Estimating Arrival Rates from the RED Packet Drop History

Sally Floyd, Kevin Fall, and Kinh Tieu*

Network Research Group
Lawrence Berkeley National Laboratory, Berkeley CA
{*floyd,kfall*}*@ee.lbl.gov*
** DRAFT ***

April 6, 1998

Abstract

This paper outlines efficient mechanisms to estimate the arrival rate of high-bandwidth flows for a router implementing RED active queue management. For such a router, the RED packet drop history constitutes a random sampling of the arriving packets; a flow with a significant fraction of the *dropped packets* is likely to have a correspondingly-significant fraction of the *arriving* packets. In this paper we quantify this statement. We distinguish between two types of RED packet drops, *random* and *forced* drops, and show how the two types of drops should be used differently in estimating the arrival rate of a high-bandwidth flow.

1 Introduction

This paper describes an efficient mechanism for a router to identify and estimate the arrival rate of high-bandwidth flows in times of congestion, using the RED packet drop history. This work is in the context of the design of mechanisms to identify and restrict the bandwidth of high-bandwidth flows that are not using end-to-end congestion control in a time of high congestion [FF98]. In this context, there is no need to estimate the arrival rate of any flows in the absence of congestion, or in times of acceptably-low congestion. Similarly, in this context there is no need to estimate the arrival rate of any of the lower-bandwidth flows even in times of high congestion. All that is required is some mechanism to estimate the arrival rate of high-bandwidth flows in times of high congestion.

RED queue management gives an efficient sampling mechanism and provides exactly the information needed for identifying high-bandwidth flows in times of congestion. This mechanism does not require the router to keep per-flow state for each active flow. Keeping per-flow counters for packet arrivals for all active flows could be an unnecessary overhead for a router handling packets from a large number of very low bandwidth flows.

Our identification mechanism uses a periodic pass in the background over information about the packets dropped at the router by the RED queue management. The mechanism is independent of the granularity used to define a flow. One possibility would be for a router to define a flow by source and destination IP addresses. This would have the advantage of not being "fooled" by an application that breaks a single TCP connection into multiple connections to increase throughput. [FF98] discusses the negative impact of "breaking up" TCP connections on the general Internet.

Another possibility for defining the granularity of a flow would be to use source and destination IP addresses and port numbers to distinguish flows. For IPv6 flows that do not use the IPv6 Encapsulating Security Header, routers could use the flow ID field to define some flows. Routers attached to high speed links in the interior of the Internet might use a coarser granularity to define a flow, rather than have each TCP connection belong to a separate flow.

The identification mechanism in this section assumes a router with RED queue management, and draws on the discussion in [Nai96] for identifying high-bandwidth flows from the RED packet drop history. Section 2 distinguishes between *forced* and *random* packet drops for RED queue management. Section 3 considers the number of packet drops that should be included in a single *sample* of packet drops to give a reasonable estimate of the arrival rate of the high-bandwidth flow in that sample. Section 4 defines both a *packet* and *byte drop metric*, and shows that a mechanism for identifying high-bandwidth flows from RED packet drops should use the packet drop metric for random packet drops, and the byte drop metric for forced packet drops. Section 4 shows simulations illustrating the identification mechanism. Appendix B shows that for queues with Drop-Tail queue management, the history of packet drops does not give suf-

*This work was supported by the Director, Office of Energy Research, Scientific Computing Staff, of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098, and by ARPA grant DABT63-96-C-0105.

ficiently reliable information for identifying high-bandwidth flows.

2 Forced and Random packet drops

This section distinguishes between *forced* and *random* packet drops.

Definitions: *forced* and *random packet drops*. We say a packet drop is *forced* if a packet is dropped because either the FIFO buffer overflowed, or the average queue size estimated by RED exceeded the RED maximum threshold parameter *maxthresh*. Otherwise a packet drop is called *random*. Random packet drops are expected to represent the majority of all packet drops for a properly-configured RED gateway, and result from RED's probabilistic sampling of the arriving packet stream.

When the average queue size exceeds some minimum threshold, indicating incipient congestion, RED queue management uses a random sampling method to choose which arriving packets to drop. [FJ93] describes two variants of the RED algorithm. In *packet mode*, for a given average queue size, each arriving packet has the same probability of being dropped regardless of the packet size in bytes. In *byte mode*, a packet's probability of being dropped is a function of its size in bytes. The simulations later in this paper use RED queue management in byte mode. RED in packet mode is preferable for those routers limited by the number of *packets* arriving from each flow, rather than the number of *bytes*. RED in packet mode would give flows an incentive to use larger packets.

RED in byte mode is designed so that a flow's fraction of the aggregate random packet drops roughly equals its fraction of the aggregate arrival rate in *bytes* per second.[1] One motivation for the design of byte-mode RED comes from the operation of TCP congestion avoidance. TCP assumes that a single packet drop indicates congestion to the end nodes, regardless of the *number* of bytes lost in any dropped packet. Thus the goal of RED queue management in *byte* mode is to have each flow's fraction of the random congestion indications correspond to its fraction of the arriving traffic in *bytes* per second, regardless of how those bytes are grouped into packets. In contrast, the goal of RED queue management in *packet* mode is to have each flow's fraction of the random congestion indications correspond to its fraction of the arriving traffic in *packets* per second,

[1] For RED in byte mode, arriving *bytes* are marked, and packets containing those bytes are dropped. We assume that the packet drop rate is sufficiently low, relative to the packet size, that, in byte mode, it is unlikely that two "bytes" in the same packet will be marked to be dropped. That is, we assume that when the packet drop rate is high, RED will not longer be probabilistically dropping packets as random packet drops.

3 The number of packet drops for a single sample

This section considers the number of packet drops that should be included in a single *sample* or *reporting interval* of packet drops to allow a reasonable estimate of the arrival rate of the high-bandwidth flow in that sample. For the purposes of this section, we assume a RED queue in packet mode with a fixed average queue size, where each arriving packet has the same fixed probability p of being dropped.

Section X of [FJ93] gives a statistical result showing that given a fixed average queue size, n packet drops in the sample (including $S_{i,n}$ packet drops from flow i), and flow i with a fraction p_i of the arriving bandwidth in packets per second, the probability a flow receives more than c times its "share" $p_i n$ of packet drops in that sample is as follows:

$$\mathrm{Prob}(S_{i,n} \geq cp_i n) \leq (e^{2n(c-1)^2(p_i)^2}).$$

This is illustrated quantitatively in [FJ93] for $n = 100$. For RED in *byte* mode, the same result applies for p the probability that a fixed-size packet is dropped, and p_i defined as flow i's fraction of the arriving bandwidth in *bytes* per second.

The result above quantifies the statement that with sufficiently many packet drops, a flow is unlikely to receive more than c times its "share" of packet drops. Thus, the RED packet drop history can be an effective aid in identifying high-bandwidth flows.

From Appendix A, we get a more precise estimate of the probability that a flow receives more than c times its "share" $p_i n$ of packet drops, for $c > 1$:

$$Prob\,(S_{i,n} \geq cp_i n) \leq \left[\left(\frac{1}{c}\right)^{cp_i} \left(\frac{1-p_i}{1-cp_i}\right)^{1-cp_i} \right]^n . \quad (1)$$

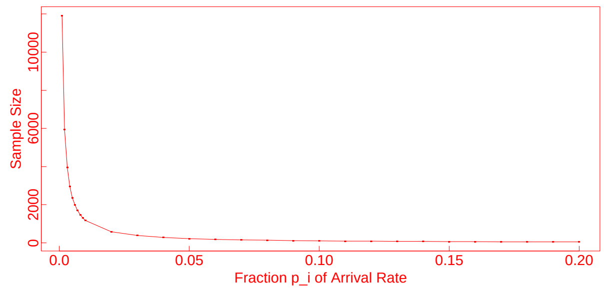

Figure 1: Sample size as a function of the high-bandwidth flow's fraction of the arrival rate, for P = 0.01, c=2.

Let P be the desired upper bound for $Prob\,(S_{i,n} \geq cp_i n)$. How large must n be (i.e., how many drops must be included in the drop history) in order to achieve the assurance that $Prob\,(S_{i,n} \geq cp_i n) \leq P$ for a given c and p_i? Solving for n

in equation (1):

$$n \geq \frac{\ln[P]}{\ln\left[\left(\frac{1}{c}\right)^{cp_i}\left(\frac{1-p_i}{1-cp_i}\right)^{1-cp_i}\right]}. \qquad (2)$$

Figure 1 shows this equation for $P = 0.01$ and $c = 2$; the x-axis shows p_i, and the y-axis shows the lower bound for n needed to satisfy equation (2). For example, if we want to know how many total packet drops RED should include in its packet history to have probability at most 1% that a flow with a fraction p_i of the arriving bandwidth receives at most twice its share of packet drops, then we substitute $P = 0.01$ and $c = 2$ into equation (2).

The approach in this paper uses equation (2) to determine how many drops n to examine to get a reasonable estimate of the bandwidth of the highest-bandwidth flow. At one-second intervals, starting from a minimum interval of three seconds, the router evaluates the sample size n (i.e., the cumulative number of packet drops in the packet drop history) and the fraction d_1 of the sample drops that belong to the flow with the highest number of drops. For given parameters P and c, the sample should be sufficiently large that there is probability at most P that the drop history overestimates the high-bandwidth flow's arrival rate by a factor of c or more. If n and p_1 satisfy the relationship in equation (2) for parameters P and c and for $p_1 = d_1/c$, then we can use this packet drop history to estimate the bandwidth of the high-bandwidth flow. If, on the other hand, n and p_1 do not satisfy the relationship in equation (2), then we continue to add to the packet drop history, and recheck one second later.

This does not give a rigorous guarantee that the probability is at most P that the flow with the most drops in the sample received more than c times its share of packet drops. However, simulations later in the paper show that in this case, the packet drop history gives a good estimate of the bandwidth of the high-bandwidth flow.

4 Packet, byte, and combined drop metrics

This section defines the *packet*, *byte*, and *combined drop metrics*, and uses simulations to show that the combined metric gives a good estimate of the arrival rate of the high-bandwidth flow.

Definition: the *packet drop metric*. We define the *packet drop metric* for a flow over some time interval as the ratio of the number of packets dropped from that flow to the total number of dropped packets from that time interval. For RED in *byte* mode, the packet drop metric for the random packet drops estimates a flow's fraction of the aggregate arrival rate in *bytes* per second (Bps). For RED in *packet* mode, the packet drop metric for the random packet drops instead estimates a flow's fraction of the aggregate arrival rate in *packets* per second.

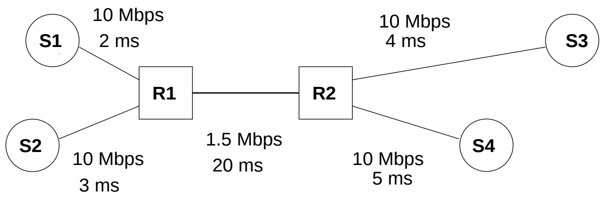

Figure 2: Simulation network.

Figure 3 shows the results of a simple simulation for the topology in Figure 2. For all simulations in this paper, the RED queue management is configured with a minimum threshold of five packets, a maximum threshold of 20 packets, and a packet drop rate approaching 10% as the average queue size approaches the maximum threshold.[2] The buffer size in router R1 for the queue for the congested link R1-R2 is set to 100 packets; packets are rarely dropped due to buffer overflow. For these simulations we use probability P set to 0.01, and share c set to 1.5 for determining the size of the drop sample. Thus, each drop sample contains enough drops that the probability that the high-bandwidth flow receives more than 1.5 times its "share" of the packet drops is at most 1%.

The simulation includes a range of two-way traffic, including bulk-data TCP and constant-bit-rate (CBR) UDP flows. The TCP connections have a range of start times, packet sizes (from 512 to 2000 bytes), receiver's advertised windows, and round-trip times. Of particular interest are the high-bandwidth flows. Flow 3 is a CBR flow with 190-byte packets and an arrival rate of 64 KBps, about one-third of the link bandwidth. Flow 4 is a TCP flow whose high bandwidth is due to its larger packet size of 2000 bytes; most of the TCP flows in the simulation use 512-byte packets. More details of the simulation scenario are available in the simulations scripts [FF98].

The upper left graph in Figure 3 shows the packet drop metric for the random packet drops in the simulation. For every drop sample, there is a mark in the graph for every flow experiencing at least one packet drop. For each flow i, the number i is plotted on the graph, with the x-axis giving i's fraction of the aggregate arrival rate in Bps over the reporting interval, and the y-axis giving i's fraction of the packet drops in that reporting interval.

If each flow's packet drop metric for random packet drops was an exact indication of that flow's arrival rate in Bps, then all marks in the upper left graph would lie on the diagonal line. A mark in the upper left quadrant of the graph indicates a flow with a larger fraction of dropped packets that arriv-

[2] This is a change from the upper bound on the packet drop rate used for simulations in [FJ93]. This change is better suited for routers that typically have high levels of congestion.

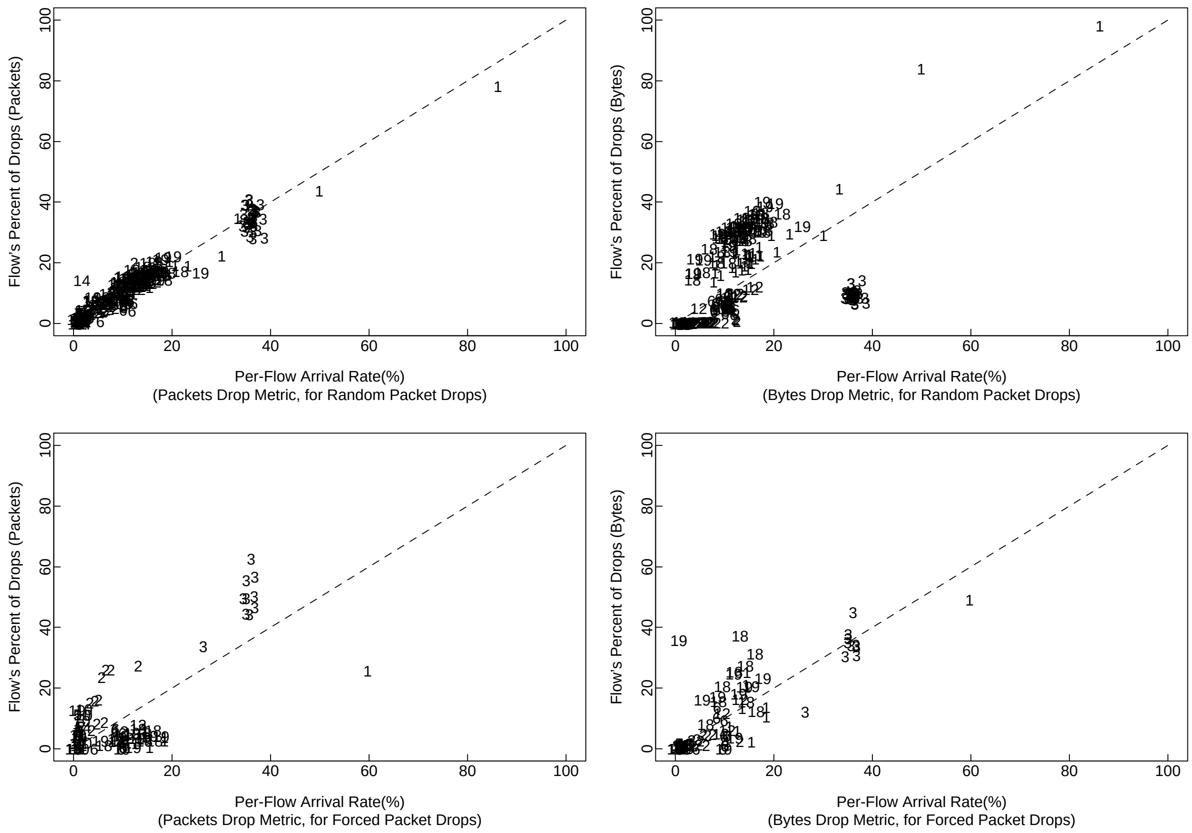

Figure 3: Comparing drop metrics for forced and random packet drops.

ing packets. As the graph shows, the packet drop metric for the random packet drops gives a reliable identification of the high-bandwidth flows.

Unlike random packet drops, with forced packet drops the RED algorithm does not get to “choose” whether or not to drop a packet. When the buffer is full, or when the average queue size exceeds the maximum threshold, RED drops *all* arriving packets until conditions change (until the buffer is not longer full, or the average queue size no longer exceeds the threshold). Thus, a flow with one large packet arriving during a forced-drop time interval will have its packet dropped, and a flow with several small packets arriving during this interval will instead have all of its small packets dropped.

The lower left graph in Figure 3 shows the packet drop metric for forced packet drops. As Figure 3 shows, the packet drop metric with forced packet drops has a systematic bias overestimating the arrival rate for flows with small packets such as Flow 3 and underestimating the arrival rate for flows with larger packets such as Flow 4.

Definition: the *byte drop metric*. The byte drop metric is defined as the ratio of the number of bytes dropped from a flow to the total number of bytes dropped. For forced packet drops, this metric gives the best estimate of a flow's arrival rate, as shown in the lower right graph of Figure 3. The upper right graph of Figure 3 shows that the byte drop metric is not adequate for random packet drops, because it overestimates the arrival rate of flows with larger packets and underestimates the arrival rate of flows with smaller packets.

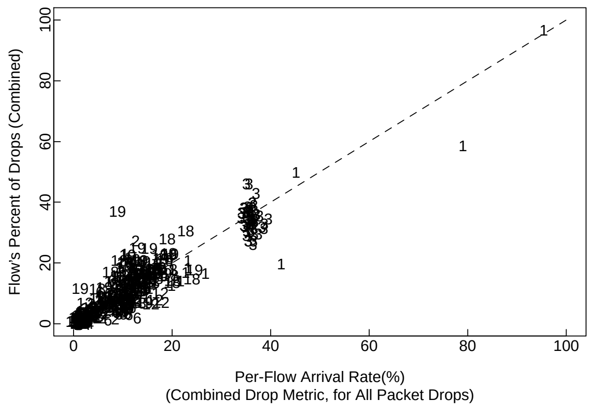

Figure 4: The combined drop metric for all packet drops, for simulation 1.

Definition: the *combined drop metric*. By weighting a

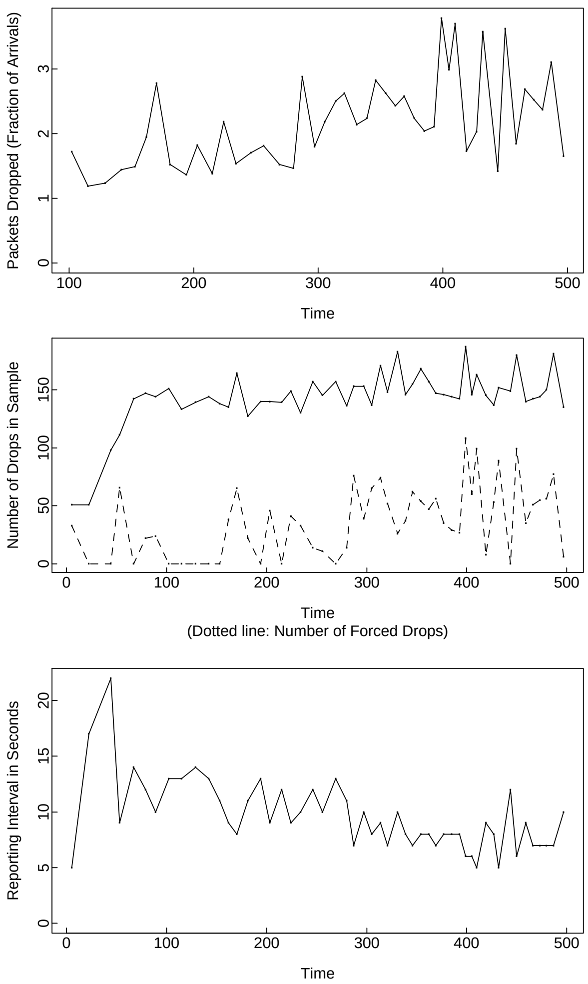

Figure 5: The percent of packets dropped, number of drops in a sample, and length of each sample, for simulation 1.

flow's byte and packet drop metrics by the ratio of forced and random packet drops, we can better estimate a flow's behavior than by using either metric alone. We define the *combined drop metric* for forced and random packet drops as follows:

$$\mathrm{M_{Forced}} * \mathrm{f_{Forced}} + \mathrm{M_{Random}} * \mathrm{f_{Random}},$$

where $\mathrm{M_{Forced}}$ is the flow's byte drop metric for the forced packet drops, $\mathrm{M_{Random}}$ is the flow's packet drop metric for the random packet drops, and $\mathrm{f_{Forced}}$ and $\mathrm{f_{Random}}$ are the fraction of the total packet drops from that sample that are forced and random, respectively.

For the simulations in this section, the buffer is sufficiently large that packets are rarely dropped due to buffer overflow; the forced packet drops in these simulations result from the average queue size exceeding the upper threshold *maxthresh*. For a queue in units of packets, where the buffer is able to accommodate a fixed number of packets regardless of packet

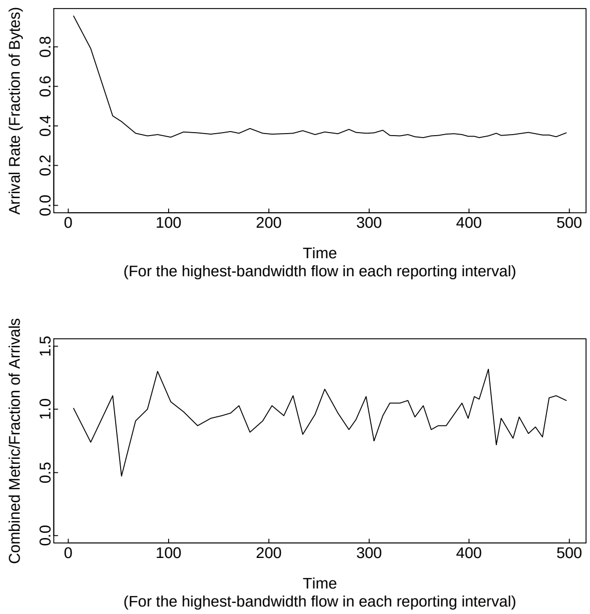

Figure 6: Statistics for the high-bandwidth flow from each sample, for simulation 1.

size, the appendix shows that packets dropped because of buffer overflow should be counted with the *random* packet drops, using the packet drop metric. In contrast, for a queue in units of bytes, packets dropped because of buffer overflow should be counted with the *forced* packet drops, using the byte drop metric. The appendix shows that routers with Drop-Tail queue management cannot use the packet drop history to reliably identify high bandwidth flows.

Figure 4 shows the combined drop metric for each flow for the simulation in Figure 3, calculated each reporting interval. As Figure 4 shows, the combined drop metric is a reasonably accurate indicator of the arrival rate for high-bandwidth flows. The graphs in Figure 5 shows the percent of arriving

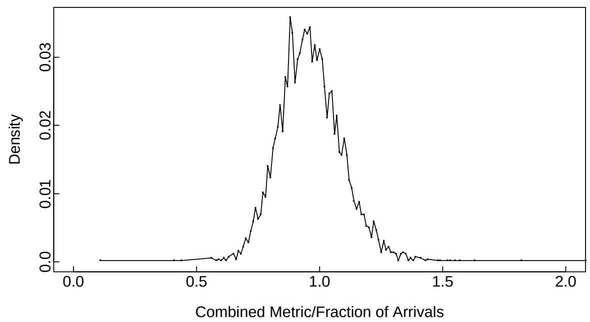

Figure 7: Statistics for the high-bandwidth flow from each sample, for 100 runs of simulation 1.

packets dropped, the number of packet drops in a sample, and the length of a reporting interval in seconds.

The two graphs in Figure 6 show the statistics for the high-bandwidth flow for each sample in the simulation; in this scenario the high-bandwidth flow is usually the CBR UDP flow. The top graph shows the arrival rate of the high-bandwidth flow as a fraction of the overall arrival rate in Bps. The second graph shows, for the high-bandwidth flow in each sample, the ratio between that flow's combined metric and that flow's fraction of the arrival rate in bytes. If the combined metric was a perfect estimate of the flow's arrival rate in Bps, then this ratio would always be one.

Figure 7 shows the density function for this ratio over 100 runs of the simulation, from more than 4,900 samples. This graph is consistent with the parameters that we used in the simulations for determining the sample size, specifying the probability that a flow received more than 1.5 times its "share" of the packet drops should be at most 0.01. Figure 7 shows that our algorithm for choosing the sample size is conservative; In 99% of these 4,900 samples, the high-bandwidth flow received at most 1.3 times its "share" of the packet drops.

Figures 8-10 show a simulation that differs from Figure 4 only in that it has none of the UDP flows, and fewer TCP flows. As Figure 8 shows, the high-bandwidth flow still receives a significant fraction of the link bandwidth, but the percent of arriving packets dropped is very low throughout this simulation, and as a result the reporting intervals are up to 30 seconds long. Figure 10 shows that for this simulation scenario, the combined drop metric is a good estimate of the arriving rate of the high-bandwidth flow.

Figures 11-13 show a simulation that differs from Figure 4 in that the bandwidth of the congested link is 45 Mbps, and there are more active flows. The level of congestion varies during the simulation, increasing up to time 350, and then decreasing again. The bottom graph of Figure 13 shows that the combined drop metric generally underestimates the arrival rate of the high-bandwidth flow. The flows that are overrepresented in terms of packet drops are the flows with smaller packet sizes. This simulation has two-way traffic, with many flows with 512-byte packets, and many other flows with 4000-byte packets. The simulation shows that in byte mode, RED has a slight bias in favor of flows with larger packets, in that flows with larger packets are somewhat less likely to have packets dropped than flows with smaller packets but the same arrival rate in Bps.

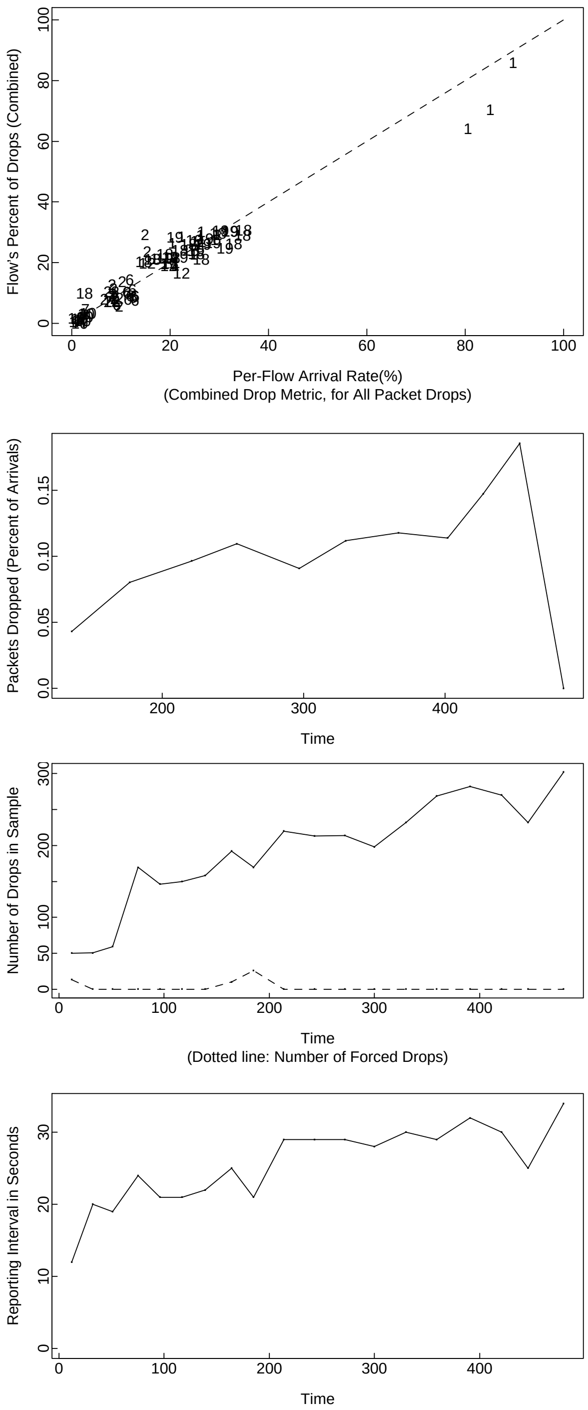

Figure 8: The combined drop metric, percent of packets dropped, number of drops in a sample, and length of each sample, for simulation 2.

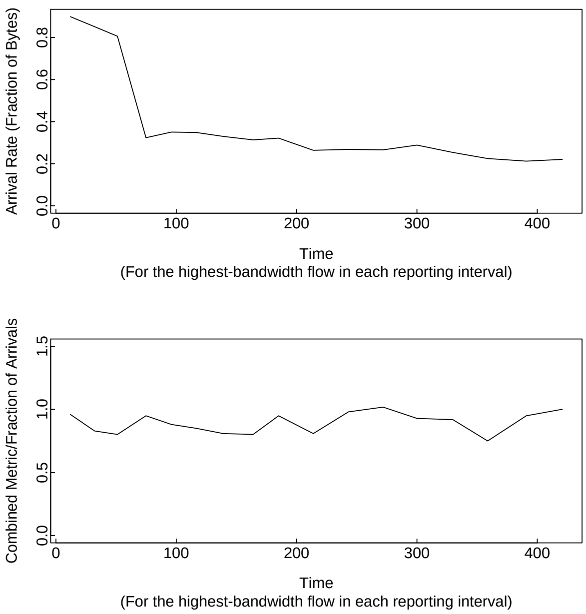

Figure 9: Statistics for the high-bandwidth flow from each sample, for simulation 2.

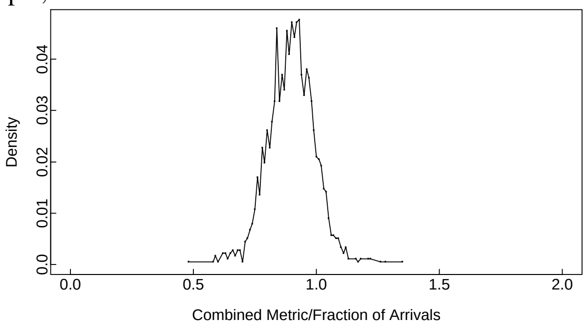

Figure 10: Statistics for the high-bandwidth flow from each sample, for 100 runs of simulation 2.

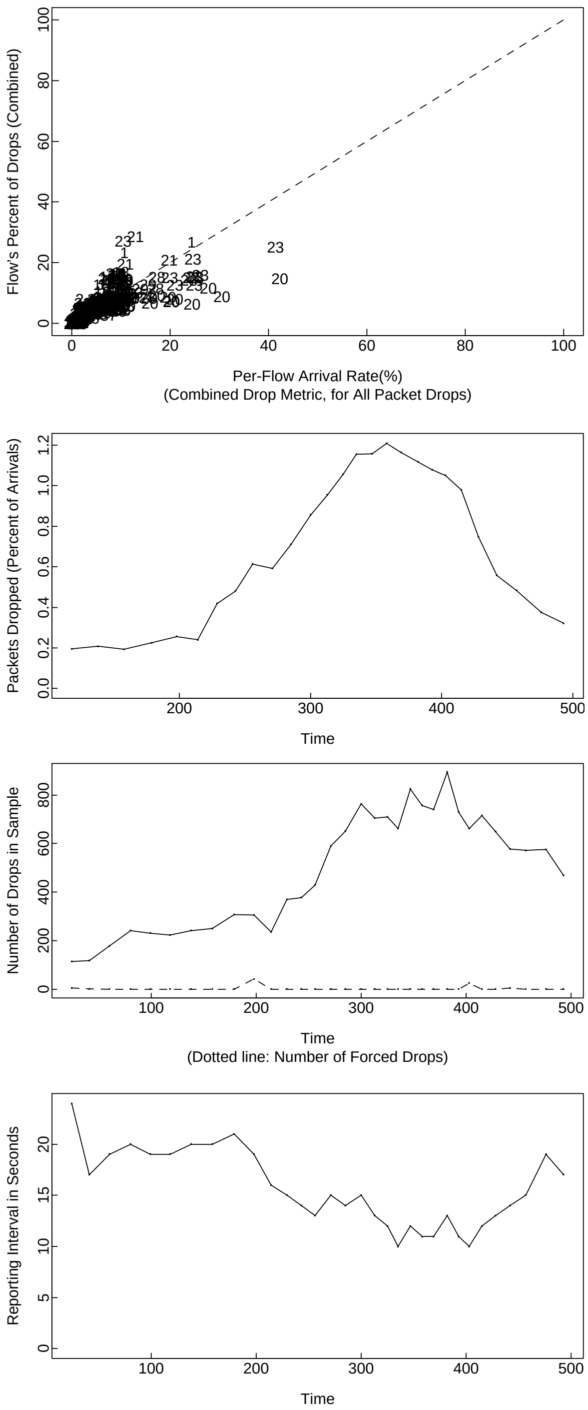

Figure 11: The combined drop metric, percent of packets dropped, number of drops in a sample, and length of each sample, for simulation 3.

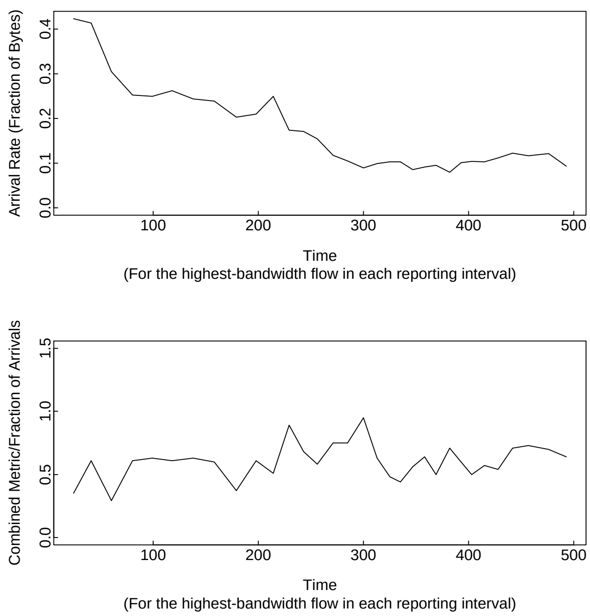

Figure 12: Statistics for the high-bandwidth flow from each sample, for simulation 3.

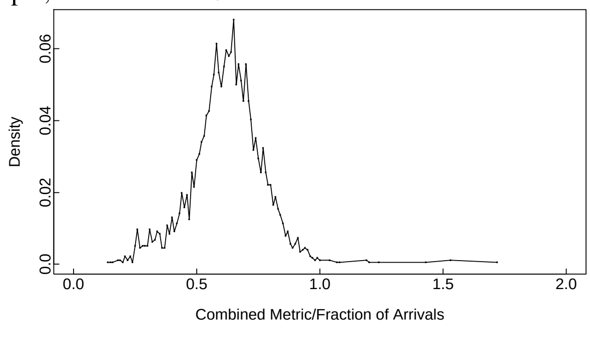

Figure 13: Statistics for the high-bandwidth flow from each sample, for 100 runs of simulation 3.

5 Conclusions

The paper has presented a mechanism for estimating the arrival rate of high-bandwidth flows based of the RED packet drop history.

This low-overhead mechanism would not be needed in a router with sufficient resources to measure directly the arrival rate of high-bandwidth flows.

In an environment with Explicit Congestion Notification [Flo94], it would be straightforward to extend this mechanism to take into account packets with the Explicit Congestion Notification bit set in the packet header. Modifications would also be needed to deal with the realities of packet fragmentation.

6 Acknowledgments

This paper is part of a series of papers on router mechanisms in support of end-to-end congestion control. This work results in part from a long collaboration with Van Jacobson. It also results from a history of discussions in the Internet End-to-End Research Group and elsewhere. We are indebted to Van Jacobson and members of the Internet End-to-End Research Group for discussions on these matters. We are also indebted to Hari Balakrishnan, Greg Minshall, Lixia Zhang, and the anonymous reviewers from SIGCOMM 97 for feedback on an earlier draft on this paper, and to Jean Bolot, Bob Braden, Jamshid Mahdavi, Matt Mathis, and Scott Shenker for discussions of related matters.

References

[FF98] S. Floyd and K. Fall. "Promoting the Use of End-to-End Congestion Control in the Internet". Submitted to IEEE/ACM Transactions on Networking, URL http://www-nrg.ee.lbl.gov/floyd/end2end-paper.html, Feb. 1998.

[FJ93] S. Floyd and V. Jacobson. "Random Early Detection Gateways for Congestion Avoidance". *IEEE/ACM Transactions on Networking*, 1(4):397–413, Aug. 1993. URL http://www-nrg.ee.lbl.gov/nrg-papers.html.

[Flo94] S. Floyd. "TCP and Explicit Congestion Notification". *ACM Computer Communication Review*, 24(5):10–23, Oct. 1994.

[HR95] T. Hagerup and C. Rub. "A Guided Tour of Chernoff Bounds". *Information Processing Letters*, 33:305–308, 1995.

[Nai96] T. Nairne. "Idenfitying High-Bandwidth Users in RED Gateways". Technical report, Oct. 1996. UCLA Computer Science Department.

A Chernoff bounds

In Section 3, we needed an upper bound on the probability that a flow receives more than c times its expected number $p_i n$ of packet drops, for $c > 1$. This Appendix shows the calculation of this bound. Let $S_{i,n}$ be the number of the n drops in a sample that are from flow i, and let p_i be flow i's fixed fraction of the arrival rate during that period. The expected value of $S_{i,n}$ is $p_i n$. We require a bound for $Prob\,(S_{i,n} \geq cp_i n)$.

Using Chernoff-type bounds [HR95], for $0 < u < 1$ and $u \geq p$:

$$Prob\,(S_{i,n} \geq un) \leq \left[\left(\frac{p}{u}\right)^u \left(\frac{1-p}{1-u}\right)^{1-u} \right]^n . \quad (3)$$

Letting $p = p_i$ and letting $u = cp_i$ in equation (3), for $c \geq 1$, we get the following:

$$Prob\,(S_{i,n} \geq cp_i n) \leq \left[\left(\frac{1}{c}\right)^{cp_i} \left(\frac{1-p_i}{1-cp_i}\right)^{1-cp_i} \right]^n .$$

It is possible also to bound the probability that a flow receives less than its share of packet drops. From [HR95], the bound in equation (3) holds for $0 < u < 1$ and $u < p$:

$$Prob\,(S_{i,n} \leq un) \leq \left[\left(\frac{p}{u}\right)^u \left(\frac{1-p}{1-u}\right)^{1-u} \right]^n .$$

B Identifying high-bandwidth flows for queues with drop-tail queue management

Figure 14 shows the results from a simulation that differs from the simulation in Figure 3 largely in that the router uses Drop-Tail rather than RED queue management. With Drop-Tail queue management, the router only drops arriving packets when the buffer overflows. For the simulation in Figure 14, the buffer is measured in packets, with a buffer size of 25 packets. That is, the buffer can store exactly 25 packets, regardless of the size of each packet in bytes.

Because this simulation was done from an older script with an older version of the simulator, it also differs from the simulation in Figure 3 in that the traffic is slightly different, and the drop samples are taken after every 100 packet drops.

The graphs in Figure 14 compare the packet and the byte drop metric. Figure 14 shows that for a Drop-Tail queue measured in packets, the byte drop metric is better than the packet drop metric in indicating a flow's arrival rate in Bps. This is because a Drop-Tail queue with a queue measured in packets drops proportionately more packets from small-packet flows than from large-packet flows with the same arrival rate in Bps. However, Figure 14 also shows that for

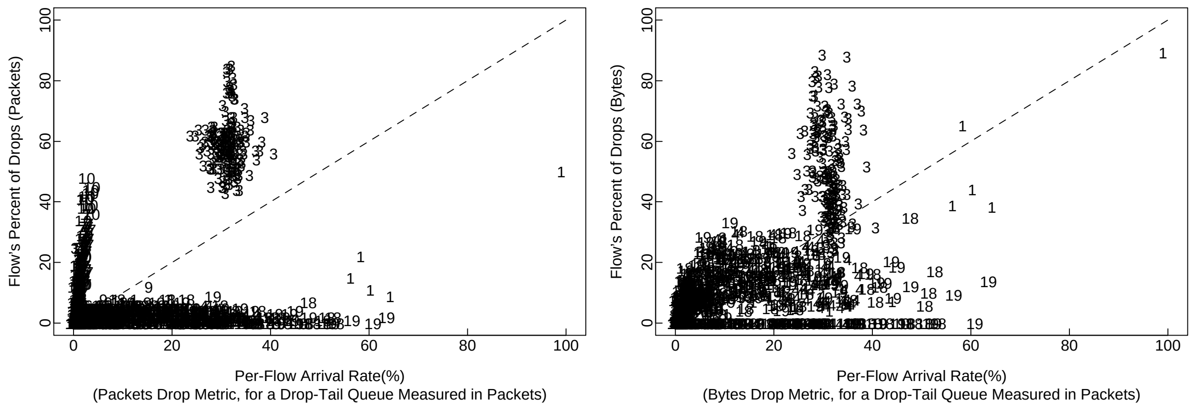

Figure 14: Comparing drop metrics for packet drops for a Drop-Tail queue measured in packets.

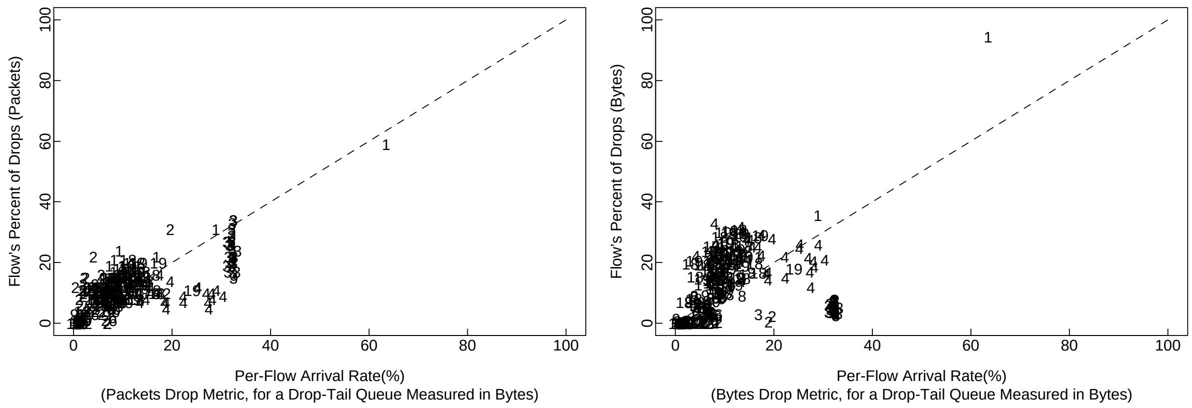

Figure 15: Comparing drop metrics for packet drops for a Drop-Tail queue measured in bytes.

a Drop-Tail queue measured in packets, neither drop metric gives a very reliable indication of a flow's arrival rate in bytes per second.

Figure 15 shows that when the Drop-Tail buffer is measured in bytes rather than packets, both drop metrics gives a plausible indication of a flow's arrival rate in Bps, with the packet drop metric doing better than the byte drop metric. These graphs show a simulation where the Drop-Tail buffer is measured in bytes, with a buffer size of 12.5 KB. In this case, an almost-full buffer might give room for a small packet but not for a larger one.

We note that with very heavy congestion and unresponsive flows, even a RED queue will no longer be dropping all packets probabilistically, but will be forced to drop many arriving packets either because of buffer overflow (for a queue with a small buffer relative to the maximum threshold for the average queue size), or because the average queue size is too high (for queues with larger buffers). As the packet drop rate increases, the computational overhead of monitoring dropped packets approaches the computational overhead of monitoring packet arrivals directly. Our hope is that the deployment of mechanisms for the identification and regulation of high-bandwidth unresponsive flows, coupled with sensible network provisioning, will in many cases be sufficient to prevent these high packet drop rates.