ON THE PERFORMANCE OF EARLY PACKET DISCARD

Maurizio Casoni and Jonathan S. Turner
Computer Science Department
Washington University
Campus Box 1045
St. Louis, MO 63130-4899
jst@cs.wustl.edu
Fax: +1-314-935-7302

Appears in *IEEE Journal on Selected Areas of Communications*, June 1997, Copyright held by IEEE.

Abstract

In a previous paper [3], one of the authors, gave a worst-case analysis for the Early Packet Discard (EPD) technique for maintaining packet integrity during overload in ATM switches. This analysis showed that to ensure 100% goodput during overload under worst-case conditions, requires a buffer with enough storage for one maximum length packet from every active virtual circuit. This paper refines that analysis, using assumptions that are closer to what we expect to see in practice and examines how EPD performs when the buffer is not large enough to achieve 100% goodput. We show that 100% goodput can be achieved with substantially smaller buffers than predicted by the worst-case analysis, although the required buffer space can be significant when the link speed is substantially higher than the rate of the individual virtual circuits. We also show that high goodputs can be achieved with more modest buffer sizes, but that EPD exhibits anomalies with respect to buffer capacity, in that there are situations in which increasing the amount of buffering can cause the goodput to *decrease*. These results are validated by comparison with simulation.

[0]This work was supported by the ARPA Computing Systems Technology Office, NEC, NTT, Samsung, Southwestern Bell, Sun Microsystems, Tektronix and University of Bologna (I).

1. Introduction

ATM networks carry many different types of traffic with diverse bandwidth requirements. At one extreme, there are continuous applications which transmit steadily at fixed rates. At the other extreme, there are highly bursty applications that transmit large blocks of information at high peak rates but whose long-term average rates are relatively low. The existence of bursty traffic means that one must either accept low network utilization or the possibility of overload periods during which the traffic sent to some network links exceeds their capacity. In addition, the fact that end-to-end transport protocols send information in packets containing many ATM cells makes the impact of overload periods worse because the loss of a single cell can lead to the loss and retransmission of the entire transport-level packet. This means that during overload periods, ATM networks can experience congestion collapse with the throughput dropping to zero as the load increases.

It is clear that overload periods will occur in ATM network from time-to-time and so it is important to improve their performance during these overload periods. In this paper we focus on a mechanism that accepts that cell loss will occur, but attempts to ensure that the available network capacity is effectively utilized by preserving the integrity of transport level packets during overload periods.

Early packet discard (EPD) [1] is an ATM buffer management technique designed ensure high end-to-end throughputs for bursty data applications during overload periods. EPD achieves this by observing the packet-level structure of each cell stream and discarding entire packets, when overload makes it necessary to do so. EPD is only one of several mechanisms that have been proposed for handling congestion in ATM networks. In particular, rate-based flow control has now been standardized by the ATM Forum to manage congestion for Available Bit Rate (ABR) traffic streams [2]. While the use of rate-based flow control reduces the need for techniques like EPD, it does not eliminate them, since buffers using ABR flow control can still experience overloads during transient periods, before rate adjustment mechanisms can react to traffic changes, and since these mechanisms will not be applied to Unspecified Bit Rate (UBR) streams. In addition, the expense of explicit rate control (in terms of memory and control complexity) is likely to limit its application to bottleneck points where bandwidth is expensive and scarce (such as at the interface between a campus network and a wide-area network).

A previous paper [3] analyzes several packet level discard mechanisms including EPD. That analysis is based on worst-case assumptions that over-state the amount of buffering required for high effective throughputs. In this paper, we refine that analysis, using more realistic assumptions and use this to determine the amount of buffering required to achieve 100% goodput (here, we define goodput to be the fraction of the link's capacity that is used to carry complete transport level packets). As an example of our results, the worst-case analysis in [3] requires a 120 Kbyte buffer to ensure 100% goodput, for an OC-3 link experiencing a five-to-one overload as a result of 15 virtual circuits sending 8 Kbyte packets at a rate of 50 Mb/s each. The new analysis shows that a 20 Kbyte buffer is sufficient to ensure 100% goodput in this case. At the same time, our analysis also shows that large buffers are needed for 100% goodput if the individual virtual circuits contributing to the overload are sending data at low rates. So for example, if the five-to-one overload in the example is the result of traffic from virtual circuits sending at a rate of 1 Mb/s each, then a buffer of almost 1 Mbyte is needed to ensure 100% goodput. (Of course, when individual virtual circuits have such small peak rates, the probability of such a large overload is extremely small.)

We also study how EPD behaves when the buffer is not large enough to ensure 100% goodput.

We find that different choices of the buffer size and threshold placement lead to very different behaviors. In general, EPD can provide reasonably high goodput even when the buffers are relatively small, but the behavior is not always intuitively obvious. In particular, we show that increasing buffer sizes for a fixed offered load can result in reducing the goodput, rather than increasing it.

As discussed more fully in section 2, the analysis is based on an idealized assumption about the "phase relationships" of packets on different virtual circuits. While the analysis is not exact, comparisons to simulation results (included below) show that it is surprisingly accurate. The principal virtue of our approach is its simplicity and the qualitative insight it provides into the behavior of EPD. The analysis naturally divides into two distinct cases. The first applies when the offered load is less than twice the link rate, and the second when the offered load is more than twice the link rate. The analyses for the two cases are given in sections 3 and 4, respectively. In section 5, we give numerical results.

2. Definitions and Assumptions

In EPD, whenever a virtual circuit begins the transmission of a new packet, the queue controller decides whether to accept the packet or to discard it. In particular, if the number of cells in the queue exceeds a specified threshold, then the packet is not propagated into the queue. If the number of cells is below the threshold, the packet is propagated. In addition, if any cell of the packet must be discarded due to queue overflow, the remainder of that packet is discarded.

We consider a homogeneous situation in which r virtual circuits transmit data continually at a normalized rate of λ (that is, λ is the fraction of the link bandwidth required by a single virtual circuit), and define the *overbooking ratio* as $r\lambda$. For an overloaded link $r\lambda > 1$. We also assume that all packets contain ℓ cells and let $k = \lfloor 1/\lambda \rfloor$ be the maximum number of virtual circuits that the link can handle without loss. For simplicity, we will also assume that $1/\lambda$ is an integer; although extension to non-integral values is straightforward, it adds little new information and obscures the key issues. We let B denote the number of cells the buffer can contain, and let b denote the threshold level. The values of b and $B - b$ largely determine the goodput that it is possible to achieve on a link that has a buffer managed by EPD. If both values are large enough, we have 100% goodput. If one or both is too small, we may have buffer *overflow* (that is, the buffer becomes full, causing a cell to be discarded), and/or buffer *underflow* (the buffer becomes empty leaving some of the link bandwidth unused).

We define a given virtual circuit to be *active* if cells from that virtual circuit are being placed in the queue on arrival. Similarly, we define a virtual circuit to be *inactive* if its cells are being discarded. We assume that the flow of cells on a given virtual circuit is smooth, ignoring the effects of jitter caused by variable delays upstream of the buffer under consideration. Under this assumption, the buffer level rises or falls in a predictable way, depending only on the number of active virtual circuits.

If we observe the number of cells in a queue managed using EPD as a function of time, we will observe a cyclic behavior in which the number of cells in the queue rises above the threshold, then as various virtual circuits become inactive (either because they've completed their packets or had cells discarded), the number of cells stops increasing and drops back down. When it falls below threshold, and new packets start arriving at the queue, the buffer level stops falling and begins to rise again.

Our prior analysis of early packet discard was based on the worst-case assumption that just before every threshold crossing, the first cell of a packet arrived on every virtual circuit, meaning that the next packet boundary was delayed as long as possible after the threshold crossing, leading to wide excursions around the threshold level. While such worst-case synchronization of packet boundaries is possible, it is hardly likely, and would certainly not be expected to persist over an extended period of time. Hence, in this paper we analyze the queue behavior under another simple assumption that more closely reflects what might be expected to happen in practice. In particular, we consider the case in which packet boundaries on different virtual circuits are offset from one another by an equal amount, leading to the occurrence of a packet boundary every $\ell/r\lambda$ cell times. This *even offset* assumption is clearly an idealization. In truth, different virtual circuits will have random phase relationships relative to one another, leading to uneven spacing of packet boundaries. Consequently, the even offset assumption is a bit optimistic, but when the number of virtual circuits is moderately large, this simple model predicts the behavior of EPD with surprising accuracy.

3. Performance With Small Overbooking Ratios

In this section we analyze the performance of EPD when $r\lambda \leq 2$. There are various different behaviors that we can observe based on the values of b and $B - b$. Figure 1 provides a map of the different regions of operation. In this section we will explore these regions and determine where the boundaries are. For now, simply notice that when b (on the horizontal axis) and $B - b$ (on the vertical axis) are large enough, a goodput of 1 (100%) is achieved. If we reduce $B - b$ sufficiently, the buffer will overflow. Alternatively, if we leave $B-b$ large but reduce b, the buffer will underflow. If we reduce both sufficiently, we can experience both overflow and underflow. The numbers within various regions of the map are the section numbers where that portion of the map is analyzed.

3.1. Buffering Needed for 100% Goodput

We start by exploring the upper right region of the goodput map. In particular, we want to determine for what values of b and $B - b$ the buffer will never overflow or underflow. To do this, we simply assume that b and $B - b$ are both very large and determine the maximum excursion above and below the threshold. When $r\lambda \leq 2$, the maximum excursion around the threshold occurs when, at each upward threshold crossing, all r virtual circuits are active. Packet boundaries occur at regular intervals following the threshold crossing. For simplicity, we assume that threshold crossings fall half-way between successive packet boundaries, meaning that packet boundaries will follow threshold crossings by $\ell/2r\lambda$, $3\ell/2r\lambda$, $5\ell/2r\lambda$ and so forth. Figure 2 illustrates how the buffer oscillates around the threshold when $k = 4$ and $r = 6$. The lines at the top indicate the packets flowing on the six virtual circuits. Heavy lines indicate packets that are accepted for transmission, while blank spaces indicate packets discarded by the buffer controller.

At each packet boundary following an upward threshold crossing, the rate at which the buffer level rises gets smaller. As shown in Figure 2, the slope of the curve is $r\lambda - 1$ at the time the threshold is crossed, and each successive packet boundary decreases the slope by λ. When there are exactly k active virtual circuits, the slope is zero and there is no further rise in the buffer level. From this discussion, we can see that the maximum excursion above the threshold is

$$\Delta U = \frac{1}{2}\frac{\ell}{r\lambda}(r\lambda - 1) + \frac{\ell}{r\lambda}\sum_{i=1}^{r-k-1}(r-k-i)\lambda = (1/2)(r-k)^2(\ell/r)$$

Note that this implies $\Delta U \le (k/4)\ell$. By the worst-case analysis given in [3], the maximum excursion above the threshold can be $(r-k)\ell$, which is always at least four times larger than the "even offset" analysis predicts.

To determine the excursion below the threshold, we need to know the number of active virtual circuits at the downward threshold crossing. This can be determined by inspection from Figure 2 and noting that the even offset assumption on the packet boundaries implies that at the downward threshold crossing, the slope of the buffer occupancy curve has the same absolute magnitude as at the upward threshold crossing. That is, the slope is $1 - r\lambda$ and the number of active virtual circuits is $2k - r$. After crossing the threshold, the buffer level continues to drop until all $2k - r$ active virtual circuits complete their packets and new packets start on the inactive virtual circuits. At this point, packet boundaries occur on the inactive virtual circuits at intervals of $\ell/r\lambda$, causing the slope of the buffer occupancy curve to increase, reaching zero when exactly k virtual circuits are active. From the threshold crossing to the first packet boundary at which the slope changes, the buffer level drops by

$$(2k - r + 1/2)(r\lambda - 1)(\ell/r\lambda) = (2k - r + 1/2)(r - k)(\ell/r)$$

From the point where the slope first changes until the buffer reaches its lowest point, the drop in the buffer level is

$$\frac{\ell}{r\lambda}\sum_{i=1}^{r-k-1} i\lambda = (1/2)(r-k-1)(r-k)(\ell/r)$$

So,

$$\Delta u = (1/2)(3k - r)(r - k)(\ell/r)$$

Note that as r increases, Δu rises initially and then falls, reaching a maximum when $r = \sqrt{3}k$. The maximum value of Δu is thus $(2 - \sqrt{3})k\ell \approx .27k\ell$. The worst-case analysis in [3] gives a maximum excursion below threshold of $k\ell$. The total buffer range is thus,

$$\Delta U + \Delta u = k(r-k)(\ell/r)$$

and since $k < r < 2k$, the buffer range is $\le \frac{k\ell}{2}$. It's interesting to note that the time duration of a cycle is exactly $2\ell/\lambda$ which is the time it takes for exactly two packets to arrive. This means that under our assumptions, the same virtual circuits have packets discarded on every cycle. So the lucky virtual circuits (the top and bottom ones in Figure 2) are able to send every packet, while the unlucky ones are able to transmit only half their packets. While in practice, we would expect differences in packet lengths and virtual circuit rates to prevent this pattern from persisting over long periods of time, it can be an issue in at least some situations.

3.2. Buffer Overflow Without Underflow

If we reduce $B - b$ sufficiently (specifically, if we let $B - b < (1/2)(r-k)^2(\ell/r)$), the buffer can overflow, causing cells to be discarded from some active virtual circuits and causing the goodput to drop below 100%. The buffer occupancy still oscillates around the threshold level, but now the top end is constrained by the limited buffer size. The goodput is determined by two things; the number of packets that are able to complete during a cycle and the time duration of the cycle. Referring to Figure 3, let p be the number of active virtual circuits at the time of an upward threshold crossing. To obtain the most conservative estimate for the goodput, we assume that the first $r - p$ packet boundaries following the threshold crossing belong to inactive virtual circuits, meaning that the buffer occupancy curve maintains a constant slope initially. After $r-p$ packet boundaries have occurred, the next packet boundaries belong to the p virtual circuits that were active at the threshold crossing. Under our modeling assumptions, exactly k of the virtual circuits that are still active when the buffer overflows, will remain active and complete their packets. The goodput is affected by which virtual circuits are selected for discarding. Again, to obtain the most conservative estimate of the goodput, we assume that the virtual circuits selected for discarding are those whose packet boundaries come first after the buffer fills. This delays the falling of the buffer level as long as possible, maximizing the period.

Note that the buffer may overflow while some of the $r-p$ initially inactive virtual circuits are still completing their packets. If $r-p>0$, then in this case, exactly k packets are completed in each cycle. If the buffer overflow occurs only after the $r-p$ initially inactive virtual circuits complete their packets, then we obtain one additional packet per cycle for each initially active virtual circuit that is able to complete its packet before the buffer fills.

As the k virtual circuits that remain active after the buffer fills, come to the ends of their packets, the buffer level starts to drop with an initial slope of $-\lambda$, then -2λ and so forth. During the time period between the first two packet boundaries belonging to these k virtual circuits, the buffer drops by ℓ/r. During the time period between the next two packet boundaries, it drops $2\ell/r$, and so forth. If we let q be the number of active virtual circuits at the time of the next downward threshold crossing, we find

$$(1+2+\cdots+(k-q-1))(\ell/r) \leq B-b < (1+2+\cdots+(k-q))(\ell/r)$$

(The validity of this observation depends on the fact that because the buffer does overflow, $B-b$ can be no larger than $(1+2+\cdots+k)(\ell/r)$.) If we define $n(z)$ to be the integer i that satisfies

$$1+2+\cdots+(i-1) \leq z < 1+2+\cdots+i$$

we can write

$$q = k - n\left(\frac{\Delta U}{\ell/r}\right)$$

where ΔU is the maximum excursion above threshold and is equal to $B-b$ in this case. If we define Δy^- to be the amount that the buffer level drops between the downward threshold crossing and the first packet boundary following the downward threshold crossing, then

$$\Delta y^- = (1/2)(k-q)(k-q+1)(\ell/r) - \Delta U$$

Following the threshold crossing, the first q packet boundaries belong to virtual circuits that were active at the threshold crossing, meaning that the slope stays constant initially, until the packets belonging to the active virtual circuits start ending. The buffer level reaches its lowest point when exactly k virtual circuits remain active. Thus, the maximum excursion below threshold is given by the following equation.

$$\Delta u = \Delta y^- + (k-q)q(\ell/r) + (1/2)(k-q-1)(k-q)(\ell/r) = \Delta y^- + (1/2)(k-q)(k+q-1)(\ell/r)$$

This expression for Δu defines the "left" boundary of the overflow only region in the goodput map of Figure 1. We now want to determine the shape of the buffer occupancy curve during the period in which it rises from its lowest point until the next threshold crossing. There are two different cases that can occur, depending on the magnitude of Δu. First, if $\Delta u \leq (1/2)(r-k)(r-k-1)(\ell/r)$ then

$$p = k + n\left(\frac{\Delta u}{\ell/r}\right)$$

(Note that the value of p in each cycle does not depend on its value in previous cycles, hence there is no ambiguity here.) In addition, if we define Δy^+ to be the amount that the buffer rises between the upward threshold crossing and the next packet boundary following the upward threshold crossing, we find

$$\Delta y^+ = (1/2)(p-k)(p-k+1)(\ell/r) - \Delta u$$

We next want to determine the time duration of the cycle in the case when $\Delta u \leq (1/2)(r-k)(r-k-1)(\ell/r)$. First, note that under our assumptions, the virtual circuit whose packet starts immediately after an upward threshold crossing is also the virtual circuit whose next packet start causes the first increase in slope following the subsequent downward threshold crossing. Thus, the time between these two packet boundaries is ℓ/λ (see Figure 3). Thus, the time duration of a cycle is given by

$$T = \frac{\ell}{\lambda} + (k-q)\frac{\ell}{r\lambda} + n\left(\frac{\Delta u}{\ell/r}\right)\frac{\ell}{r\lambda} = (\ell/\lambda)(1+(p-q)/r)$$

Now, let p_{ok} denote the number of packets that are able to complete during a cycle. As mentioned earlier, if the buffer overflows before any of the virtual circuits which are active at the upward threshold crossing are able to complete their packets, exactly k packets are completed per cycle. That is, if $\Delta U < \Delta y^+ + (r-p)(p-k)(\ell/r)$, then $p_{ok} = k$. If $\Delta U \geq \Delta y^+ + (r-p)(p-k)(\ell/r)$ let w denote the number of packets belonging to active virtual circuits that complete before the buffer overflows and note that

$$\begin{aligned} [((p-k)-1)+\cdots+((p-k)-(w-1))](\ell/r) &\leq \Delta U - (\Delta y^+ + (r-p)(p-k)(\ell/r)) \\ &< [((p-k)-1)+\cdots+((p-k)-w)](\ell/r) \end{aligned}$$

Also, note that $p_{ok} = k + w$. Define $\overline{n}_x(z)$ to be the integer i that satisfies

$$(x-1)+\cdots+(x-(i-1)) \leq z < (x-1)+\cdots+(x-i)$$

for all x and z that satisfy $1 < x-1 \leq z < 1+\cdots+(x-1)$. For $z \leq 0$, let $\overline{n}_x(z) = 0$ and for $0 \leq z < x-1$, let $\overline{n}_x(z) = 1$. With this definition, we can write

$$p_{ok} = k + \overline{n}_{p-k}\left(\frac{\Delta U - (\Delta y^+ + (r-p)(p-k)(\ell/r))}{\ell/r}\right)$$

Note that the definition of $\overline{n}_x(z)$ allows us to use this expression for p_{ok} even when $\Delta U < \Delta y^+ + (r-p)(p-k)(\ell/r)$.

We now turn to the case where $\Delta u > (1/2)(r-k)(r-k-1)(\ell/r)$. In this case, when the buffer reaches its lowest point, there are $r-k$ virtual circuits that remain inactive, and all $r-k$ become active before the buffer rises above threshold. Thus, $p = r$ in this case. To determine Δy^+, note that after all virtual circuits become active, the buffer level rises in a straight line, increasing by $(r-k)(\ell/r)$ between each successive pair of packet boundaries. So Δy^+ is between 0 and $(r-k)(\ell/r)$, and is given by

$$\Delta y^+ = (r-k)(\ell/r) - \left[\left(\Delta u - \frac{1}{2}(r-k)(r-k-1)(\ell/r)\right) \bmod (r-k)(\ell/r)\right]$$

where $x \bmod y$ is the remainder when x is divided by y. The duration of the cycle in this case is given by

$$\begin{aligned} T &= \frac{\ell}{\lambda} + (k-q)\frac{\ell}{r\lambda} + (r-k)\frac{\ell}{r\lambda} + \left\lceil \frac{\Delta u - (1/2)(r-k-1)(r-k)(\ell/r)}{(r-k)(\ell/r)} \right\rceil (\ell/r\lambda) \\ &= (\ell/\lambda)(1+(p-q)/r) + \left\lceil \frac{\Delta u - (1/2)(r-k-1)(r-k)(\ell/r)}{(r-k)(\ell/r)} \right\rceil (\ell/r\lambda) \end{aligned}$$

The last term is the length of the time period from when the last virtual circuit becomes active until the first packet boundary following the upward threshold crossing. When $\Delta u > (1/2)(r-k)(r-k-1)(\ell/r)$, some packets are able to complete during the period when the buffer is below threshold. In particular, during the period when all virtual circuits are active, and the buffer level is rising but below threshold, each packet boundary corresponds marks the end of a packet belonging to an active virtual circuit. Thus,

$$p_{ok} = k + \overline{n}_{p-k}\left(\frac{\Delta U - (\Delta y^{+} + (r-p)(p-k)(\ell/r))}{\ell/r}\right) + \left\lceil\frac{\Delta u - (1/2)(r-k-1)(r-k)(\ell/r)}{(r-k)(\ell/r)}\right\rceil$$

Now, the goodput is the ratio of the number of packets that complete in a cycle, to the maximum number of packets that could complete in a cycle if no packets were discarded. Since, in a time period of length T (where the time unit is the time to send a single cell), we can send T/ℓ packets on the link, the goodput is given by

$$goodput = \frac{p_{ok}}{T/\ell}$$

Notice that as $B-b$ is decreased, q increases, causing Δu to decrease. That is, the excursion below threshold decreases as the amount of buffering above the threshold decreases. Later in the paper we will plot the results of the analysis and we will note that the goodput exhibits anomalous behavior with respect to the amount of buffering and we'll observe that while decreasing $B-b$ from the point at which no loss occurs, initially leads to lower goodput, as we continue to decrease it, we reach a minimum goodput after which further decreases in $B-b$ lead to increasing goodput. The explanation for this behavior is that once $B-b$ becomes small enough so that $p_{ok}=k$, further decreases in $B-b$ do not cause any further reduction in p_{ok} but do reduce T, causing the goodput to rise. We shall observe similar anomalies in other cases as well.

Let's consider an example to illustrate the use of the equations. Suppose $k=4$, $r=8$ and $B-b=3.5(\ell/r)$. Applying the various equations, we find $q=1$, $\Delta y^{-}=2.5(\ell/r)$, and $\Delta u = 8.5(\ell/r)$. Since $\Delta u \leq (1/2)(4)(5)(\ell/r)$, we apply the equations for the "small" Δu case, giving $p=8$, $\Delta y^{+}=1.5(\ell/r)$, $T=4(15/8)\ell$ and $p_{ok}=5$. This yields a goodput of 2/3, as illustrated in Figure 3.

3.3. Buffer Underflow without Overflow

We now consider what happens when the threshold level is small enough to cause the buffer to underflow, but the buffer is large enough so that overflow does not occur. Note that the maximum excursion below the threshold $\Delta u = b$. To obtain the most conservative goodput assumption, we assume that the first $r-p$ packet boundaries immediately following an upward threshold crossing belong to inactive virtual circuits. This yields

$$\begin{aligned}\Delta U &= \Delta y^{+} + (r-p)(p-k)(\ell/r) + (1/2)(p-k-1)(p-k)(\ell/r)\\ &= \Delta y^{+} + (1/2)(2r-p-k-1)(p-k)(\ell/r)\end{aligned}$$

(This expression for ΔU, together with the equations for p and Δy^{+} below, define the "bottom" boundary of the overflow only region of the goodput map in Figure 1.) Using this and the fact that $r \leq 2k$, it's easy to show that $\Delta U < (1/2)k(k+1)(\ell/r)$. Consequently, as the buffer level falls from it's maximum value down to the threshold, every packet boundary belongs to an active virtual circuit, meaning that the slope changes at each packet boundary. Consequently,

$$\begin{aligned} q &= k - n\left(\frac{\Delta U}{\ell/r}\right) \\ \Delta y^- &= (1/2)(k-q)(k-q+1))(\ell/r) - \Delta U \end{aligned}$$

Now, consider the situation following the downward threshold crossing. To obtain the most conservative estimate of the goodput, we assume that the first q packet boundaries following the threshold crossing belong to the initially active virtual circuits, meaning that the buffer level will start to rise only after $k+1$ packet boundaries have occurred during the below threshold interval.

If $\Delta u < (1/2)(r-k-1)(r-k)(\ell/r)$ not all the virtual circuits become active before the upward threshold crossing and we find that

$$\begin{aligned} p &= k + n\left(\frac{\Delta u}{\ell/r}\right) \\ \Delta y^+ &= (1/2)(p-k)(p-k+1)(\ell/r) - \Delta u \\ T &= \frac{\ell}{\lambda} + (p-k)\frac{\ell}{r\lambda} + n\left(\frac{\Delta u}{\ell/r}\right)\frac{\ell}{r\lambda} = (\ell/\lambda)(1+(p-q)/r) \\ p_{ok} &= k + n\left(\frac{\Delta u}{\ell/r}\right) = p \end{aligned}$$

The equation for p_{ok} is based on the observation that since there is no overflow, every packet boundary during the below threshold period corresponds to a packet that is successfully sent on the link.

If $\Delta u \geq (1/2)(r-k-1)(r-k)(\ell/r)$, all the virtual circuits become active before the upward threshold crossing and consequently

$$\begin{aligned} p &= r \\ \Delta y^+ &= (r-k)(\ell/r) - [(\Delta u - (1/2)(r-k-1)(r-k)(\ell/r)) \bmod (r-k)(\ell/r)] \\ T &= (\ell/\lambda)(1+(p-q)/r) + \left\lceil\frac{\Delta u - (1/2)(r-k-1)(r-k)(\ell/r)}{(r-k)(\ell/r)}\right\rceil(\ell/r\lambda) \\ p_{ok} &= p + \left\lceil\frac{\Delta u - (1/2)(r-k-1)(r-k)(\ell/r)}{(r-k)(\ell/r)}\right\rceil \end{aligned}$$

Finally, the goodput is again given by $p_{ok}/(T/\ell)$.

3.4. Buffer Underflow and Overflow

We now consider the case where both b and $B-b$ are both small enough that overflow and underflow occur together. We first note that $\Delta u = b$ and $\Delta U = B - b$ and that $\Delta U \leq (1/2)k(k+1)(\ell/r)$ since otherwise we could not have overflow. Consequently, as the buffer level drops down to the threshold, each packet boundary belongs to an active virtual circuit, giving

$$\begin{aligned} q &= k - n\left(\frac{\Delta U}{\ell/r}\right) \\ \Delta y^- &= (1/2)(k-q)(k-q+1)(\ell/r) - \Delta U \end{aligned}$$

If $\Delta u < (1/2)(r-k-1)(r-k)$ not all the virtual circuits become active before the upward threshold crossing and we find that

$$\begin{aligned}
p &= k + n\left(\frac{\Delta u}{\ell/r}\right)\\
\Delta y^{+} &= (1/2)(p-k)(p-k+1)(\ell/r) - \Delta u\\
T &= (\ell/\lambda)(1+(p-q)/r)\\
p_{ok} &= k + \overline{n}_{p-k}\left(\frac{\Delta U - (\Delta y^{+} + (r-p)(p-k)(\ell/r))}{\ell/r}\right)
\end{aligned}$$

If $\Delta u \geq (1/2)(r-k-1)(r-k)(\ell/r)$ all the virtual circuits become active before the upward threshold crossing, giving

$$\begin{aligned}
p &= r\\
\Delta y^{+} &= (r-k)(\ell/r) - \left[\left(\Delta u - \frac{1}{2}(r-k)(r-k-1)(\ell/r)\right) \bmod (r-k)(\ell/r)\right]\\
T &= (\ell/\lambda)(1+(p-q)/r) + \left\lceil\frac{\Delta u - (1/2)(r-k-1)(r-k)(\ell/r)}{(r-k)(\ell/r)}\right\rceil \frac{\ell}{r\lambda}\\
p_{ok} &= k + \overline{n}_{p-k}\left(\frac{\Delta U - (\Delta y^{+} + (r-p)(p-k)(\ell/r))}{\ell/r}\right) + \left\lceil\frac{\Delta u - (1/2)(r-k-1)(r-k)(\ell/r)}{(r-k)(l/r)}\right\rceil
\end{aligned}$$

4. Performance With Large Overbooking Ratios

We now consider the case of overbooking ratios greater than two. As in the last section, we will first determine the maximum excursion around the threshold when buffers are large enough to avoid overflow and underflow and then consider what happens when we decrease the amount of buffering above and below the threshold. Figure 4 is a map of the various performance regions.

4.1. Buffering Needed for 100% Goodput

When the overbooking ratio is larger than two, the queue behavior changes qualitatively. In particular, the largest buffer excursion occurs in this case when no virtual circuits remain active at a downward threshold crossing. Thus, following such a threshold crossing, the slope of the buffer occupancy curve is initially -1, and at successive packet boundaries, the slope increases until, when k virtual circuits have become active, the slope is zero and the buffer level drops no further. We can determine the excursion below the threshold using an analysis much like that in the last section. We find

$$\Delta u = \frac{1}{2}\frac{\ell}{r\lambda} + \frac{\ell}{r\lambda}\sum_{i=1}^{k-1}(1-i\lambda) = (1/2)k^2(\ell/r)$$

Note that when $r \to \infty$, $\Delta u \to 0$ so that Δu is in the range $\left[0, \frac{k\ell}{4}\right]$.

After the buffer level reaches its lowest level, it starts to rise as packet boundaries occur on additional virtual circuits. The time interval during which the buffer occupancy is below the threshold is $2k(\ell/r\lambda)$. Packet boundaries occurring after the buffer rises above the threshold do not immediately cause new virtual circuits to become active, so the slope of the buffer occupancy curve remains constant until the first virtual circuit that became active following the previous downward threshold crossing comes to an end of its packet. This occurs ℓ/λ cell times after it began. At this point the slope changes to $(2k-1)\lambda - 1$ and as additional packet boundaries occur it drops to zero. Thus, the total excursion above the threshold is:

$$\Delta U = \frac{\ell}{\lambda} - (2k - 1/2)\frac{\ell}{r\lambda} + \frac{\ell}{r\lambda}\sum_{i=1}^{k-1}[(2k-i)\lambda - 1] = (1/2)(2r-3k)k(\ell/r)$$

When $r \to \infty$, $\Delta U \to k\ell$ so ΔU lies within the range $\left[\frac{k\ell}{4}, k\ell\right]$. Adding Δu and ΔU gives a total buffer range of $(r-k)k(\ell/r)$. The worst-case analysis of [3] gives a maximum buffer range of $r\ell$.

4.2. Buffer Overflow Without Underflow

When we have large overbooking ratios, we observe somewhat different behavior than with small overbooking ratios. In particular, we find that with large overbooking ratios, the portion of the cycle during which the buffer is rising from its low point to the threshold contains packet boundaries only for inactive virtual circuits, meaning that each packet boundary during this part of the cycle is accompanied by an increase in the slope. On the other hand, during the part of the cycle in which the buffer level is dropping from its maximum to the threshold, we can have all active virtual circuits turn off and subsequent packet boundaries then belong to virtual circuits that are already inactive, meaning that the buffer level drops with constant slope during this period.

In the case of overload only, we have $\Delta U = B - b$. If $\Delta U < (1/2)(k-1)k(\ell/r)$ then each packet boundary during the part of the cycle where the buffer level is dropping to the threshold belongs to an active virtual circuit. Consequently

$$\begin{aligned} q &= k - n\left(\frac{\Delta U}{\ell/r}\right) \\ \Delta y^- &= (1/2)(k-q)(k-q+1)(\ell/r) - \Delta U \\ T &= (\ell/\lambda) + (k-q)(\ell/r\lambda) + n\left(\frac{\Delta u}{\ell/r}\right)(\ell/r\lambda) = (1+(p-q)/r)(\ell/r) \end{aligned}$$

If, on the other hand $\Delta U > (1/2)(k-1)k(\ell/r)$ then we can have packet boundaries belonging to inactive virtual circuits during the period when the buffer level is dropping toward the threshold, giving

$$\begin{aligned} q &= 0 \\ \Delta y^- &= k(\ell/r) - [(\Delta U - (1/2)(k-1)k(\ell/r)) \bmod k(\ell/r)] \\ T &= (1+(p-q)/r)(\ell/\lambda) + \left\lceil \frac{\Delta U - (1/2)(k-1)k(\ell/r)}{k(\ell/r)} \right\rceil (\ell/r\lambda) \end{aligned}$$

In either case, we have

$$\begin{aligned} \Delta u &= \Delta y^- + (k-q)q(\ell/r) + (1/2)(k-q-1)(k-q)(\ell/r) \\ &= \Delta y^- + (1/2)(k-q)(k+q-1)(\ell/r) \\ p &= k + n\left(\frac{\Delta u}{\ell/r}\right) \\ \Delta y^+ &= (1/2)(p-k)(p-k+1)(\ell/r) - \Delta u \\ p_{ok} &= k + \overline{n}_{p-k}\left(\frac{\Delta U - (\Delta y^+ + (r-p)(p-k)(\ell/r))}{\ell/r}\right) \end{aligned}$$

Finally, of course the goodput is $p_{ok}/(T/\ell)$.

4.3. Buffer Underflow Without Overflow

In this case

$$\begin{aligned}
\Delta u &= b \\
p &= k + n\left(\frac{\Delta u}{\ell/r}\right) \\
\Delta y^{+} &= (1/2)(p-k)(p-k+1)(\ell/r) - \Delta u \\
\Delta U &= \Delta y^{+} + (r-p)(p-k)(\ell/r) + (1/2)(p-k-1)(p-k)(\ell/r) \\
&= \Delta y^{+} + (1/2)(2r-p-k-1)(p-k)(\ell/r) \\
p_{ok} &= k + n\left(\frac{\Delta u}{\ell/r}\right) = p
\end{aligned}$$

If $\Delta U < (1/2)(k-1)k(\ell/r)$

$$\begin{aligned}
q &= k - n\left(\frac{\Delta U}{\ell/r}\right) \\
\Delta y^{-} &= (1/2)(k-q)(k-q+1)(\ell/r) - \Delta U \\
T &= (\ell/\lambda) + (p-k)(\ell/r\lambda) + n\left(\frac{\Delta U}{\ell/r}\right)(\ell/r\lambda) = (1+(p-q)/r)(\ell/\lambda)
\end{aligned}$$

If, on the other hand $\Delta U > (1/2)(k-1)k(\ell/r)$,

$$\begin{aligned}
q &= 0 \\
\Delta y^{-} &= k(\ell/r) - [(\Delta U - (1/2)(k-1)k(\ell/r)) \bmod k(\ell/r)] \\
T &= (1+(p-q)/r)(\ell/\lambda) + \left\lceil \frac{\Delta U - (1/2)(k-1)k(\ell/r)}{k(\ell/r)} \right\rceil (\ell/r\lambda)
\end{aligned}$$

As usual, the goodput is $p_{ok}/(T/\ell)$.

4.4. Buffer Underflow and Overflow

In this case

$$\begin{aligned}
\Delta u &= b \qquad \Delta U = B - b \\
p &= k + n\left(\frac{\Delta u}{\ell/r}\right) \\
\Delta y^{+} &= (1/2)(p-k)(p-k+1)(\ell/r) - \Delta u \\
p_{ok} &= k + \overline{n}_{p-k}\left(\frac{\Delta U - (\Delta y^{+} + (r-p)(p-k)(\ell/r))}{\ell/r}\right)
\end{aligned}$$

If $\Delta U < (1/2)(k-1)k(\ell/r)$

$$\begin{aligned}
q &= k - n\left(\frac{\Delta U}{\ell/r}\right) \\
\Delta y^{-} &= (1/2)(k-q)(k-q+1)(\ell/r) - \Delta U \\
T &= (1+(p-q)/r)(\ell/\lambda)
\end{aligned}$$

If, on the other hand $\Delta U > (1/2)(k-1)k(\ell/r)$,

$$\begin{aligned} q &= 0 \\ \Delta y^- &= k(\ell/r) - [(\Delta U - (1/2)(k-1)k(\ell/r)) \bmod k(\ell/r)] \\ T &= (1 + (p-q)/r)(\ell/\lambda) + \left\lceil \frac{\Delta U - (1/2)(k-1)k(\ell/r)}{(k(\ell/r)} \right\rceil (\ell/r\lambda) \end{aligned}$$

Finally, the goodput is $p_{ok}/(T/\ell)$.

5. Numerical Results

Figure 5 shows results from the analysis from sections 3.1 and 4.1. Two cases are shown, one with $k = 4$ and one with $k = 16$. In the simulations, packet lengths are fixed, as in the analysis, but the relative phases of the packets in different virtual circuits were randomized at the start of each simulation run and the maximum excursion above and below threshold over the entire run was recorded. The values plotted are the average values from multiple simulation runs (more than 100 runs per data point).

Notice that when the overbooking ratio is ≤ 2, the maximum excursion above the threshold is always smaller than the excursion below the threshold and that when $r\lambda \geq 2$ this is reversed. Also notice that the analysis tracks the simulation results most closely when k is large. This is to be expected, since for small k, there is a greater likelihood that randomly selected phase offsets will differ substantially from the idealized even offset assumption of the analysis.

The analysis reported here gives much better estimates of the required buffer size than the worst-case analysis, allowing the buffers and the required threshold to be more closely tailored to the real needs. For the two cases shown, the worst-case analysis requires a total buffer size of 20 and 80 when the overbooking ratio is five. Notice however that when k is large (meaning that the peak virtual circuit rate is much smaller than the link rate), the amount of buffering required can still be substantial. This can be problematical for high speed links, for which cell buffers can be an expensive resource.

We now consider the goodput that results when the buffer is too small to ensure no overflow or underflow. Figures 6 and 7 are three-dimensional plots showing how the goodput varies with b and $B - b$ for small and large overbooking ratios, respectively. In each figure, the left-hand plot shows the goodput with approximately the same orientation as the goodput maps in Figure 1 and 4. The regions in the goodput maps are clearly identifiable in the plots. The right-hand plot in each figure shows the same data, but from a different viewpoint which more clearly shows how the goodput varies with b and $B - b$. The non-monotonic behavior of the goodput with respect to the amount of unavailable buffering is a somewhat unexpected but characteristic behavior of the early packet discard technique.

Figure 8 shows how the goodput varies as a function of the overbooking ratio, $r\lambda$. Two sets of analytical results are given. The one labeled worst-case corresponds to the analysis developed earlier. The one labeled best-case is a variation in which, when the buffer overflows, we assume that the k virtual circuits that remain active are those that have the earliest upcoming packet boundaries, rather than the latest, as assumed in sections 3 and 4. For these results, the packet was taken to be 8 Kbytes long or 171 cells, while b and $B - b$ were fixed at 128 and 64 cells in the left and right-hand plots, respectively. Note that as the overbooking ratio is increased from 1, the goodput first decreases due to underflow, reaching a minimum when the overbooking ratio is between 1.5 and 2. As the overbooking ratio increases further, the buffer swings above threshold become larger, and the swings below threshold shrink leading to buffer overflow. Once the overbooking ratio has increased to the point where k packets are propagated each cycle, further increases simply affect the duration of the cyclic variation in the buffer occupancy.

If we fix b and B with $B - b \leq k\ell$, the asymptotic goodput is simply $1/(1 + (B - b)/k\ell)$. This is based on the observation that if $r \to \infty$ while b and $B - b$ are held fixed, then the excursion below threshold approaches zero and so the overflow-only analysis of Section 4.2 is applicable. Notice that the asymptotic goodput can't drop below 1/2 since $B - b \leq k\ell$ (with a larger buffer, there would be no overflow). Also note that the largest asymptotic goodput is obtained when $B - b$ is small. However, this "advantage" of small buffers is offset by the fact that larger buffers can achieve 100% goodput for larger overbooking ratios than can small buffers. Using the analysis of Section 3.1 one can show that if $b = B - b$ and $b \leq (2 - \sqrt{3})k\ell \approx .27k\ell$, then 100% goodput is achieved when

$$r\lambda \leq 2 - (b/k\ell) - \sqrt{1 + (b/k\ell)^2 - 4(b/k\ell)}$$

Similarly, using the analysis of Section 4.1, one can show that if $(2 - \sqrt{3})k\ell < b < k\ell$, then 100% goodput results so long as

$$r\lambda \leq \frac{3}{2(1 - b/k\ell)}$$

Thus, we can characterize the goodput as having three distinct regions, as a function of offered load. In the *small overload region* a goodput of 100% is achieved, in the *large overload region* the goodput approaches the asymptotic value and in the *transition region* it moves between these two levels. If the small overload region ends when the overbooking ratio is < 2, the transition region is characterized by a dip below the asymptotic goodput (as in Figure 8), while if the small overload region extends to overbooking ratios > 2, the goodput simply decays to its asymptotic value in the transition region. Figure 9 tabulates the extent of the small overload region and the asymptotic goodput using the equations above. Notice that decreasing k, like increasing the buffer size, increases the extent of the small overload region while reducing the asymptotic goodput.

6. Closing Remarks

In this paper we have refined the analysis of the well-known Early Packet Discard technique for maintaining high throughput in overloaded ATM switches. The analysis is easy to apply, provides insight into the performance of Early Packet Discard in a wide variety of situations and agrees well with simulation results, especially when k and r are large. The results show that EPD requires large buffers for 100% goodput when the ratio of the line rate to the virtual circuit rate is large, but can achieve acceptable goodputs even for small buffers. With smaller buffers, it is subject to degraded goodput at small to medium overloads, but for heavy overloads, the goodput rises to an asymptotic value ensuring that congestion collapse is avoided. Reference [3] describes variants of EPD that provide both better performance and provide more nearly fair treatment of different virtual circuits.

References

[1] Floyd, Sally and Allyn Romanow. "Dynamics of TCP Traffic over ATM Networks," *IEEE Journal on Selected Areas in Communications*, 5/95.

[2] Jain, Raj. "Congestion Control and Traffic Management in ATM Networks: Recent Advances and a Survey," *Computer Networks and ISDN Systems*, 10/96.

[3] Turner, Jonathan S. "Maintaining High Throughput During Overload in ATM Switches", *Proceedings of Infocom*, 3/96.

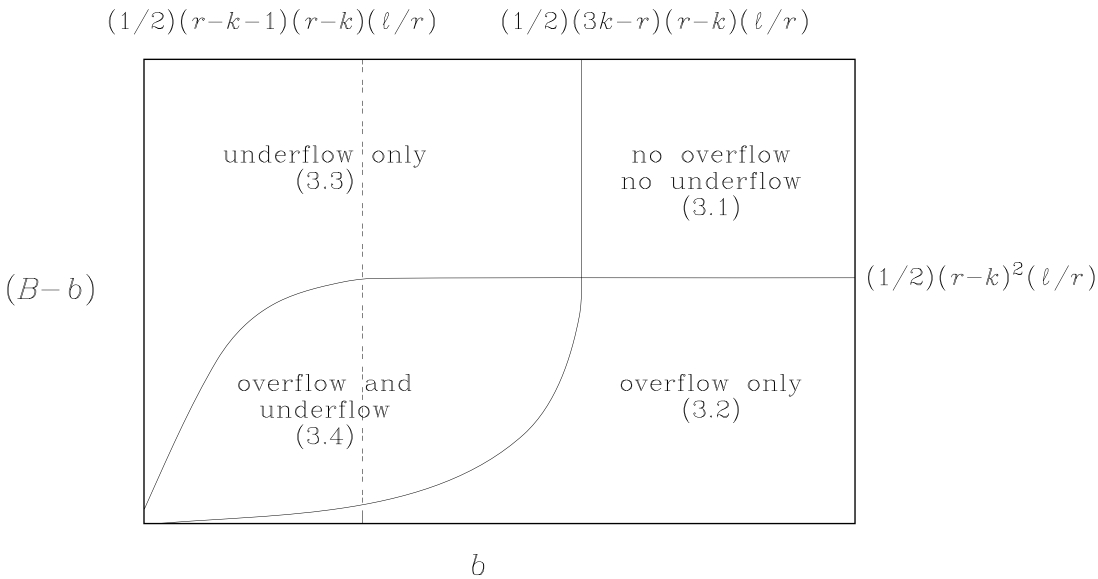


Figure 1: Goodput Map for $r\lambda \leq 2$

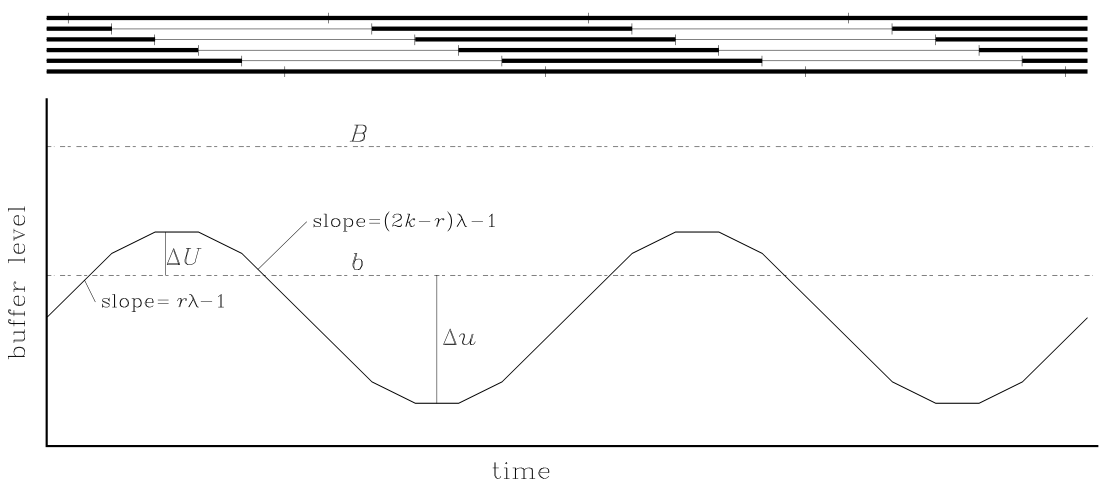


Figure 2: Buffer Range In Absence of Overflow and Underflow

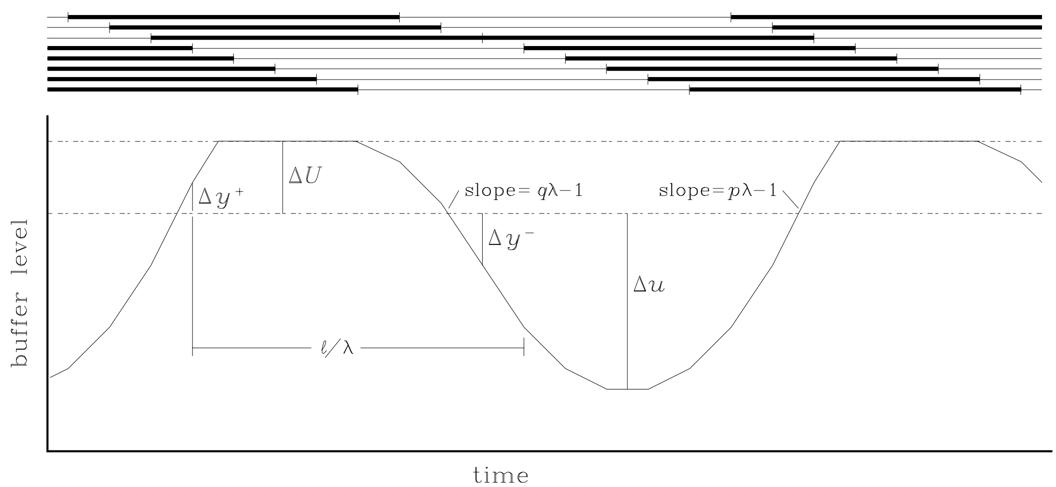


Figure 3: Buffer Overflow Without Underflow

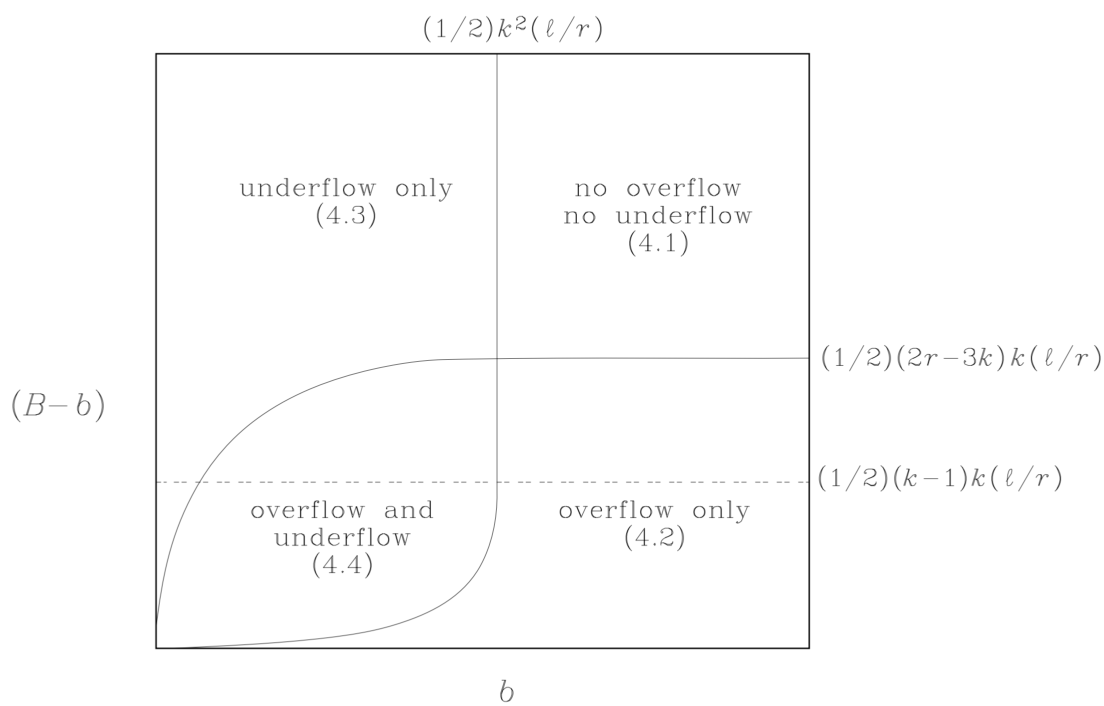
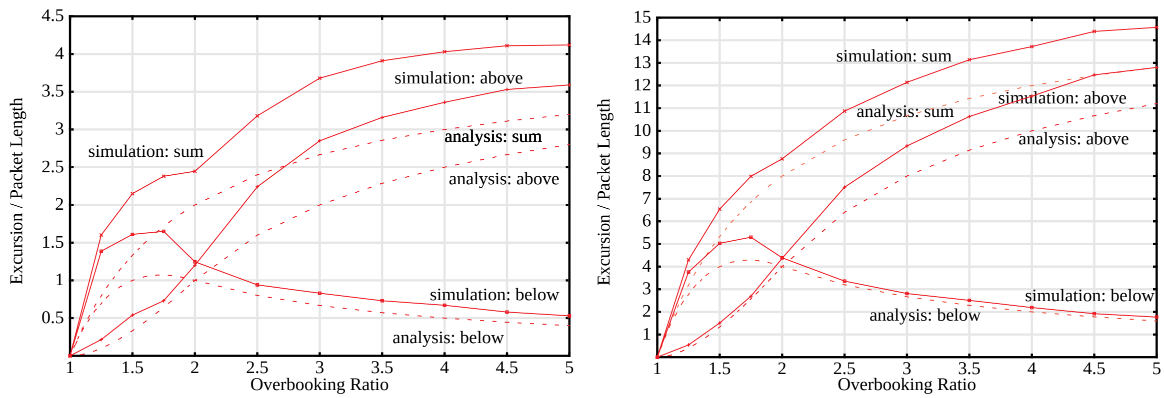


Figure 4: Goodput Map for $r\lambda \geq 2$



(a) k = 4

(b) k = 16

Figure 5: Normalized excursion above and below the threshold

Figure 6: Goodput as a function of b and $(B-b)$ for Small Overbooking Ratio

Figure 7: Goodput as a function of b and $(B - b)$ for Large Overbooking Ratio

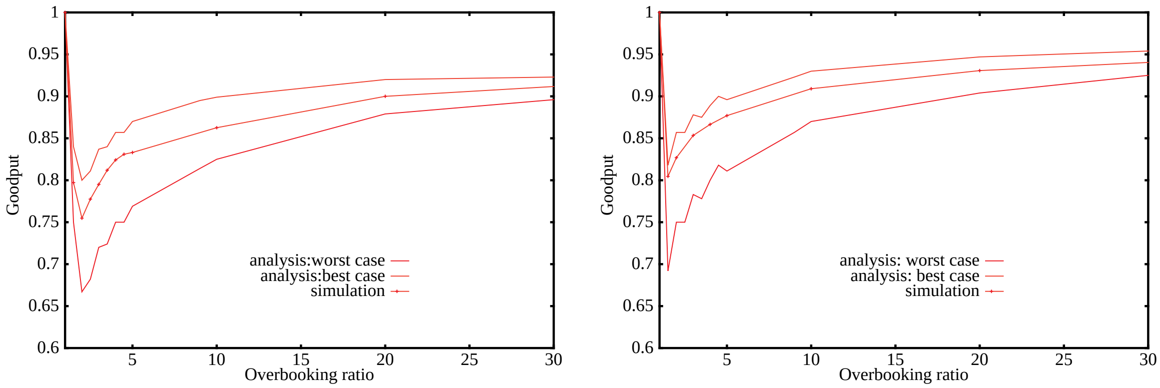


Figure 8: Goodput as a function of overbooking ratio ($k = 12$, $\ell = 171$)

		max $r\lambda$ for 100% goodput						asymptotic goodput					
		b/ℓ						b/ℓ					
		.5	1	2	3	4	5	.5	1	2	3	4	5
k	1	3	∞	∞	∞	∞	∞	.67	1	1	1	1	1
	2	1.5	3	∞	∞	∞	∞	.80	.67	1	1	1	1
	3	1.23	2.25	4.5	∞	∞	∞	.86	.75	.60	1	1	1
	4	1.16	1.5	3	6	∞	∞	.89	.80	.67	.57	1	1
	5	1.12	1.3	2.5	3.75	7.5	∞	.91	.83	.71	.63	.56	1
	6	1.1	1.23	2.25	3	4.5	9	.92	.86	.75	.67	.60	.55
	8	1.07	1.16	1.5	2.4	3	4	.94	.89	.80	.73	.67	.62
	10	1.05	1.12	1.3	2.14	2.5	3	.95	.91	.83	.78	.71	.67

Figure 9: Extent of Small Overload Region and Asymptotic Throughput