Supporting Dynamic Inter-Class Resource Sharing: A Multi-Class QoS Routing Algorithm

Qingming Ma
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134
qma@cisco.com

Peter Steenkiste
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213, USA
prs@cs.cmu.edu

Abstract

In an integrated services network, resources are shared by multiple traffic classes. Service classes that deliver Quality-of-Service (QoS) to applications have priority over others that do not. In such a multi-service network, routing decisions for high priority QoS traffic will affect what resources are available for lower priority traffic: Poor route selection can result in congestion for, or even starvation of, lower priority traffic. Whereas many studies have focused on routing algorithms that optimize the network throughput for individual service classes, little effort has been devoted to routing algorithms that address inter-class resource sharing. In this paper, we propose a routing algorithm that allows dynamic sharing of link resources among multiple traffic classes. The algorithm is based on the concept of "virtual residual bandwidth," which is derived from the link residual bandwidth by taking the congestion condition of low priority traffic into account. By using the virtual residual bandwidth in the link cost function for QoS sessions, we discourages QoS sessions from using links that are heavily loaded with low priority traffic. Our approach is simple in the sense that besides changing the link cost function, no other changes to the routing algorithms for individual service classes are required. An extensive simulation study shows that when the traffic load is unevenly distributed, significant performance improvements can be achieved for low priority traffic without sacrificing performance for high priority traffic. The result demonstrates that QoS routing is important even when the QoS traffic load is light and call blocking is not an issue.

1 Introduction

The increased capacity of packet-switched networks makes it feasible to support new applications such as video conferencing, Internet telephony, and distance learning. In the mean time, the volume of traditional best-effort data traffic continues to grow with Web applications and I/O intensive scientific applications. To meet these diverse application needs, the Internet is evolving from a network that provides a single best effort service to a network that supports multiple classes of services. Tremendous effort has been devoted to defining new service models that deliver better end-to-end Quality-of-Service (QoS) than current best-effort service can deliver. Example service models include *Guaranteed Service* [6, 27], *Controlled Load Service* [29], and *Differentiated Services* [8, 24].

To successfully support multiple classes of service, network resources must be managed effectively to ensure end-to-end QoS while at the same time sustaining high network throughput. New resource management mechanisms being proposed include scheduling disciplines, resource reservation protocols, admission control, and QoS routing. QoS routing refers to the routing mechanisms that achieve two goals [9, 16, 14, 10, 11, 18]: selecting feasible paths that meet QoS constraints and, if more than one feasible path is available, selecting an efficient path that achieves high network throughput. In addition to routing protocols that manage routing information (e.g. link state or distance vector), routing algorithms that select feasible as well as efficient paths are key components of a QoS routing architecture. In this paper we focus on QoS routing algorithms for networks that support multiple classes of traffic.

Existing routing studies [3, 30, 7, 4, 28, 23, 19, 22, 21, 26] have primarily focused on improving the network throughput of individual service classes while assuming the service class being studied is the sole service class in the network. For example, studies for routing traffic with QoS guarantees focus on minimizing call blocking rate, while studies for routing best effort traffic focus on avoiding congested links and improving per-session throughput. In contrast, routing in a network that supports multiple classes of service has received little attention. Routing in a multi-class network is not as simple as selecting for each flow an "optimal" path, assuming that the traffic class of this flow is the only service class in the network. Such an "optimal" path can worsen the congestion condition of flows in other traffic classes. For example, in a network that supports both QoS traffic requiring bandwidth guarantees and best-effort traffic, which resources are available to best-effort traffic depends on how QoS flows are routed. QoS flows can consume all bandwidth on certain links, thus creating congestion for, or even starvation of, best effort sessions. Statically partitioning the link resources can result in low network throughput if the traffic mix changes over time. Thus, a mechanism that dynamically distributes link resources across traffic classes based on the current load conditions in each traffic class is critical for performance.

A common belief is that QoS routing is only important when the load of QoS traffic exceeds what the network can accommodate. When the QoS traffic load is light, call blocking is not a concern, and the choice of routing algorithm for QoS traffic in general has little impact on network throughput. This suggests that in an integrated services network where best-effort traffic dominates, network performance may not be influenced much by QoS routing. However, we have observed in our work [19, 21] that this is not correct. For example, we have found that using the short-widest path algorithm for QoS traffic can lower the throughput of best effort traffic significantly compared with other algorithms, even when the call blocking was zero in all cases. This initial result indicates that the choice of routing algorithm for QoS traffic does have an impact on the overall network throughput.

In this paper, we further demonstrate that QoS routing can have a significant impact on overall network throughput even when the QoS traffic load is light. We design and evaluate a multi-class QoS routing algorithm that supports efficient inter-class resource sharing. The algorithm discourages QoS (high priority) traffic from using links that are already loaded with best effort (low priority) traffic. This is achieved by replacing the residual link bandwidth in the link cost for high priority traffic by a "virtual residual bandwidth," which is derived from the link residual bandwidth by taking into account of the congestion conditions for low priority traffic. The proposed mechanism does not require any changes to the path selection algorithms employed for individual service classes: the only change is in the link cost. Through an extensive set of simulations, we show the effectiveness and robustness of the mechanism. The distribution of network resources across traffic classes adapts dynamically and automatically to the mix of traffic in the traffic classes, even if the traffic distribution is uneven across the network. When the traffic load is concentrated, significant performance improvements are achieved for low priority sessions without penalizing high priority sessions. To the best of our knowledge, our study is the first that addresses inter-class resource sharing while routing traffic in multiple classes.

The rest of this paper is organized as follows. We begin with a brief discussion in Section 2 on why routing for an intergrated services network cannot be addressed by simply combining routing algorithms for each traffic class. We present our routing framework in Section 3. In Section 4 we describe our simulator design. We evaluate the performance of our inter-class resource sharing architecture in Sections 5 and 6. Related work is discussed in Section 7 and we summarize in Section 8.

2 Routing and Inter-Class Resource Sharing

This section motivates our study on inter-class resource sharing in multi-class routing and presents our approach.

2.1 Multi-Class Routing: Challenges

Let us consider a simple network that supports two traffic classes: bandwidth-guaranteed (or QoS) and best-effort traffic. In such a network, resources are shared by the two service classes: QoS sessions have priority over best-effort sessions. Given a fixed traffic load of QoS sessions, what resources are available to best effort traffic will be influenced by the routing for QoS sessions. If the routing algorithm for QoS sessions ignores the load or even the presence of lower priority traffic, as is commonly done, it optimizes the throughput of QoS traffic while ignoring the performance of best-effort sessions. Considering that best-effort traffic is likely to continue to be the dominant traffic class in the network in the near future, optimizing the throughput for QoS sessions while ignoring the performance of best effort traffic does not seem appropriate. Moreover, since for low to medium traffic loads, many routing algorithms for QoS traffic will have the same performance (a zero call blocking rate), the performance of best effort traffic will be the main performance metric.

Since the network is designed in accordance with a long-term average of the network load, the short-term local traffic load of the network can often be quite different. As a result, the amount of bandwidth left over for best effort sessions can be highly variable (Figure 1). For example, links close to a powerful Web server can get heavily loaded with best-effort traffic, but may have very little QoS traffic. A naive QoS routing algorithm would direct many QoS sessions to these congested links, which would worsen the congestion.

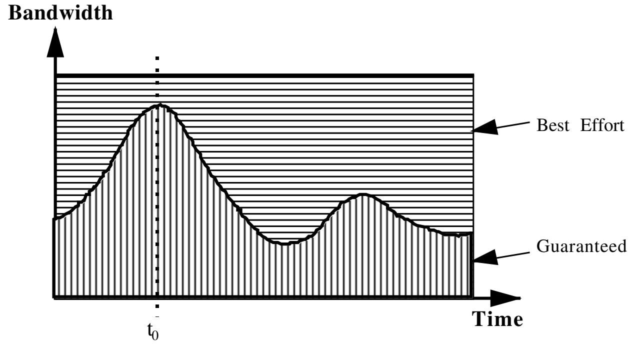


Figure 1: Link resources are shared across guaranteed (QoS) traffic and best effort traffic

Because both guaranteed traffic and best-effort traffic can be either evenly or unevenly distributed, there are four combinations of load distribution to consider (Figure 2). Of these four scenarios, the ones where the best-effort traffic is unevenly distributed are the most challenging ones. In these scenarios, routing decisions regarding QoS traffic can easily increase the level of congestion experienced by best effort traffic. On the other hand, the throughput for best effort traffic can be improved if QoS sessions can be routed around the best-effort hot spots.

2.2 Inter-Class Sharing: Alternative Approaches

There are several ways to route traffic in a multi-class network. The simplest approach is to combine the algorithms for individual traffic classes. As we discussed above, this approach does not address the issue of resource distribution across service classes, and as a result, it can cause significant congestion or even starvation of best-effort traffic. We compare the performance of such an "isolated" routing approach to that of our integrated approach in Section 5.

A second strategy is to use a static link sharing policy as has been studied in [12] for class based queueing. With this approach, the capacity of a link is statically divided between QoS and best-effort sessions. The problem is how to determine what fraction of the link capacity should be used for each traffic class because the ratio of QoS sessions over best-effort sessions changes over time. Allocating too much bandwidth for best effort sessions can cause a high blocking rate for QoS sessions, while allocating too little bandwidth for best effort sessions can cause serious congestion. Moreover, the traffic in each class may not be evenly distributed. For example, QoS sessions may be concentrated in one part of the network, while best-effort sessions are concentrated in another part of the network. Static sharing policies cannot adapt to such an imbalance.

A semi-dynamic link sharing policy may be employed to overcome the problems of static link sharing. Under this sharing policy, the measured link utilization for different traffic classes is used to periodically update what fraction of the link capacity is assigned to each traffic class. While this provides some degree of adaptation,

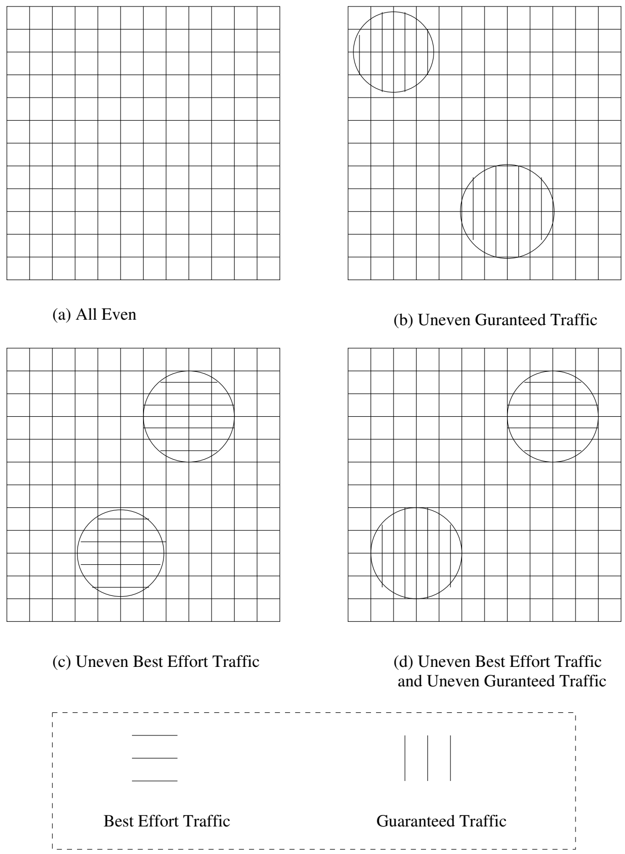


Figure 2: Four possible scenarios of load distribution

this strategy may not work well if there are sudden changes in the utilization. Often these changes are caused by external phenomena which are difficult or impossible to predict and whose effects are acute. Because the semi-dynamic algorithms are reactive rather than pro-active, latencies associated with their adaptive operation may make their reaction times too long.

Another approach is to use the measured link utilization of both QoS traffic as well as best-effort traffic as the link state for both traffic classes. This means that the routing algorithm for QoS traffic considers the load conditions of best effort traffic, and is likely to route QoS sessions along the links with low utilization. However, link utilization is not always a good indicator of how much bandwidth is available to reserve. For example, very bursty best-effort sessions can consume all link bandwidth, which leads to high utilization of the link, even though the link in question has reservable bandwidth. The routing algorithm for QoS traffic may thus reject a request even though plenty of reservable bandwidth is available.

In our multi-class routing algorithm, link resources are dynamically partitioned between QoS traffic and best-effort traffic. By adjusting link costs, we discourage the routing algorithm for QoS traffic from selecting links congested with best effort traffic, if alternative paths are available.

3 An Inter-class Sharing Architecture

We develop a multi-class routing algorithm that supports dynamic resource sharing across traffic classes.

3.1 Network Model

We use a flow-based network model. A flow can be a hard-state virtual circuit as in an ATM network, a soft-state flow as in an IP network, a stateless TCP connection, or the aggregation of several such micro-flows.

As discussed in Section 1, a variety of service models have been proposed to deliver different levels of end-to-end QoS to applications. As diverse as these models are, there is a common theme: the network gives preferential treatment to flows in one class over flows in other classes and bandwidth is the key QoS parameter. To simplify our discussion, we assume that there are two types of flows in the network: QoS flows and best-effort flows. QoS flows require certain kind of bandwidth guarantees. The intent is that our QoS flows cover the concepts of the Guaranteed Service, Controlled Load Service, and Differentiated Service classes. We discuss this point in more detail in Section 6. The packet scheduler in a router or switch ensures that the QoS traffic will at least get the amount of bandwidth that it reserved. Best effort sessions use the bandwidth left unused by the QoS sessions.

Best-effort flows in general want the network to deliver their packets to the destination as quickly as possible. Following [22], we distinguish between *high-bandwidth* traffic and *low latency* best-effort traffic. Since the low-latency sessions use more or less fixed minimum-hop 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. For this reason, we focus on high-bandwidth traffic. We will assume that best effort sessions share resources according to the max-min fairness principle (see [15, 17] for details).

3.2 Algorithm overview

Since we expect best-effort traffic to continue to be the dominant traffic class in the Internet, we will focus on that case, so our design has two goals. First, it seeks to admit the same number of QoS sessions as would be admitted by a network dedicated to QoS traffic. Second, it seeks to improve the performance of best-effort traffic by optimizing the placement of QoS sessions.

The idea behind our approach is to discourage QoS sessions from using links that already carry a heavy best-effort traffic load by increasing their link cost. This is achieved by using a link cost for QoS traffic that is based on the "virtual residual bandwidth" instead of the actual residual bandwidth. The virtual residual bandwidth captures the congestion conditions of best effort traffic, i.e. it is lower than the actual residual bandwidth on links that, relative to the rest of the network, carry a lot of best effort traffic and higher on links that have little best effort traffic.

In the remainder of this section, we describe how to identify links that carry heavy or light best effort traffic loads. We then define the virtual residual bandwidth and describe how it is used in the design of our multi-class routing algorithm.

3.3 Relative Link Congestion

As a first step we must be able to identify congested links. Our algorithm uses the max-min fair share rate as a barometer of network congestion. The thresholds that are used to identify the level of congestion are calculated dynamically based on network state information. This permits a more accurate measure of link congestion than would be possible with a static threshold.

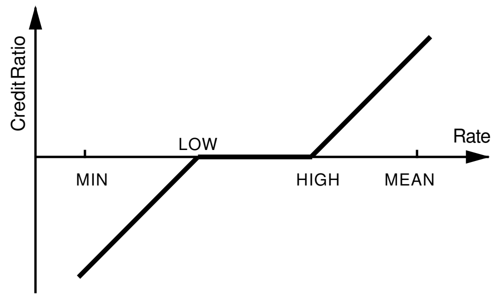


Figure 3: An inter-class load balance algorithm

We use link-state routing and assume that the routing protocol distributes the max-min fair share rates for the links in the network. From this rate information we then calculate the minimum and the mean max-min rates, which we represent as min and mean. While the mean max-min fair rate seems like an intuitively obvious threshold for separating heavily loaded from lightly loaded links, in practice it does a poor job. The reason is that congested links should have traffic loads well above average, so we want to select a threshold that is lower than, but tracks, the mean. We achieve this by dividing link rates into the following three intervals: $[\mathsf{min}, \mathsf{low})$, $[\mathsf{low}, \mathsf{high}]$, and (high, ∞), where

$$\mathsf{low} = \mathsf{min} + \frac{\mathsf{mean} - \mathsf{min}}{3} \quad (1)$$

$$\mathsf{high} = \mathsf{mean} - \frac{\mathsf{mean} - \mathsf{min}}{3} \quad (2)$$

This divides the interval $[\mathsf{min}, \mathsf{mean}]$ into three equal subintervals (Figure 3). When the max-min fair share rate of a link is in the lower subinterval $[\mathsf{min}, \mathsf{low}]$, the link is considered congested and, therefore, the QoS routing algorithm should be discouraged from using it. If the max-min fair rate of a link is greater than high, the link is not congested and its use should be encouraged.

3.4 Virtual Residual Bandwidth

We define the "virtual residual bandwidth" as the sum of the actual residual bandwidth and certain amount of "credit bandwidth", as defined below.

$$\text{virtual residual BW} = \text{residual BW} + \text{credit BW} \quad (3)$$

The amount of credit that is assigned to a link depends on its congestion status. When the link is lightly loaded, we add a positive credit, making the link more attractive to the QoS routing algorithm. When the link is congested, we add a negative credit, making the link less attractive for QoS sessions. We use the following formulas:

$$\text{credit ratio} = \begin{cases} \frac{r}{\mathsf{low}} - 1 & \text{if } r < \mathsf{low} \\ 0 & \text{if } \mathsf{low} \le r \le \mathsf{high} \\ \frac{r}{\mathsf{low}} - \frac{\mathsf{high}}{\mathsf{low}} & \text{if } r > \mathsf{high} \end{cases}$$

$$\text{credit BW} = \text{credit ratio} * \mathsf{MAX\_CREDIT} * \text{residual BW}$$

where $\mathsf{MAX\_CREDIT}$ is the maximal ratio of adjustment when the max-min fair share rate of a link approaches 0. With this definition, the credit ratio is positive when the max-min fair rate r of the link is in the interval (high, ∞) and is negative when r is in the interval $[\mathsf{min}, \mathsf{low})$ so the bandwidth credit and thus the virtual residual bandwidth track the congestion conditions. An important property of the virtual residual bandwidth is that it automatically adapts to changes in the network load, so there is no need to statically split link capacity between QoS sessions and best-effort sessions. The virtual residual bandwidth will adjust properly, both if the best effort hot spots move around in the network and if the ratio between best-effort and QoS traffic changes.

Other definitions for the credit ratio are of course possible. Our experiments suggest the credit ratio has to meet the following criteria to be effective:

- For a link heavily loaded with best-effort traffic, the difference between real residual bandwidth and virtual residual bandwidth should be large enough so that the congested link becomes unattractive.
- The change of the value of virtual link residual bandwidth should be smooth. That is the reason why we create a bandwidth interval where the credit bandwidth is 0.

3.5 A Multi-Class Routing Algorithm

We show in [19] that when routing traffic with bandwidth guarantees, the shortest-distance path based on the link residual bandwidth performs consistently well for both evenly and unevenly distributed loads. The distance of a path P is given by the equation

$$\text{dist}(P) = \sum_{i \in P} \frac{1}{R_i} \quad (4)$$

where R_i is the residual bandwidth of link i. For best effort traffic, we shown in [22] that a polynomial distance function

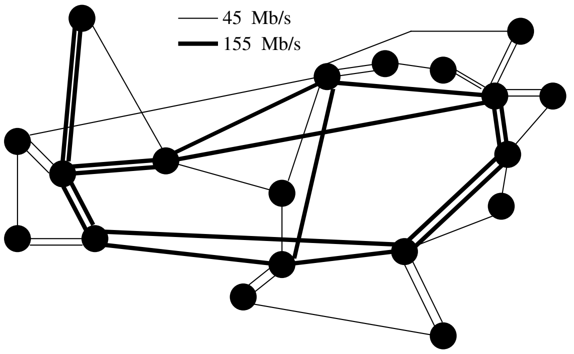
$$\text{Shortest-dist}(P, n) = \sum_{i \in P} \frac{1}{r_i^n} \quad (5)$$

where r_i is the max-min fair share rate for a new connection on link i, performs consistently better than either the minimum-hop path or the widest path.

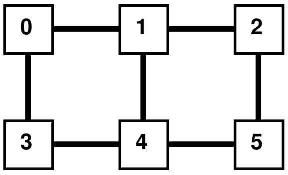
Given the two algorithms for the individual service classes, our multi-class routing algorithm is simple. For a best-effort session, we use the shortest-path routing algorithm with the distance function of Equation (5) as the path cost. For a QoS session, we use the shortest-path algorithm with the distance function of Equation (4) as the path cost, but we use the virtual residual bandwidth R_i, defined in Equation (3), as the rate:

```
MultiClass_Routing(alg_BE, alg_QoS, F, G)
void *(alg_BE)(), *(alg_QoS)();
flow F;
topology G;
{
    topology G_virtual;

    if (F is a best-effort traffic flow)
        alg_BE(F, G);
    else (F is a flow of QoS traffic) {
        G_virtual = credit_adjustment(G);
        alg_QoS(F, G_virtual);
    }
    return;
}
```



(a) MCI topology



(b) Switched cluster topology

Figure 4: Topologies used in the simulation

Hence, the proposed routing algorithm requires no changes to and is independent of the routing algorithms for the individual service classes. The only change needed is the use of the virtual residual bandwidth instead of the real residual bandwidth when routing QoS sessions. Note that the impact on throughput of this approach can of course be different when different algorithms are used for individual service classes.

As we show below, the multi-class routing algorithm based on dynamic bandwidth sharing works well for most scenarios. However, it can potentially create some problems. First, even though we focus on the case in which best effort sessions dominate the traffic load, it is still important to consider what happens when the QoS traffic load is high. Under these conditions, it is possible that QoS sessions reserve all the capacity on certain links, which may result in starvation of the best effort traffic. To avoid this problem, we set aside a certain fraction of the capacity of each link (e.g., 5% or 10%) for best-effort traffic. We evaluate the impact of this design decision later in the paper.

Another potential problem is that the dynamic link sharing algorithm may force QoS sessions to use longer physical paths. This increases resource consumption and could lead to higher call blocking rates. However, as was shown in [19], for light loads of QoS traffic, the blocking rates for different routing algorithms are all zero or close to zero, so this should not be a concern for the scenarios in which we are interested. Moreover, our simulation results in Section 5 show little increase in the blocking rate, even when the QoS traffic load is quiet heavy.

4 Simulation Design

Our evaluation is based on an extensive simulation study using an event-driven simulator that models activities at 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 (Figure 4): one is an 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: QoS sessions and best-effort sessions. The total traffic load is split between the two traffic classes according to a simulation parameter. Sessions arrive according to a Poisson distribution with the mean determined by the overall network load.

For QoS sessions, we assume that the requested bandwidth is uniformly distributed from 1 to 5 megabits/second. Most simulation studies of real-time traffic use an exponential call holding time distribution, while others assumed that all incoming calls last forever [25]. Recent studies [5] show that the call holding time distribution for conversations, facsimile, and voice mail connections have a large portion of very short calls and a log-normal long-tail distribution. We use a long-tail distribution following the model suggested in [5]: 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.

For best effort sessions, we assume that the traffic source specifies the total number of bytes to be sent. The link capacity that is not reserved by QoS sessions is shared among all high-bandwidth sessions following the max-min fair sharing principle.

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 the sessions select a source and destination pair from a preselected subset of nodes and the remaining sessions randomly pick any pair of switches as the source and destination.

4.2 Performance Metrics

Our performance metrics for traffic with bandwidth guarantees is the call blocking rate. A QoS 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. We use the *bandwidth blocking rate* as defined in [19], which takes into account the bandwidth of rejected sessions:

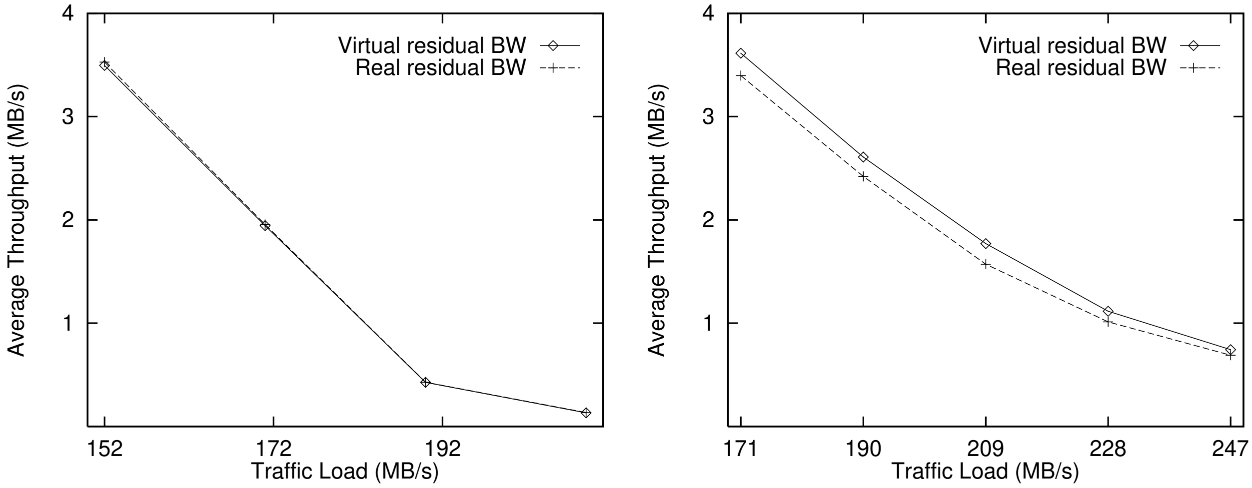
$$\text{bandwidth blocking rate} = \frac{\sum_{i \in \text{BG\_blk}} \text{bandwidth}(i)}{\sum_{i \in \text{BG}} \text{bandwidth}(i)}$$

where BG_blk is the set of blocked sessions and BG is the set of requests for QoS sessions.

Our performance metric for best-effort traffic is the average throughput of all best-effort sessions [22]:

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

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



(a) Evenly distributed load (b) Unevenly distributed load

Figure 5: Average throughput as a function of network load: MCI topology and 50% high bandwidth traffic, 90% of maximal reservation ratio. The Shortest-dist(P,1) path is used for best-effort traffic.

4.3 Routing Information

In addition to static topology information, the link state information includes the dynamic residual bandwidth and the max-min fair 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 Performance Evaluation

Our evaluation of the proposed multi-class routing algorithm characterizes the effects of topology, traffic load, degree of unevenly distributed load, maximum reservation ratio for a link, routing algorithms used for best-effort sessions, and the ratio of QoS to best-effort traffic. We compare several algorithms. Most of the evaluation focuses on comparing the performance of the dynamic inter-class resource sharing algorithm described in Section 3.5 with an approach in which the traffic of each class is routed independently. We will sometimes refer to the latter approach as isolated routing. We also compare our multi-class routing algorithm with an algorithm that uses static link partitioning.

Most of our simulation results are for the small cluster topology, since it is easier to understand inter-class sharing behavior with a small topology than with a larger one. It also has more manageable simulation times. As we described above, we expect that large topologies, such as the MCI topology, will have regions with uneven traffic loads; the cluster topology should be viewed as an example of such a region. Obviously, in a large topology, the average performance results will mix the performance of regions with even and uneven load, so the performance effects of inter-class routing may be less pronounced, depending on the mix of traffic. Given the many parameters, it is impractical to try to get a single number characterizing performance. Instead, our goal is to show when and under what circumstances dynamic inter-class resource sharing is desirable and to evaluate its ability to achieve its goal without reducing performance in network regions that have an even load.

To simplify our evaluation, we assume that all QoS sessions require bandwidth guarantees, although we expect that the discussion on the performance of inter-class sharing can also be applied to the case of QoS sessions with delay guarantees. The routing algorithm for QoS sessions is the shortest distance path algorithm that uses the link cost of Equation (4). We initially use the Shortest-dist$(P, 1)$ path routing algorithm for best-effort traffic. In Section 5.5 we evaluate the Shortest-dist$(P, 0.5)$ path algorithm [22] to see how the performance of our dynamic resource sharing framework is influenced by the choice of routing algorithm.

5.1 Impact of Traffic Load

Figure 5 shows the average throughput (MB/s) as a function of the network load for the MCI topology. The traffic is split evenly between traffic requiring bandwidth guarantees and best effort traffic. For all traffic loads considered, no QoS sessions are blocked.

When the traffic load is evenly distributed, we see virtually identical performance for the best-effort sessions, regardless of whether the actual residual bandwidth or virtual residual bandwidth is used. This is not unexpected. The function of the integrated routing framework is to distribute the QoS traffic so as to avoid using links that are congested with best-effort traffic. With an even load, the traffic mix is going to be similar on all links, and the dynamic inter-class sharing should, by design, have little or no impact on the overall performance.

The uneven load distribution (Figure 5(b)) corresponds to the case where 80% of the best-effort traffic is distributed over the upper half of the network and 70% of the QoS traffic is concentrated on the lower half of the network. We see an increase of up to 13% in the average throughput for best-effort sessions. Without dynamic link sharing, QoS sessions can be directed to links that are already congested with best-effort traffic. With dynamic link sharing, links that are congested with best-effort sessions are avoided by the routing algorithm for QoS sessions. For an uneven load, this improvement

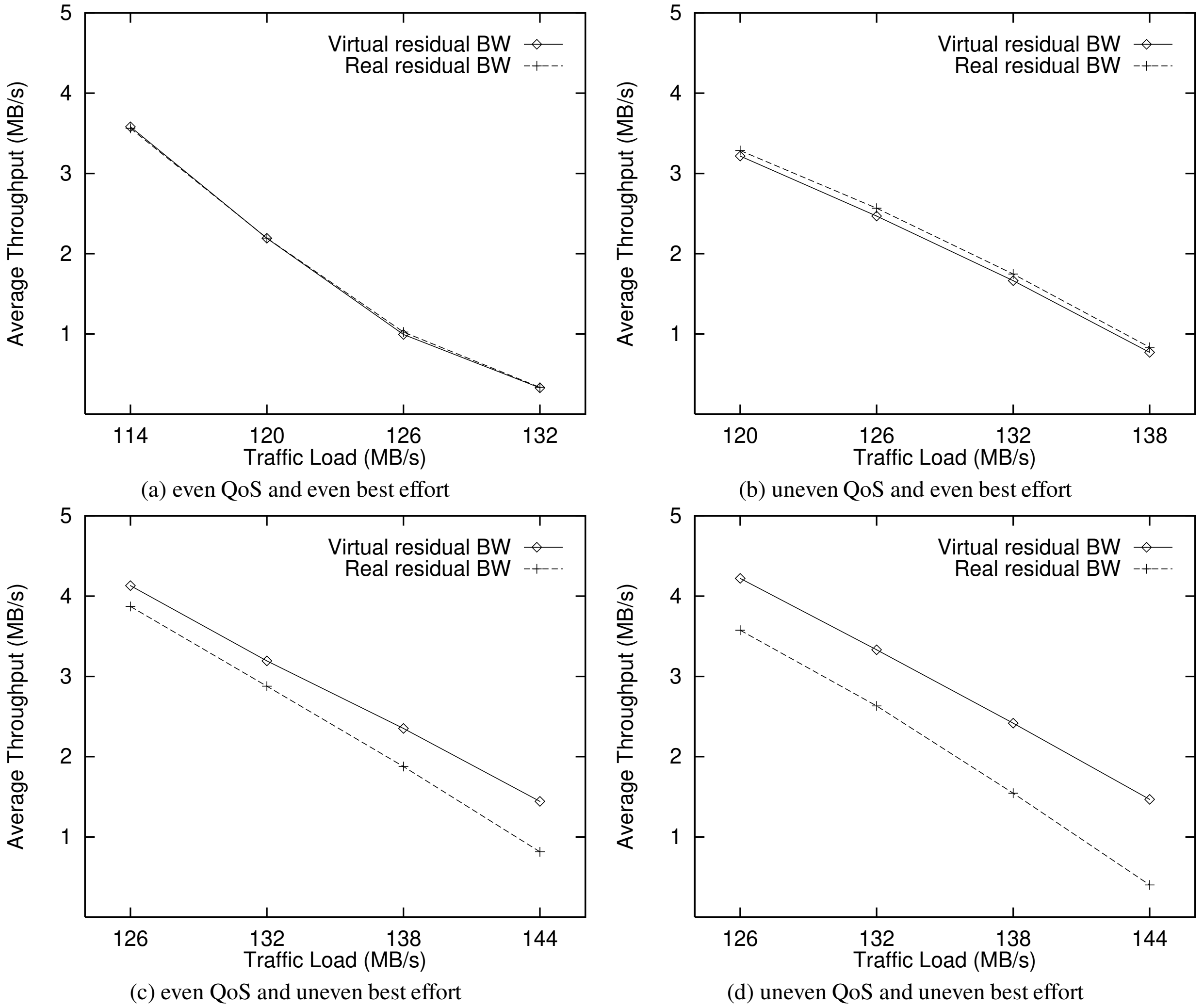


Figure 6: Average throughput as a function of network load: cluster topology, 50% QoS traffic and 50% best-effort traffic, and 90% of maximal reservation ratio. The Shortest-dist$(P, 1)$ path is used for best-effort traffic.

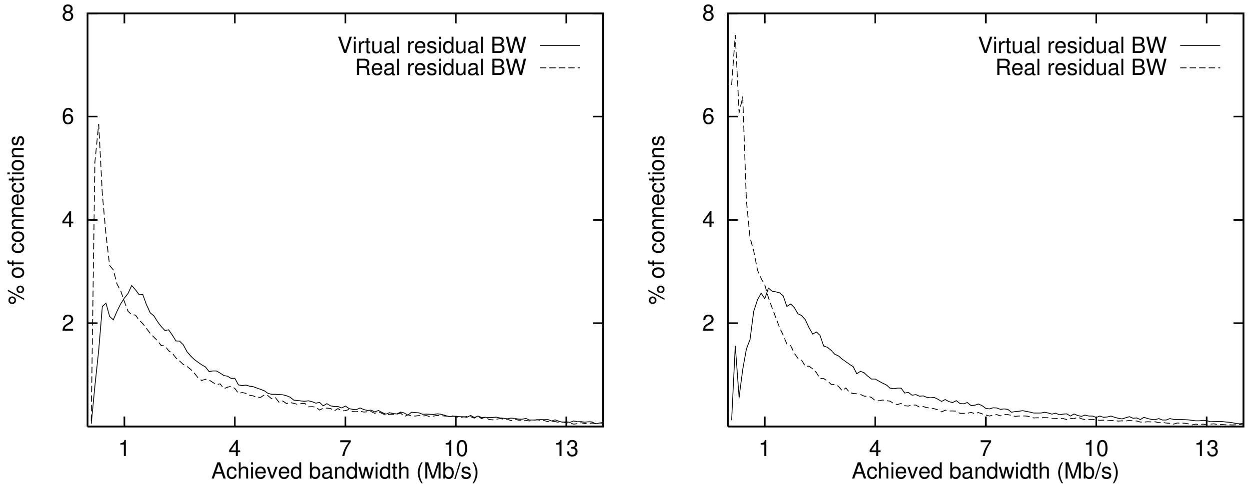
is significant since the performance index is average throughput. We will discuss the distribution of the per-flow bandwidth later.

Figure 6 shows the average throughput for the six-node cluster topology, where 50% of traffic is best-effort traffic and 50% is QoS traffic. The four graphs in Figure 6 correspond to different combinations of evenly and unevenly distributed loads. For the case of unevenly distributed best-effort traffic, 72% of best-effort sessions are concentrated on the links between node pairs (1, 4) and (3, 5). For the case of unevenly distributed QoS traffic, 72% of the traffic is distributed between node pairs other than (1, 4) and (3, 4), (4, 5), and (3, 5).

For all scenarios in Figure 6, no QoS flows are blocked so the throughput for best effort traffic is the only relevant performance metric. For evenly distributed loads (Figure 6 (a)), we see similar performance regardless of whether real or virtual residual bandwidth is used. Thus, our inter-class sharing mechanism does not hurt the performance of either traffic class when the load is evenly distributed, which was one of our design goals. When the QoS traffic is unevenly distributed (Figure 6 (b)), we see a very small performance degradation for best-effort traffic. The reason for this degradation is that, as a result of the inter-class sharing mechanism, QoS sessions sometimes use a slightly longer and more resource-intensive path, which takes resource away from best effort sessions.

When the best-effort traffic is unevenly distributed (Figure 6 (c) and (d)), we observe performance gains for best effort traffic. In some scenarios, *e.g.*, when the load is as heavy as 144 MB/s, the performance improvement for best effort traffic is as high as a factor of 3, because QoS sessions are routed around links (1, 4), (3, 4), and (4, 5), which are heavily loaded with best-effort traffic. Note that we are not suggesting that this is a typical performance improvement. Rather, our results show that the dynamic inter-class resource sharing algorithm can result in significant performance improvements in the most congested regions of the network, when the traffic load is unevenly distributed.

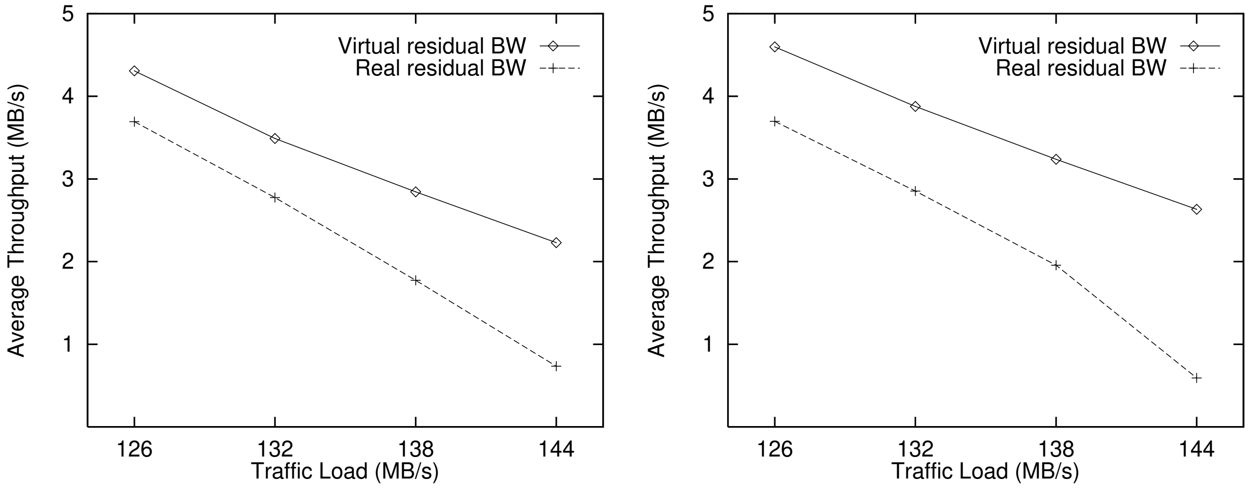
Figure 7 shows the throughput distribution of best-effort sessions for the load of 144 MB/s in Figure 6(c) and (d), i.e. for uneven best effort traffic. We see that the sessions that benefit the most from inter-class routing are the sessions with the worst performance, so inter-class routing results in a more even load distribution. Dynamic inter-class routing shifts many connections that achieve an average throughput between 0 and 1 Mb/s to a higher bandwidth region. This is done by directing QoS sessions to less



(a) even QoS and uneven best effort

(b) uneven QoS and uneven best effort

Figure 7: Variation of achieved throughput: cluster topology, 50% QoS traffic and 50% best-effort traffic (144 MB/s), and 90% of maximal reservation ratio. The Shortest-dist$(P, 1)$ path is used for best-effort traffic.



(a) even QoS and uneven best effort

(b) uneven QoS and uneven best effort

Figure 8: Average throughput as a function of network load: cluster topology, 25% QoS sessions and 75% high-bandwidth sessions, and 90% of maximal reservation ratio. The Shortest-dist$(P, 1)$ path is used for best-effort traffic.

Traffic load (MB/s)	126	132	138	144
Virtual residual BW	0.408%	0.789%	1.306%	2.196%
Real residual BW	0.381%	0.725%	1.160%	1.85%

(a) even QoS and uneven best effort

Traffic load (MB/s)	126	132	138	144
Virtual residual BW	0.031%	0.179%	1.151%	2.174%
Real residual BW	0.008%	0.071%	0.432%	1.088%

(b) uneven QoS and uneven best effort

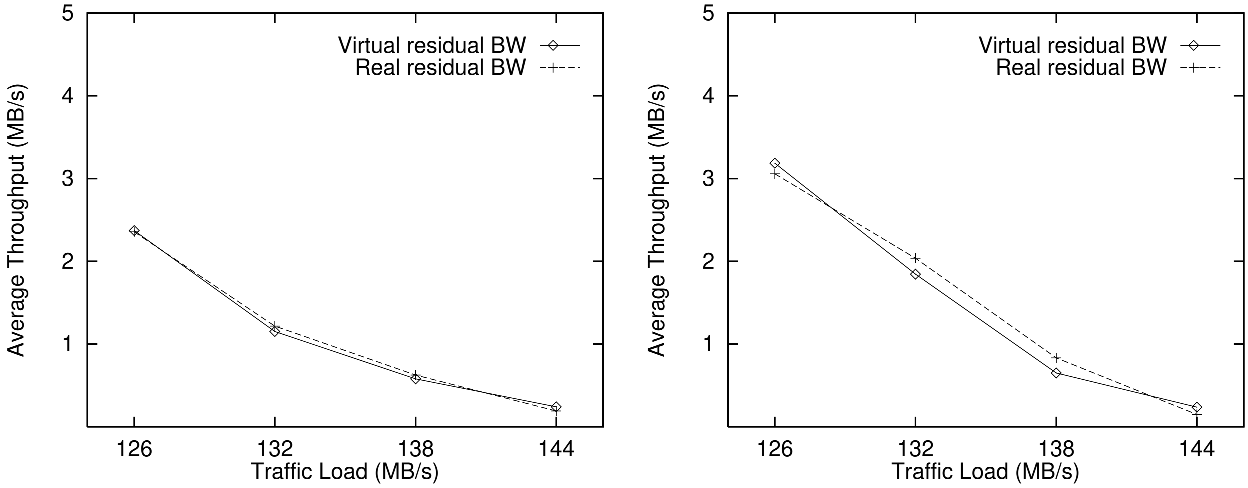
Table 1: Bandwidth blocking rate: cluster topology, 75% QoS sessions, 25% best-effort sessions, and 90% maximal reservation rate. The Shortest-dist$(P, 1)$ path is used for best-effort traffic.

congested links.

5.2 Impact of Best-effort Traffic Volume

In the previous experiments, the traffic was equally distributed between best-effort and QoS sessions. For the results in Figure 8, 75% of the traffic is best-effort. With the increased volume of best-effort traffic, links where best-effort traffic is concentrated become even more congested. As a result, it becomes even more important to route QoS sessions to other links whenever possible. Thus, dynamic inter-class resource sharing results in a higher performance improvement when compared with the case of a 50/50 workload.

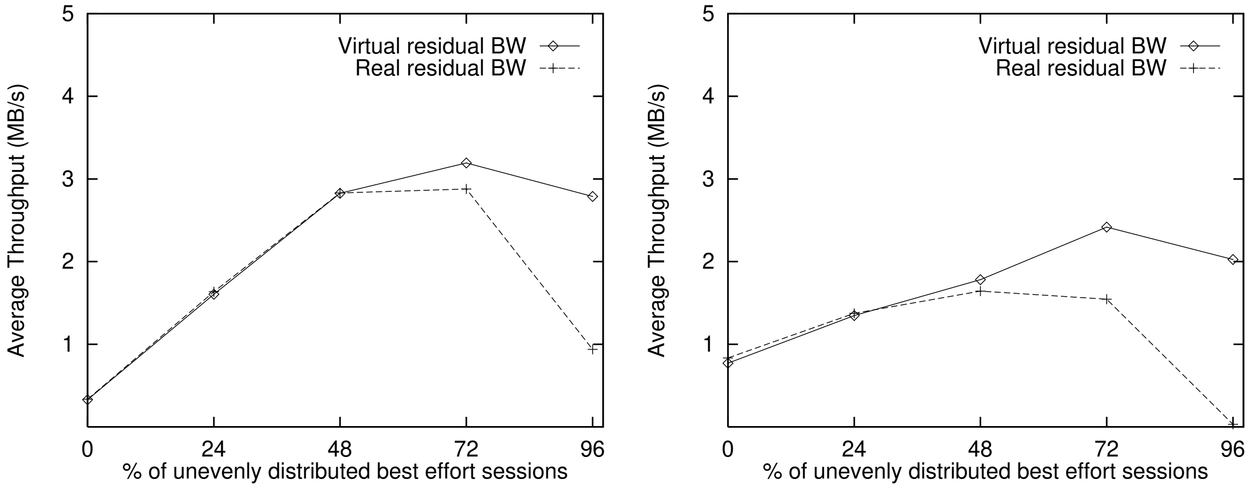
The proposed scheme is intended for use in an environment where the QoS traffic load is low compared to the overall network capacity. In such cases, the bandwidth blocking rate is not a concern. When the volume of QoS traffic is higher, using the virtual residual bandwidth can potentially increase the bandwidth blocking rate. Figure 9 shows the performance when 75% of the traffic is QoS traffic. We see that inter-class routing and isolated routing result in similar average throughputs for best effort traffic. Table 1 lists



(a) even QoS and uneven best effort

(b) uneven QoS and uneven best effort

Figure 9: Average throughput as a function of network load: cluster topology, 75% QoS sessions and 25% high-bandwidth sessions, and 90% maximal reservation ratio. The Shortest-dist$(P, 1)$ path is used for best-effort traffic.



(a) even QoS and uneven best effort, 132 MB/s

(b) uneven QoS and uneven best effort, 138 MB/s

Figure 10: Average throughput: cluster topology, 50% QoS sessions and 50% high-bandwidth sessions, and 90% maximal reservation rate. The Shortest-dist$(P, 1)$ is used for best-effort traffic.

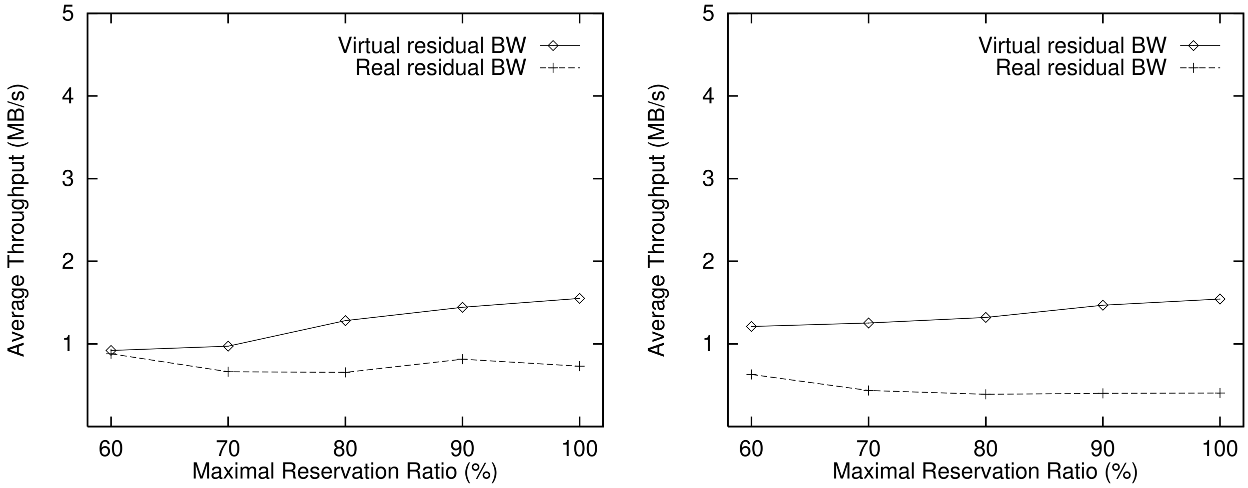
the bandwidth blocking rates for the QoS sessions. We observe a slightly higher blocking rate for QoS sessions when the dynamic inter-class resource sharing algorithm is used. The reason is that, when the network load is heavy, the inter-class resource sharing algorithm can cause QoS flows to use longer, resource-intensive paths, which can increase their call blocking rate.

5.3 Impact of the degree of Imbalance

In Figure 10, we show the average bandwidth of best-effort traffic as a function of the fraction of best effort sessions that are concentrated between nodes 1-4 and 3-5 of the cluster topology, i.e. higher values mean that the traffic distribution is more uneven. The results are for a distribution of 50% traffic in the QoS class and 50% in the best effort class and the total traffic load is 132 MB/s for Figure 10(a) and 138 MB/s for Figure 10(b). We see that the performance gain from using dynamic inter-class resource sharing increases as the traffic distribution becomes more uneven.

5.4 Impact of the Maximum Reservation Ratio

To avoid starvation of best effort sessions, QoS sessions can only reserve a certain fraction of the capacity of each link (Section 3.5). We will call this fraction the *maximum reservation ratio*. In the previous experiments, we used a maximum reservation ratio of 90%. In Figure 11, we show the average throughput as a function of the maximum reservation ratio for a fixed traffic load. For different maximum reservation ratios, the average bandwidth stays relatively constant when isolated routing is used. When using the multi-class routing algorithm, we see an increase in performance of as much of 50% to 60% as the maximum reservation ratio is increased from 60% to 100%. Increasing the maximum reservation ratio basically gives the routing algorithms more freedom. The multi-class routing algorithm can use this additional flexibility to move QoS traffic session away from links that have high loads of best effort traffic. In fact, the use of a multi-class routing algorithm seems to eliminate the need for a static maximum reservation ratio. The isolated routing



(a) even QoS and uneven best effort

(b) uneven QoS and uneven best effort

Figure 11: Average throughput as a function of the maximal reservation rate: cluster topology, 144 MB/s, 50% QoS sessions and 50% high-bandwidth sessions. The Shortest-dist$(P, 1)$ is used for best-effort traffic.

Max resv ratio (%)	60	70	80	90	100
Virtual residual BW	3.4%	0.4%	0.0%	0.0%	0.0%
Real residual BW	2.5%	0.3%	0.0%	0.0%	0.0%

(a) even QoS and uneven best effort

Max resv ratio (%)	60	70	80	90	100
Virtual residual BW	5.1%	0.4%	0.0%	0.0%	0.0%
Real residual BW	3.4%	0.3%	0.0%	0.0%	0.0%

(b) uneven QoS and uneven best effort

Table 2: Bandwidth blocking rate: cluster topology, 144 MB/s, 50% QoS sessions and 50% high-bandwidth sessions.

algorithm cannot take advantage of this additional freedom.

Table 2 shows the bandwidth blocking rates for QoS traffic for the same scenarios as Figure 11. The blocking rates are higher when only a small fraction of the link capacity can be reserved by QoS traffic. This is not surprise: a low maximum reservation ratio limits the freedom of routing algorithms and results in poor performance for both traffic classes. The results in Table 2 clearly show that a higher maximum reservation ratio translates into higher network throughput. Since these results are for a 50/50 distribution of the traffic between the QoS and best effort classes, the low maximum reservation ratio can be viewed as a simple static partioning of network resources between the two traffic classes. In this case, we observe poor performance for both best effort and QoS sessions. Figure 11 combined with Table 2 show that dynamic inter-class resource sharing combined with a high maximum reservation ratio results in the best performance.

5.5 Impact of Routing Algorithms for Best Effort Traffic

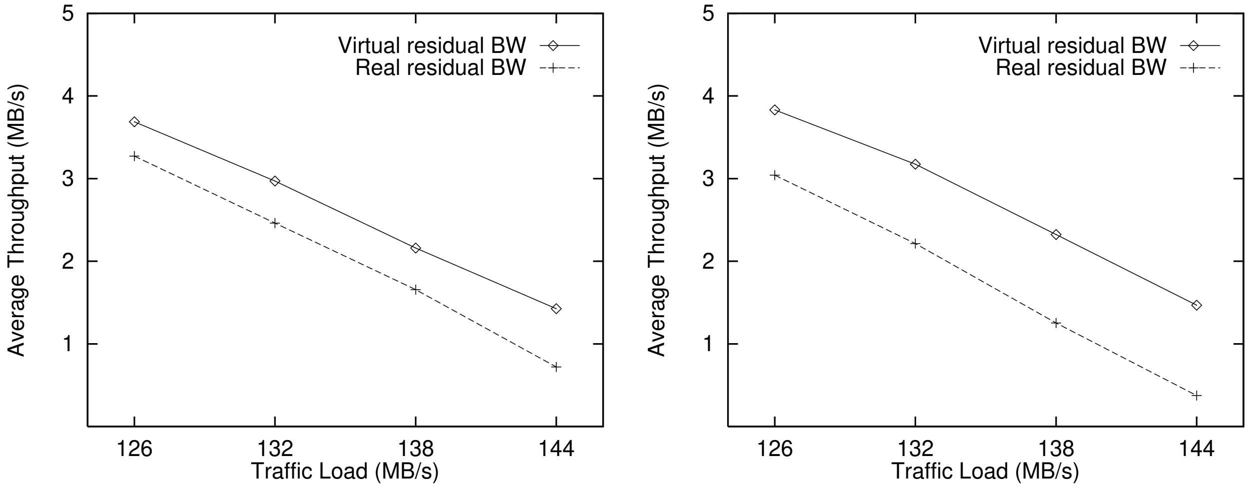
In all the previous evaluations, we used the Shortest-dist$(P, 1)$ path routing algorithm for best effort traffic. Figure 12 shows the performance for the same scenarios as Figure 6, but the routing algorithm used for best-effort traffic is the Shortest-dist$(P, 0.5)$ path algorithm. Note that the Shortest-dist$(P, 0.5)$ path algorithm is less sensitive to bandwidth availability, and more sensitive to hop count than the Shortest-dist$(P, 1)$ path algorithm. A comparison of Figures 6 and 12 shows that the Shortest-dist$(P, 0.5)$ path has lower performance than the Shortest-dist$(P, 1)$ path both for link costs based on the virtual and the real residual bandwidth. The reason is that the Shortest-dist$(P, 0.5)$ path algorithm is more likely to use the minimum-hop path than the Shortest-dist$(P, 1)$ path algorithm and is therefore less "agile", i.e. it cannot avoid links with low residual bandwidth.

Further comparison of Figures 6 and 12 also shows that the use of multi-class routing results in a higher average throughput improvement for the Shortest-dist$(P, 0.5)$ path algorithm than for the Shortest-dist$(P, 1)$ path algorithm. The reason is that inter-class routing is less likely to create links that are bottlenecks for best effort traffic, i.e. with high best effort traffic loads relative to the residual bandwidth, which is the scenario that the Shortest-dist$(P, 0.5)$ path algorithm does not handle well.

6 Discussion

In the previous sections, we focused on the case where the high priority traffic class requires bandwidth guarantees. However, we believe that the algorithm and insight should be useful for a broader set of service classes. The reason is simple: when multiple service classes coexist in the network, the notion of differentiating between these service classes by giving one class the first chance to claim bandwidth seems both typical and intuitive. For example, when allocating connections with delay guarantees according to the IETF guaranteed service class, delay is controlled through bandwidth. In [20], an iterative Bellman-Ford (IBF) routing algorithm for traffic with delay guarantees is described, and the link residual bandwidth used there could be replaced by the virtual residual bandwidth.

The more recently proposed differentiated services model specifies [8, 24] that core routers maintain aggregated state and that only edge routers can keep per-flow state. Most proposals for differentiated services offer at least one traffic class that is given some kind of preference (e.g. higher priority) over best effort flows. To admit such a differentiated flow, the network must know the resource availability inside the network. Hence, source routing based on link residual bandwidth is likely to be adopted in the network. Using the



(a) even QoS and uneven best effort

(b) uneven QoS and uneven best effort

Figure 12: Average throughput as a function of network load: cluster topology, 50% QoS sessions and 50% high-bandwidth sessions, 90% of maximal reservation ratio. The Shortest-dist$(P, 0.5)$ is used for best-effort traffic

inter-class sharing model, we can route the flows of differentiated services around links that are congested with best effort sessions.

In this paper we have defined and evaluated a multi-class routing algorithm for a network with two classes of service. For a network that supports more than two classes of service, our approach should be directly applicable when all but the best effort service require resource reservation. If the network supports more than two service classes with strict priority, then our approach does not directly apply. However, we can extend the credit calculation scheme of Equation (4) in Section 3.4 to consider the traffic load of service classes with lower priority. For example, we can assign credit for each of these service classes, but use different weights (i.e. the slopes in Figure 3) for different service classes. Further work is needed to define and evaluate such extensions.

7 Related Work

While several studies have addressed routing support for multiple classes of service, inter-class resource sharing for data networks has not been addressed.

In [23], Matta and Shankar study routing for a network that supports delay-sensitive and throughput-sensitive classes of service. It is shown that class-based routing improves performance significantly if one exploits the scheduling discipline used at links. IBM's System Network Architecture (SNA) introduced the concept of Class of Service (COS)-based routing [1, 13] where several classes of service (interactive, batch, and network control) were used. Users could also define additional classes. When a data session is started, an application or device would request a COS, which is then mapped into a statically configured route that identifies a path through the physical network. These studies do not address the issue of inter-class resource sharing.

More recently, a routing scheme called Real-Time Network Routing (RTNR) [2] is proposed for fully-connected circuit-switched networks. RTNR handles multiple classes of service, including voice and data at fixed rates. It utilizes a sophisticated per-class trunk reservation mechanism with dynamic bandwidth sharing between classes. When alternate routing is required, RTNR utilizes the load on all trunks in the network to select a path.

In contrast to these studies, we address routing support for resource sharing among service classes with different priorities. Our inter-class sharing scheme is simple and general. It does not require data traffic to be sent at a fixed rate, nor does our scheme require a trunk reservation mechanism with dynamic bandwidth sharing between classes.

8 Summary

In this paper we showed that multi-class routing is more than simply putting together routing algorithms designed for individual traffic classes. Careful inter-class resource distribution is essential to achieving high network throughput when the network load is unevenly distributed. Instead of statically partitioning link capacity across the service classes, we developed a dynamic link sharing approach that lets the routing algorithm determine how to partition link resources among the service classes.

At the core of our multi-class routing algorithm is the concept of "virtual residual bandwidth," which is the actual residual bandwidth adjusted to account for the link congestion conditions experienced by best effort traffic. The key idea is to direct QoS sessions to links that are less congested with best-effort traffic. The algorithm does not require any changes to the routing algorithms used for individual traffic classes and it is applicable to routing algorithms that use residual bandwidth as link state.

Our simulation results show the effectiveness of the proposed dynamic inter-class resources sharing approach. Compared with a multi-class routing algorithm that does not address inter-class resource sharing, we observe significant performance improvement for best-effort traffic when the load is heavy and unevenly distributed. This improvement is more significant when the best-effort traffic load is high, when the best effort traffic distribution is very uneven, when a higher portion of link capacity is reservable, and when the routing algorithm for best-effort traffic is more likely to select minimum-hop paths. For many scenarios, the local performance improvements in regions with uneven load distribution are significant. Moreover, our approach does not result in any performance degradation in regions of the network where the load is evenly distributed. Our results demonstrate that, contrary to the common belief, QoS routing is desirable even when the QoS traffic load is light and the call blocking for QoS sessions is not a concern.

References

[1] V. Ahuja. Routing and Flow Control in SNA. *IBM Systems Journal*, 18(2):298–314, 1979.

[2] G.R. Ash, J.S. Chen, A.E. Frey, and B. D. Huang. Realtime network routing in a dynamic class-of-service network. In *Proceedings of the 13h ITC*. ITC, 1992.

[3] S. Bahk and M. Elzarki. Dynamic Multipath Routing and How it Compares with Other Dynamic Routing Algorithms for High Speed Wide Area Networks. *ACM SIGCOMM 92*, September 1992.

[4] J. Behrens and J.J. Garcia-Luna-Aceves. Distributed, Scalable Routing Based on Link-State Vectors. *ACM SIGCOMM*, September 1994.

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

[6] R. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture: an Overview. *IETF RFC 1633*, June 1994.

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

[8] D. Clark and J. Wroclawski. An Approach to Service Allocation in the Internet. *IETF Internet Draft <draft-clark-diff-svc-alloc-00.txt>*, July 1997.

[9] Eric Crawley, Raj Nair, Bala Rajagopalan, and Hal Sandick. A Framework for QoS-based Routing in the Internet, August 1998. IETF RFC 2386.

[10] Eric Crawley, Raj Nair, Bala Rajagopalan, and Hal Sandick. Implementation and Performance Measurements of QoS Extensions to OSPF, 1998. Unpublished draft. Available at http://www.seas.upenn.edu/ guerin/public ations/dated_qospf.ps.gz.

[11] PNNI SWG (Doug Dykeman ed.). PNNI Draft Specification. ATM Forum 94-0471R10, October 1995.

[12] S. Floyd and V. Jacobson. Link-sharing and Resource Management Models for Packet Networks. *IEEE/ACM Transactions on Networking*, 3(4):365–386, August 1995.

[13] J.P. Gray and T.B. McNeil. SNA Multi-System Networking. *IBM Systems Journal*, 18(2):263–297, 1979.

[14] R. Guerin, D. Williams, T. Przygienda, S. Kamat, and A. Orda. QoS Routing Mechanisms and OSPF Extensions. *IETF Internet Draft <draft-guerin-qos-routing-ospf-03.txt>*, March 1998. Work in progress.

[15] J.M. Jaffe. Bottleneck flow control. *IEEE Transactions on Communications*, COM-29(7):954–962, July 1981. Correspondence.

[16] 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.

[17] D. Lin and R. Morris. Dynamics of Random Early Detection. In *ACM SIGCOMM '97*, pages 115–126, Cannes, France, Sepetember 1997.

[18] Q. Ma. *Quality of Service Routing in Integrated Services Networks*. PhD thesis, Department of Computer Science, Carnegie Mellon University, 1998.

[19] Q. Ma and P. Steenkiste. On Path Selection for Traffic with Bandwidth Guarantees. In *IEEE International Conference on Network Protocols*, Atlanta, Georgia, October 1997.

[20] 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.

[21] Q. Ma and P. Steenkiste. Routing Traffic with Quality-of-Service Guarantees in Integrated Services Networks. In *The 8th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 98)*, Cambridge, England, July 1998.

[22] Q. Ma, P. Steenkiste, and H. Zhang. Routing High-Bandwidth Traffic in Max-Min Fair Share Networks. In *ACM SIGCOMM '96*, pages 206–217, Stanford, CA, August 1996.

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

[24] K. Nichols, V. Jacobson, and L. Zhang. A Two-bit Differentiated Services Architecture for the Internet. *IETF Internet Draft <draft-nichols-diff-svc-arch-00.tex>*, Nov 1997.

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

[26] A. Shaikh, J. Rexford, and K. Shin. Efficient Precomputation of Quality-of-Service Routes. In *The 8th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 98)*, Cambridge, England, July 1998.

[27] S. Shenker, C. Partridge, and R. Guerin. Specification of guaranteed quality of service, September 1997. IETF RFC 2212.

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

[29] J. Wroclawski. Specification of the Controlled-Load Network Element Service, September 1997. IETF RFC 2211.

[30] William Zaumen and J.J. Garcia-Luna Aceves. Dynamics of distributed shortest-path routing algorithms. *ACM SIGCOMM*, September 1991.