Load Profiling for Efficient Route Selection in Multi-Class Networks*

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

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

Abstract

High-speed networks, such as ATM networks, are expected to support real-time Quality of Service (QoS) guarantees required by many applications such as those that involve voice and video communication. To support such services, routing algorithms that allow for the reservation of the needed bandwidth over a Virtual Circuit (VC) have been proposed. Commonly, these algorithms assign VCs to routes using the least-loaded concept, and thus result in balancing the load over the set of all candidate routes. In this paper, we show that for such reservation-based VC routing algorithms—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 VC requests. We present an on-line VC routing scheme that is based on the concept of "load profiling", which allows a distribution of "available" bandwidth across a set of candidate routes to match the characteristics of incoming VC QoS requests. We show the effectiveness of our load-profiling approach when compared to traditional load-balancing and load-packing VC routing schemes.

1 Introduction

High-speed integrated networks, such as Asynchronous Transfer Mode (ATM) networks [13], are expected to carry traffic of a wide variety of applications (e.g., multimedia, voice, and mail) with heterogeneous Quality of Service (QoS) requirements. To meet these requirements, resource allocation algorithms for scheduling, admission and routing are needed to control the sharing of resources among the different service classes. *Scheduling algorithms* are responsible for the allocation of link resources (bandwidth, buffers, etc.) among the different services. *Admission algorithms* are responsible for accepting or rejecting a new incoming application/call, based on the requested QoS and the available resources. *Routing algorithms* are responsible for the selection of the particular route—which should have sufficient resources to satisfy the application's QoS requirements—to be taken by application packets (or cells) to reach their destination. In this paper, we address the issue of routing real-time traffic (e.g., voice and video) requiring QoS guarantees (e.g., bandwidth and delay guarantees).

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

VC Routing in Multi-Class Networks

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 application or protocol entity requesting the establishment of the VC. Over the last few years, several routing protocols based on the VC model have been proposed (e.g. [2, 5]).

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" [6] 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.

Routing Algorithms

Routing schemes are commonly based on the least-loaded concept (e.g., [7, 5, 1, 3]). When a new VC arrives, it is setup on the least utilized candidate route provided it can support the VC's bandwidth requirement. Thus, the scheme attempts to evenly distribute the load among the candidate routes. We call such scheme *Least Loaded Routing* (LLR).

Recently [8], it has been recognized 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). This packing strategy achieves two desired properties: (1) it minimizes the fragmentation of available bandwidth, which in turn results in (2) improved fairness by increasing the chances of admittance for wideband VCs.

A routing scheme based on the packing concept was proposed in [8]. The scheme attempts to pack class-s VCs in order to reduce blocking for only the next higher class of VCs. In [9], 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.

This Research

In this paper, we investigate a scheme based on the probabilistic selection of routes, where probabilities are chosen to match the distribution of available resources on the candidate routes (i.e. resource availability profile) with the distribution of traffic demand of different classes (i.e. the load profile). We call this scheme *Load Profiling Routing* (LPR).

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. In this paper, we compare MLR, LPR and LLR assuming accurate feedback. We show that MLR and LPR are competitive and significantly outperform the traditional LLR. This indicates that LPR is a promising routing approach and would perform 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 [12].

The paper is organized as follows. Section 2 motivates load profiling by comparing it to load balancing and load packing. Section 3 presents simulation results to demonstrate the superiority of load profiling and the effectiveness of our proposed LPR routing strategy. We conclude in Section 4 with a summary and with directions for future work.

2 Balancing, Packing, and Profiling

In this section we show that for reservation-based traffic—which requires a preset fraction of a resource's bandwidth for an extended period of time—load balancing is not desirable. In particular, we propose a load-profiling strategy that controls the distribution of load amongst the various resources in the system in such a way so as to maximize the chances of finding resources that would satisfy the needs of *future* incoming requests.

Overview

Load balancing is often used to ensure that resources in a distributed system are equally loaded. In [14], 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[1] 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 scheme we will present below 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*.

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

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.

Figure 1 illustrates the advantage of load profiling when compared to load balancing. In particular, when a request with high capacity requirement is submitted to the system, the likelihood of accepting this request in a load-profiled system is higher than that in a load-balanced system.

We next motivate load profiling by comparing load packing and load balancing. We analytically show the optimality of load packing. 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. Finally we introduce our load profiling heuristic, denoted by LPR, as an attractive alternative to MLR.

MLR versus LLR: An Analytical Comparison

Consider a system with N different paths between a particular source and 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. That is $f(u)$ is the probability that the bandwidth requirement of a VC request will be u, where $0 \leq u \leq 1$. Furthermore, let W denote the overall load of the system, expressed as the sum of the reserved bandwidth over all paths (i.e. $N \geq W \geq 0$). 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/N. 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.

Let $\mathcal{C}$ denote the set of N 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 $\mathcal{F}$.

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:

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

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. $L_{\mathcal{C}}(u) = \int_0^u l_{\mathcal{C}}(u)du$. 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:

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

Thus, the probability that a VC request will be accepted is given by $P = \int_{(1-W/N)}^1 f(u)\ 1\ du$.

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 $\int_0^1 u\ l_{\mathcal{C}}(u)du = (1 - W/N)$. One solution to this optimization problem is for $l_{\mathcal{C}}(u)$ to be chosen as $l_{\mathcal{C}}(u) = (W/N).\delta_u(0) + (1-W/N).\delta_u(1)$ 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/N of the paths in the system are 100% utilized, and thus have *no* extra bandwidth to spare, whereas a fraction $(1-W/N)$ 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:

$$L_{\mathcal{F}}(u) = \begin{cases} 0 & \text{if } 0 \leq u < 1 \\ 1 & \text{if } u = 1 \end{cases} \quad (3)$$

Plugging these values into equation (1), we get $P = \int_0^1 f(u)(1 - 0)du = 1$, which is obviously optimal.

The *perfect fit* implied in equation (3) 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,

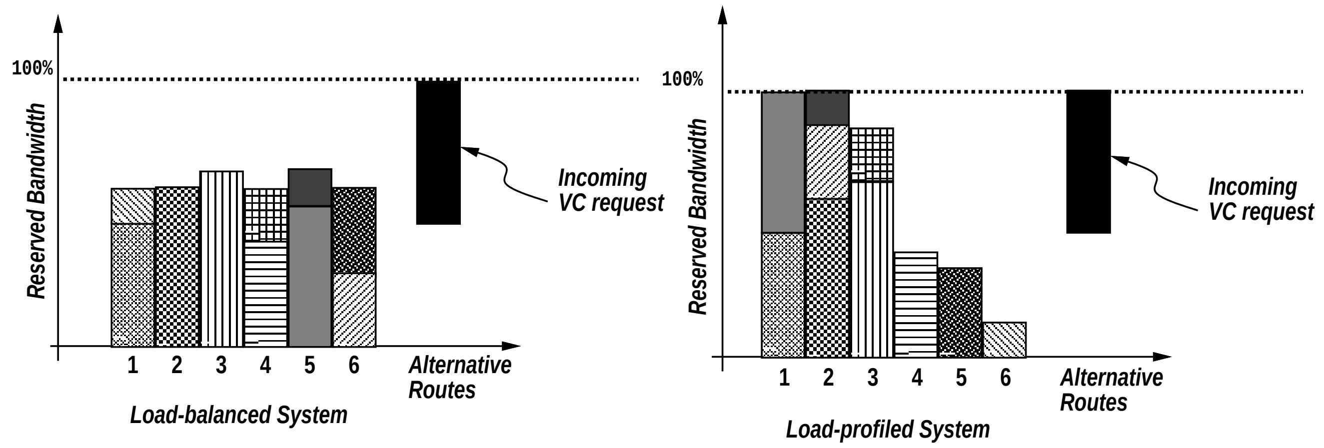


Figure 1. Load-Packing/Profiling (MLR/LPR) versus Load-Balancing (LLR): An illustration.

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 [11]. However, for a small value of N—which is likely to be the case in network routing problems—best-fit outperforms first-fit.

MLR versus LLR: Simulation Experiments

To quantify the benefits of load packing versus load balancing, we performed a number of simulation experiments to compare the acceptance rate of VC requests under two load distribution strategies. The first is a *load-balancing* strategy, whereby a requested VC is assigned to the least loaded route (LLR) out of all the routes capable of satisfying the bandwidth requirement of that VC. If none exist, then the VC request is deemed inadmissible. The second is a *load-packing* strategy, whereby a VC request is assigned to the most loaded route (MLR) (i.e. the route that provides the best fit) out of all routes capable of satisfying the bandwidth requirement of that VC. If none exist, then the VC request is deemed inadmissible.

In our simulations, VC requests were continually generated until the overall reserved bandwidth across all routes in the system (W) reaches a certain level. Two experiments were conducted. In the first, 5 routes were available between the source and destination, whereas in the second 10 routes were available. In both experiments, all routes were identical in terms of their capacity (total bandwidth).

Subsequent VC requests were assumed to be identically and independently distributed. In particular, VC requests were generated so as to request bandwidth uniformly from the range $[0, 1]$, where 1 indicates 100% of the total bandwidth available on a single route. Once a VC is accepted, it is assumed to hold its reserved bandwidth indefinitely. For each one of the load distribution strategies, the percentage of the VC requests successfully admitted is computed. We call this metric the *VC Admission Ratio*.

Figure 2 shows example results from our simulations. These results suggest that as the reserved bandwidth across all paths increases, the performance of both LLR (load balancing) and MLR (load packing) degrades as evidenced by the lower admission ratio. However, the degradation for LLR starts much earlier than for MLR. This is to be expected, since the bandwidth availability profile in a load-balanced system is not as diverse as that in a load-packed system. Figure 2 also shows that the advantage of using MLR is more pronounced when the number of alternative paths is small (i.e. 5 routes versus 10 routes).

MLR versus LPR

First-fit and best-fit heuristics work well when accurate information about the available bandwidth on all N 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 (3) 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 could be explained by noting that the best-fit heuristic is the

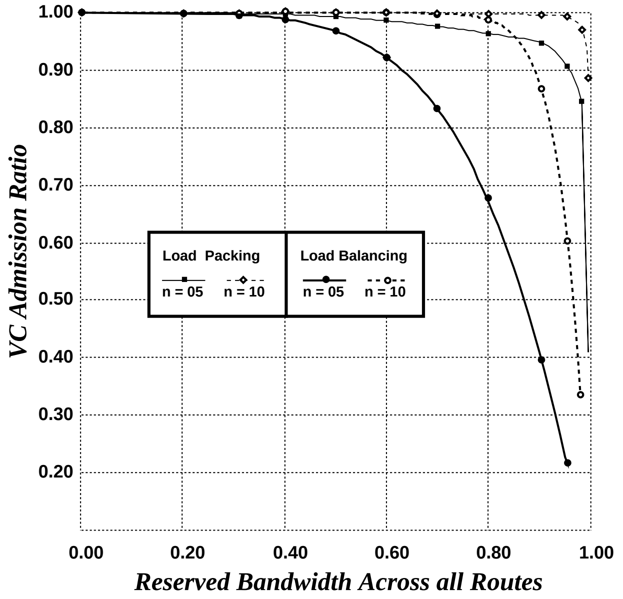


Figure 2. Load-Packing (MLR) versus Load-Balancing (LLR): Simulation results.

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 next 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 bandwidth availability profile of the system is maintained as close as possible to the expected profile of incoming VC bandwidth requests. This process is illustrated in the next section.

3 Performance Evaluation

In this section, we compare MLR, LPR and LLR in terms of how well they distribute VCs from multiple classes. A simulator was written in C to study the behavior and performance of the algorithms. MLR selects the feasible path (i.e. a path with enough bandwidth to support the incoming VC) with the *least* available bandwidth. LLR selects the feasible path with the *most* available bandwidth. LPR selects a feasible path *probabilistically* by matching the expected load profile of incoming VC requests and the bandwidth availability profile of the system as exemplified below.

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. 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).

Illustration of the LPR Scheme

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 3 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.

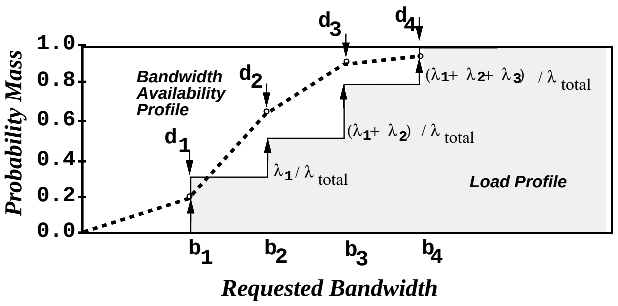


Figure 3. Example load profile and bandwidth availability profile.

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.

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.

Let d_i $(i = 1, 2, 3, 4)$ be the differences between the load profile and the bandwidth availability profile (see Figure 3). We now assign a weight to each path according to the smallest route set it belongs to as shown in Table 1.[2] 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:

$$W(p,k) = \sum_{i=k}^{S} (d_i - d_{min}) \quad (4)$$

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.

Figures 4 and 6 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 available bandwidth on each path is 20, 25, 30, 35 and 40.

Figures 5, 7 and 8 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 available bandwidth on each path is 20, 25, 30, 35, 40, 45, 50, 55, 60 and 65.

The following observations can be made from the results in Figures 4 through 8.

- In terms of total number of accepted VCs, MLR and LPR are competitive and they both significantly outperform LLR. For equal class arrival rates and 5 candidate paths, MLR outperforms LLR by about 45%, whereas LPR outperforms LLR by about 22%. With 10 candidate paths, MLR outperforms LLR by about 42%, whereas LPR outperforms LLR by about 44%. Consistent with results in Section 2, the advantage of using MLR is more pronounced with a smaller number of candidate paths. 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.

- 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 7(a) and 8(a), where $\lambda_4 = 0.25$ and 0.1, respectively.

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

4 Conclusion and Future Work

We presented a novel approach to routing real-time virtual circuits in multi-class networks. The approach is based on the 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 QoS requests (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]. We found LPR competitive to the asymptotically optimal most-loaded-based routing (MLR) assuming accurate feedback information. Furthermore, LPR is less sensitive to inaccuracy in the feedback information that is inherent in a distributed network system because of its probabilistic selection of routes.

A recent study of LPR routing in virtual path based networks can be found in [10]. LPR is found to provide higher revenue and to improve fairness among VC classes when compared to LLR. 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".

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 - IC-CCN '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(12):1–27, 1997.

[5] 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.

[6] 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.

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

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

[9] I. Matta and M. Krunz. Packing and Least-Loaded Based Routing in Multi-Rate Loss Networks. In Proc. *IEEE ICC*, pages 132-139, 1997.

[10] I. Matta and A. Bestavros. Evaluation of a Load Profiling Approach to Routing Guaranteed Bandwidth Flows. Technical Report NU-CCS-97-10, BU-CS-97-013 (available from http://www.ccs.neu.edu/home/matta/).

[11] 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.

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

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

[14] Songnian 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.

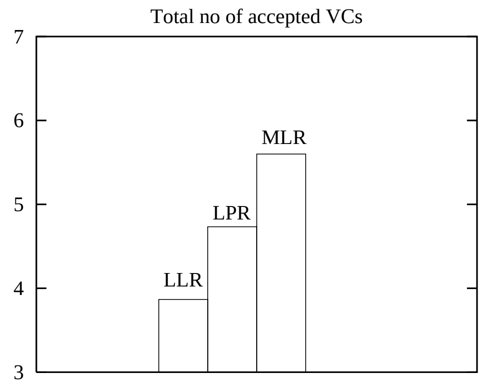


Figure 4. Total number of accepted VCs until first VC blocking occurs for the 5-path simulation experiments with equal class arrival rates.

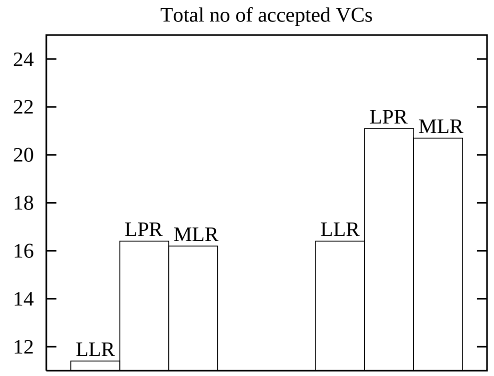
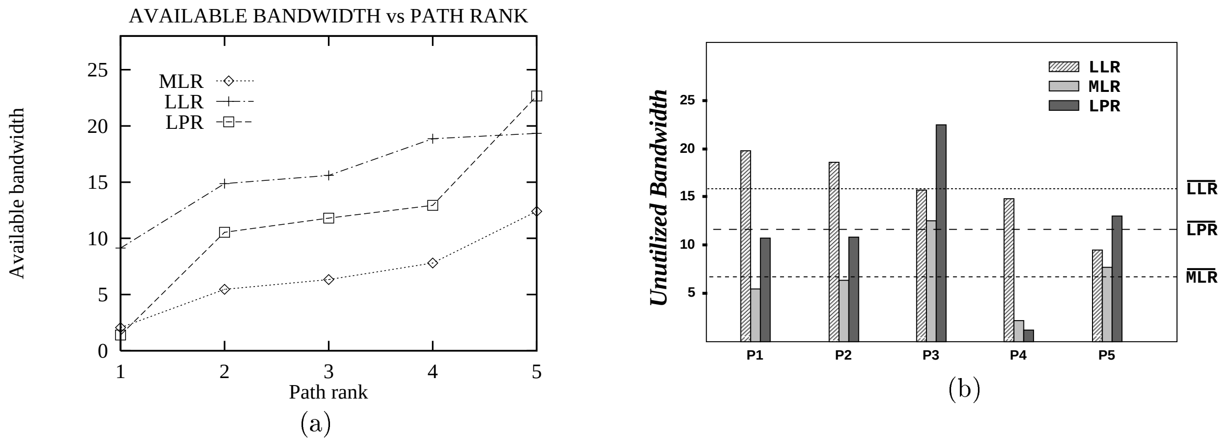
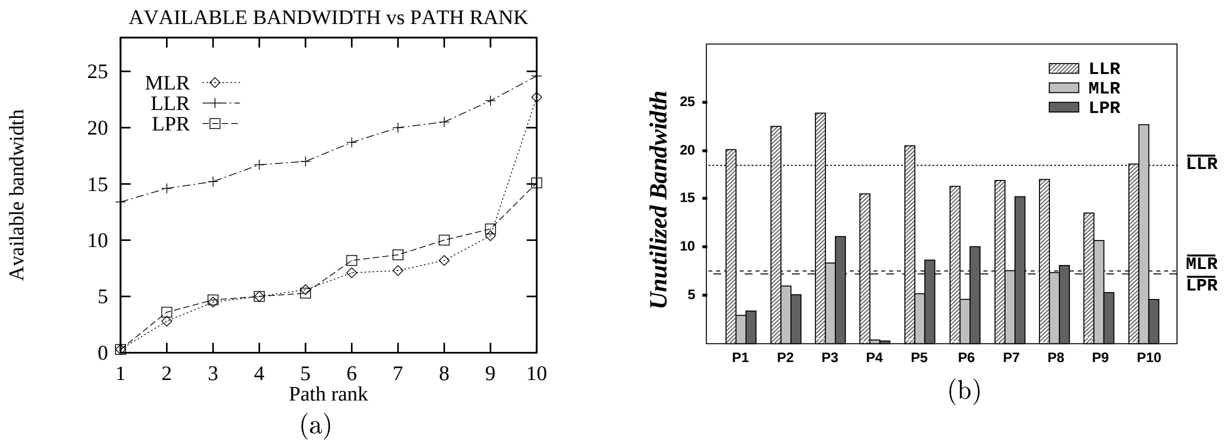


Figure 5. Total number of accepted VCs until first VC blocking occurs for the 10-path simulation experiments with equal class arrival rates (left) and unequal class arrival rates (right).



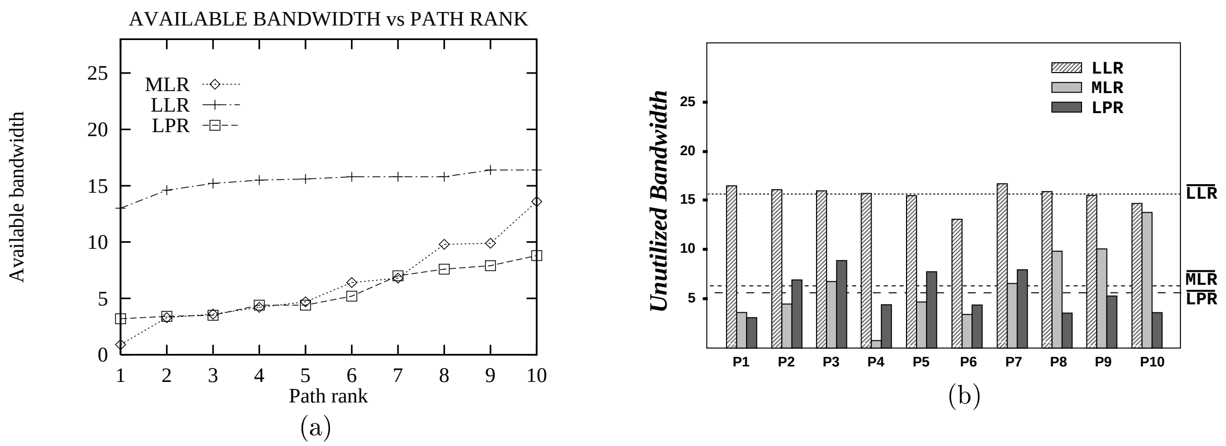
(a) (b)

Figure 6. 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 7. 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 8. 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.