show that our load-profiling VC routing scheme performs better or as well as the traditional load-balancing VC routing in terms of revenue under both skewed and uniform workloads. Furthermore, load-profiling routing improves routing fairness by proactively increasing the chances of admitting high-bandwidth connections.

Organization

The remainder of this paper is organized as follows. Section 2 motivates load profiling by comparing it to load balancing and load packing—it extends the analysis in [5] to emphasize the effect of VC request granularity. The comparison is done both analytically and via a pilot baseline simulation experiment using a simplified model of a single source-destination node pair connected by multiple paths, where the cost of a path is defined by its current available bandwidth. In Section 3 a comprehensive comparative evaluation of LPR versus LLR is presented using simulation of a fully-connected virtual path based network, where routing algorithms consider one-link and two-link paths. Here, the cost of a path is defined by not only its current available bandwidth but also its length; establishing a VC on a two-link path consumes twice as much bandwidth as a one-link path. We conclude in Section 4 with a summary and with directions for future work.

2 Load Profiling: On Neither Balancing nor Packing VC Requests

In this section we show that for reservation-based routing of guaranteed flows: (1) load balancing is not a desirable policy as it results in serious fragmentation of network resources, especially when the granularity of VC requests is large; and (2) load packing, while optimal, is not desirable due to its susceptibility to the inaccuracies about global state inherent in a distributed environment. We propose a load-profiling strategy that combines the advantages of both load balancing (namely tolerance to inaccuracies about feedback information) and load packing (maximal VC admission rates), while avoiding their disadvantages.

2.1 Overview

Load balancing is often used to ensure that resources in a distributed system are equally loaded. In [33], load balancing was found to reduce significantly the mean and standard deviation of job response times, especially under heavy or unbalanced workloads.

For best-effort systems, reducing the mean and standard deviation of the metric used to gauge performance (e.g. job response times or throughput) *is* indicative of better performance. This, however, is not necessarily the case for systems that require an "all or nothing" (quality of) service[2] such as for the bandwidth-reservation-based routing protocols that we consider in this paper.

In order to maximize the probability that an incoming request for a VC will be accepted, the routing protocol has to keep information about each source-destination path that could be used for the VC. The routing scheme we present in this paper does not use this information to achieve a load-balanced system. On the contrary, it allows paths to be unequally loaded so as to get a broad spectrum of available bandwidth across the various paths. We call this spectrum of available bandwidth, the *bandwidth availability profile*. By maintaining a bandwidth availability profile that resembles the expected characteristics of incoming requests for VC, the likelihood of succeeding in honoring these requests increases. We use the term *load profiling* to describe the process through which the availability profile is maintained.

We denote by MLR a load packing heuristic that assigns an incoming VC request to the most loaded path provided it can support the VC. We denote by LLR a load balancing scheme that assigns an incoming VC request to the least loaded path provided it can support the VC. In the remainder of this section, we motivate LPR as an attractive alternative to LLR and MLR. We start with an analysis that shows the optimality of MLR and the conditions under which LLR's

[2] Examples of such systems include bandwidth reservation for guaranteed QoS, and periodic or aperiodic real-time computational tasks [4].

performance degenerates. Next, we illustrate an example LPR technique and we present simulation results that confirm the premise of LPR when compared to MLR and LLR.

2.2 MLR versus LLR: An analytical comparison

Consider a system with M different paths between a particular source and a particular destination. Without loss of generality, we assume that the capacity of all such paths is identical and is normalized to a unit. Let $f(u)$ denote the probability density function for the utilization requirement of requests for VCs between the same source and destination considered above. That is $f(u)$ is the probability that the bandwidth requirement of a VC request will be u, where $0 \leq u \leq U$, where U is the largest possible bandwidth request. By virtue of the capacity assumption, $U \leq 1$.

Let W denote the overall load of the system, expressed as the sum of the reserved bandwidth over all paths (i.e. $M \geq W \geq 0$). A load-balanced system would tend to distribute its load (i.e. reserved bandwidth) equally amongst all paths, making the reserved bandwidth on each path as close as possible to W/M. A load-profiled system would tend to distribute its load in such a way that the probability of satisfying the QoS requirements of incoming VC requests is maximized. We explain a particular way of achieving such a goal next.

Let $\mathcal{C}$ denote the set of M paths in the system between a particular source-destination pair. For routing purposes, we assume the availability of a *routing policy* that allows the routing protocol to select a subset of routes from $\mathcal{C}$ that are *believed* to be capable of satisfying the QoS requirement u of an incoming VC request. We denote this *feasible set* by $\mathcal{F} \subseteq \mathcal{C}$.

Let $l_{\mathcal{F}}(u)$ denote the fraction of paths in a feasible set $\mathcal{F}$, whose *unused* (i.e. unreserved/available) bandwidth is equal to u. Thus, $L_{\mathcal{F}}(u) = \int_0^u l_{\mathcal{F}}(u)du$ could be thought of as the (cumulative) probability that the available bandwidth for a path selected at random from $\mathcal{F}$ will be less than or equal to u. Alternatively, $1 - L_{\mathcal{F}}(u)$ is the cumulative probability that the available bandwidth for a path selected at random from $\mathcal{F}$ will be larger than or equal to u, and thus enough to satisfy the demand of a VC request of u (or more) bandwidth.

Thus, the probability that a VC request will be accepted on a path selected randomly out of $\mathcal{F}$ is given by:[3]

$$P \quad = \quad \int_0^U f(u)(1 - L_{\mathcal{F}}(u))du \tag{1}$$

Let $l_{\mathcal{C}}(u)$ denote the fraction of paths in the system candidate set $\mathcal{C}$, whose unused bandwidth is equal to u. Denote by $L_{\mathcal{C}}(u)$ the cumulative distribution of available bandwidth for $\mathcal{C}$, i.e. $L_{\mathcal{C}}(u) = \int_0^u l_{\mathcal{C}}(u)du$.

[3] The integration is from 0 to U since U is the largest possible bandwidth request, *i.e.* $f(u) = 0$ for $U < u \leq 1$.

Load Balancing: In a perfectly load-balanced system, any feasible set of routes will be identical in terms of its bandwidth profile to the set of all routes in the system. Thus, in a load-balanced system $L_{\mathcal{F}}(u) = L_{\mathcal{C}}(u) = L(u)$. Moreover, we have:

$$L(u) = \begin{cases} 0 & \text{if } 0 \leq u < (1 - W/M) \\ 1 & \text{if } (1 - W/M) \leq u \leq 1 \end{cases} \tag{2}$$

The probability that a VC request will be accepted is given by $P = \int_0^V f(u)\ 1\ du$, where $V = \min(U, (1 - W/M))$. Thus,

$$P = \begin{cases} F(1 - W/M) & \text{if } 1 < \frac{U}{1-W/M} \\ 1 & \text{if } \frac{U}{1-W/M} \leq 1 \end{cases} \tag{3}$$

Equation (3) indicates that the performance of LLR is *dependent* on the system load. In particular, equation (3) predicts that LLR's performance will be optimal as long as the utilization of the system (W/M) is less than $1-U$, but that it will degenerate as soon as (W/M) bypasses that bound. The manner in which such a degeneration occurs will depend heavily on the distribution of requests $f(u)$.

Load Packing: A load-profiling algorithm would attempt to *shape* $L_{\mathcal{C}}(u)$ in such a way that the choice of a feasible set $\mathcal{F}$ would result in minimizing the value of $L_{\mathcal{F}}(u)$, thus maximizing the value of P in equation (1) subject to the boundary constraint $\int_0^1 u\ l_{\mathcal{C}}(u) du = (1 - W/M)$. One solution to this optimization problem is for $l_{\mathcal{C}}(u)$ to be chosen as $l_{\mathcal{C}}(u) = (W/M).\delta_u(0) + (1 - W/M).\delta_u(1)$ where $v.\delta_u(x)$ is an impulse function of magnitude v applied at $u = x$.

The above solution corresponds to a system that *packs* its load (or reserved bandwidth) using the minimal possible number of routes. In other words, a fraction W/M of the paths in the system are 100% utilized, and thus have *no* extra bandwidth to spare, whereas a fraction $(1-W/M)$ of the paths in the system are 100% idle, and thus able to service VC requests with *any* QoS requirements. The choice of any feasible set $\mathcal{F}$ from the set of unused routes in $\mathcal{C}$ would result in $L_{\mathcal{F}}(u)$ being a step function given by:

$$L_{\mathcal{F}}(u) = \begin{cases} 0 & \text{if } 0 \leq u < 1 \\ 1 & \text{if } u = 1 \end{cases} \tag{4}$$

Plugging these values into equation (1), we get

$$\begin{aligned} P &= \int_0^U f(u)\ (1 - 0)\ du \\ &= 1 \end{aligned} \tag{5}$$

Equation (5) shows that choosing $l_{\mathcal{C}}(u) = (W/M).\delta_u(0) + (1 - W/M).\delta_u(1)$ is obviously optimal. Furthermore, this optimality is *independent* of the system load or the request distribution $f(u)$.

The *perfect fit* implied in equation (4) may require that VCs already in the system be reassigned to a different path upon the submission and acceptance of a new VC request, or the termination of an existing VC. Even if such reassignment is tolerable, achieving a perfect fit is known to be NP-hard. For these reasons, heuristics such as *first-fit* or *best-fit* are usually employed for on-line resource allocation. Asymptotically, both the first-fit and best-fit heuristics are known to be optimal for the on-line *bin packing* problem [23]. However, for a small value of M—which is likely to be the case in network routing problems—best-fit outperforms first-fit.

2.3 MLR versus LLR: The effect of VC request granularity

An important distinction between LLR and MLR—evident from equations (3) and (5)—is the sensitivity (insensitivity) of LLR (MLR) to the request distribution $f(u)$. LLR's sensitivity to request distributions is pronounced most when the granularity of the requests is large—*i.e.* U approaches 1—and is insignificant when the granularity of the requests is small—*i.e.* U approaches 0.

To demonstrate the susceptibility of LLR, consider a uniform request distribution over the $[0-1]$ interval. According to equation (3), only one half of all VC requests will be possible to honor when the system utilization is 50%, and only one tenth when the system utilization is 90%.[4]

2.4 Load Profiling: A robust alternative to MLR

First-fit and best-fit heuristics work well when accurate information about the available bandwidth at all M paths between a source and a destination is available. This is not the case in a networking environment, where knowledge at the periphery of the network about reserved bandwidth on various paths within the network is often imprecise, and approximate at best.

In particular, equation (4) shows analytically that best-fit (or an MLR policy)—as an approximation of a perfect fit—is an appropriate heuristic for selecting a route from amongst a set of routes that satisfy the bandwidth requirement of a VC request. However, in a networking environment, the performance of best-fit is severely affected by the inaccuracy of knowledge about reserved bandwidth on various routes. The inadequacy of best-fit in a distributed environment

[4] For a request distribution with half the granularity–*i.e.* a uniform distribution over the $[0-0.5]$ interval—all VC requests will be possible to honor when the system utilization is 50%, and one fifth when the system utilization is 90%.

could be explained by noting that the best-fit heuristic is the *most* susceptible of all heuristics to even minor inaccuracies in knowledge about reserved bandwidth on various routes. This is due to best-fit's minimization of the slack on the target route—a minimal slack translates to a minimal tolerance for imprecision.

In the remainder of this section, we examine the details of a probabilistic load-profiling heuristic (LPR) that is more appropriate for the imprecision often encountered in distributed and networking environments. Using this LPR protocol, the process of choosing a target route from the set of feasible routes is carried out in such a way so as to maximize the probability of admitting future VC requests. The probability of picking a route from the set of feasible routes is adjusted in such a way that the availability profile of the system is maintained as close as possible to the expected profile of incoming VC bandwidth requests.

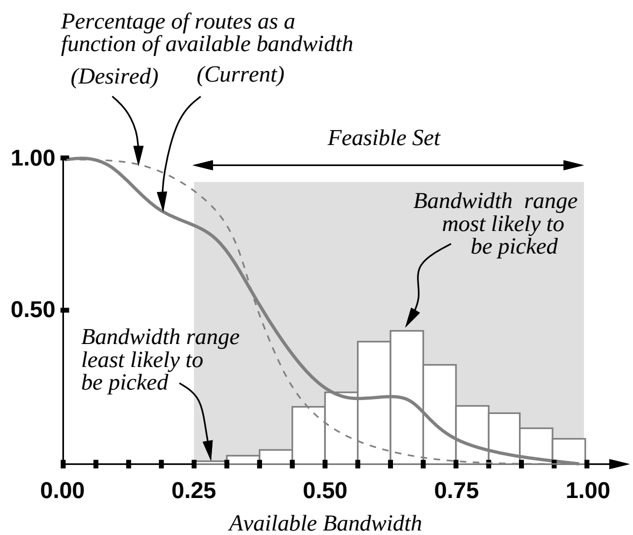


Figure 1: Maintaining a bandwidth availability profile that matches the characteristics of VC requests.

Figure 1 illustrates this idea. It shows two availability profile distributions. The first is the current availability profile of the system, which is constructed by computing the percentage of routes in the system with *available* (i.e. unused) bandwidth larger than a particular range. The second is the desired availability profile, which is constructed by matching the characteristics of incoming VC requests. From these two availability profiles, a probability density function (shown as a histogram in Figure 1) is constructed and a route is probabilistically chosen according to that density function.

2.5 Illustrative LPR Example

We explain our implementation of LPR through an illustrative example. Consider four classes of VCs with bandwidth requirements b_1, b_2, b_3 and b_4. Without loss of generality, assume $b_1 < b_2 < b_3 < b_4$. Assume the arrival rates are λ_1, λ_2, λ_3 and λ_4. Figure 2 shows the corresponding load profile, i.e. the distribution of requested bandwidths, Prob[requested bandwidth $\leq$ B]. It also shows the bandwidth availability profile, i.e. the frequency of routes with available bandwidth $\leq$ B.

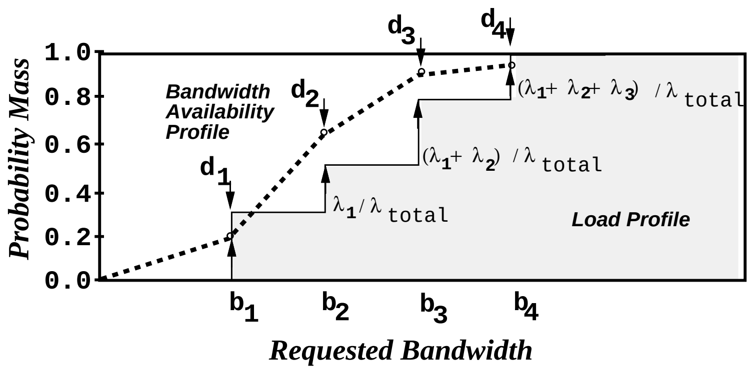


Figure 2: Example load profile and bandwidth availability profile.

The goal of LPR is to make the two profiles match as closely as possible. Denote by R_s the set of paths whose available bandwidth $\leq b_s$, $s = 1, 2, 3, 4$. These sets of routes are related as follows: $R_1 \subseteq R_2 \subseteq R_3 \subseteq R_4$. For a new incoming VC, we want to assign it a route from one of these sets. To do so, we compute the probability of choosing a path from each of the route sets. Let d_i $(i = 1, 2, 3, 4)$ be the differences between the load profile and the bandwidth availability profile (see Figure 2). We now assign a weight to each path according to the smallest route set it belongs to as shown in Table 1.[5] To compute a probability distribution, we scale the second column in Table 1 such that all values are non-negative. From the set of feasible paths we select a path probabilistically according to the resulting distribution.

In general, for S classes of VC requests, if R_k is the smallest route set to which a path p belongs, then the weight given to select p, $W(p,k)$, is given by:

$$W(p,k) \quad = \quad \sum_{i=k}^{S}(d_i - d_{min}) \tag{6}$$

where $d_{min} = min_j(\{d_j : j = 1, \cdots, S\})$. The complexity of this computation is proportional to the number of VC classes and candidate paths.

[5]Note that if a path $p \in R_i$ then $p \in R_j$ for all $j > i$.

Smallest route set	Weight of choosing the path
R_1	$d_1 + d_2 + d_3 + d_4$
R_2	$d_2 + d_3 + d_4$
R_3	$d_3 + d_4$
R_4	d_4

Table 1: Weight assigned to various routes.

2.6 Performance of LLR, MLR, and LPR

In the remainder of this section we summarize the results of a pilot performance evaluation study [5] that compared MLR, LPR and LLR in terms of how well they distribute VCs from multiple classes over a set of candidate paths between a given source and destination. As predicted in our analytical characterization in Section 2.2, these results confirm the superiority of a packing VC routing methodology over a load-balancing counterpart. In particular, our simulations show that MLR and LPR are competitive and that they both significantly outperform LLR. In section 3, we present a much more detailed simulation study that pits LPR to LLR in a more realistic networking environment.

Simulation Model and Setup: A class-s VC requires the reservation of b_s units of bandwidth. Each class-s VC, once it is successfully setup, has an infinite lifetime during which it holds b_s units of bandwidth.[6] The simulation run is stopped whenever an arriving VC blocks because none of the candidate paths is feasible. In other words, once an incoming request for a VC cannot be honored, the simulation is stopped and statistics are collected so as to examine the load distribution on the various paths that caused the system to start blocking VC requests. The performance metrics we report are the *total number of accepted VCs* and the *unutilized bandwidth*—the amount of bandwidth available on each path when the first VC blocking occurs. The results shown are the average of 15 independent runs (i.e. each run starts with a different random number seed).

Simulation Results: Figures 3 and 5 show our simulation results for 4 VC classes and 5 candidate paths. The requested bandwidths for the four VC classes are $b_1 = 10$, $b_2 = 16$, $b_3 = 22$ and $b_4 = 35$. The arrival rates for these classes are assumed equal—$\lambda_i = 0.25$ for $i = 1, 2, 3, 4$. The initial

[6] This infinite VC lifetime assumption is relaxed in the next section.

capacities of the 5 candidate paths are 20, 25, 30, 35, and 40.

Figures 4, 6 and 7 show our simulation results for 4 VC classes and 10 candidate paths. The requested bandwidths for the four VC classes are $b_1 = 10$, $b_2 = 16$, $b_3 = 22$ and $b_4 = 35$. We considered both equal and unequal class arrival rates. As before, for equal class arrival rates, $\lambda_i = 0.25$ for $i = 1, 2, 3, 4$. For the unequal class arrival rates, we set $\lambda_1 = 0.4$, $\lambda_2 = 0.3$, $\lambda_3 = 0.2$ and $\lambda_4 = 0.1$. The initial capacities of the 10 candidate paths are 20, 25, 30, 35, 40, 45, 50, 55, 60, and 65.

Observations: The results shown in Figures 3 through 7 lead to the following observations and conclusions.

- In terms of the total number of accepted VCs, MLR and LPR significantly outperform LLR. The advantage of using MLR becomes more pronounced with a smaller number of candidate paths as the gain from packing becomes more significant. This is also true with LPR although here the advantage of using LPR is more pronounced with more candidate paths as LPR is able to better distribute the load on the various paths to match the desired load profile before the first VC blocking occurs.

- For MLR, the first blocking occurs when the bandwidth utilization across all candidate paths (for both the 5-path and 10-path experiments) is around 85%. For LLR this number drops to around 50%. According to our analytical characterization for equal class arrival rates (*i.e.* uniform request distribution) a 50% utilization would result in a 50% VC admission rate for LLR and in a 100% VC admission rate for a perfect packing heuristic. While MLR (*i.e.* best-fit packing) approximates perfect packing only asymptotically [23], our results show that MLR's performance advantage is evident even at the small number of candidate paths we considered (namely 5 and 10).[7]

- In terms of the distribution of VCs, LLR balances the load over the candidate paths. This load balancing is clearly not a primary goal when routing real-time VCs. LPR and MLR have the more important goal of increasing the chance that future incoming VCs are accepted even at the expense of load balancing. This load imbalance is more pronounced with a higher load of large VCs. This can be seen by comparing Figures 6(a) and 7(a), where $\lambda_4 = 0.25$ and 0.1, respectively.

[7] In particular, at 50% utilization, while perfect packing is expected to outperform LLR by a factor of two, MLR outperforms LLR by a factor of about 1.6.

3 Simulation of LPR and LLR in Virtual Path Based Networks

In this section, we compare LLR and LPR in a network that uses the Virtual Path (VP) concept. This concept is often used to simplify network management and to increase the apparent direct connectedness of the network [8, 17, 30]. Typically, a VP is installed between two nodes (switches) over a sequence of physical links, and bandwidth is allocated to it. Thus a virtual fully-connected network can be overlayed over the physical network, where the VPs constitute the (virtual) links connecting the network nodes. Simple routing schemes that only consider paths with one link (called *direct routes*) and two links (called *alternative routes*) are then used. For a fully-connected network with N nodes, each pair of nodes has one direct route and $N-2$ two-link alternative routes. A number of such routing schemes were designed for telephone networks [13, 12, 3, 14] and recently for ATM networks [32, 17, 20, 21, 18, 19].

3.1 Simulation Model and Setup

We consider a fully-connected logical VP network, which could be carved out over an arbitrary underlying physical topology. We assume all VP links have the same total bandwidth. The network is used by a number of VC classes. A class-s VC requires the reservation of b_s units of bandwidth. We classify bandwidth demands into two categories: 1) *aggregate flow demands*, where the establishment of a VC requires the reservation of a large fraction of the total link bandwidth; and 2) *small flow demands*, where a VC bandwidth requirement is a small fraction of the total link bandwidth. As pointed out earlier, aggregate flow demands could constitute the workload on a multi-class backbone network where a node/router would request the establishment of a high-bandwidth VC to carry a type of real-time traffic coming from an internet service provider or a large number of sources. Class-s VC setup requests arrive to the network according to a Poisson process of rate λ_s. Each class-s VC, once it is successfully setup, has a lifetime of exponential duration with mean $1/\mu_s$.

We consider both uniform and skewed workloads. For a *uniform* workload, the source and destination nodes of an arriving VC are chosen randomly. Each VC class has the same arrival rate and average lifetime. Thus, on average, each node pair has the same VC traffic intensity for each class. In practice, workload is naturally skewed and each node pair may have different VC traffic intensities. To model a *skewed* workload, we assume each VC class has different arrival rate and average lifetime. Furthermore, the network is partitioned into two equal groups, each containing half of the total number of nodes N. The source and destination nodes of a VC are chosen randomly from the same group. The group is chosen with some specified probability, p_{skew}.

A node in another group may be chosen by the routing algorithm to act as the intermediate node in a two-link path. We consider routing algorithms that choose from the set of one-link and two-link paths. An arriving VC request rejected by the admission control algorithm—because resources are either unavailable or being reserved for future incoming VCs—is considered blocked and lost.

3.2 Routing and Admission Control Algorithms

Since we are considering routing over paths with different length (in terms of number of links), we have to take into consideration the fact that a VC established over a two-link alternative route consumes twice as much bandwidth compared to when the VC is established over the one-link direct route. The trunk reservation concept [3, 24] is often used to address this issue. Here each link has a Trunk Reservation (TR) value associated with it. A two-link alternative route is said to be *TR-permissible* if, for each of its links, there is still a certain amount of idle bandwidth available beyond the corresponding trunk reservation level. For example, consider a link (on an alternative route) that has idle bandwidth of 100 units and TR value of 10 units, then the idle bandwidth considered available is 100 - 10 = 90 units.

Consider a traditional LLR scheme with trunk reservation. When a new VC arrives, it is setup on the direct route between the VC's source and destination provided it can support the VC's bandwidth requirement. Otherwise, the VC is setup on the *least-loaded* TR-permissible alternative route if there is at least one that can support the VC. Thus, the scheme attempts to evenly distribute the load among the alternative routes. If the direct route and all the two-link alternative routes are unavailable, the VC is blocked. Trunk reservation is used in order to discourage using two-link routes, and thus reserve some amount of bandwidth for future direct VCs.

Before we present more formally the LLR and LPR algorithms with trunk reservation, we first introduce the following definitions.

Idle Capacity: The *idle capacity* of a link is defined as the amount of link bandwidth that is currently not in use. We define the idle capacity of a route as the minimum idle capacity of all its links.

QOS-permissibility: A route is said to be *QOS-permissible* if it has sufficient idle capacity to carry the VC.

TR-permissibility: In this paper, we use two definitions for the *TR-permissibility* of a two-link alternative route. For simplicity, we will assume that all links have the same TR value.

Definition 1. An alternative route is said to be TR-permissible if its idle capacity minus the reservation threshold is greater than or equal to the requested bandwidth of the incoming VC [17].

Note that the idle capacity should then exceed a certain amount of bandwidth that varies depending on the class of the incoming VC. This further discourages higher VC classes (with higher bandwidth requirements) from using alternative routes. We thus refer to this as "class-dependent reservation". Also note that if an alternative route is TR-permissible then it is also QOS-permissible, and hence allowable (see Lemma A.1 in Appendix).

Definition 2. An alternative route is said to be TR-permissible if *only* when it carries at least one direct VC on one of its links, the idle capacity must be greater than or equal to a reservation threshold that is independent of the class of the incoming VC.

This definition of TR-permissibility requires that switches keep track of the number of direct VCs on outgoing links. This avoids unnecessary reservations for direct VCs when not present. Also, since the reservation does not depend on the class, we ensure that all classes are treated fairly concerning the use of alternative routes. We refer to this as "class-independent reservation".

Allowable Alternative Routes: A two-link alternative route is said to be *allowable* if it is both QOS-permissible and TR-permissible.

3.2.1 Least-Loaded Routing (LLR)

The following steps are executed when a new VC arrives:

1. Set up the VC along the direct route if the direct route is QOS-permissible. Otherwise, go to step 2.

2. If no allowable alternative routes are available, then the VC request is rejected. Otherwise, set up the VC on the allowable alternative route with the largest idle capacity, i.e. the least loaded.

3.2.2 Load Profiling Routing (LPR)

LPR constructs the bandwidth availability profile from the current bandwidth available on the direct and alternative routes between the source and destination. It constructs the desired load profile from the class arrival probabilities of incoming VC requests. The following steps are executed when a new VC arrives:

1. Set up the VC along the direct route if the direct route is QOS-permissible. Otherwise, go to step 2.

2. If no allowable alternative routes are available, then the VC request is rejected. Otherwise, assign selection probabilities to allowable alternative routes according to the difference between the bandwidth availability profile and the desired load profile. Select an allowable alternative route probabilistically to setup the VC.

3.3 Performance Measures

To evaluate the performance of the algorithms, our main measure is *revenue*, which is defined as

$$revenue = \sum_{k=1}^{S} \rho_k \, (1 - B_k) \, b_k$$

where $\rho_k = \frac{\lambda_k}{\mu_k}$, and B_k is the blocking probability of class k.
The revenue measure reflects the fact that a commercial network provider's earnings depend not only on the number of VCs admitted, but also on the total amount of VC bandwidth in use.

We also define the *carried load* to be the average number of VCs carried by the network.

$$carried\ load = \sum_{k=1}^{S} \rho_k \, (1 - B_k)$$

The length of each simulation run is 200,000 events (an event is either a VC arrival or departure). We ignore the first 20,000 events to account for transient effects. Results are obtained by averaging five independent runs (i.e. each run starts with a different random number seed).

3.4 Simulation Results for Aggregate Flows

Figures 8 and 9 show results for a 20-node network, i.e., $N = 20$. Each VP link has a total of C units of bandwidth. Here we take $C = 20$. We have four classes of VC with $b_1 = 1.0$, $b_2 = 5.0$, $b_3 = 10.0$ and $b_4 = 15.0$. Trunk reservation is not used in these experiments. Figure 8 shows results for a skewed workload. The arrival rates are $\lambda_1 = 0.4\lambda$, $\lambda_2 = 0.3\lambda$, $\lambda_3 = 0.2\lambda$ and $\lambda_4 = 0.1\lambda$, where λ is the total VC arrival rate. The departure rates are $\mu_1 = 0.004$, $\mu_2 = 0.003$, $\mu_3 = 0.002$ and $\mu_4 = 0.001$. We take $p_{skew} = 0.8$. We observe that LPR outperforms LLR in terms of revenue while maintaining about the same carried load.

Figure 9 shows results for a uniform workload. The arrival rates are $\lambda_i = 0.25\lambda$ for $i = 1, 2, 3, 4$, where λ is the total VC arrival rate. The departure rates are $\mu_i = 0.002$ for $i = 1, 2, 3, 4$. We observe that LPR still has a higher revenue, although the gain from load profiling is less than that obtained in the skewed workload case. The reason is that this gain is reduced due to the negative effect LPR may have on direct VCs as it tends to load two-link alternative paths nonuniformly and may overload some links resulting in some VCs being alternately routed instead of being directly routed over those (overloaded) links. This leads to increased bandwidth consumption.

Figure 10 shows the class blocking probabilities for LPR and LLR under the skewed workload with $\lambda = 1$. LPR reduces the unfairness seen by high-bandwidth (class-4) VCs by reducing their blocking by about 7% at the expense of slight increase in blocking for lower classes.

3.5 Simulation Results for Small Flows

Figures 11 and 12 show results for a network with $N = 20$, $C = 96$, and without trunk reservation. We have four classes of VC with $b_1 = 1.3$, $b_2 = 4.1$, $b_3 = 6.7$ and $b_4 = 9.9$. As in Section 3.4, for skewed workload (Figure 11), the arrival rates are set to $\lambda_1 = 0.4\lambda$, $\lambda_2 = 0.3\lambda$, $\lambda_3 = 0.2\lambda$ and $\lambda_4 = 0.1\lambda$, where λ is the total VC arrival rate. The departure rates are $\mu_1 = 0.004$, $\mu_2 = 0.003$, $\mu_3 = 0.002$ and $\mu_4 = 0.001$. Notice that we have chosen the parameters such that the highest class of VC, which might represent large video connections requiring the largest amount of bandwidth, arrives less often and holds on longer. For uniform workload (Figure 12), the arrival rates are set to $\lambda_i = 0.25\lambda$ for $i = 1, 2, 3, 4$, where λ is the total VC arrival rate. The departure rates are $\mu_i = 0.002$ for $i = 1, 2, 3, 4$. We also compare the LLR and LPR algorithms to a simple DIRECT routing algorithm that uses *only* direct (one-link) paths.

We observe that LLR performs better than LPR in terms of both revenue and carried load. The gain from load profiling is offset by the loss from overloading some links on alternative routes causing VCs to be alternately routed instead of being directly routed on those (overloaded) links. As pointed out earlier, the gain from load profiling in terms of reduced resource fragmentation is less pronounced with smaller demands. In the skewed workload case, both LLR and LPR are significantly superior to DIRECT (as expected) as they make use of available bandwidth on alternative routes.

However, in the uniform workload case, DIRECT significantly outperforms both LLR and LPR. This is due to the uniformity of the traffic, which implies that all node pairs have, on average, equal VC traffic intensity. Thus, it is more beneficial to minimize using alternative routes whose links are then used by direct VCs, thus conserving network bandwidth. To overcome this drawback of adaptive routing, link reservation thresholds should be used so that an adaptive routing algorithm would converge to direct routing as the load on alternative routes increases.

Routing with Trunk Reservation

Optimal reservation thresholds have often been determined assuming a fixed (known) input traffic pattern (e.g. [31]). For simplicity, we assume all links have the same reservation threshold. We set the reservation threshold such that revenue is maximized. Figure 13 shows revenue versus reservation threshold for LLR under the skewed workload with $\lambda = 11$, where TR-permissibility is defined as given by Definition 1. It illustrates that there exists an optimal reservation threshold that maximizes revenue. This optimal value depends on the algorithm used and the workload. For example, the optimal value here is 4. This suggests that the reservation threshold should be dynamically varied (see Section 4). In the following, for each algorithm, we plot the results corresponding to the reservation threshold that maximizes revenue.

We denote by LLR_res1 (LLR_res2) the LLR algorithm with TR-permissibility given by Definition 1 (Definition 2). Figure 14 shows that LLR_res2 outperforms both LLR_res1 and LLR (without trunk reservation). This is because LLR_res2 uses a class-independent reservation giving *all* classes an equal chance at using alternative routes. Thus henceforth we only use Definition 2 for TR-permissibility. Figure 15 shows that under skewed workload, LPR_res2 is competitive to LLR_res2 in terms of revenue, albeit a decrease in carried load as it tends to accept fewer low-bandwidth VCs and more bandwidth-intensive VCs, thus reducing unfairness. Figure 16 shows that under uniform workload, LPR_res2 and LLR_res2 schemes exhibit similar performance. As expected, DIRECT[8] is not significantly worse than both schemes as is the case under skewed workload. In fact, DIRECT starts to provide similar revenue at high λ, where it is more advantageous to completely avoid using alternative routes.

Although in the case of small flows, the gain from LPR_res2 due to load profiling is overshadowed by its negative effect on direct VCs resulting in similar revenue as LLR_res2, load profiling is still beneficial in reducing unfairness seen by high-bandwidth VCs. This is demonstrated here by a lower carried load. Figure 17 illustrates this by showing the class blocking probabilities for LPR_res2 and LLR_res2 under the skewed workload with $\lambda = 11$. LPR_res2 reduces the blocking probability of the highest class at the expense of increased blocking for lower classes. This improves fairness among traffic classes by bringing the blocking probability of different classes within a smaller range.

[8] Note that with DIRECT, no reservation threshold is used since alternative paths are not used.

4 Conclusion and Future Work

We presented a novel approach to routing guaranteed bandwidth flows in virtual path networks. The approach is based on our recently proposed concept of load profiling. We showed that a probabilistic routing scheme based on load profiling (LPR) performs better than the traditional least-loaded-based routing (LLR) scheme. LPR relies on actively matching the distribution of available resources (resource availability profile) with the distribution of Virtual Circuit (VC) QoS requirements (VC load profile). The VC load profile may be known a priori (based on traffic forecasts) or dynamically estimated as is often done in telephone networks [3].

Our findings (both analytically and via simulations) confirm that for routing guaranteed bandwidth flows in Virtual Path (VP) networks—which allow for the exclusive use of a preset fraction of a VP's bandwidth for an extended period of time—LLR is not desirable as it results in VP bandwidth fragmentation, which adversely affects the likelihood of accepting new VC requests. This fragmentation is more pronounced when the granularity of VC requests is large. Typically, this occurs when a common VC is established to carry the *aggregate* traffic flow of many high-bandwidth real-time sources.

As an alternative to LLR, our simulations have shown that LPR's performance is competitive to the asymptotically optimal [23] most-loaded-based routing (MLR), while being much less susceptible to (more tolerant of) the inaccuracies in the feedback information inherent in a distributed network system because of its probabilistic selection of routes. LPR's use of probabilistic route selection also results in using *multiple* paths simultaneously *during* a routing information update as opposed to using a single path (the least-loaded) when LLR is employed. This multi-path routing further improves performance, and allows for using even a longer routing update interval, thus reducing routing (processing and communication) overheads. In VP networks, LPR provides better revenue for aggregate VC requests. Also, it reduces unfairness among VC classes by reducing blocking for high-bandwidth classes at the expense of increased blocking for low-bandwidth classes.

Future work remains to further improve LPR routing. One issue we are pursuing is to consider the "length" of the VC request, i.e. the lifetime of the VC. In many applications, the lifetime of the VC may be known (or possible to estimate/predict a priori). Taking into consideration the lifetime of the VC may be useful in achieving a better "profiling". We are also developing mechanisms for the dynamic control of reservation thresholds. In particular, we are currently investigating a dynamic scheme that increases reservation thresholds as direct VCs are blocked, and decreases them as direct VCs are admitted. This is of practical interest when the input traffic is time-varying.

Appendix

Lemma A.1 *If an alternative route P is TR-permissible then it is also QOS-permissible, and hence allowable.*

Proof:

Denote by $IC[P]$ the idle capacity of route P, and by TR its reservation threshold. Then,

$$\begin{aligned} P \text{ is TR} - \text{permissible} &\implies IC[P] - TR \geq b_s \\ &\implies IC[P] \geq b_s \\ &\implies P \text{ is QOS} - \text{permissible} \\ &\implies P \text{ is allowable} \quad \Box \end{aligned}$$

References

[1] H. Ahmadi, J. Chen, and R. Guerin. Dynamic Routing and Call Control in High-Speed Integrated Networks. In Proc. *Workshop on Systems Engineering and Traffic Engineering, ITC'13*, pages 19–26, Copenhagen, Denmark, June 1991.

[2] C. Alaettinoglu, I. Matta, and A.U. Shankar. A Scalable Virtual Circuit Routing Scheme for ATM Networks. In Proc. *International Conference on Computer Communications and Networks - ICCCN '95*, pages 630–637, Las Vegas, Nevada, September 1995.

[3] G. Ash, J. Chen, A. Frey, and B. Huang. Real-time Network Routing in a Dynamic Class-of-Service Network. In Proc. *13th ITC*, Copenhagen, Denmark, 1991.

[4] A. Bestavros. Load Profiling in Distributed Real-Time Systems. *Journal of Information Sciences*, 101(1-2):1–27, 1997.

[5] A. Bestavros and I. Matta. Load Profiling for Efficient Route Selection in Multi-Class Networks. In Proc. *IEEE ICNP*, 1997. To appear. Also, Technical Report NU-CCS-97-08.

[6] B. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture: An Overview. Internet Draft, October 1993.

[7] L. Breslau, D. Estrin, and L. Zhang. A Simulation Study of Adaptive Source Routing in Integrated Services Networks. Available by anonymous ftp at catarina.usc.edu:pub/breslau, September 1993.

[8] I. Chlamtac, A. Faragó, and T. Zhang. Optimizing the System of Virtual Paths. *IEEE/ACM Transactions on Networking*, 2(6):581–587, December 1994.

[9] S-P. Chung, A. Kashper, and K. Ross. Computing Approximate Blocking Probabilities for Large Loss Networks with State-Dependent Routing. *IEEE/ACM Transactions on Networking*, 1(1):105–115, February 1993.

[10] A. Elwalid and D. Mitra. Effective Bandwidth of General Markovian Traffic Sources and Admission Control of High-Speed Networks. *IEEE/ACM Transactions on Networking*, 1(3):329–343, June 1993.

[11] Y. Rekhter et al. Tag Switching Architecture Overview. Internet Draft, September 1996.

[12] F. Le Gall and J. Bernussou. An Analytical Formulation for Grade of Service Determination in Telephone Networks. *IEEE Transactions on Communications*, COM-31(3):420–424, March 1983.

[13] A. Girard. *Routing and Dimensioning in Circuit-Switched Networks*. Addison-Wesley Publishing Company, 1990.

[14] A. Girard and M. Bell. Blocking Evaluation for Networks with Residual Capacity Adaptive Routing. *IEEE Transactions on Communications*, COM-37:1372–1380, 1989.

[15] R. Guerin, H. Ahmadi, and M. Naghshineh. Equivalent Capacity and its Application to Bandwidth Allocation in High-Speed Networks. *IEEE J. Select. Areas Commun.*, SAC-9(7):968–981, September 1991.

[16] R. Guerin, A. Orda, and D. Williams. QoS Routing Mechanisms and OSPF Extensions. Internet Draft, November 1996.

[17] S. Gupta. *Performance Modeling and Management of High-Speed Networks*. PhD thesis, University of Pennsylvania, Department of Systems, 1993.

[18] S. Gupta, K. Ross, and M. ElZarki. Routing in Virtual Path Based ATM Networks. In Proc. *GLOBECOM '92*, pages 571–575, 1992.

[19] S. Gupta, K. Ross, and M. ElZarki. On Routing in ATM Networks. In Proc. *IFIP TC6 Task Group/WG6.4 Workshop on Modeling and Performance Evaluation of ATM Technology*. H. Perros, G. Pujolle, and Y. Takahashi (Editors). Elsevier Science Publishers B.V., Amsterdam, The Netherlands, 1993.

[20] R.-H. Hwang. *Routing in High-Speed Networks*. PhD thesis, University of Massachusetts, Department of Computer Science, May 1993.

[21] R-H. Hwang, J. Kurose, and D. Towsley. MDP Routing in ATM Networks Using Virtual Path Concept. In Proc. *IEEE INFOCOM*, pages 1509–1517, Toronto, Ontario, Canada, June 1994.

[22] I. Matta and M. Krunz. Packing and Least-Loaded Based Routing in Multi-Rate Loss Networks. In Proc. *IEEE ICC*, pages 827–831, 1997.

[23] C. McGeoch and J. Tygar. When are best fit and first fit optimal? In Proc. *1988 SIAM Conference on Discrete Mathematics*, 1988. Also, Technical report, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA, October 1987.

[24] D. Mitra, R. Gibbens, and B. Huang. Analysis and Optimal Design of Aggregated-Least-Busy-Alternative Routing on Symmetric Loss Networks with Trunk Reservation. In Proc. *13th ITC*, Copenhagen, Denmark, 1991.

[25] M. Mitzenmacher. Load Balancing and Density Dependent Jump Markov Processes. In Proc. *FOCS '96*, 1996.

[26] P. Newman, T. Lyon, and G. Minshall. Flow Labelled IP: A Connectionless Approach to ATM. In Proc. *IEEE INFOCOM '96*, pages 1251–1260, San Francisco, CA, March 1996.

[27] C. Parris and D. Ferrari. A Dynamic Connection Management Scheme for Guaranteed Performance Services in Packet-Switching Integrated Services Networks. Technical Report TR-93-005, International Computer Science Institute, Berkeley, California, January 1993.

[28] G. Parulkar, D. Schmidt, and J. Turner. IP/ATM: A Strategy for Integrating IP with ATM. In Proc. *ACM SIGCOMM '95*, page 9, Cambridge, MA, September 1995.

[29] M. Prycker. *Asynchronous Transfer Mode - Solution for Broadband ISDN*. Prentice Hall, 1995.

[30] Y. Sato and K. Sato. Virtual Path and Link Capacity Design for ATM Networks. *IEEE J. Select. Areas Commun.*, SAC-9:104–111, January 1991.

[31] S. Sibal and A. DeSimone. Controlling Alternate Routing in General-Mesh Packet Flow Networks. In Proc. *ACM SIGCOMM '94*, pages 168–179, September 1994.

[32] E. Wong, A. Chan, S. Chan, and K. Ko. Bandwidth Allocation and Routing in Virtual Path Based ATM Networks. In Proc. *IEEE ICC*, pages 647–652, 1996.

[33] S. Zhou. *Performance Studies of Dynamic Load Balancing in Distributed Systems*. PhD thesis, University of California Berkeley, Computer Science Department, 1987. Also TR: CSD-87-376.

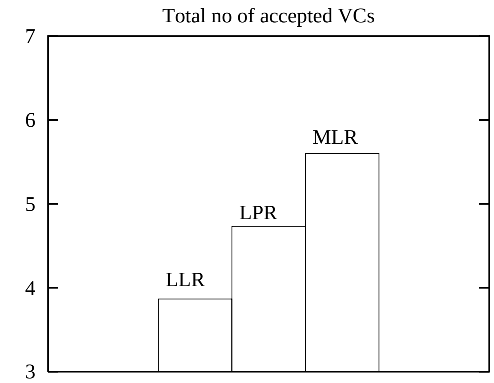


Figure 3: Total number of accepted VCs until first VC blocking occurs for the 5-path simulation experiments with equal class arrival rates.

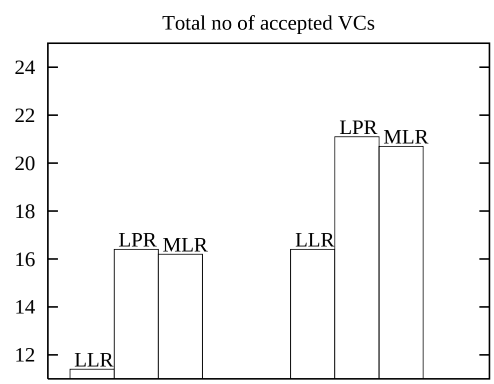
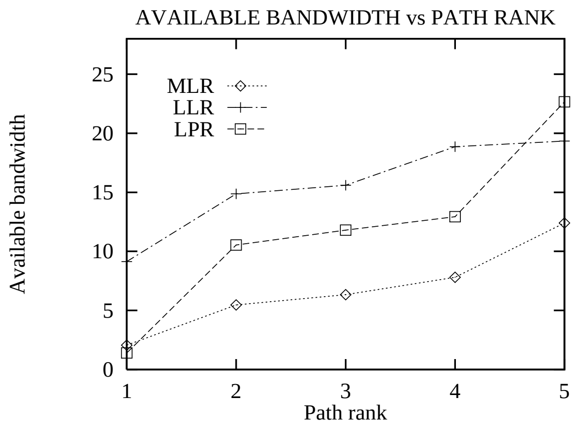
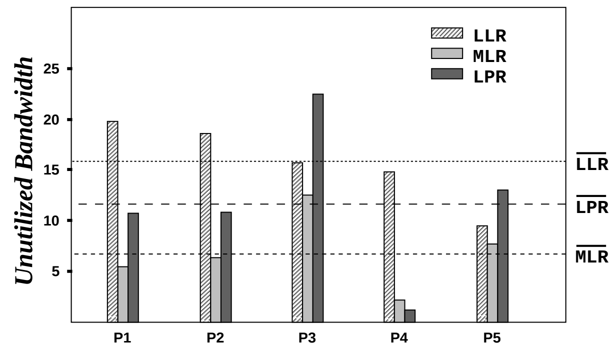


Figure 4: Total number of accepted VCs until first VC blocking occurs for the 10-path simulation experiments with equal class arrival rates (left) and unequal class arrival rates (right).

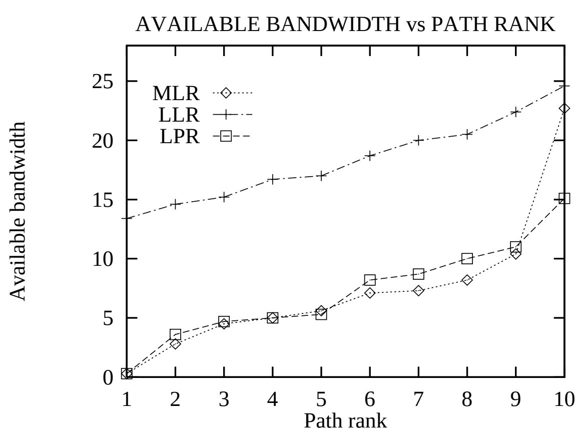


(a)

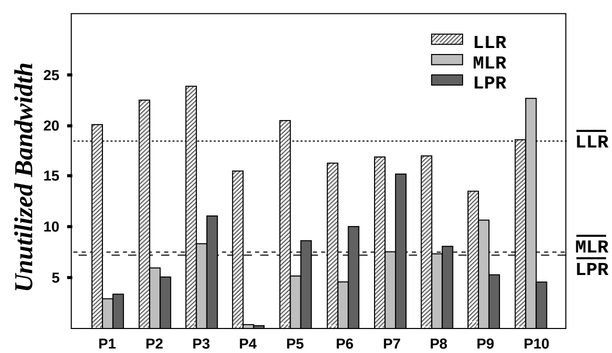


(b)

Figure 5: Unutilized bandwidth after first VC blocking occurs for the 5-path simulation experiments with equal class arrival rates: (a) Ranked unused bandwidth (b) Unused bandwidth per path.

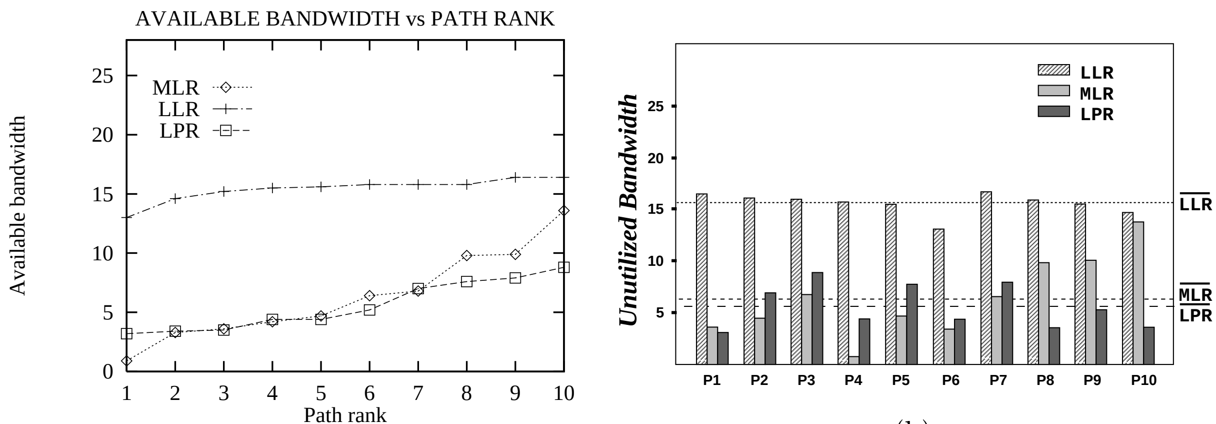


(a)



(b)

Figure 6: Unutilized bandwidth after first VC blocking occurs for the 10-path simulation experiments with equal class arrival rates: (a) Ranked unused bandwidth (b) Unused bandwidth per path.



(a) (b)

Figure 7: Unutilized bandwidth after first VC blocking occurs for the 10-path simulation experiments with unequal class arrival rates: (a) Ranked unused bandwidth (b) Unused bandwidth per path.

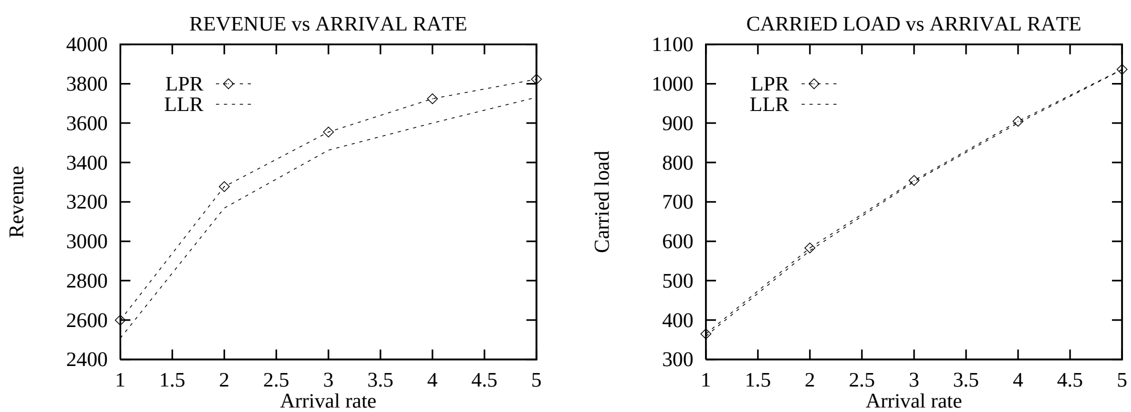


Figure 8: Revenue and carried load versus total VC arrival rate. Aggregate flows, skewed workload.

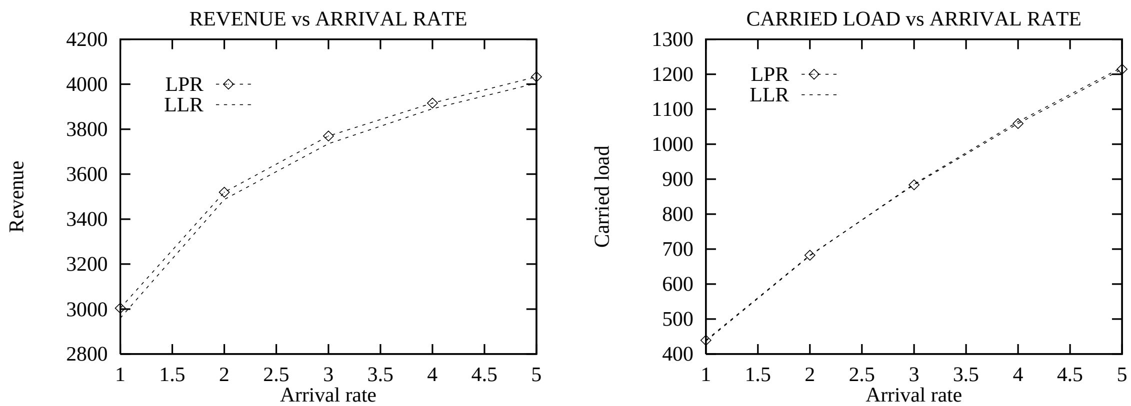


Figure 9: Revenue and carried load versus total VC arrival rate. Aggregate flows, uniform workload.

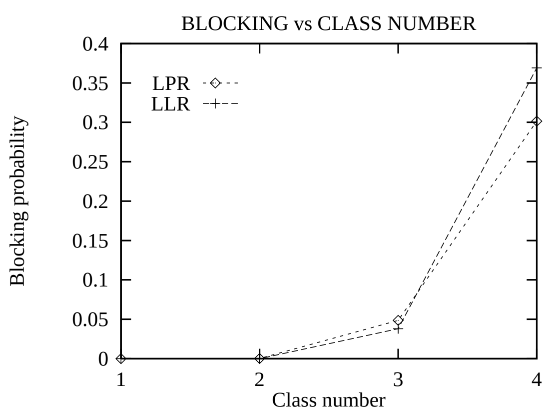


Figure 10: Class blocking probability versus class number. Aggregate flows, skewed workload. VC arrival rate = 1.

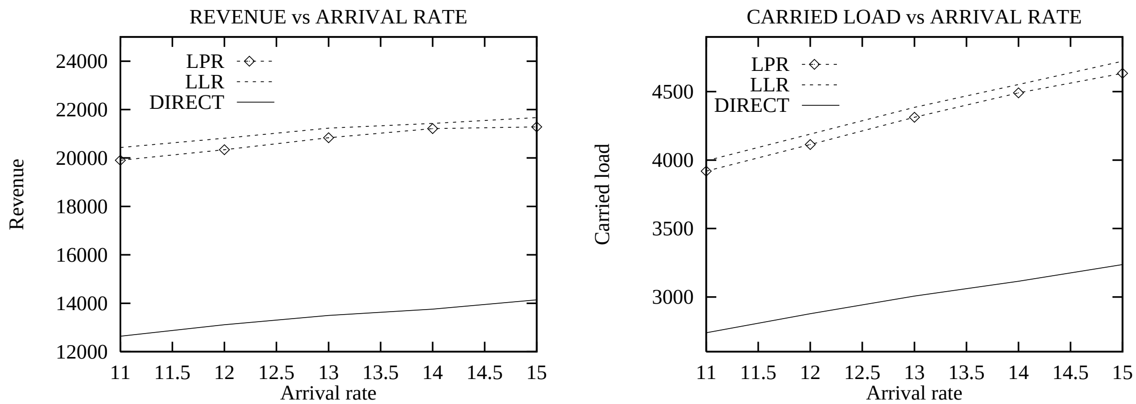


Figure 11: Revenue and carried load versus total VC arrival rate. Small flows, skewed workload.

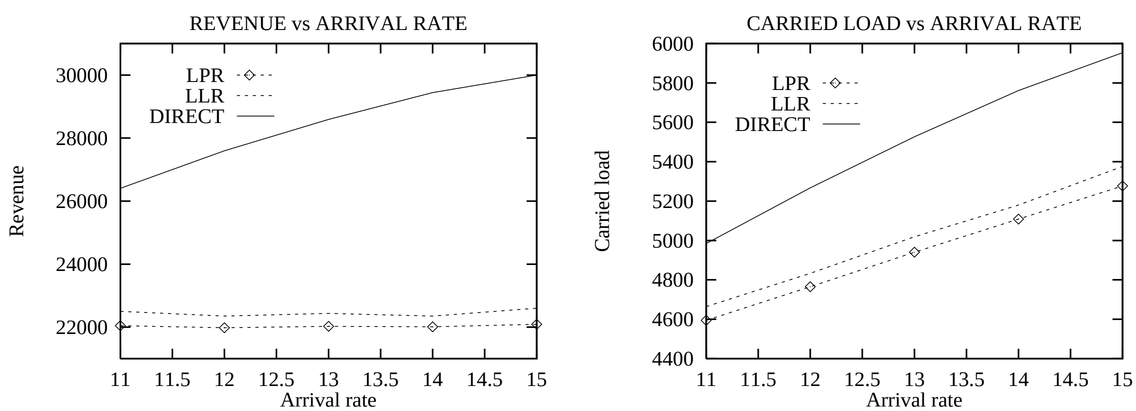


Figure 12: Revenue and carried load versus total VC arrival rate. Small flows, uniform workload.

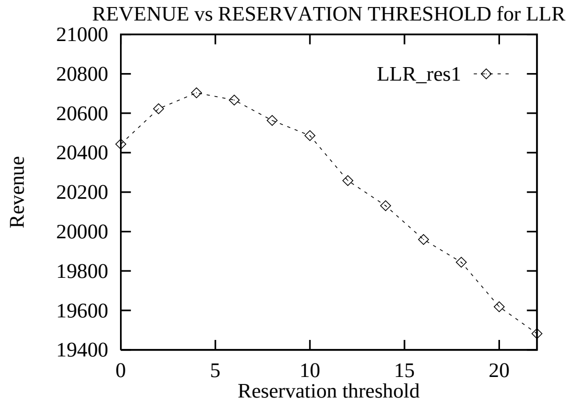


Figure 13: Revenue versus reservation threshold for LLR. Small flows, skewed workload. VC arrival rate = 11.

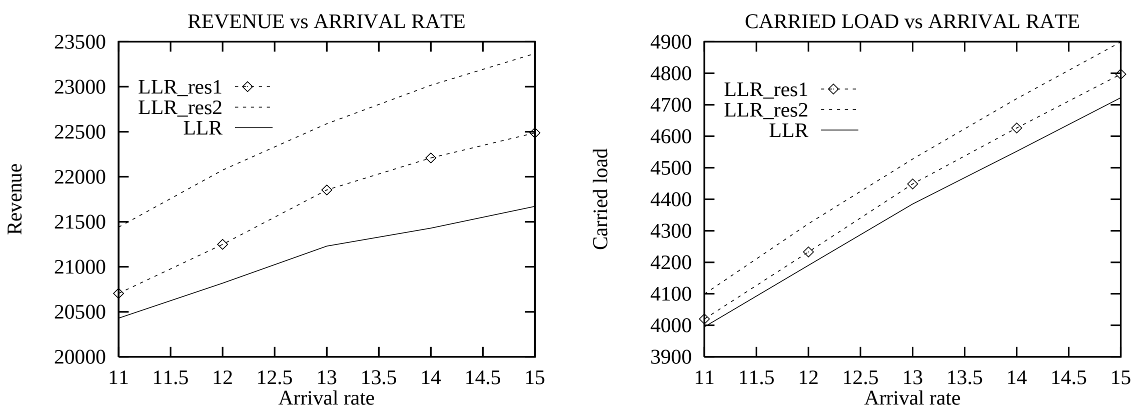


Figure 14: Revenue and carried load versus total VC arrival rate. Small flows, skewed workload.

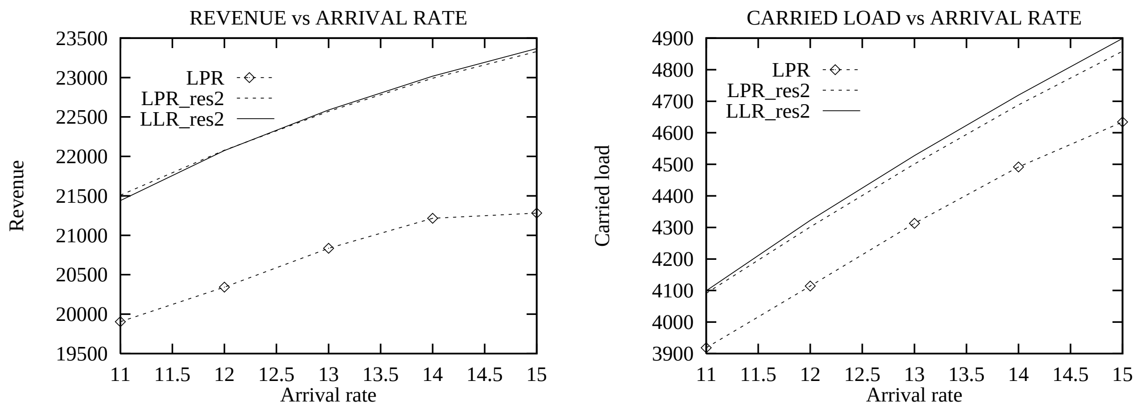


Figure 15: Revenue and carried load versus total VC arrival rate. Small flows, skewed workload.

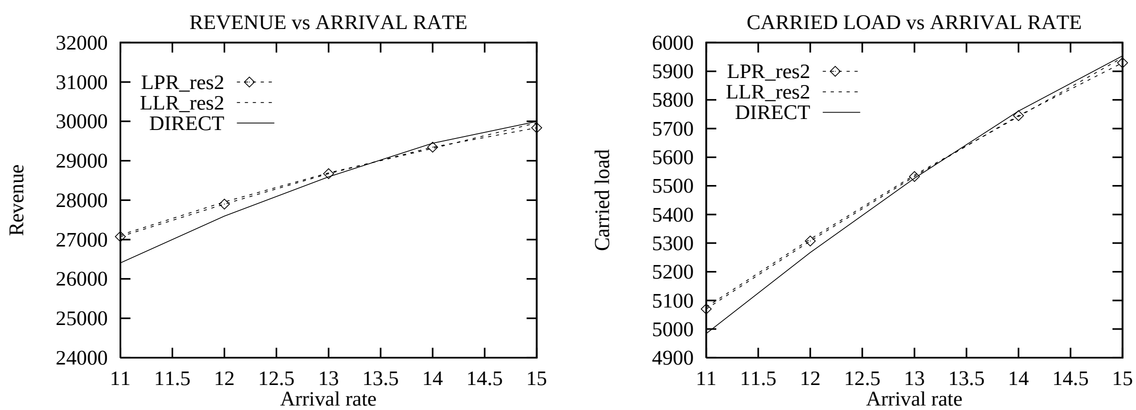


Figure 16: Revenue and carried load versus total VC arrival rate. Small flows, uniform workload.

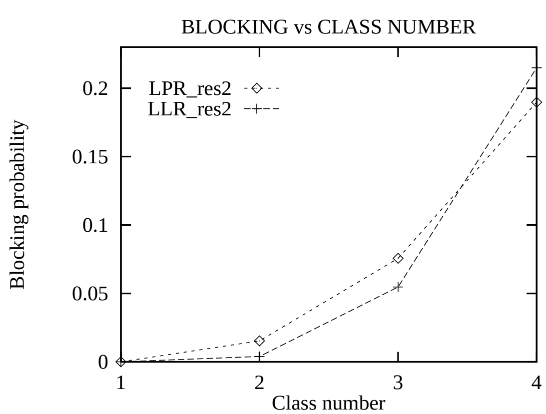


Figure 17: Class blocking probability versus class number. Small flows, Skewed workload. VC arrival rate = 11.