FAIR BANDWIDTH ALLOCATION FOR DIFFERENTIATED SERVICES

Anindya Basu
Bell Laboratories
basu@research.bell-labs.com

Zheng Wang
Bell Laboratories
zhwang@dnrc.bell-labs.com

Abstract

In the past year, there have been several proposals for providing differentiated services in the Internet. In this paper, we investigate how fairly bandwidth is allocated by three of these proposals, the RIO scheme, the two-bit scheme and the User-Share Differentiation (USD) scheme. We simulate the three schemes for a number of different scenarios under the same network configuration. The results show that matching the network bottleneck bandwidth to the expected bandwidth profiles is an important issue for RIO and two-bit schemes. While the RIO and two-bit schemes perform well when there is no such mismatch, it is not the case if the bottleneck bandwidth available is in excess of the aggregated expected bandwidths. The bandwidth allocation in such a situation is not commensurate with the user expectations. On the other hand, the USD scheme is able to do a fair allocation of bandwidth that is in keeping with user expectations. Short-lived flows such as the web traffic are also an interesting scenario. The RIO and two-bit schemes need to be modified in some way to react more quickly to changes in network conditions. The USD scheme is able to deal with such traffic much better since it does not rely the edge to enforce admission control. In the case of non-responsive sources, RIO performs better than the two-bit scheme when the non-responsive sources are CBR, however, malicious sources that flood the network are able to starve low bandwidth sources. The USD scheme is able to provide better traffic isolation under such circumstances. For high bandwidth flows, all the schemes perform well, except that in the two-bit scheme case, an over-committed network may deprive some of the sources from getting any bandwidth, instead of allocating the available bandwidth evenly among all active sources. One implication of our findings is that some form of coarse level of signaling for aggregated flows will be necessary in order to provision a network in a more efficient fashion.

1. INTRODUCTION

The Internet is currently based on the best-effort model where all data packets are treated equally and the network tries its best to ensure data delivery. The best-effort model has been successful till now because the majority of the traffic on the Internet is TCP-based [12]. The TCP end-to-end congestion control mechanism forces traffic sources to back off whenever congestion is detected in the network. However, such dependence on the end systems' cooperation is increasingly becoming unrealistic. Given the current best-effort model with FIFO queuing inside the network, it is relatively easy for non-adaptive

sources to gain greater shares of network bandwidth and thereby starve other, well-behaved, TCP sources. For example, a greedy source may simply continue to send at the same rate when faced with congestion while other TCP sources back off. Even today, many applications such as Netscape web browsers take advantage of the best-effort model by opening up multiple connections to web servers in an effort to grab as much bandwidth as possible. The best-effort model is also inadequate for applications such as real time audio and video that require explicit bandwidth and delay guarantees. Moreover, the best-effort model treats all packets equally once they have been injected into the network. Thus, it is difficult for Internet Service Providers (ISPs) to provide services that are commensurate with the expectations of consumers who are willing to pay more for a better class of services.

Over the last ten years, considerable efforts have been made to provide Quality-of-Services (QoS) assurance in the Internet. Much of the work has focused on the end-to-end per-session reservation approach where an application makes an end-to-end reservation before starting a session [15]. While it makes sense to set up a reservation for a long-lasting session, this approach is not suitable for highly transaction-oriented applications such as browsing the World Wide Web. The overheads and latency involved in setting up a reservation for each session become prohibitively high. There are also concerns regarding the scalability of providing fine-grained guarantees at the per-session level in the core of the Internet.

More recently, a number of proposals have been put forward for providing differentiated services in the Internet [1]. In contrast to the per-session reservation approach, the differentiated services approach allows service providers to offer different levels of services to a few (typically, a small number such as two or three) classes of aggregated traffic flows. For example, an ISP may offer two levels of services - a premium service for some customers who are willing to pay more and a best-effort service at a lower price. The differentiated services approach attempts to allocate bandwidth for aggregated flows without per-session signaling or maintaining per-session state in the network core, and to offer better services for users willing to pay more with long-term service contracts.

Some of these proposals, such as the RIO scheme by Clarke *et al* [4] and the two-bit scheme by Jacobson *et al* [7], advocate pushing traffic policing to the edge of the network and leave the core of the network as simple as possible. Incoming traffic from a user is monitored at the edge of the network against the profile that the user has agreed with the service provider. The packets that are out of the profile are either shaped or tagged and the core routers process such tagged (or untagged) packets according to predefined (packet drop) policies. Previous work has shown how to set the packet drop policies at the core routers to obtain desired throughputs for both tagged and untagged packets [17]. Such an approach has the advantage of simplicity since core routers do not need to store state for each traffic flow; instead, they can process packets using policies based on a small number of bits in the packet header. On the other hand, since the traffic is only policed at the edge, substantial over-provisioning in the core of the network may be required to ensure that there is sufficient bandwidth for the aggregated flows. In addition, as the core of the network can only see aggregated traffic, fairness among individual traffic flows belonging to the same class may not be maintained when there is congestion.

A different set of proposals that is exemplified by the User-Share Differentiation (USD) scheme proposed by Wang [13, 14] advocate putting modestly sophisticated control mechanisms in core routers

to provide proportional fair sharing of resources and per-prefix traffic isolation. The advantage of such schemes is that they do not require a priori estimation of resources that a particular user or an application expects to use. At any point in the network, the USD scheme can ensure that the bandwidth allocated to a flow is commensurate with some predefined metric, such as the price paid by the originator of the flow to the ISP. For example, it is possible to guarantee that user A who pays twice as much as user B, gets twice as much bandwidth as user B anywhere in a network. The major disadvantage of such schemes lie in the increased complexity of the routers in the network core and the concern that such complexity may not be amenable to scaling.

The issue of fairness among users is important in this context since we expect that the differentiation in service will be based on some kind of pricing mechanism, thus it is necessary that the resource allocation be commensurate with how the service is priced. One of the key resources in a network is network bandwidth. In this paper, we study how fairly these three schemes allocate bandwidth. To analyze the performance of these schemes, we implement them on a simulated network using the REAL network simulator [10] and study their performances under different network configurations. We find that although the RIO and two-bit schemes are easy to implement and function well when sources are TCP friendly and the user profiles match the bottleneck bandwidths, they do not perform well if user profiles do not match bottleneck bandwidths or if sources that are unresponsive to network congestion are present. Furthermore, if connections are short-lived, bandwidth allocations by these schemes are not commensurate with user profiles. In contrast, the USD scheme provides a better match of performance to user expectations and is more robust with respect to short-lived flows (such as web traffic) despite the added cost of implementing proportional fair sharing in the network core. Finally, we outline the need for a network provisioning mechanism that dimensions the network based on different service level guarantees.

The rest of the paper is organized as follows: Section 2 describes the three schemes in detail. Section 3 evaluates the performance of the three schemes under different scenarios. Section 4 summarizes the results and we conclude the paper in Section 5 with a discussion of the implications of our findings.

2. THREE PROPOSALS FOR DIFFERENTIATED SERVICES

In this section, we describe in detail three differentiated services schemes, namely, the RIO scheme, the two-bit scheme and the USD scheme.

The RIO Scheme

The RIO scheme (RIO stands for Random Early Detection with In/Out bit) is also referred to as the assured service scheme. This scheme uses some form of packet tagging to indicate the drop priority of the packet to the core network routers. Each user (or traffic flow, depending upon the granularity) is assigned a service profile by the ISP based on the expected bandwidth utilization by the user. At the edge of the network domain managed by the ISP (i.e., at the ingress points), user traffic is monitored by a profile meter to ensure that it stays within the profile. Any packets that are out of profile are marked as "out" while those that conform to the user profile are marked as "in". In the network core, the "in" and

"out" packets are treated with different drop priorities using RED. In short, "in" packets start being dropped only when the queue size crosses a higher threshold than in the case of "out" packets and with a lower probability. This ensures that traffic that conforms to its profile has less chance of getting dropped than the out-of-profile traffic, and therefore gets predictable levels of service so long as it stays within profile. At the same time, the RIO scheme makes it possible to use statistical multiplexing to make use of any excess bandwidth that may be available since it does not prevent out-of-profile packets from entering the network.

It is important to note here that the RIO scheme requires the profile meters to estimate the instantaneous sending rates. To do that, the authors propose a Time Sliding Window (TSW) scheme for TCP traffic that provides an exponentially averaged estimate of the TCP sending rate over time. The TSW scheme estimates the instantaneous sending rate upon each packet arrival and decays the past history over time in order to adapt to changing network conditions. If the estimated sending rate on packet arrival exceeds a certain threshold, the packet is marked as out of profile, otherwise it is in profile. The details are described in [3].

The Two-Bit Scheme

The two-bit differentiated services architecture combines the RIO scheme with Jacobson's premium service [5] to come up with three different classes of service, namely premium, assured and best-effort. The premium service is based on explicit resource reservations and a simple priority queuing model where the premium traffic is transmitted prior to other traffic. Such a service is useful for time-critical applications that require minimum delay as well as applications that require explicit delay and bandwidth guarantees. The premium traffic is strictly policed at the edge of the network. Packets that are out of the service profile are dropped or delayed until they are within the service profile. The assured service class corresponds to the "in" traffic in the RIO scheme and has a lower drop priority than the ordinary best-effort service. In times of congestion, the chances are high that the assured packets would receive predictable levels of service if they conform to their service profile, which is typically based on average bandwidth utilization. This service is appropriate for users who are willing to pay somewhat more in order to get better performance in times of congestion.

In order to implement the two-bit scheme, the authors propose two distinct bit patterns in the packet headers that act as tags – the P bit for the premium service and the A bit for the assured service. Routers in the network are classified into two types, the edge routers that implement some form of profile metering and the core routers that use the packet tags to assign priorities to packets. The core routers do not need to store any per-session state: they only need to know how to process packets with the P tag, packets with the A tag and untagged packets. The profile meter on an edge router uses a leaky bucket algorithm to meter premium and assured packets. For premium packets, the leaky bucket consists of tokens that fill up the bucket at the peak rate that has been reserved for the premium service. When a new packet arrives, it is passed on to the forwarding engine or dropped depending on whether a token is available or not. In the case of assured packets, the leaky bucket is filled at the average rate based on the expected user profile. On arrival of an assured packet, the A bit is reset if there are no tokens available and the packet is sent to the forwarding engine.

The forwarding engine is common to both the leaf and core routers. It implements two sets of priority queues, a high priority queue for premium packets and a low priority queue for assured and best-effort packets. The high priority queue is always serviced before the low priority queue. The low priority queue is serviced using the RIO scheme, treating the packets with the A bit set as "in" packets and the rest as "out" packets. As in the RIO case, this ensures that the assured traffic flows get predictable levels of service when they stay within profile.

The USD Scheme

The USD scheme takes an approach that is quite different from that of the RIO and two-bit schemes. First, instead of taking a minimalist approach with a small number of distinguishable classes, the USD scheme allows traffic isolation on a per-customer basis. This allows much more sophisticated service contracts to be supported. Second, in contrast to the RIO and two-bit schemes where the traffic policing happens at the edge of the network, the USD scheme manages bandwidth allocation directly on the bottlenecks where the congestion takes place. Finally, the USD scheme works with traffic of both outgoing and incoming directions whilst both RIO and two-bit schemes primarily deal with traffic going from customers' networks to their ISP.

The USD scheme introduces two terms for bandwidth allocation, user and share. The *user* refers to the customer to which the bandwidth is allocated. In USD, a user can be a network or a group of networks identified by a prefix or an individual end user. Each user is assigned a number called *share*, based on some predefined metrics, for example, the amount that a user has paid for the service. The USD scheme operates with the *proportional fair sharing (PFS)* principle, whereby at any point in a network, the bandwidth is allocated in proportion to the traffic from or to a user using its share as weight. In other words, the exact amount of bandwidth allocated to flow from or to a user is determined by the total bandwidth available at the bottleneck and the relative weights (or shares) of active users competing for the bottleneck resources.

Proportional allocation ensures that the allocation is always fair with respect to the allocated shares. Such an approach is much more flexible when we try to accommodate different traffic dynamics. For example, traffic volume fluctuates throughout the day. The PFS scheme can adapt to the changing conditions. Whilst the amount of bandwidth that a user gets depends on the competing traffic, its fair share is always guaranteed. In a network consisting of links that have different speeds, proportional allocation can scale in accordance to the speed of the link. If an ISP has a "thin" international link to remote ISP and a "fat" local link to a neighboring ISP, proportional allocation functions correctly on both the bottlenecks. Such flexibility is particularly useful for transaction-oriented data applications, where end-to-end mechanisms may not react in time to ensure that the bandwidth allocations at bottlenecks are done in accordance with user profiles. When network failure occurs, even an over-provisioned network may still experience congestion, for example, during some link failures. Proportional allocation ensures that there is still explicit policing even under such circumstances.

The USD scheme can be implemented with a classifier capable of prefix lookup (to identify flows belonging to individual customers) and a scheduler that supports PFS. Prefix lookup has been extensively studied for route lookups and several novel algorithms have been proposed for this purpose

[5]. Hardware-based route lookup implementations that can support over 1 millions entries [11] exist and the same techniques can be used in the USD for identifying which customer a packet belongs to. A large number of scheduling algorithms can support PFS. Among them, weighted fair queuing (WFQ) [8] can support PFS with the most stringent fairness guarantees. Many variations of WFQ, such as Deficit Round Robin (DRR) and Self-Clock Fair Queuing (SCFQ), have lower implementation complexity and yet provide reasonable fairness guarantees [6, 9]. For example, the Bell Labs Router implements a variation of WFQ with 64,000 queues at 622 Mb/s speed [11].

3. SIMULATING DIFFERENTIATED SERVICES

In this section, we present the simulation results with the three proposals for differentiated services we have described in the previous section. The simulation was done with the REAL network simulator [10]. For purposes of the simulation, we used a network with the configuration shown in Figure 1. In the simulation, we have 10 sources (1 through 10 counting downwards) that communicate with one of two different destinations, X and Y. The sources are connected to destination X through a bottleneck link of capacity 33Mbps and to destination Y through a bottleneck link of capacity 155Mbps. Each of the links that connects a source to the first router in the diagram has a capacity of 30Mb/s. The one-way propagation delays for the sources (counting from top downwards) were set at 20, 20, 40, 40, 50, 50, 70, 70, 100, and 100 milliseconds respectively. The sources are all TCP-Reno sources (unless specified otherwise) that use a packet size of 500 bytes and each of them sends 500,000 packets to the destination. For the RIO implementation, the routers use RED with the values of 400 packets, 800 packets, and 0.02 for *min_in*, *max_in*, and P_{max_in}, and 100 packets, 200 packets and 0.5 for *min_out*, *max_out*, and P_{max_out} [3] where *min_in* and *max_in* represent the upper and lower bounds for the average queue size for in profile packets when RED kicks in and P_{max_in} is the maximum drop probability for an in profile packet when the average queue size is in the [*min_in*, *max_in*] range. The *min_out*, *max_ou*t and P_{max_out} are the corresponding parameters for the out of profile packets. Note that the total queue sizes are used when calculating the drop probability for out of profile packets, as suggested in [3]. For the USD scheme, we used the weighted fair queuing (WFQ) discipline on the router to provide PFS.

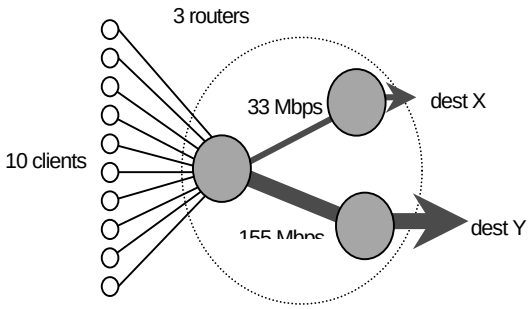


Figure 1: Simulated Network

Effect of Expected Bandwidth Profiles

We first evaluated the effect of expected bandwidth profiles and bottleneck bandwidths on actual bandwidth allocation. For this purpose, we ran three sets of simulations for each of the two bottleneck links (33Mb/s to destination X and 155Mb/s to destination Y). In each of the simulations, the odd

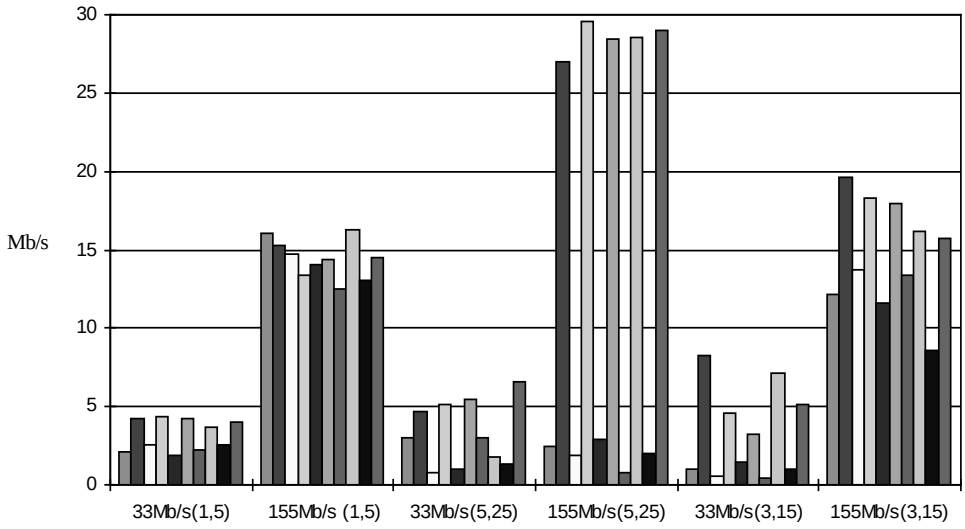


Figure 2: Bandwidth share for each of the 10 sources, for two different bottleneck bandwidths and 3 different expected profiles using the RIO scheme

numbered sources (1,3,5,7,9) had the lower expected bandwidth and the even numbered sources (2,4,6,8,10) had the higher expected bandwidth. In the first set of simulations, we set the expected lower and higher bandwidths to 1Mb/s and 5Mb/s, respectively, such that the aggregate expected bandwidth matched the 33Mb/s bottleneck bandwidth. In the second set, the expected bandwidths were chosen to be 5 and 25Mb/s, respectively to match the 155Mb/s bottleneck and in the third set, they were set to 3 and 15Mb/s, respectively, which is the average of expected bandwidths in the first two sets.

The performance of the RIO scheme is shown in Figure 2. The figure consists of six clusters with 10 bars in each cluster. Each bar has a label of the form *x*Mb/s (*y*, *z*), which means that the particular cluster represents the bandwidth allocation for the 10 sources when the bottleneck bandwidth is *x*Mb/s and the higher and lower expected bandwidths are *y* and *z*Mb/s, respectively. The results indicate that if the aggregate expected bandwidth closely matches the bottleneck bandwidth, the RIO scheme is able to allocate bandwidth in accordance with the expected bandwidth profiles. This is clear from the graphs for the 33Mb/s (1,5) and 155Mb/s (5,25) cases. In the first case, the sources with a lower expected bandwidth are allocated a slightly higher bandwidth than their expected profiles at the expense of sources with a higher expected bandwidth and the situation is the reverse when we have the 155Mb/s (5,25) case. We expect that it is possible to tune the RED parameters to obtain a bandwidth allocation that is more commensurate with the expected bandwidth profiles. If the bottleneck bandwidth is in excess of the aggregated expected bandwidth, the excess bandwidth is more or less evenly allocated among all the sources. This is evident from the figures for the 155Mb/s (3,15) and the 155Mb/s (1,5) cases. In the 155Mb/s (3,15) case, the sources with the higher expected bandwidth do get more bandwidth than the others, but the relative difference is smaller than those in the 33Mb/s (1,5) and the 155Mb/s (5, 25) cases where the bottleneck bandwidth matches the aggregate expected bandwidth. In

the 155Mb/s (1,5) case (where the difference between the bottleneck bandwidth and the aggregated expected bandwidth is higher), the bandwidth allocation is fairly even among all the sources. This is because the excess bandwidth is only allocated to the out of profile packets, and the RED scheme for out of profile packets ensures that there is some fairness in allocation of this excess bandwidth.. Finally, if the aggregated expected bandwidth is less than the bottleneck bandwidth, the RIO scheme is able, for the most part, to allocate the existing bandwidth in a manner that is commensurate with existing user expectations. There is one exception in the 33Mb/s (5, 25) case, where the bandwidth allocated to source 8 is less than that allocated to sources 1 and 7 although the latter have lower expected bandwidths. In summary, the RIO scheme performs well if the bottleneck bandwidth matches or is less than the aggregate expected bandwidth, but is unable to allocate bandwidth in proportion to user expectations if there is excess bandwidth at the bottleneck.

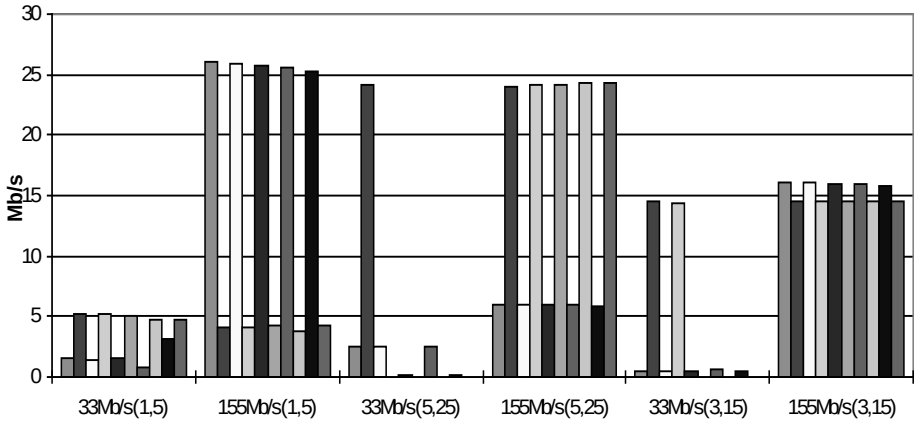


Figure 3: Bandwidth share for each of the 10 sources, for two different bottleneck bandwidths and 3 different expected profiles using the two-bit scheme.

To evaluate the two-bit scheme, we set the higher bandwidth traffic as the premium traffic. For example, in the 33Mb/s (1,5) case, the 5Mb/s sources were all considered premium sources. The low bandwidth sources were subjected to the RIO scheme where all in profile packets had their "A"-bit set while the out of profile packets did not. The RIO parameters in this case were the same as in the previous scheme. Figure 3 shows the bandwidth allocation for the two-bit scheme. We see that the bandwidth allocation matches the expected bandwidth in the 33Mb/s (1,5) case and the 155Mb/s (5,25) case. This implies that the two-bit scheme performs well when the bottleneck bandwidth matches the aggregate expected bandwidth. However, we see that in the 155Mbit/s (1,5) and the 155Mb/s (3,15) cases, the bandwidth share allocated to the sources with lower expected bandwidth is higher. This is because of the nature of the premium traffic: excess bandwidth is never allocated to traffic belonging to this category because any out of profile premium packet is dropped by the profile meter. Consequently, all the excess bandwidth in these cases goes to the sources with lower expected bandwidth. In this respect, the RIO scheme does a bandwidth allocation that is more in keeping with user expectations. In the 33Mb/s (5,25) and 33Mb/s (3,15) cases, some of the high expected bandwidth sources do not get any bandwidth at all. This is

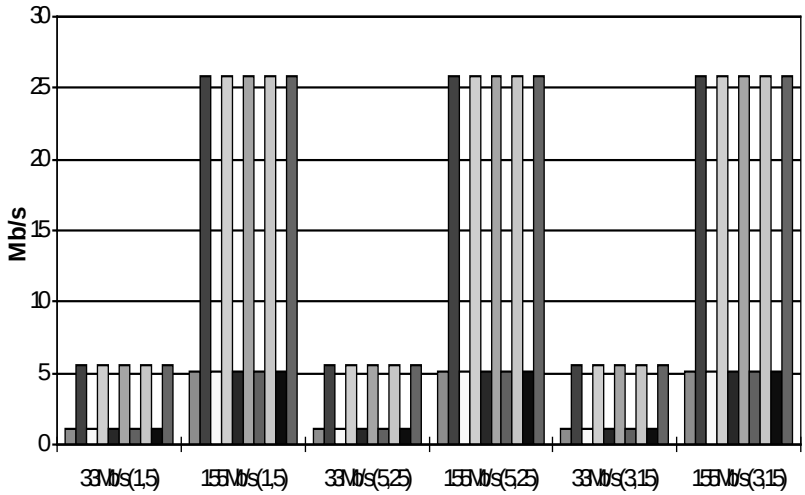


Figure 4: Bandwidth share for each of the 10 sources, for two different bottleneck bandwidths and 3 different expected profiles using the USD scheme

because their connection set up calls get rejected due to unavailable bandwidth. Thus, for the two-bit scheme with premium traffic, it is important that the aggregated expected bandwidth match the bottleneck bandwidth, else, some premium traffic may get denied bandwidth because the network resources have been over committed.

Finally, we simulated the USD scheme using weighted fair queuing in the network core. The bandwidth allocation is shown in Figure 4. In this case, we see that the relative bandwidth allocated to all the sources remain the same in all cases. In other words, the sources with high expected bandwidth (which presumably have paid more for such a privilege) get a higher relative share of the bandwidth than the others, independent of the bottleneck bandwidth or the absolute values of the expected bandwidth profiles.

Short Duration Traffic

A significant component of Internet traffic today is web traffic [12] that is characterized by short connection lifetimes. In order to study the effect of short connection lifetimes on the three different schemes, we ran simulations with the same settings as before except that the sources now sent only 200 packets (corresponds to a web transfer of 100Kbytes, a moderately sized web page with embedded graphics) to the destination. Figure 5 shows the performance for the 33Mb/s (1,5) (or matching bottleneck bandwidth) scenario and Figure 6 shows the performance for the 155Mb/s (1, 5) (or excess bottleneck bandwidth) scenario. The numbers for the 33Mb/s (1, 5) scenario show that on such shorter time scales, the bandwidth allocated by the RIO scheme is not commensurate with the expected bandwidth profiles. This is because the RED technique used in the routers for estimating average queue lengths (that determines when to drop a packet) and the Time Sliding Window technique used by a

profile meter to estimate the sending rate for a source (that determines whether a packet is in or out of profile) depend on long term averages. In the shorter term, the Time Sliding Window technique tends to overestimate the sending rate and mark more packets as out of profile than is necessary. We have performed some experiments to test how sensitive the calculated sending rates are to the length of the

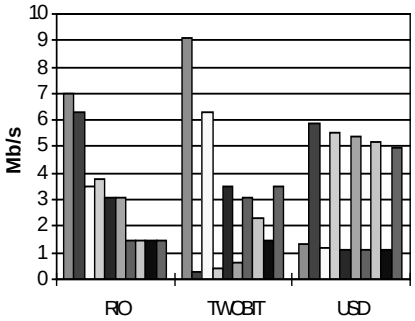


Figure 5: Effect of short-duration traffic with 33Mb/s (1,5)

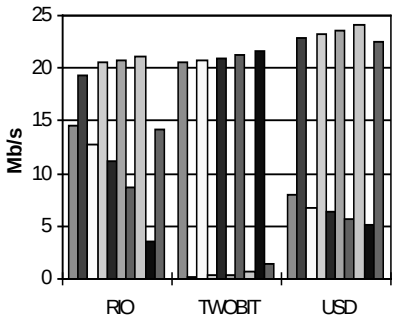


Figure 6 : Effect of short-duration traffic with 155Mb/s (1,5)

time window (the results are not shown here for lack of space), and found that the closer the time windows are to the round trip time, the more accurate the estimates are. This implies that it may be difficult to estimate sending rates accurately if the profile meter is de-coupled from the sender. In the 155Mb/s (1, 5) case, the bandwidth allocation is more reasonable in the sense that the higher bandwidth sources receive greater bandwidth than the lower bandwidth ones, except for source 10 that gets slightly lower bandwidth than source 1. However, the bandwidth allocations are not in proportion to the respective expected average bandwidths. We have seen earlier that the bandwidth at the bottleneck that is in excess of the aggregate expected bandwidths is more or less evenly allocated among all the sources – in this case, where all the traffic flows are of much shorter duration, RED is unable to enforce fair sharing of the excess bandwidth. The higher bandwidth sources, in the short term, are able to grab a higher share of the excess bandwidth since their congestion windows open up faster (due to lower drop rates at the routers that implement RIO) than those of the lower bandwidth sources.

The numbers for the two-bit scheme are interesting. The even sources (i.e., those with expected bandwidths of 5Mb/s), for the most part, show a lower bandwidth allocation compared to the odd sources (i.e. those with expected bandwidths of 1Mb/s) in both the scenarios. This anomalous behavior is caused by the way premium traffic is managed under the two-bit scheme: the monitor at the edge of the network generates premium tokens at a fixed rate, based on the expected bandwidth reserved by the call set up mechanism. Since TCP-Reno initially doubles the size of the congestion window every round trip time, the traffic generated is bursty and the rate can exceed the peak reserved bandwidth in the short term. As a result, the premium traffic, on reaching the monitor faces an absence of premium tokens and gets dropped. This happens in the initial stages of the opening up of the congestion window (before *ssthresh* is reached) and fast recovery does not occur because there are not enough lost packets. Thus, TCP times out multiple times and causes the congestion window to open up very slowly. In the long run, this effect is absent because over longer time scales, enough tokens are generated to absorb such bursts. The sources with the lower expected bandwidth are now able to make use of the excess bandwidth that is

freed up by a lack of premium traffic and therefore show higher bandwidth utilization. In the case of the USD scheme, the short term bandwidth allocations match the long term case: the figures in each of the two scenarios show that the relative bandwidth shares for the sources reflect the expected bandwidth profiles. Thus, for short-lived flows such as web traffic, where end-to-end mechanisms are harder to implement because they need at least one round trip time to react, USD is an attractive option for providing differentiated services.

Non-Responsive Sources

We used two different kinds of non-responsive sources, CBR sources that send at a fixed rate and malicious sources that flood the network by sending as fast as they can. None of these sources back off in the face of congestion. We first investigated the effects of CBR on each of the three schemes. The simulations were repeated using bottleneck links of capacity 33Mb/s and 155Mb/s. In the simulations, sources 1 and 2 were CBR sources transmitting at 5Mb/s and the other sources were TCP-Reno sources where the odd numbered sources had an expected bandwidth of 1Mb/s and the even numbered sources had an expected capacity of 5Mb/s. All CBR traffic was marked as out of profile.

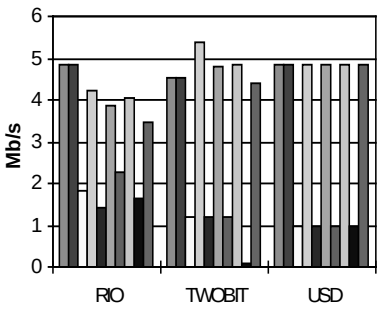


Figure 7: Effect of two non-responsive (CBR) sources with 33Mb/s (1,5)

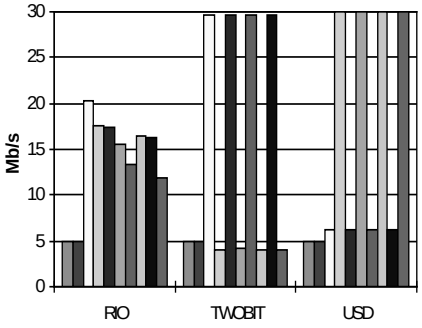


Figure 8: Effect of two non-responsive (CBR) sources with 155Mb/s (1,5)

Figure 7 shows the bandwidth allocation in each of the three schemes when the bottleneck bandwidth is 33Mb/s and Figure 8 shows the same when the bottleneck bandwidth is 155Mb/s. In both figures, the first two bars in each cluster are for the CBR sources and the rest are for the TCP-Reno sources. The numbers for the 33Mb/s (1,5) case indicate that in the case of RIO, the CBR sources are able to grab close to their expected bandwidth (i.e., about 5Mb/s) despite all of their traffic being marked out of profile. However, the TCP-Reno sources perform reasonably well, the sources with higher expected bandwidth show a bandwidth share of about 4Mb/s which is close to what the expected bandwidth profile is. The sources with a lower expected bandwidth show bandwidth shares that are higher than their expected bandwidth profiles, closer to about 2Mb/s rather than 1Mb/s. As before, we postulate that this can be remedied by an appropriate adjustment of the RIO parameters. In the 155Mb/s (1,5) case, the CBR sources get about 5Mb/s as in the previous case. All the other sources (that are all TCP-Reno

sources) receive bandwidths that are in excess of their expected average bandwidths, and it is clear that the excess bandwidth at the bottleneck is allocated fairly evenly among the TCP sources by the RED mechanism, as was the case for the bottleneck bandwidth mismatch scenario.

For the two-bit scheme, the bandwidth allocation is slightly skewed. As in the RIO scheme, the CBR sources get close to their expected bandwidth in both the scenarios, which is about 4.5Mb/s. The sources with the expected high bandwidth traffic, which is all marked as premium traffic also get close to 5Mb/s bandwidth (about 4.8Mb/s from the graphs). In the 33Mb/s (1, 5) case, 3 of the 4 sources that have a low expected bandwidth profile of 1Mb/s, get slightly higher bandwidth allocation than their expected profiles. All of this comes at the expense of the fourth low bandwidth source, which gets almost close to zero bandwidth. Thus, it appears that in the two-bit scheme, if there are non-cooperative, bandwidth hungry sources (even if all their packets are marked out of profile) other co-operative low bandwidth sources may get starved. The situation is slightly different when there is excess bandwidth at the bottleneck, i.e., the 155Mb/s (1,5) case. In this scenario, the premium bandwidth sources do not get any more than their expected bandwidths because premium traffic that is out of profile gets dropped. Consequently, the excess bandwidth at the bottleneck is shared evenly by the low bandwidth TCP-Reno sources, that now show a higher bandwidth allocation than the higher bandwidth sources. Finally, the numbers for the USD scheme show that the bandwidth allocations are commensurate with the expected bandwidth profiles, which implies that the USD scheme is effective in the presence of non co-operative sources.

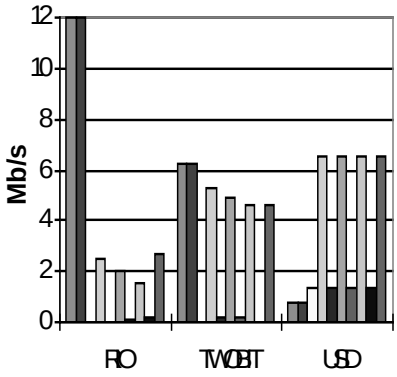


Figure 9: Effect of two malicious sources with 33Mb/s (1,5)

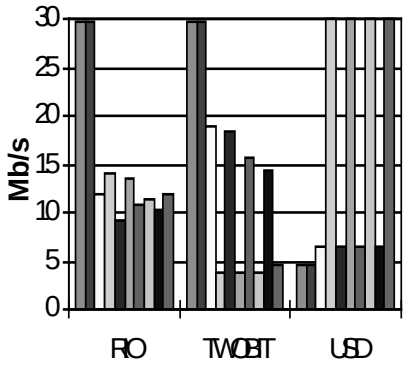


Figure 10: Effect of two malicious sources with 155Mb/s (1,5)

To investigate the effect of malicious sources, we used a bottleneck bandwidths of 33Mb/s and 155Mb/s. Sources 1 and 2 were set to be malicious sources that declare an expected bandwidth of 1Mb/s and then try to flood the network by sending as fast as they can. The other sources are TCP-Reno sources where the odd numbered sources have an expected bandwidth of 1Mb/s while the even numbered sources have an expected bandwidth of 5Mb/s. Note that these scenarios are different from the CBR cases described earlier, because in the previous cases, the sources did not try to flood the network in addition to not

responding to congestion. The purpose of simulating malicious sources here is to see if they have a negative effect on low bandwidth traffic by filling up the queues allocated for such traffic.

The results for the 33Mb/s (1,5) scenario are shown in Figure 9. In the RIO case, the low bandwidth sources get very little bandwidth – sources 3 and 5 get nothing and sources 7 and 9 get about 0.1Mb/s. The high bandwidth sources do not do well either, none of the sources are able to get more than 2.5Mb/s. The reason for this is that the packets for the malicious sources fills up the queues on the bottleneck routers very quickly and causes packets from other sources to be dropped more often. Consequently, the TCP windows do not open up to the extent required to get the expected average bandwidth. For the two-bit scheme, the low bandwidth sources get very little bandwidth (as in the RIO case) for the same reason and the bulk of the bandwidth goes to the malicious sources. However, in this case, the malicious sources grab about 6Mb/s each as opposed to 12Mb/s each in the RIO case. The reason is that the packets from the high bandwidth sources in the two-bit scheme are enqueued in a separate premium traffic queue. As a result, the high bandwidth sources are able to get close to their expected bandwidths of 5Mb/s while the low bandwidth sources get very little – sources 3, 5 and 7 get less than 0.2Mb/s and source 9 gets nothing. Thus, in both RIO and two-bit schemes, a malicious source can easily deprive low bandwidth sources of their fair share, but high bandwidth sources tend to do better in the two-bit scheme because premium traffic is enqueued in a separate higher priority queue. In case of the USD scheme, the malicious sources are penalized and actually receive slightly less than their average expected bandwidth (0.79Mb/s in place of 1Mb/s) while the other sources get slightly more than their expected bandwidth – the low bandwidth sources get 1.31Mb/s instead of 1Mb/s and the high bandwidth sources get 6.54Mb/s instead of 5Mb/s.

The results for the 155Mb/s (1,5) scenario are shown in Figure 10. In this scenario, the malicious sources for the RIO and the two-bit schemes are able to grab almost the full bandwidth of their outgoing links, which is set to 30Mb/s in the simulation. In the RIO case, the TCP-Reno sources are able to get more than their expected bandwidth profiles, but not in proportion to their expected bandwidths. Again, the reason is that the excess bandwidth (that is consumed only by out of profile packets) is more or less evenly distributed by the RED scheme among all the sources. In the two-bit case, the premium sources are able to get slightly less than their expected bandwidths (in the range 3.5 – 4Mb/s instead of 5Mb/s) which is somewhat surprising since premium traffic gets higher priority than best effort traffic. We conjecture that this is because the malicious sources constantly fill up the lower priority queues and therefore the routers spend more time in processing the lower priority queues than the higher priority ones. Consequently, the congestion windows for the higher bandwidth premium traffic does not open up fast enough to utilize the requisite expected bandwidth. The excess bottleneck bandwidth is more or less evenly allocated only among the low bandwidth sources because any premium traffic that is out of profile is dropped. This causes the low bandwidth sources to get a higher bandwidth allocation than the high bandwidth sources. Lastly, the USD scheme again penalizes the malicious sources and the allocation of bandwidth among the well behaved TCP-Reno sources is in proportion to their expected average bandwidths – the figures show that the low bandwidth sources get about 6Mb/s while the high bandwidth sources get about 30Mb/s.

The performance of the three schemes indicate that in the presence of non-responsive sources, low bandwidth traffic could get starved when the RIO or two-bit schemes are used. The two-bit scheme ensures that the high bandwidth sources (which send premium traffic) are able to utilize their average

profile bandwidth whereas the RIO scheme is not as effective in restricting non-responsive sources (especially malicious ones) from grabbing as much bandwidth as possible, even at the cost of high bandwidth sources. It is necessary to develop stricter monitoring mechanisms for the RIO and two-bit schemes in order to identify and penalize non responsive traffic flows (for example, the TCP-friendly approach outlined in [3]) for such schemes to be more effective. The USD scheme performs well in that it is able to effectively penalize non-responsive and malicious sources, but at the cost of increased complexity in the core routers.

Flows with High Expected Bandwidth

Another set of simulations was performed to investigate the effects of over allocation of bandwidth. In this case, we set the expected bandwidth of all the sources to 5Mb/s and used a 33Mb/s bottleneck link. The performance is shown in Figure 11. For the RIO scheme, the bandwidth allocation is fairly even, which seems to indicate that the RED scheme is nearly fair if all connections have the same weight. In the case of the two-bit scheme, only 6 of the 10 connections are able to get any bandwidth, the others connection requests are denied at set up time. The graphs indicate that the monitoring scheme for the premium packets is effective in ensuring that the connections that have been allowed stay within their profiles. The graphs for the USD scheme show that the bandwidth allocation is equal for each of the 10 connections, which is to be expected. The lesson here is that although RIO and USD schemes adapt well in the case when the network is under provisioned and all connections have equal weights, the two-bit scheme can only do so by denying service to some of the connections. In a setting where organizations or individuals have already paid an ISP for service guarantees, this may not be a feasible option. Hence in the two-bit case, networks perforce have to be over provisioned, a solution that may not be sustainable in the long run.

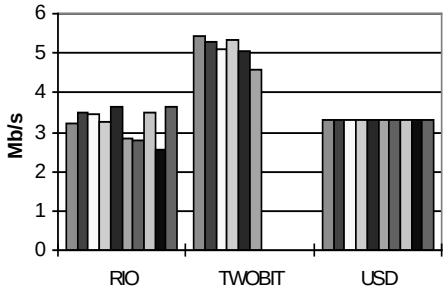


Figure 11: Effect of high bandwidth flows

4. SUMMARY

In this paper, we have evaluated the bandwidth allocation performance of the RIO, two-bit and USD schemes. We have shown that matching the network bottleneck bandwidth to the expected bandwidth profiles for sources is an important issue. While the RIO and two-bit schemes perform well when there is no such mismatch, it is not the case if the bottleneck bandwidth available is in excess of the aggregated expected bandwidths. This does not mean that the sources get less than their expected bandwidths, it is just that the bandwidth allocation in such a situation is not commensurate with the user expectations. On the other hand, the USD scheme with PFS is able to do a fair allocation of bandwidth that is in keeping with user expectations. Short term flows is another issue that needs to be addressed in view of the high percentage of web traffic on the Internet – the RIO and two-bit schemes need to be modified in some way to react more quickly to changes in network conditions to make this possible. The USD scheme is

able to achieve this since it does not rely on end-to-end congestion control schemes to enforce fairness. In the case of non-responsive sources, RIO performs better than the two-bit scheme when the non-responsive sources are CBR, however, malicious sources that flood the network are able to starve low bandwidth sources. The USD scheme is able to provide traffic isolation under such circumstances. For high bandwidth flows, all the schemes perform well, except that in the two-bit scheme case, an over-committed network may deprive some of the sources from getting any bandwidth, instead of allocating the available bandwidth evenly among all active sources. Finally, we note that independent work by Nandy et. al. [18] have confirmed our findings about RIO.

5. DISCUSSION

The differentiated services model embodies two fundamentally conflicting goals. On one hand, it needs to provide some quantitative guarantees to the users. On the other hand, it tries to avoid any signaling through the network. In another word, it tries to provide guarantee without knowing where the packets may go. Provisioning for such guarantees is a non-trivial problem. In the case of the RIO and the two-bit proposals, the assumption is that with admission control at the edges of the network, it is possible to limit the total traffic for one class. However, traffic flows inside the network tend to be dynamic with respect to bandwidth demands as well as source/destination pairs. In order to perform proper admission control, it is necessary for the edge network nodes to have complete knowledge about all bottlenecks inside the network.

Alternatively, the network has to provision enough bandwidth for the worst-case traffic distribution. However, over-provisioning can lead to low utilization of network resources. In a large network, network wide over-provisioning may not be feasible. The USD scheme tackles the problem by making the guarantee relative. Because the bandwidth is allocated according to the weights, the USD scheme can adopt automatically to different situation. Normally, the bandwidth lookup table consisting of user's network prefix and its associated share is distributed via a network management system as part of the provisioning.

We believe that some form of coarse level of signaling for aggregated flows will be necessary in order to provision a network in a more efficient fashion. For the purpose of the discussion, we call such a protocol as Trunk Provisioning Protocol, or TPP. TPP can be built with a revised RSVP or as a new light-weight protocol. The main purpose of TPP is to provide a mechanism for provisioning network trunks between ingress and egress routers around the network. As it only deals with aggregated flows, its operation is decoupled with individual microflows. The provisioning of trunks needs to be adjusted only when there are significant and persistent changes among the aggregated flows. Thus, the overheads of TPP is likely to be very low.

TPP can be designed to provide explicit provisioning or proportional provisioning. TPP can be used with USD for provisioning different shares along different paths. For example, if a customer has a subsidiary in another country, it can trade a higher share on the international path to that country with lower share on a path to a neighbor ISP. For RIO or two-bit schemes, it can be used to set up egress-specific guarantees. For example, instead of providing a fixed amount of premium bandwidth anywhere in the network, one can guarantee bandwidth between the access point of the customers to major egress

points of the network. Working in this mode, TPP is more like trunk reservation protocol. If there are not sufficient bandwidth on the specific paths, the provisioning will fail.

References

[1] D. Black, S. Blake, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An Architecture for Differentiated Services", Internet Draft, may 1998.

[2] D. Clark, S. Shenker, and L. Zhang, "Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism", In Proceedings of *ACM SIGCOMM'92*, Aug 1992.

[3] D. Clark and W. Fang, "Explicit Allocation of Best Effort Packet Delivery Service", Draft, 1998.

[4] D. Clark, J. Wroclawski, "An Approach to Service Allocation in the Internet," Internet Draft, July 1997.

[5] M. Degermark, A. brodnok, S. Carlsson, S. Pink, " Small Forwarding Tables for Fast Routing Lookups", In Proceedings of *ACM SIGCOMM'97*, Sept. 1997.

[6] S. Golestani, "A self-clocked fair queuing scheme for broadband applications," In Proceedings of *IEEE INFOCOM'94*, pp. 636-646, April, 1994.

[7] K. Nichols, V. Jacobson, L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet", Internet Draft, Nov 1997.

[8] A. K. Parekh, R. G. Gallager, "A generalized processor sharing approach to flow control - the single node case", In Proceedings of *IEEE INFOCOM'92*, May 1992.

[9] M. Shreedhar and G. Varghese, "Efficient Fair Queuing using Deficit Round Robin," In Proceedings of *ACM SIGCOMM'95*, Sept 1995.

[10] S. Keshav, "REAL 5.0 Manuals", http://www.cs.cornell.edu/skeshav/real/man.html, Aug. 1997.

[11] V. P. Kumar, T. V. Lakshman, and D. Stiliadis, "Beyond Best Effort: Router Architectures for the Differentiated Services of Tomorrow's Internet", IEEE Communications Magazine, May 1998.

[12] K. Thompson, G. Miller, R. Wilder, "Wide-Area Internet Traffic Patterns and Characteristics", IEEE Network, Nov 1997.

[13] Z. Wang, "User-Share Differentiation – scalable service allocation for the Internet", Internet Draft, Nov 1997.

[14] Z. Wang, "USD: Scalable bandwidth allocation for the Internet", Proceedings of HPN'98, Sept. 1998.

[15] L. Zheng, S. Deering, S. Estrin, S.Shenker, D. Zappala, "A New Resource Reservation Protocol", IEEE Network, Sept 1993.

[16] D. Lin and R. Morris. Dynamics of Random Early Detection. In Proceedings of *ACM SIGCOMM'97*, Cannes, France, September 1997.

[17] M. May, J-C. Bolot, A. J-Marie, and C. Diot. Simple Performance Models of Differentiated Services Schemes for the Internet. In Proceedings of *IEEE INFOCOM'99*, New York, NY, March 1999.

[18] B. Nandy, N. Seddigh, and P. Pieda. Diffserv's Assured Forwarding PHB: What Assurance does the Customer Have? In Proceedings of *NOSSDAV'99*, Basking Ridge, NJ, June 1999.