On Path Selection for Traffic with Bandwidth Guarantees

Qingming Ma Peter Steenkiste
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213, USA
{qma,prs}@cs.cmu.edu

Abstract

Transmission of multimedia streams imposes a minimum-bandwidth requirement on the path being used to ensure end-to-end Quality-of-Service (QoS) guarantees. While any shortest-path algorithm can be used to select a feasible path, additional constraints that limit resource consumption and balance the network load are needed to achieve efficient resource utilization. We present a systematic evaluation of four routing algorithms that offer different tradeoffs between limiting the path hop count and balancing the network load. Our evaluation considers not only the call blocking rate but also the fairness to requests for different bandwidths, robustness to inaccurate routing information, and sensitivity to the routing information update frequency. It evaluates not only the performance of these algorithms for the sessions with bandwidth guarantees, but also their impact on the lower priority best-effort sessions. Our results show that a routing algorithm that gives preference to limiting the hop count performs better when the network load is heavy, while an algorithm that gives preference to balancing the network load performs slightly better when the network load is light. We also show that the performance of using pre-computed paths with a few discrete bandwidth requests is comparable to that of computing paths on-demand, which implies the feasibility of class-based routing. We observe that the routing information update interval can be set reasonably large to reduce routing overhead without sacrificing the overall performance, although an increased number of sessions can be misrouted.

1 Introduction

New network service models are being proposed to support multimedia applications. These new services, e.g., the controlled-load service by the IETF [22] and the available-bit-rate (ABR) with minimal cell rate (MCR) by ATM Forum [1], impose a minimum-bandwidth requirement on the path being used, ensuring an end-to-end Quality-of-Service (QoS) guarantee. A path is *feasible* if the unserved bandwidth of all links on the path is higher than the requested bandwidth. The goal of QoS routing is to select a feasible path if one exists, and to select one leading to better overall resource efficiency if more than one path is available.

While a feasible path can be selected using any shortest-path algorithms, additional optimality constraints need to be imposed to achieve efficient resource utilization. The most common way for a routing algorithm to achieve resource efficiency is to limit resource consumption and to keep the network load balanced. For traffic with bandwidth guarantees, resource consumption can be reduced by restricting the hop count of the path being selected, while the network load can be balanced by selecting the least loaded path. However, these two "optimality" constraints can conflict and routing algorithms with different optimality criteria can be obtained by attaching different weights to the two constraints. Understanding the performance tradeoffs between these routing algorithms essential to the successful deployment of QoS in future networks.

Several path selection algorithms have been proposed in the literature, including widest-shortest path [9], shortest-widest path [21], and utilization-based path selection algorithms [15]. However, a systematic evaluation of these algorithms is missing. In this paper, we consider four routing algorithms based on the following "optimality" criteria: minimum hop count, maximum residual bandwidth, minimum path cost based on link utilization, and a variant of the dynamic-alternative path used in telecommunication networks. Our evaluation considers not only the call blocking rate but also the fairness to requests for different bandwidths, robustness to inaccurate routing information, and sensitivity to the routing information update frequency. Finally, the choice of routing algorithm for traffic with bandwidth guarantees can have a significant impact on the performance of lower priority best-effort sessions. To best of our knowledge, our study is the first to evaluate the performance impact of the routing of algorithm used for guaranteed sessions on best-effort traffic.

Our results show that, when the network load is heavy, routing algorithms that give preference to limiting resource consumption perform better than others giving preference to balancing the network load. On the other hand, when the network is light, they perform slightly worse. We also show that routing update interval can be set reasonably large without increasing blocking rate, although an increased number of sessions can be misrouted. As a result, routing information can be distributed less frequently, reducing routing overhead. The performance of using pre-computed paths with discrete bandwidth requests is comparable to that of computing paths on-demand, which implies the feasibility of class-based routing. Finally, we show the performance impact of these routing algorithms on best-effort traffic.

The rest of the paper is organized as follows. After providing some background in Section 2, we present four routing algorithms in Section 3. Section 4 summarizes our methodology. The next three sections examine the performance of different routing approaches: dynamic on-demand routing in Section 5, static routing in Section 6, and class-based routing in Section 7. The performance impact on best-effort traffic is examined in Section 8. We discuss related work in Section 9 and conclude in Section 10

0-8186-8061-X/97 $10.00 © 1997 IEEE

2 Background

2.1 Service Models

We study a network with two classes of traffic: sessions requiring bandwidth guarantees, or guaranteed sessions, and best-effort sessions. Our focus is on routing algorithms for guaranteed sessions. For best-effort sessions we use the results in [15].

For guaranteed sessions, we assume that before communication starts, the traffic source specifies its traffic characteristics and desired performance guarantees. The network performs routing and admission control to decide if the traffic can be admitted. The admission control can be either a measurement-based mechanism [12] or based on some analytic model, e.g., equivalent capacity [7]. Once a flow is admitted, resources along its path are reserved through a reservation protocol, e.g., RSVP [23].

2.2 Route Computation

What routing information is used and **when** routes are computed have a direct impact on routing overhead.

We study both static routing and dynamic load-sensitive routing. With *static routing*, a path is computed using only static network information such as the topology and link bandwidth, regardless of the network load. Paths are fixed unless more than one path has the same "distance". In such cases, one is randomly selected. The admission control algorithm determines if a path is feasible, which depends on the dynamic network load at that particular moment.

With *dynamic routing*, link load information is advertised periodically and a different path may be selected for a given source and destination pair at different times. For simplicity, we assume that link-state routing is used for dynamic routing [19], although the path selection algorithms discussed below can also be implemented as distance-vector routing algorithms.

Paths can either be selected on-demand or they can be precomputed. With on-demand routing, the path selection algorithm is executed for every request. With precomputed paths, the path selection algorithm is executed periodically as new routing information is received. The precomputed paths are stored in the routers, and routing requests result in a simple table look up. In practice, source routing [3] often selects path on demand, while hop-by-hop routing uses precomputed paths.

We will evaluate both the dimensions of static versus dynamic routing, and of on-demand routing versus using precomputed paths. Our evaluation assume that routing information is updated periodically, as is the case in existing routing protocols such as OSPF and PNNI. As a result, we incorporate in our evaluation the impact of inaccurate routing information.

3 Routing Algorithms

In this section, we discuss routing algorithms for traffic requiring bandwidth guarantees. The goal of the routing algorithm is to find a feasible path if one exists, and to select one that achieves efficient resource utilization if more than one path is available.

3.1 Selecting Feasible Paths

Given a path $\mathbf{p} = \{i_1, \ldots, i_k\}$, the maximal reservable bandwidth (**mrb**) on the path **p** is the minimum of the reservable bandwidth of all links on the path:

$$\mathbf{mrbp} = \min\{R_{i_j} | i_j \in \mathbf{p}\}.$$

The path **p** is feasible if the $\mathbf{mrbp}$ is no less than the requested bandwidth **b**: $\mathbf{mrbp} \geq \mathbf{b}$.

To select a feasible path, either Dijkstra's shortest-path algorithm or the Bellman-Ford shortest-path algorithm (see [4]) can be used. For example, we can define the cost of a link i_j as R_{i_j}—the link residual bandwidth—and the cost of a path **p** the $\mathbf{mrbp}$. Using either shortest-path algorithm, a path with the maximum **mrb** can be selected. If the **mrb** is no less than the request bandwidth **b**, a feasible path is found. An alternative is to prune all links whose residual bandwidth is less than the request bandwidth and then to select a shortest-path in the remaining graph using any cost function.

3.2 Selecting Efficient Paths

It is often the case that several feasible paths are available, raising the question of what path uses resources most efficiently.

3.2.1 Alternative Paths

Routing algorithms can achieve resource efficiency by limiting resource consumption while balancing the network load. For traffic with bandwidth guarantees, since the requested bandwidth has to be reserved, a natural way to limit resource consumption is to select a path with as few hops as possible. The network load can be balanced by selecting the least loaded path. However, these two "optimality" criteria sometimes conflict, for example a path with the fewest hops may contain heavily loaded links.

Several path selection algorithms that put different weight on limiting hop count and on balancing the network load have been proposed in the literature. They include the widest-shortest path [9], the shortest-widest path [21], and a utilization-based shortest path [15]. In the literature on telecommunication routing, where the network is fully connected, it has been shown [6] that dynamic alternative path routing combined with trunk reservation performs well. Since data networks are rarely fully connected, we introduce a simple variant of dynamic alternative path routing that does not use trunk reservation. Based on this earlier work, we selected the following four candidate paths for evaluation:

- **Widest-shortest path**: a path with the minimum hop count among all feasible paths. If there are several such paths, the one with the maximum reservable bandwidth is selected. If there are several such paths with the same bandwidth, one is randomly selected.
- **Shortest-widest path**: a path with the maximum bandwidth among all feasible paths. If there are several such paths, the one with the minimum hop count is selected. If there are several such paths with the same hop count, one is randomly selected.
- **Shortest-distance path**: a feasible path with the shortest distance. The distance function is defined by

$$\text{dist}(\mathbf{p}) = \sum_{j=1}^{k} \frac{1}{R_{i_j}}$$

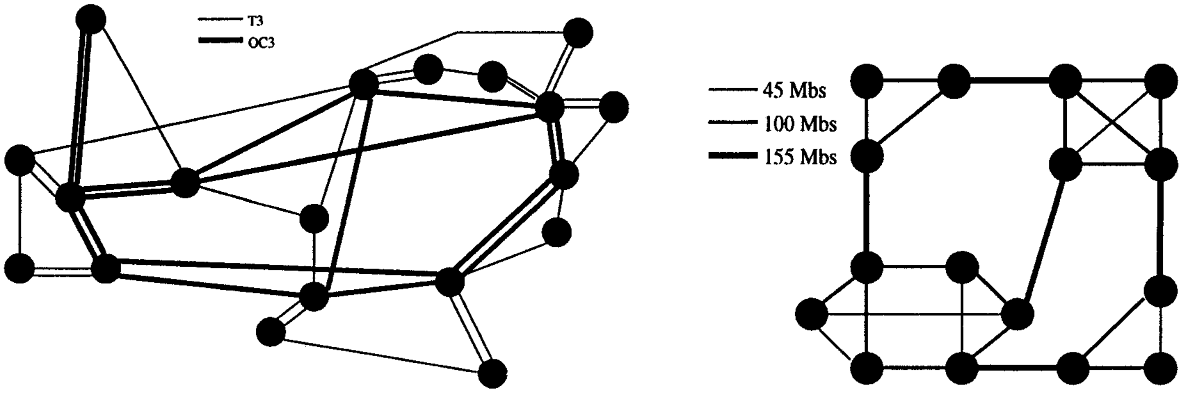


Figure 1: Topologies used in simulation: the MCI topology (left) and a switched cluster topology (right)

where R_{i_j} is the bandwidth available on link i_j. It has been shown that this algorithm performs consistently well when routing best-effort sessions [15].

- **Dynamic-alternative path**: Let n be the hop count of a minimum-hop path when the network is idle. A dynamic-alternative path is a widest-shortest path with no more than $n+1$ hops.

The widest-shortest path gives high priority to limiting the hop count, while the shortest-widest path gives high priority to balancing the network load. The shortest-distance path uses the distance function to dynamically balance the impact of the hop count and the path load. For dynamic-alternative path, we do not consider trunk-reservation, since the minimal-hop path may not be unique and the performance may be sensitive to the trunk-reservation rate. Compared with the shortest-widest path, the dynamic-alternative path puts a upper bound ($n+1$) on the widest-shortest path.

3.2.2 Algorithms

A widest-shortest path algorithm based on the Bellman-Ford algorithm is proposed in [9]. We can also modify Dijkstra's algorithm to find the widest-shortest paths by defining two distance functions: the hop count is the primary function and the **mbr** is the secondary function. When selecting the next node to mark, one selects the node with the minimal hop count. If several nodes have the same minimal hop count, one selects the node with the largest **mbr**. The algorithm terminates when the destination is reached.

A simple shortest-widest path algorithm applies Dijkstra's algorithm twice. First find a widest path; assume its bandwidth is B. Then find a shortest path with bandwidth B by using Dijkstra's algorithm on a network that only includes links of bandwidth B or higher. Note that the single-pass link-state shortest path algorithm given in [21] does not always find the shortest-widest path. For example for the topology in Figure 2, the algorithm will select the lower path in the graph using links with bandwidth 4. The reason is that, when a link (e.g., the link connected to the node D) with low bandwidth has to be added to the path, the earlier shortest-widest segment may no longer be the shortest-widest one.

The shortest-distance path can be selected by any shortest-path algorithm using the distance function as the cost function. The dynamic-alternative path can be selected by using the widest-shortest path algorithm while imposing hop count restriction on the nodes being selected.

Each of these algorithms may select a path that is not feasible, either because of stale routing information or because of changes in the network state while the connection is being established. If that happens, the request is rejected. Similarly, it is possible that the algorithms do not find a feasible path, although one exists.

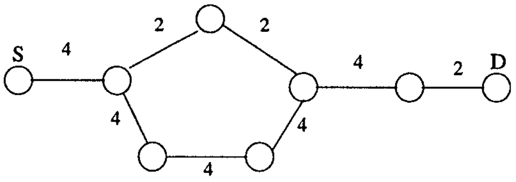


Figure 2: Finding a path from S to D with bandwidth 2

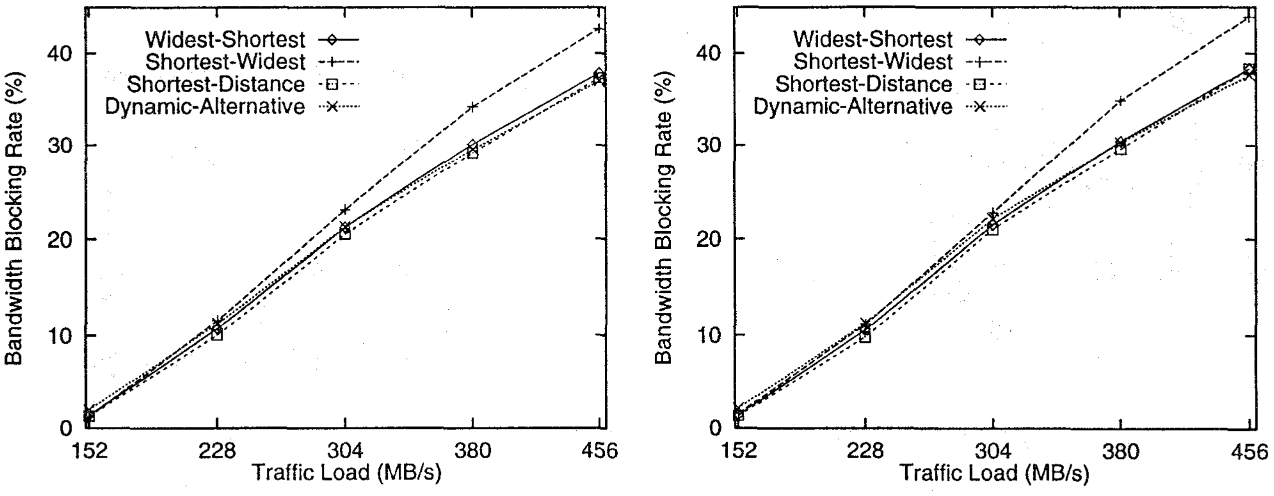
4 Simulation Design

Our evaluation is based on an event-driven simulator that consists of two levels. At the session-level, it selects routes and does admission control and resource reservation. At the packet level, it does connection setup and tear-down and routing information distribution. We use two topologies (see Figure 1): one is the MCI Internet topology and the other is a switch-based cluster topology. For the MCI topology, the propagation delay between two nodes is calculated from their physical distance, and for the cluster topology, we use a fixed propagation delay of 20 microseconds for every link.

4.1 Traffic Load

Our traffic load consists of two classes of traffic: guaranteed sessions and best-effort sessions. The total traffic load is split between the two traffic classes according to a predetermined session ratio. Sessions arrive according to a Poisson distribution with the mean determined by the overall network load.

For guaranteed sessions, we distinguish between audio sessions and video sessions. We assume that the requested bandwidth is uniformly distributed from 16 to 64 kilobits/second for audio sessions, and from 1 to 5 megabits/second for video sessions. Most simulation studies of real-time traffic use an exponential call holding time distribution, while others assumed that all incoming calls last forever [18]. Recent studies [2] show that the call holding time distribution for conversations, facsimile, and voice mail connections have a large portion of very short calls and a lognormal long-tail distribution. We use a long-tail distribution following the model



(a) 60% voice and 40% video sessions

(b) 100% video sessions

Figure 3: Bandwidth blocking rate as a function of network load: MCI topology, evenly distributed load, and path selection on demand

suggested in [2]: most of our simulations use a holding time distribution that is a mixture of two normal distributions (F_1 and F_2) on a logarithmic time scale with the mixing probability α (0.25 used for video sessions and 0.4 for voice sessions in our simulation).

$$F(x) = \alpha \cdot F_1(x) + (1 - \alpha) \cdot F_2(x)$$

The average mean holding time used in our simulation is 20 minutes for video sessions and 260 seconds for voice sessions, except the cases when we study the impact of call holding time distributions. For the purpose of comparison, we also consider exponentially distributed call holding times.

In [15] we considered two types of best-effort traffic: high-bandwidth sessions and low-latency sessions. Since the low-latency sessions use more or less fixed paths and require a small amount of bandwidth, their performance is fairly insensitive to how higher priority sessions are routed, assuming sufficient link capacity is set aside for these sessions. Thus, we consider only high-bandwidth sessions, which typically transfer large amounts of data and consume as much bandwidth as is available. We assume that the traffic source specifies the total amount of bytes to be sent. The link capacity that is not reserved by guaranteed sessions is shared among all high-bandwidth sessions following the max-min fair share principle. This sharing model is not only representative of the ATM ABR service, but under some conditions it also approximates IP networks. For example, in [11] it is shown that Round-Robin scheduling combined with window-based congestion control supports max-min fairness.

We consider traffic loads that are either evenly or unevenly distributed. For evenly distributed loads, a new session selects with equal probability any pair of switches as its source and destination. For unevenly distributed loads, a percentage of sessions use a preselected set of source and destination pairs and the rest of the sessions randomly pick any pair of switches as the source and destination.

4.2 Performance Metrics

The two main performance metrics used in our evaluation of routing algorithms for traffic with bandwidth guarantees are the blocking rate and the routing inaccuracy.

A guarantees session can be rejected either because no path with sufficient resources can be found by the routing algorithm or because the elected path does not have sufficient resources. The latter can be caused either by the delay of hop-by-hop connection setup or by stale routing information. The *call blocking rate* is defined as the percentage of rejected sessions over all arrivals. However, when sessions can request different amounts of bandwidth, a low call blocking rate does not necessary reflect high efficiency. Thus, we introduce the *bandwidth blocking rate*, which takes into account the bandwidth of rejected sessions:

$$\text{bandwidth blocking rate} = \frac{\sum_{i \in \text{BG\_blk}} \text{bandwidth}(i)}{\sum_{i \in \text{BG}} \text{bandwidth}(i)}$$

where BG_blk is the set of blocked sessions and BG is the set of requests for guaranteed sessions. We also evaluate the fairness of the routing algorithm: how are sessions requesting different amounts of bandwidth treated?

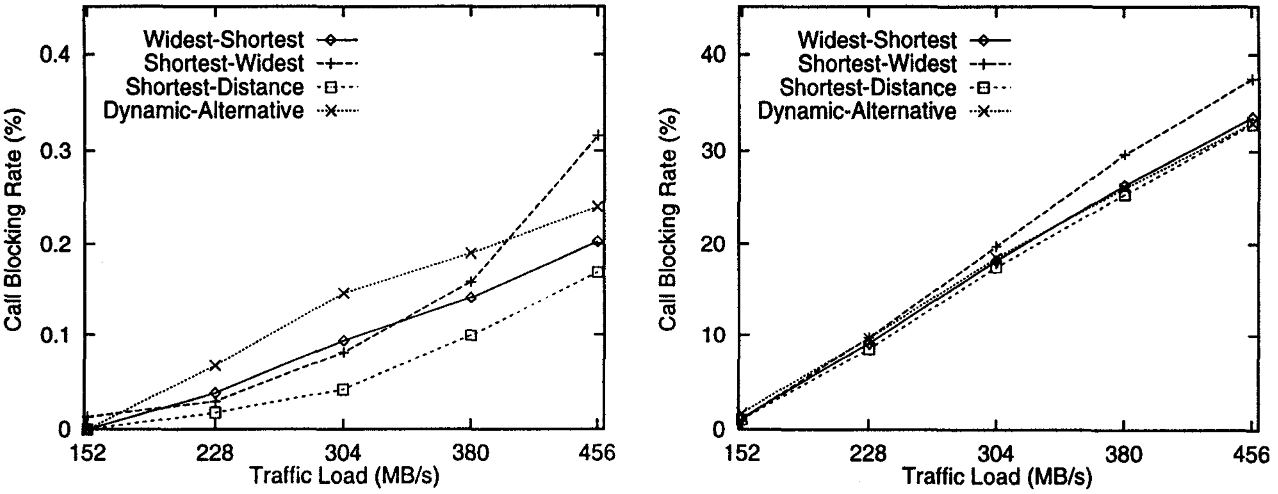
As a result of inaccurate routing information and connection setup delays, routing algorithms can generate an incorrect result: either a path with insufficient bandwidth is selected or no feasible path can be found although one exists. To capture this aspect of routing algorithm performance, we define the *routing inaccuracy* metric:

$$\text{routing inaccuracy} = \frac{\text{number of incorrect route selections}}{\text{total number of sessions requests}}$$

Finally, using different routing algorithms for guaranteed sessions can have a significant impact on the performance of lower priority best-effort sessions. For example, when the blocking rate is close to 0, the preferred routing algorithm for the guaranteed sessions will allow the best effort sessions to achieve high throughput. In our earlier study [15] on routing algorithms for best-effort traffic, we used the average throughput of all best-effort sessions as the performance index:

$$\text{Average throughput} = \frac{\sum_{i \in \text{BE}} \text{bytes}(i)}{\sum_{i \in \text{BE}} \text{time}(i)}.$$

where BE is the set of all best-effort sessions.



(a) Audio sessions

(b) Video sessions

Figure 4: Blocking rate as a function of network load: MCI Topology, 60% voice sessions, evenly distributed load, and path selection on demand

4.3 Routing Information

In addition to static topology information, the link state information includes the dynamic residual bandwidth and the max-min share rate. Routing protocols, such as OSPF, use a reliable flooding mechanism to periodically exchange routing information. We implemented a simple version of flooding in our simulator: each node updates its link state (bandwidth) at regular time intervals (default value of 30 seconds) and sends it to all its neighbors. When receiving a routing information update, a node updates its database and forwards the information to all its neighbors, except the one from which the update was received. Duplicate routing updates are discarded.

5 Dynamic On-demand Routing

In this section, we expand our preliminary investigation in [14] and examine the performance of the four routing algorithms described in Section 3. We assume that guaranteed sessions are the only traffic class in the network. Paths are selected on-demand using dynamic load information, which is updated asynchronously every 30 seconds.

5.1 Blocking Rate

We examine the blocking rate of the four routing algorithms for different loads, topologies, and call holding time distributions.

5.1.1 Evenly Distributed Load

Figure 3 shows the bandwidth blocking rate as a function of the traffic load. Traffic is evenly distributed and results are shown for 60% audio and 40% video sessions (left), and for 100% video sessions (right). Overall, we see that the bandwidth blocking rate is linearly proportional to the network load. When the load is heavy, the shortest-widest path performs poorly because it tends to allocate long expensive paths, which penalizes later arrivals. As the network load decreases, the performance difference between the four algorithms becomes smaller; the shortest-distance path performs slightly better and the shortest-widest path slightly worse than the others. When the network load is very light, all four algorithms have similar performance; the shortest-widest path now performs best and the shortest-distance path is a close second. The following table lists the bandwidth blocking rate (in %) with the same simulation configuration as in Figure 3 when the the network load is 114 MB/s.

	WS	SW	SD	DA
40% video sessions	0.2388	0.1665	0.1836	0.6860
100% video sessions	0.1505	0.0604	0.0678	0.4806

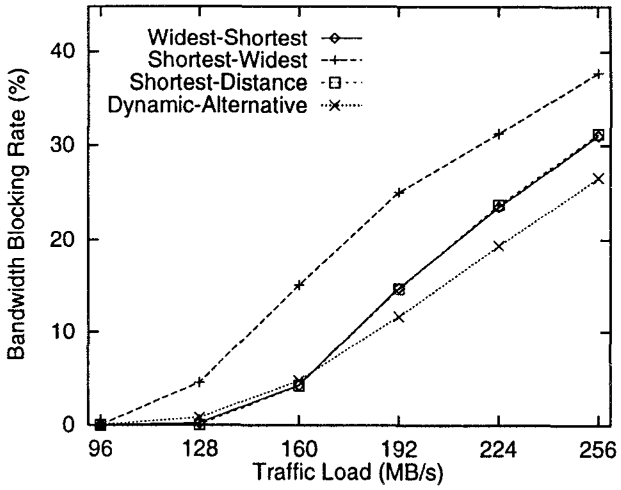
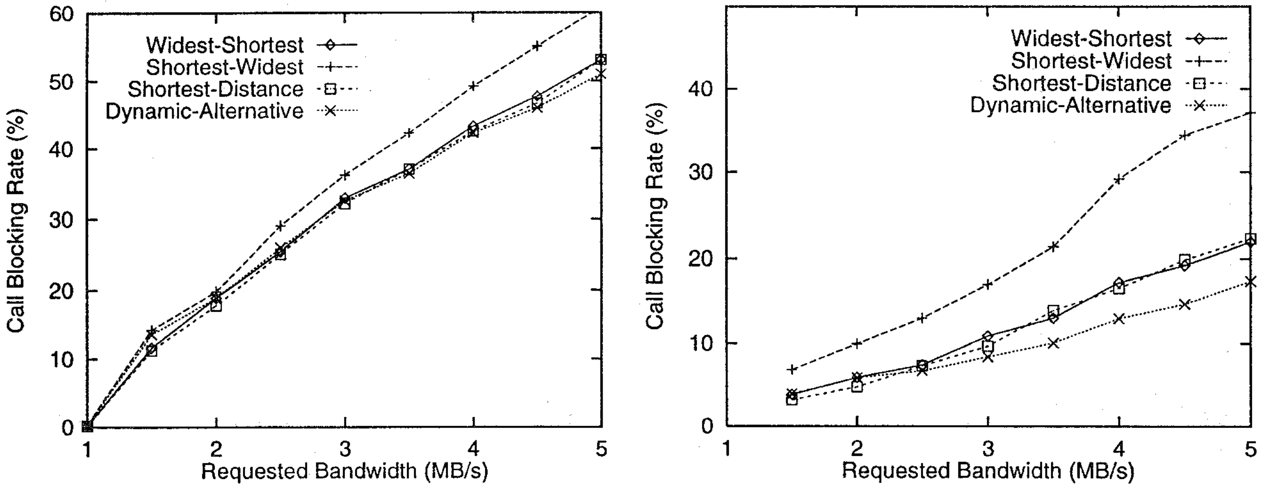


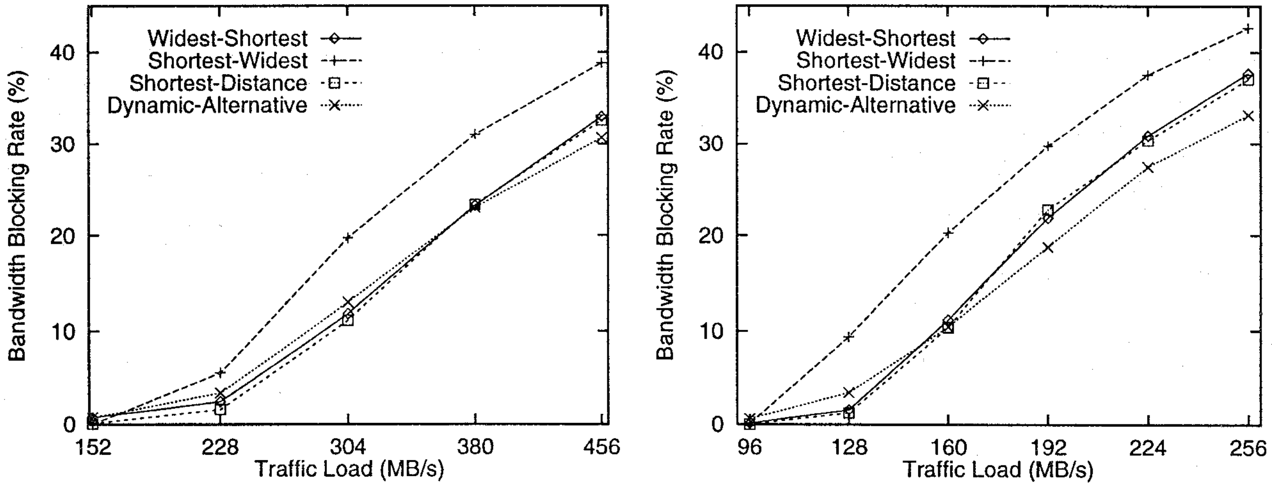
Figure 5: Bandwidth blocking rate as a function of network load: cluster topology, 100% video sessions, Evenly distributed load, and path selection on demand

This result is different from the result obtained for best effort traffic in [15], where we found that the impact of the routing algorithm on performance was often significant. This difference is caused by the different sharing rules in the two traffic classes. For best effort traffic, link capacity is shared among all sessions, and all paths are feasible, even those that are heavily congested compared with other parts of the network. Compared with the widest-shortest



(a) MCI topology, 40% video sessions, and 456MB/s (b) Cluster topology, 100% video sessions, and 192MB/s

Figure 6: Call blocking rate as a function of requested bandwidth: evenly distributed load and path selection on demand



(a) MCI topology (b) Cluster topology

Figure 7: Bandwidth blocking rate as a function of network load: 100% video sessions, unevenly distributed load, and path selection on demand

path, which is strictly a minimum-hop path, the shortest distance path is able to route around congested links. For traffic with bandwidth guarantees, the heavily congested links are no longer feasible and all algorithms will route around them. For example, a widest-shortest path is not necessarily a minimum-hop path, but the shortest one among all feasible paths.

In Figure 4, we show the call blocking rate for audio and video traffic separately. We see that audio traffic has a much lower blocking rate (under 0.4%) than video traffic, as one would expect given its lower bandwidth requirements.

Figure 5 show the bandwidth blocking rate as a function of traffic load for the cluster topology. We observe that the shortest-widest path performs much worse than the other three, and the dynamic alternative path performs much better when the load is heavy. This suggests that with a more symmetric topology, restricting resource consumption becomes more important when the network load is heavy. The dynamic-alternative path does not consider "expensive" paths that are two or more hops longer than the minimal hop path, i.e. it rejects requests that would require a relatively expensive path, favoring later "cheaper" requests. The shortest-distance path performs slightly better than the dynamic-alternative path when the load is light. The reason is that the restriction on the hop count of a path limits the degree to which the algorithms can route around congested links.

In Figure 6 we show the call blocking rate as a function of the requested bandwidth; all requests for less than 1 MB/s (i.e. audio) are combined in a single data point (MCI topology). We see that the call blocking rate is almost a linearly function of requested bandwidth, confirming that the algorithms favor sessions that ask for less bandwidth.

5.1.2 Unevenly Distributed Load

Figure 7 shows bandwidth blocking rates for an uneven load distribution. Most traffic is between the west coast and east coast in the MCI topology, and between the left corner building and other buildings in the cluster topology. For both topologies, our earlier observations hold: the shortest-widest path performs worse than the

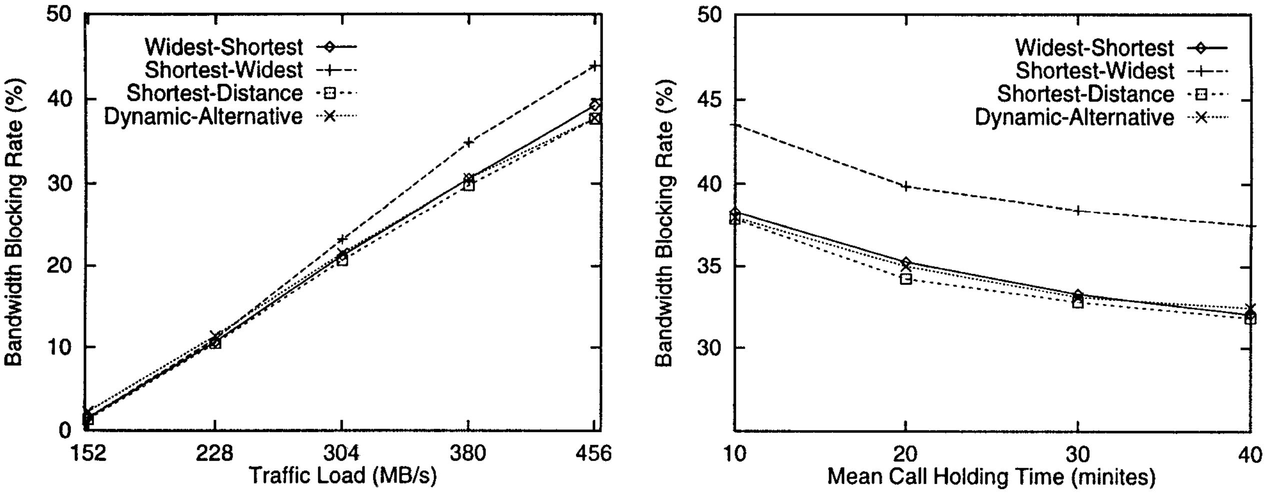
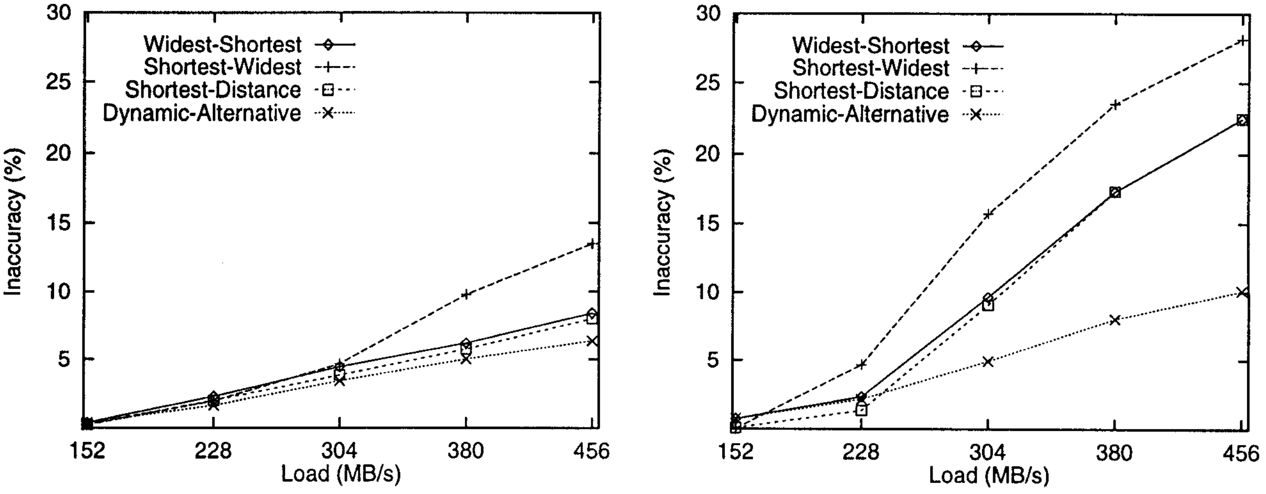


Figure 8: Bandwidth blocking rate as a function of network load (left) and mean call holding time (right, 380MB/s): MCI topology, 60% voice sessions, evenly distributed load, exponential call holding time, and path selection on demand



Evenly distributed load

Unevely distributed load

Figure 9: Routing inaccuracy as a function of network load: MCI topology, and 100% video sessions

other three algorithms, the dynamic-alternative path performs better when the load is heavy, and the shortest distance path performs better than dynamic alternative path when the load is light.

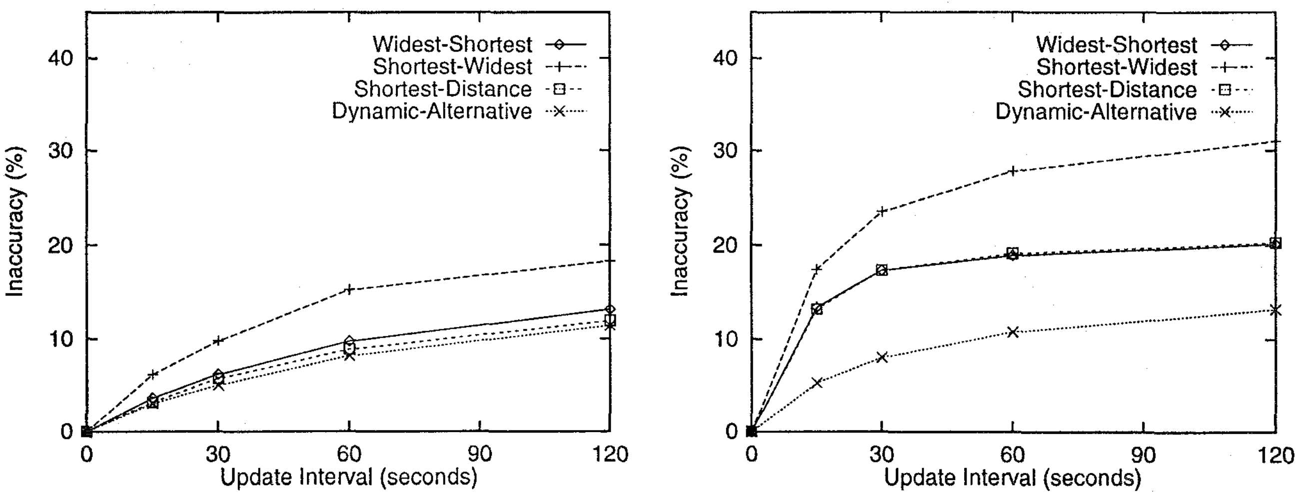
5.1.3 Impact of Call Holding Time Distribution

Figure 8 (left) shows the bandwidth blocking rate for the same configuration used in Figure 3, but using an exponential call holding time distribution with the same mean as the long tail distribution used in Figure 3. We observe a slightly higher bandwidth blocking rate when using an exponential call holding time distribution. The reason is that with a long-tail distribution there are more sessions with a short call holding time. These shorter sessions can more easily use the bandwidth left unused by long sessions, minimizing the impact of a poor routing decision.

Figure 8(right) shows the call blocking rate as a function of the mean call holding time. We see that the bandwidth blocking rate is higher for shorter mean call holding times. The reason is that, with the same traffic load but lower mean call holding times, more sessions arrive during a routing update interval, thus increasing the degree of inaccuracy in the routing information which results in more rejected sessions.

5.2 Routing Inaccuracy

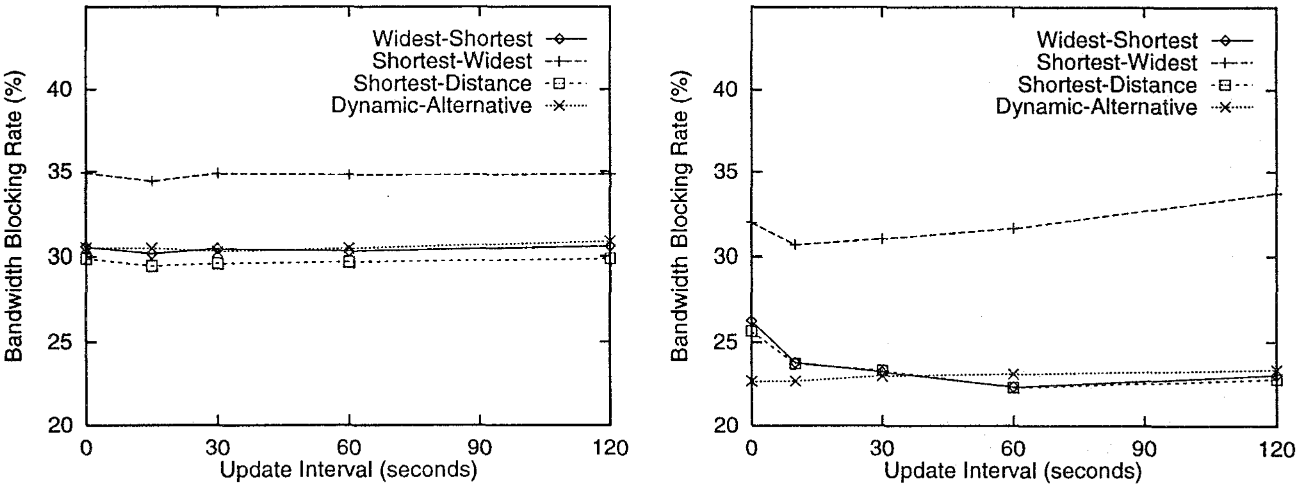
In Figure 9, we show the routing inaccuracy as a function of the network load for both evenly and unevely distributed traffic load. We observe that routing inaccuracy increases with the network load for all algorithms, and the shortest-widest path is most sensitive to inaccurate routing information. The widest-shortest path and the shortest-distance path have similar performance with the shortest-distance path doing slightly better. The dynamic-alternative path algorithm is the most robust one when the load is heavy but it loses ground to the shortest-distance path algorithm when the load is uneven and light. We also see a significant difference in routing inaccuracy between evenly and unevenly distributed load. The reason is that when the traffic is concentrated (unevenly distributed), during a routing update interval more sessions will be routed to the area in the network where traffic is concentrated, which results in a faster rate of changes in the network state and more incorrectly



Evenly distributed load

Unevely distributed load

Figure 10: Inaccuracy as a function of routing update time: MCI topology, 100% video sessions, and 380MB/s



Evenly distributed load

Unevenly distributed load

Figure 11: Bandwidth blocking rate as a function of routing information update interval: MCI topology, 100% Video sessions, and 380MB/s

routed sessions. We can see this more clearly by comparing the routing inaccuracy metric (Figure 9) with the blocking rate (Figure 3 (right) and 7 (left)). We observe that for evenly distributed traffic most of the blocked sessions are rejected by the routing algorithm. For uneven traffic, except for the dynamic-alternative path, blocked sessions have typically been routed, but they are rejected by the admissions control module on one of the switches.

The routing inaccuracy metric covers both sessions that are routed but later rejected and sessions that are not routed when a paths does exist. Our simulations show that the sessions that are routed but are later rejected dominate routing inaccuracy. Among the sessions for which no paths are found, 6-10% percent for even load and 10-40% for uneven load would have a found a path if accurate routing information had been available.

Figure 10 explores the sensitivity of the routing accuracy to the routing update interval, which was 30 seconds in the earlier simulations. We see that the routing inaccuracy increases with the update interval. However, the pace of increasing slows down above a threshold of about 30 seconds. The shortest-widest path is most sensitive and the dynamic-alternative path is least sensitive to increases in the routing update interval.

Figure 11 shows the bandwidth blocking rate as a function of routing information update interval. It is interesting to see that, while the routing inaccuracy increases with the routing update interval, the bandwidth blocking rate remains quite stable. In some cases, the bandwidth blocking rate is even slightly higher when the routing information is more accurate. The reason is that, with more accurate information, the network allocates resource more "conservatively" in the sense that it discourages sessions from trying if there is no path available. With less accurate information, it can allocate resources: even though no path can be found at routing time if accurate routing information is used, sessions can successfully set up a connection, using resources that are being relinquished by connections that are terminating.

We conclude that increasing the routing information update interval does not affect much the overall blocking rate, even though more sessions are routed but rejected by the admission control algorithm or are not routed while a path exists.

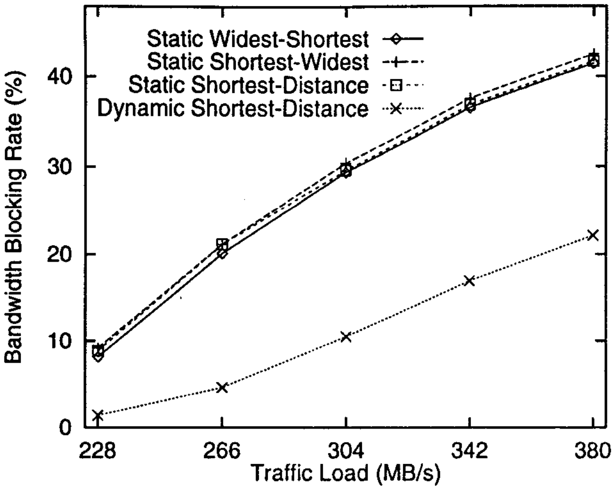


Figure 12: Bandwidth blocking rate as a function of network load: MCI topology, 100% video sessions, and unevenly distributed load

6 Static Routing

Dynamic routing can be expensive both in terms of operational costs and implementation complexity. Static routing uses static link capacity as link-state. Our simulation results show that in a network with evenly distributed traffic load, the performance difference between static and dynamic routing is very small. However, when the load is unevenly distributed, the performance difference between dynamic and static routing can be significant (see Figure 12). The reason for the significant difference is that loaded links cannot be avoided with static routing, and selecting a loaded path leads to a session being rejected.

7 Class-based Routing

An alternative to on-demand dynamic routing is to use pre-computed paths: each router updates its paths regularly when new link-state information is received. Since the requested bandwidths can be very diverse, using a path that meets all bandwidth requests results in using the widest path, which does not always perform well, as we have shown above. An alternative is to use class-based routing: several paths, each for a different bandwidth range, are precomputed. A new session selects the path with the lowest bandwidth satisfying the request.

Figure 13 considers three ways of selecting pre-computed paths and compares them with on-demand path selection; we consider widest-shortest path and shortest-distance path which had the best overall performance. The "type 3" algorithm uses three classes serving bandwidth requests falling in the ranges (0, 1], (1, 3], and (3, 5]. The "type 2" algorithm uses two classes, serving audio and video traffic, and the "type 1" algorithm uses only one class. We observe that the efficiency increases with the number of classes. However, the difference is not very high, which implies that class-based routing is feasible. We also see that the use of class-based routing has less effect on the widest-shortest path algorithm, while the shortest-distance path gives up the small performance advantage it had over the widest-shortest path when paths are selected on demand.

Figure 14 shows the call blocking rate of class-based routing and on-demand routing as a function of requested bandwidth. Since only video sessions are considered, there is no difference between the "type 1" and "type 2" schemes. The data point for a bandwidth x represents the call blocking rate for sessions with bandwidth requests in the range $(x - 0.5, x]$. We see that for all algorithms, the call blocking rate increases slowly with the requested bandwidth. We also see that class-based routing treats sessions with different bandwidth demands more evenly than on-demand routing. The reason is that while using pre-computed paths may be less efficient, it reduces bandwidth fragmentation, which benefits larger requests. We also notice that with the "type 3" algorithm, there is a jump in the call blocking rate between the two classes. The reason is that while the bandwidth demands are uniformly distributed, the pre-computed path for the high-bandwidth class can accommodate fewer sessions, increasing the blocking rate. Moreover, the high-bandwidth path is likely to have more hops, increasing the risk of interference from other sessions.

8 Performance Impact on Best-effort Sessions

In a network with multiple classes of traffic, what paths are used for high priority sessions significantly affects the performance of lower priority traffic. In this section, we consider two classes of traffic: guaranteed sessions and best-effort sessions. Best-effort traffic uses resources left unused by the guaranteed sessions. The routing algorithm used for best effort sessions is the shortest-distance path algorithm with the distance for a path defined as ([15])

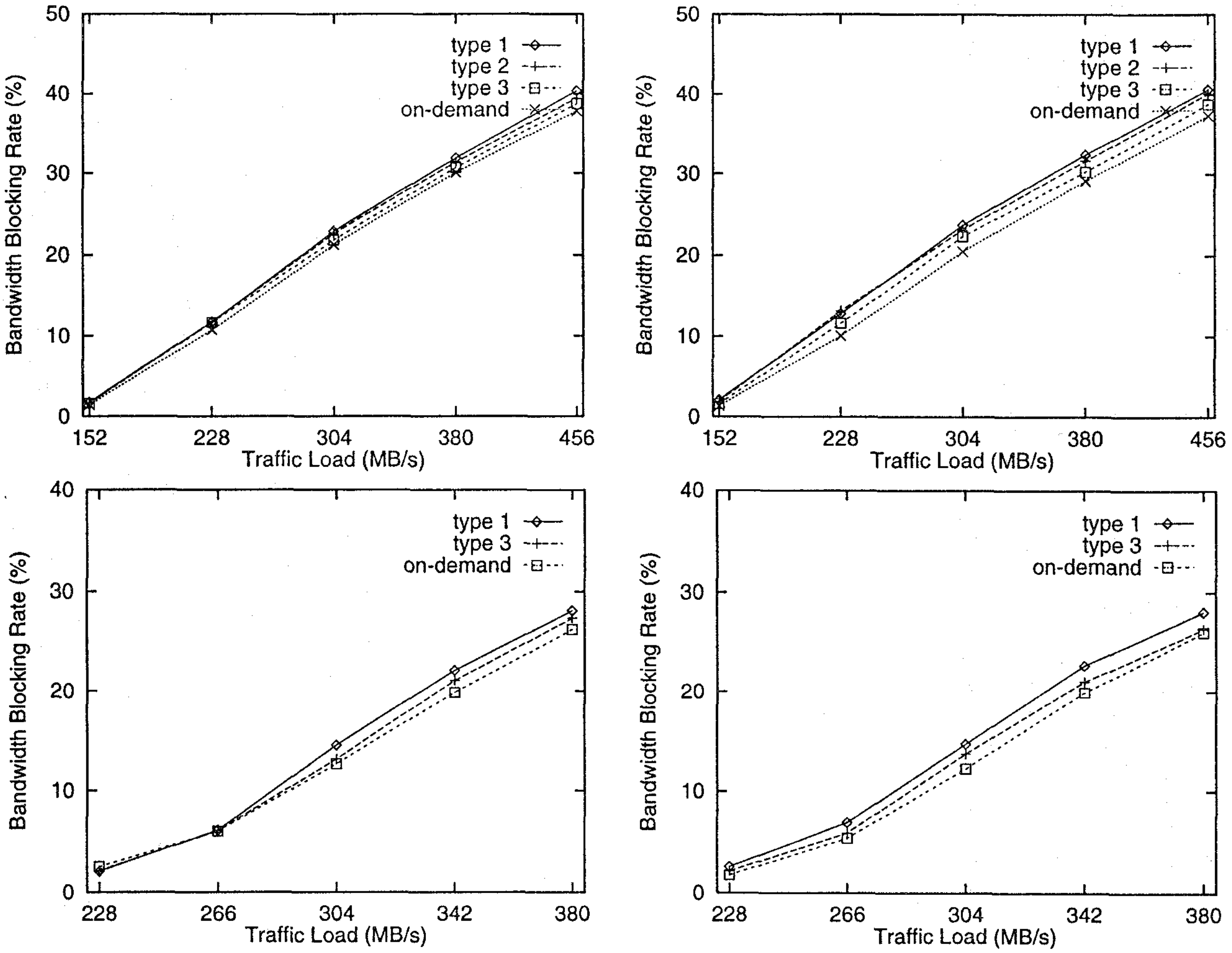
$$\text{dist}(p) = \sum_{i=1}^{k} \frac{1}{r_i}$$

where r_i is the max-min rate of link i for a new best-effort session.

Figure 15 shows the average throughput of all best-effort sessions as a function of the total network load. 60% of traffic is generated by best-effort sessions and we assume that up to 90% of the capacity of each link can be reserved by guaranteed sessions. We observe that the impact of the choice of routing algorithm for guaranteed sessions on the best-effort sessions is small when the network load is very light; the reason is that there are abundant resources. For higher network loads, we start seeing a more significant impact. The widest-shortest path and the dynamic-alternative path have very similar performance because they tend to pick the same path when the guaranteed traffic load is low, as is the case here. The shortest-distance path performs consistly better than the widest-shortest path and the dynamic-alternative path. While the improvement is usually small, it is significant in some cases, e.g., for uneven load in the cluster topology.

The fourth algorithm, the shortest-widest path, has very uneven performance. For the MCI topology, it performs extremely well. The reason is that by selecting the "widest" path, it spreads the load over the network and avoids links that already have little bandwidth left for best-effort sessions. This is especially clear in the case of an uneven traffic load. For the cluster topology, the shortest-widest path performs poorly. The reason is that for this more symmetric topology, the disadvantage of the shortest-widest path (the use of more expensive paths for guaranteed sessions) outweighs the potential advantage of distributing the load.

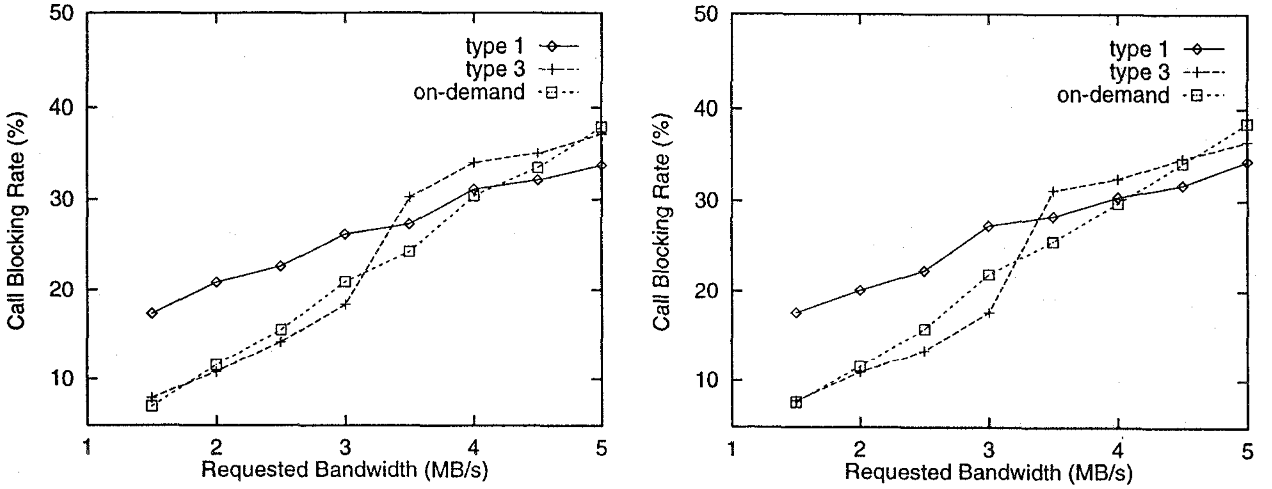
We conclude that the use of the shortest-distance path algorithm for guaranteed sessions results in good performance for best-effort sessions. While the shortest-widest path algorithm sometimes gives better performance, its performance is very uneven across different topologies and network loads. More work is needed to better un-



(a) Widest-shortest path

(b) Shortest-distance

Figure 13: Bandwidth blocking rate as a function of network load: MCI Topology, 60% voice sessions, evenly distributed load (top); 100% video sessions, unevenly distributed load (bottom)



(a) Widest-shortest path

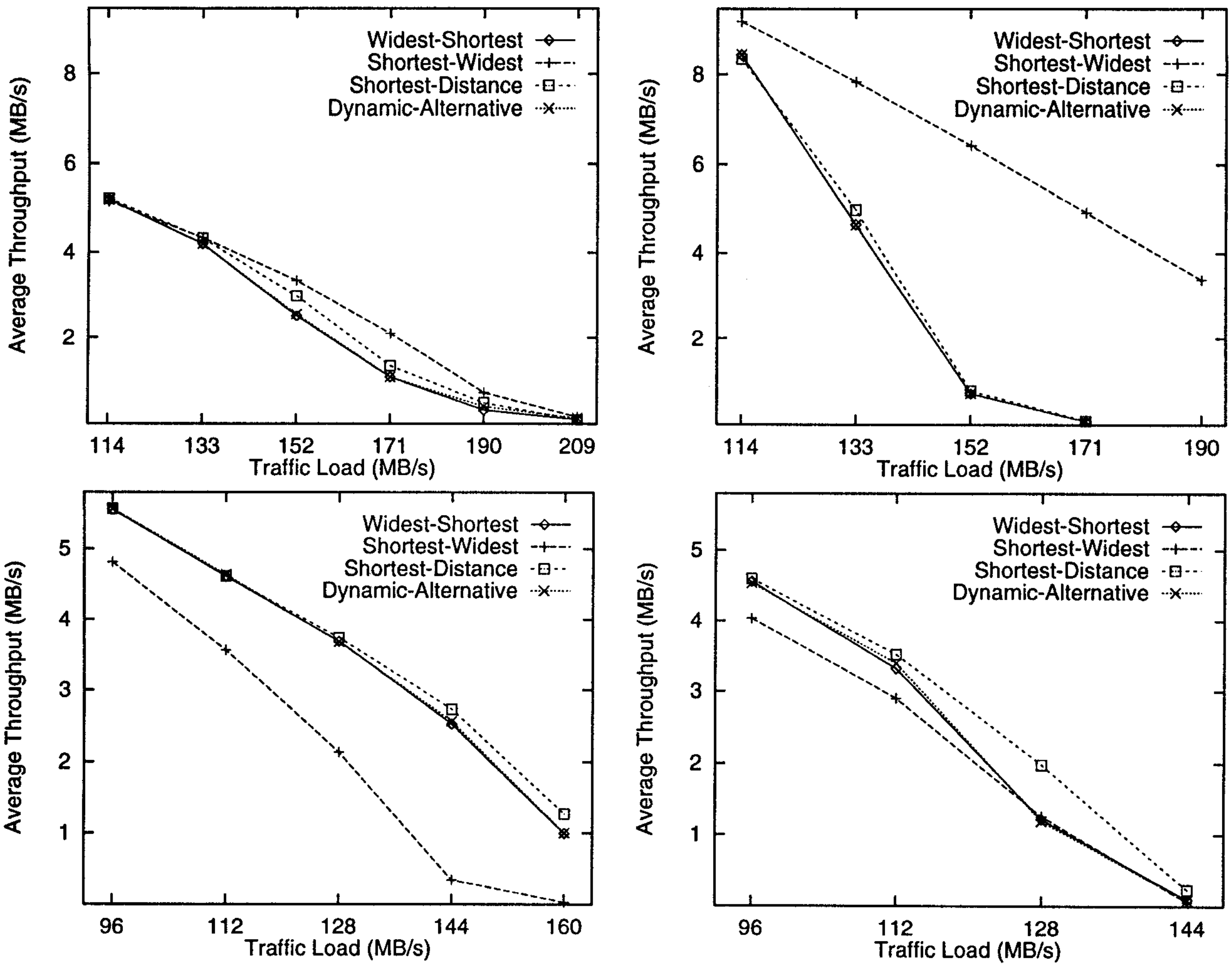
(b) Shortest-distance path

Figure 14: Call blocking rate as a function of requested bandwidth: MCI Topology, 100% video sessions, unevenly distributed load (380MB/s)

derstand the impact of the routing algorithm for high priority traffic on the performance of best-effort traffic.

9 Related Work

Many papers in the literature have studied QoS routing and path selection algorithms. We list the results that are most relevant



Evenly distributed load

Unevenly distributed load

Figure 15: Average throughput as a function of network load: MCI Topology (top), cluster topology (bottom), 60% high-bandwidth best-effort traffic, and 40% video traffic

to our study. A good introduction to QoS routing and routing in general can be found in [13, 20]. QoS routing in telecommunication networks has been an active area of research for a long time. Trunk Reservation and Dynamic Alternative Routing [6], and Least Load Routing [10] are some of the algorithms that have been studied.

In [21], Wang and Crowcroft study the complexity of QoS routing with multiple QoS constraints. They propose the shortest-widest path algorithm as a way of minimizing the call blocking rate, but no performance evaluation is given. In [3], an adaptive load-based source routing algorithm for traffic requiring bandwidth guarantees is suggested by Breslau, Estrin, and Zhang. Its performance is compared with the static minimal-hop path and the alternative path algorithms. In [5], Gawlick, Kalmanek, and Ramakrishnan present an evaluation study of several routing algorithms, including minimal-hop path, exponential path, and max-min path, for permanent connections with bandwidth guarantees. In his thesis [18], S. Rampal evaluates the performance of several path selection algorithms, including static minimal-hop path, dynamic shortest distance paths based on link utilization and residual bandwidth, and shortest delay path. The study assumes that the session holding time is infinite. Matt and Shankar study delay and throughput based type-of-service routing [17] and dynamic routing of real-time virtual circuits [16].

Guerin, Orda, and Williams [9] propose to use the widest-shortest path to extend OSPF for QoS routing. In another paper [8], Guerin and Orda study routing with inaccurate information and show that, using a shortest path algorithm, it can find the feasible path that is most likely to accommodate the requested bandwidth.

Compared with these studies, we evaluate a wider range of algorithms under more realistic traffic loads: audio and video traffic, long-tail distribution for the call holding times, and even and uneven traffic load distributions. We also consider the effect of routing information propagation and the use of pre-computed paths on performance.

10 Conclusions

This paper presents a simulation study of QoS routing for traffic requiring bandwidth guarantees. While selecting a feasible path, i.e. a path that meets the bandwidth requirement, can be done using any shortest path algorithm, selecting paths that also optimize overall network performance by utilizing resources efficiently has not been well understood. The two main methods to achieving resource efficiency are limiting resource consumption and balancing the network load. Our evaluation considers four routing algorithms that attach different weight to each of these criteria: widest-shortest path, shortest-widest path, shortest-distance path, and dynamic-alternative path. To compare these algorithms for traffic with diverse bandwidth requirements, we introduced the bandwidth blocking rate as our main performance metric.

Our results show that for dynamic on-demand routing, limiting the hop count (e.g., dynamic alternative path) gives better performance when the network load is heavy, while putting more emphasis on balancing the laod (e.g., shortest-distance path and shortest-widest path) pays off when the load is light. This result is different from what we observed for best-effort traffic [15], where the shortest-distance path algorithm has a clear performance edge over the other algorithms. The reason for this is the difference in sharing policy in the two traffic classes. Heavily loaded links become automatically ineligible for guaranteed sessions, causing any dynamic algorithm to route around them, but they remain eligible for best effort traffic so the algorithm has to explicitly avoid them. Our results also show that algorithms that minimize the hop count result in a more even blocking rate across sessions with diverse bandwidth requirements. When comparing static and dynamic routing, we observed that using dynamic information can reduce the blocking rate significantly in cases of unevenly distributed load, because dynamic information makes it possible to route around (infeasible) bottleneck links. We also examined a class-based routing algorithm and found that its performance is comparable to that of computing paths on-demand.

To evaluate the impact of inaccurate routing information and connection setup delay, we introduced a routing inaccuracy metric. We showed that the overall blocking rate is fairly insensitive to the increase in routing information update interval although more sessions are misrouted. Algorithms that minimize the hop count result in more robustness to inaccurate routing information and increases in the routing information update interval.

For the performance impact on the lower priority best-effort sessions, we showed that the shortest-distance path consistently results in better performance for best-effort sessions than the widest-shortest path and the dynamic-alternative path, while the performance impact of the shortest-widest path is very inconsistent across network topologies and traffic load distributions.

References

[1] ATM Forum Traffic Management Specification Version 4.0, October 1995. ATM Forum/95-0013R8.

[2] V.A. Bolotin. Modeling Call Holding Time Distributions for CCS Network Design and Performance Analysis. *IEEE JSAC*, 12(3):433–438, April 1994.

[3] L. Breslau, D. Estrin, and L. Zhang. A Simulation Study of Adaptive Source Routing in Integrated Service Networks. *USC CSD Technical Report*, Sep., 1993.

[4] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorihtms*. The MIT Press, Cambridge, MA, 1990.

[5] Rainer Gawlick, Charles Kalmanek, and K.G. Ramakrishnan. On-line Routing for Permanent Virtual Circuits. In *Proc. IEEE INFOCOM'95*, 1995.

[6] R.J. Gibbens, F.P. Kelley, and P.B. Key. Dynamic Alternative Routing — Modelling and Behaviour. In *Proceedings of the 12th International Teletraffic Congress*, June 1988.

[7] R. Guerin, H. Ahmadi, and M. Naghshineh. Equivalent Capacity and Its Application to Bandwidth Allocation in High-Speed Networks. *IEEE JSAC*, 9(7):968–981, September 1991.

[8] R. Guerin and A. Orda. QoS-based Routing in Networks with Inaccurate Information: Theory and Algorithms. In *Proc. IEEE INFOCOM'97*, 1997.

[9] R. Guerin, A. Orda, and D. Williams. QoS Routing Mechanisms and OSPF Extensions. *IETF Internet Draft <draft-guerin-qos-routing-ospf-00.txt>*, November 1996.

[10] S. Gupta, K.W. Ross, and M. El Zarki. On Routing in ATM Networks. In M. Steenstrup, editor, *Routing in Communications Networks*, pages 49–74. Prentice Hall, 1995.

[11] E. Hahne. Round-Robin Scheduling for MaxMin Fairness in Data Networks. *IEEE Journal on Selected Areas in Communications*, 9(7), September 1991.

[12] S. Jamin, S. Shenker, and P. Danzig. Comparison of Measurement-based Admission Control Algorithms for Controlled-Load Service. In *Proceedings of IEEE INFOCOM'97*, Kobe, Japan, April 1997.

[13] W.C. Lee, M.G. Hluchyj, and P.A. Humblet. Routing Subject to Quality of Service Constraints in Integrated Communication Networks. *IEEE Network*, 9(4):14–16, July/August, 1995.

[14] Q. Ma and P. Steenkiste. Quality-of-Service Routing for Traffic with Performance Guarantees. In *IFIP Fifth International Workshop on Quality of Service*, pages 115–126, NY, NY, May 1997.

[15] Q. Ma, P. Steenkiste, and H. Zhang. Routing High-Bandwidth Traffic in Max-Min Fair Share Networks. In *ACM SIGCOMM96*, pages 206–217, Stanford, CA, August 1996.

[16] I. Matta and A. U. Shanka. Dynamic Routing of Real-Time Virtual Circuits. In *Proc. IEEE INCP'96*, 1996.

[17] I. Matta and A. U. Shankar. Type-of-Service Routing in Datagram Delivery Systems. *IEEE JSAC*, 13(8):1411–1425, October 1995.

[18] S. Rampal. *Routing and End-toend Quality of Service in Multimedia Networks*. PhD thesis, North Carolina State University, August, 1995.

[19] M. Steenstrup. Inter-Domain Policy Routing Protocol Specification: Version 1. *IETF Internet Draft*, May 1992.

[20] M. Steenstrup. *Routing In Communications Networks*. Prentice Hall, Englewood Cliffs, NJ 07362, 1995.

[21] Z. Wang and J. Crowcroft. Quality-of-Service Routing for Supporting Multimedia Applications. *IEEE JSAC*, 14(7):1288–1234, September 1996.

[22] J. Wroclawski. Specification of the Controlled-Load Network Element Service. *IETF Internet Draft <draft-ietf-intserv-ctrl-load-svc-04.txt>*, November 1996.

[23] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala. RSVP: A New Resource Reservation Protocol. *IEEE Communications Magazine*, 31(9):8–18, September 1993.