Patterns of Congestion Collapse

Tom Kelly, Sally Floyd, and Scott Shenker*
International Computer Science Institute,
and University of Cambridge

June 3, 2003

Abstract

In this paper we consider the potential for congestion collapse in a range of network scenarios. In particular, we are interested in the effect of the topology, the scheduling discipline (FIFO or FQ scheduling), the level of statistical multiplexing, the traffic characteristics, and other factors. We consider topologies more complex than a single congested link, or a single string of congested links. This paper first shows that it is possible to have high equilibrium loss rates with rational greedy senders sharing a FIFO link or with greedy senders sharing an FQ link with bursty cross-traffic. We then consider specific topologies with a range of senders to calculate steady-state packet loss rates and derive the goodput of the congested links. In particular, we find some scenarios where goodput is higher with FQ scheduling, and other scenarios where goodput is higher with FIFO scheduling.

1 Introduction

Previous work on congestion collapse has looked at simple topologies with "dumb" sources [2].[1] Our goal is to explore how more complex and realistic congestion collapse scenarios could arise, and to explore the effect of topologies, scheduling mechanisms, and other parameters on this congestion collapse.

End-to-end congestion control is required not only to help the end user, but also to prevent congestion collapse in the network. Without end-to-end congestion control, links could be busy transmitting packets that will only be dropped later downstream, thereby wasting scarce bandwidth that could have been used productively. For a given link, let a *dead packet* be a packet that ends up being dropped downstream, before reaching its intended receiver. The *dead packet ratio* for a link is then the fraction of the link bandwidth consisting of dead packets. This paper explores how the dead packet ratio on busy links can be affected by factors such as the topology; the scheduling discipline; the number of concurrent flows; the utility function being optimized by greedy flows; the unpredictability of available per-flow bandwidth; and other factors.

There are other types of packets that are not dropped downstream, but that do not contribute to the overall goodput of the flow or of the network. These include duplicate packets; dummy packets that carry no information of interest to the receiver (e.g. DDoS, junk mail, etc.); fragments of packets where some other fragment has been dropped in the network; etc. We do not include any of these in our definition of dead packets, and we do not consider any of these other forms of unproductive packets in this paper. Thus, in an environment where no packets are ever dropped, the dead packet ratio as defined in this paper would be zero. Similarly, given a definition of 'congested links' that includes all links that drop packets, then the dead packet ratio on congested links would be zero in an environment where each flow traversed at most one congested link.[2]

The approach taken here consist of two separate parts. The first part, described in Section 1.1, consists of exploring simple scenarios that result in high steady-state packet drop rates on a single congested link. We pay particular attention to scenarios with greedy flows, where sources are free to change their sending rate to optimize their own utility function. The second part of the paper, described in Section 1.2, addresses the sometimes subtle relationship between high packet drop rates, high dead packet ratios on congested links, and the loss of aggregate goodput.

1.1 Packet drop rates

In the first part of the paper we explore a range of scenarios that can result in high steady-state packet drop rates. It is well-known that dumb senders can result in high packet drop rates, so we don't explore this case further.

*This material is based in part upon work supported by the National Science Foundation under Grant No. 0205519. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

[1] We define a "dumb" source as any source that does not modify its sending rate as a function of the packet drop rate along the path.

[2] The dead packet ratio on uncongested links would not necessarily be zero, but since these links were uncongested, these dead packets would not represent a waste of scarce bandwidth, and would not be a concern.

In Section 3 we explore high packet drop rates arising from greedy senders optimizing their own utility functions against a cost incurred due to a lost packet. For scenarios with FIFO scheduling and many greedy flows competing on a single congested link, we consider a locally stable state, where no flow has an incentive to change its sending rate, and derive the equilibrium packet drop rate in this stable state. The equilibrium packet drop rate, of course, depends directly on the utility functions used by the greedy flows. For these scenarios the use of FQ scheduling would be sufficient to ensure a drop rate of zero with greedy senders.

In Section 4 we consider scenarios with an unpredictable available bandwidth, where greedy senders have an incentive to maintain a high steady-state packet drop rate even with FQ[3] scheduling. So far, we have only considered flows with concave utility functions, but future work could also consider non-concave utility functions.

1.2 Congestion collapse and the dead packet ratio

In the second part of the paper we show that in an environment with multiple congested links, high steady-state packet drop rates can result in degraded overall goodput. That is, we explore the relationship between high steady-state packet drop rates and the dead packet ratio for congested links in the network. In particular, we explore the effect of the topology and of scheduling mechanisms on the dead packet ratio.

Section 5 considers the dead packet ratio in a scenario with dumb CBR sources on a cyclic topology with FIFO scheduling. This illustrates that the dead packet ratio on congested links could be high in an environment of small fixed-rate flows, such as telephony traffic, if the demand is sufficiently large. One could argue that long-term provisioning will be sufficient to avoid sustained overload in the steady-state. Even if this is true, it is desirable to avoid high dead-packet ratios that could occur in atypical periods of high congestion, such as the two hours after an earthquake, when network bandwidth can be particularly precious. Or more explicitly, it is precisely when demand is unusually high that one would particularly like to avoid high dead packet ratios on congested links.

Section 6 considers greedy sources in a cyclic topology, with a range of utility functions. In this perfectly symmetric environment, the use of FQ scheduling would be sufficient to reduce the dead packet ratio on the congested links to zero. An open question is to explore how the unpredictability of available bandwidth would influence the dead packet ratio in this case, with both FIFO and FQ scheduling.

In contrast to the cyclic topology, where FQ scheduling is sufficient to prevent a high dead packet ratio on congested links in the steady-state case, Sections 7 and 8 consider "rail-road" topologies. In these scenarios, with dumb senders, the dead packet ratio is sometimes greater with FIFO scheduling and other times is greater with FQ scheduling, depending on the details of the topology.

It has been shown elsewhere that FQ scheduling mechanisms aren't sufficient to prevent congestion collapse [2]. However, the question of when FQ scheduling would significantly decrease the danger of congestion collapse, and when it would not, remains unanswered, particularly for a range of topologies and scenarios more realistic than the ones studied so far. One goal would be to characterize how the scheduling mechanism affects the dead packet ratios on congested links for a wide range of topologies and traffic scenarios. A specific open question is whether there are scenarios with congestion collapse when there are greedy senders, a predictable, steady-state available bandwidth as a result of long-lived flows, and FQ scheduling.

1.3 Building in robustness against congestion collapse

A general theme of this work is to explore network scenarios with some form of tragedy of the commons [3], where users optimizing their own individual utility functions can lead away from optimizing the common good, and to consider how the network can build in robustness in these cases. To prevent the tragedy of the commons that can result from high dead packet ratios on congested links, it would generally suffice for end users to agree not to maintain high sending rates in the face of high steady-state packet drop rates.

It is generally agreed that routers need mechanisms to drop packets from flows that persist in maintaining high sending rates in the face of high packet drop rates; these mechanism can provide a useful deterrent to misbehaving flows. However, local dropping mechanisms at routers to control anti-social flows would not be sufficient to prevent congestion collapse. In order to prevent congestion collapse when faced with users that maintain high sending rates in the face of high packet drop rates, congested routers need a mechanism to reduce their own dead packet ratios. To achieve this packets which are likely to be dropped downstream should be dropped in preference to those that would reach their destination. However, this would require a high degree of coordination between congested routers, and this does not appear to be a realistic goal for datagram networks in the general case. (In extreme cases, such as large-scale DDoS attacks, there may be few other alternatives.)

Instead of attempting to build in robustness against congestion collapse by introducing router mechanisms to reduce dead packet ratios in the presence of high packet drop rates, it could be argued that router mechanisms should deter high packet drop rates in the first place. One goal of this work is to explore the kinds of scenarios, traffic mixes, transport pro-

[3] In this paper we concentrate on FQ (Fair Queuing) scheduling, but it is hoped that the results would also apply to any packet scheduling algorithm that results in a max-min bandwidth allocation.

tocols, and utility functions that could result in high steady-state packet drop rates.

A second goal of this work is to begin to understand the effect of topologies, scheduling mechanisms, and the like on the dead packet ratio, given a possibly-transient scenario with high packet drop rates. This can help us to understand if some scenarios are naturally more resistant to congestion collapse than others. (As a trivial example, scenarios where all paths have at most one congested link are naturally immune to the congestion collapse that comes from a high dead packet ratio on congested links.) This paper is a first step towards the goal of understanding the susceptibility of different topologies, scheduling mechanisms, and the like to congestion collapse.

2 Greedy source model

Let x_r be the sending rate of source r, in packets per second, and let p_r be the end-to-end drop rate experienced by that source. There are then three possible modelling assumptions:

(a) Independent packet drop rate and sending rates:
$\frac{\partial p_r}{\partial x_r} = 0$ and $\frac{\partial x_i}{\partial x_j} = 0$ for all $i \neq j$.
(b) Independent sending rates:
No constraints on $\frac{\partial p_r}{\partial x_r}$ but $\frac{\partial x_i}{\partial x_j} = 0$ for all $i \neq j$.
(c) No independence assumptions:
No constraints on $\frac{\partial p_r}{\partial x_r}$ or $\frac{\partial x_i}{\partial x_j}$.

These three possible assumptions are ordered in sophistication and can be intuitively considered as the following. For assumption (a) of an independent packet drop rate and independent sending rates, the source approximates that its own sending rate doesn't affect either its own packet drop rate or the sending rate of other flows. This assumption could be appropriate for a large number of small flows sending in a FIFO environment.

For assumption (b) of independent sending rates, the source assumes that its sending rate does affect its own packet drop rate, but that it doesn't affect the sending rates of other flows. This assumption could be appropriate for many scenarios in a FQ environment, or for a large flow in a FIFO environment competing against CBR flows.

For the most general assumption, assumption (c) of no independence, the source assumes that its sending rate could affect its own packet drop rate as well as the sending rates of other flows. This would be the assumption in a game-theoretic model, where a source can try to make actions taking into account what happens to the drop rate and the response of other sources to its own actions.

2.1 Greedy senders with general utility functions

Let R be the set of all flows, and let $p_r(x)$ be the end-to-end drop rate for flow r. The cost C_r and the packet drop rate p_r for a flow are both functions of the sending rates x of all of the flows. One might think of the cost C_r as a function of the packet drop rate p_r, which is in turn a function of the sending rates x, but for simplicity we will express C_r directly in terms of x. The cost function C_r can be thought of as expressing the cost of packet drops to the flow, apart from their effect on the received rate. This cost could be caused by the need to retransmit packets or need for forward error correction.

We assume that each greedy sender r is trying to optimize its utility function $U(y)$, where $U(y)$ is a concave function of the received rate y, given that the sender r has a generalized cost function of $C_r(x)$. The sender r attempts to solve the following optimization for the sending rate x_r:

$$\begin{array}{ll} \text{maximize} & U\left(x_r\left(1-p_r(x)\right)\right) - C_r(x) \\ \text{subject to} & x_r \geq 0 \text{ for } r \in R. \end{array}$$

Differentiating with respect to x_r gives the partial derivative

$$U'\left(x_r\left(1-p_r\right)\right)\left(\left(1-p_r\right) - x_r\frac{\partial p_r}{\partial x_r}\right) - \frac{\partial C_r}{\partial x_r}. \quad (1)$$

We assume a FIFO environment, and define an equilibrium point to be that at which no sender has an incentive to alter its rate.

Equating the partial derivative in Equation (1) to zero for each flow r shows that an equilibrium point will have rates x_r such that

$$U'\left(x_r\left(1-p_r\right)\right) = \frac{\partial C_r}{\partial x_r}\frac{1}{1-p_r - x_r\frac{\partial p_r}{\partial x_r}}. \quad (2)$$

Notice that solving such an equation can be difficult for the general case, with the functions $p_r(x)$ and $C_r(x)$ being determined by the queuing disciplines and network routes.

2.2 Packet drop rates with TCP

It will be useful to compare the effects of greedy users against TCP users which are a particular class of greedy senders. The TCP response function which models a sources sending rate, S in packets per second, is given by [7]:

$$S = \frac{1}{T\sqrt{\frac{2p}{3}} + t_{RTO}\left(3\sqrt{\frac{3p}{8}}\right)p(1+32p^2)} \quad (3)$$

where p is the drop rate, T is the round trip time, and t_{RTO} is the round trip timeout value.

3 Packet drop rates with a single resource

In this section we determine the equilibrium drop rate p_e in an environment with greedy senders sharing a single FIFO bottleneck link, assuming a linear cost function $C_r(x) = \alpha x_r p_r$. Section 4 discusses the drop rate with FQ, for this and other environments.

We assume that all connections pass through a single FIFO bottleneck link of bandwidth B in packets per second. In this case

$$p_r = \begin{cases} 0 & \text{if } \sum_{i\in R} x_i \le B; \\ 1 - \frac{B}{\sum_{i\in R} x_i} & \text{if } \sum_{i\in R} x_i > B. \end{cases}$$

For tractability, we assume independent sending rates; that is, we assume that $\frac{\partial x_i}{\partial x_j} = 0$ for $i \neq j$. We are trying to determine a locally stable state, where a greedy sender would not have an incentive to change its sending rate given that other sources do not change their own sending rates in response to each other. No constraints are placed on $\frac{\partial p_r}{\partial x_r}$. For $\sum_{i\in R} x_i > B$, Equation (2) becomes:

$$U'\left(\frac{Bx_r}{\sum_{i\in R} x_i}\right) = \alpha\left(\frac{\sum_{i\in R} x_i}{B}\left(\frac{1}{1-\frac{x_r}{\sum_{i\in R} x_i}}\right) - 1\right). \tag{4}$$

Using this expression, we can analyze the effect of multiple greedy senders with the same utility curves on a single resource. Assume that all senders are sending at the equilibrium rate x_e. If $x_e > \frac{B}{n}$, then the equation above holds, and becomes:

$$U'\left(\frac{B}{n}\right) = \alpha\left(\frac{nx_e}{B}\left(\frac{n}{n-1}\right) - 1\right). \tag{5}$$

Thus the equilibrium rate x_e at which all senders do not have an incentive to change is

$$x_e = \begin{cases} \frac{B}{n} & \text{if } (n-1)U'\left(\frac{B}{n}\right) \le \alpha; \\ \frac{B}{n}\frac{(n-1)}{n}\left(\frac{U'(B/n)}{\alpha} + 1\right) & \text{if } (n-1)U'\left(\frac{B}{n}\right) > \alpha. \end{cases} \tag{6}$$

In the first case, with a larger value for α, drops are costly to the user, and the stable state has no loss.

In the second case, with $(n-1)U'\left(\frac{B}{n}\right) > \alpha$, the resulting equilibrium drop rate p_e will be

$$p_e = 1 - \frac{n}{(n-1)}\frac{1}{\left(\frac{U'(B/n)}{\alpha} + 1\right)}. \tag{7}$$

Holding n constant and letting $\alpha \to 0$, so that the cost of drops becomes arbitrarily small to the user, the equilibrium drop rate rises to 1, as one would expect.

The optimal state, for almost any definition of optimal, would be the state where all users send at rate $\frac{B}{n}$, with a zero drop rate. However, if α is sufficiently small, or n sufficiently large, then this optimal state is not stable with drop-tolerant greedy users and FIFO scheduling.

3.1 Linear utility functions

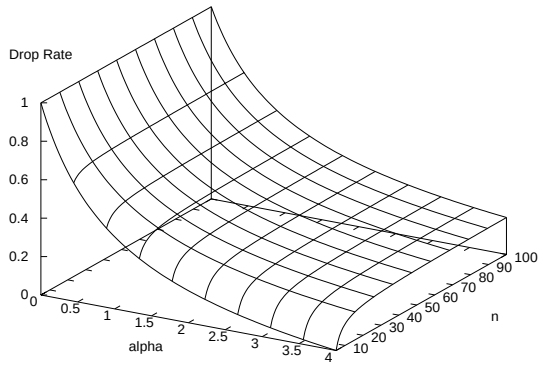

Figure 1: Equilibrium drop rate with the linear utility function.

Given the linear utility function $U(y) = y$, the equilibrium rate at which all senders do not have an incentive to change rates, assuming the link is saturated, is:

$$x_e = \begin{cases} \frac{B}{n} & \text{if } n-1 \le \alpha; \\ \frac{B}{n}\frac{(n-1)}{n}\left(\frac{1}{\alpha}+1\right). & \text{if } n-1 > \alpha; \end{cases}$$

and the equilibrium drop rate p_e is:

$$p_e = \begin{cases} 0 & \text{if } n-1 \le \alpha; \\ 1 - \frac{n}{(n-1)}\frac{1}{1/\alpha+1} & \text{if } n-1 > \alpha. \end{cases}$$

Figure 1 shows the equilibrium drop rate as a function of n and α, for α from 0 to 4, and n from 5 to 100.

Holding α constant and letting $n \to \infty$, the equilibrium drop rate approaches $\frac{1}{1/\alpha+1}$, and the sending rate will approach $\frac{B}{n}\left(\frac{1}{\alpha}+1\right)$. For a linear utility function, p_e is independent of the link capacity B. So when there is a negligible cost for each dropped packet the equilibrium drop rate will remain high for a given number of sources even when more bandwidth is installed!

3.2 Log utility functions

Given the log utility function $U(y) = log(y)$, the equilibrium rate at which all senders do not have an incentive to change rates, assuming the link is saturated, is:

$$x_e = \begin{cases} \frac{B}{n} & \text{if } \frac{n(n-1)}{B} \le \alpha; \\ \frac{B}{n}\frac{(n-1)}{n}\left(\frac{n/B}{\alpha}+1\right). & \text{if } \frac{n(n-1)}{B} > \alpha; \end{cases}$$

and the equilibrium drop rate is:

$$p_e = \begin{cases} 0 & \text{if } \frac{n(n-1)}{B} \le \alpha; \\ 1 - \frac{n}{(n-1)}\frac{1}{\left(\frac{n/B}{\alpha}+1\right)} & \text{if } \frac{n(n-1)}{B} > \alpha. \end{cases}$$

For simplicity, consider both x and B in units of packets per second. Letting $B = 100$ pps, the drop rate p_e is plotted in Figure 2. Figure 3 shows the drop rate for $B = 5000$ pps.

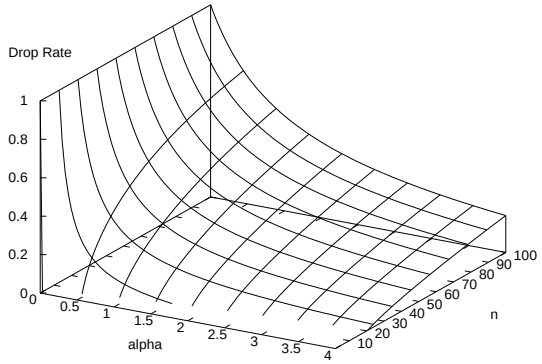

Figure 2: Equilibrium drop rates for the log utility function, $B = 100$ pps.

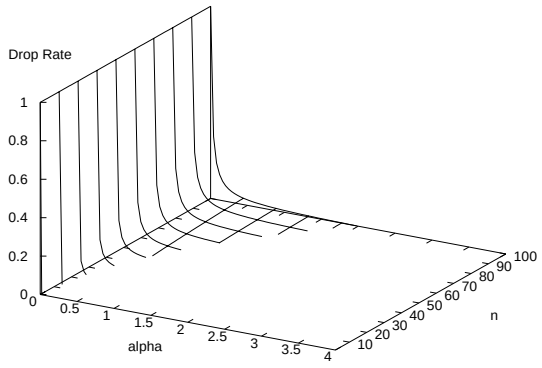

Figure 3: Equilibrium drop rates for the log utility function, $B = 5000$ pps.

Holding α constant and letting $n \to \infty$ the drop rate will rise to 1.

3.3 Polynomial utility functions

Given the polynomial utility function $U(y) = -y^{-m}$ for $m > 0$, the equilibrium rate at which all senders do not have an incentive to change rates, assuming the link is saturated, is

$$x_e = \begin{cases} \frac{B}{n} & \text{if } \frac{mn^{m+1}(n-1)}{B^{m+1}} \le \alpha; \\ \frac{B}{n}\frac{(n-1)}{n}\left(\frac{m(n/B)^m}{\alpha}+1\right) & \text{if } \frac{mn^{m+1}(n-1)}{B^{m+1}} > \alpha; \end{cases}$$

and the equilibrium drop rate is

$$p_e = \begin{cases} 0 & \text{if } \frac{mn^{m+1}(n-1)}{B^{m+1}} \le \alpha; \\ 1 - \frac{n}{(n-1)}\frac{1}{\left(\frac{m(n/B)^m}{\alpha}+1\right)} & \text{if } \frac{mn^{m+1}(n-1)}{B^{m+1}} > \alpha. \end{cases}$$

Holding α constant and letting $n \to \infty$ the equilibrium drop rate will rise to 1.

3.4 TCP drop rates

Assume that n TCP connections, all with the same round-trip time, go through a single link of size B packets per second. Then the following equation holds, with either FIFO or FQ scheduling:

$$\frac{B}{n} = \frac{1}{T\sqrt{\frac{2p}{3}} + t_{RTO}\left(3\sqrt{\frac{3p}{8}}\right)p(1+32p^2)} \qquad (8)$$

Figure 4 shows packet drop rate p as a function of n, for a specific B, T, and t_{RTO}. For these low to moderate loss rates the drop rate is primarily dependent on bandwidth delay product BT/n available to each flow.[4]

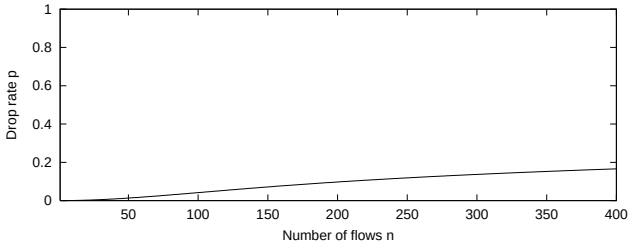

Figure 4: The packet drop rate p for TCP, with n senders ($B = 5000$ packets per second, $T = 0.1$ seconds, and $t_{RTO} = 0.2$s).

Figure 5 displays the same packet drop rate as in Figure 4, with the x-axis extended to show up to 10,000 flows. For this regime of heavy congestion, the packet drop rate is largely determined by TCP's round trip timeout behaviour.

4 Greedy senders in an unpredictable environment

The previous sections explored scenarios where, with FIFO scheduling, the stable state might be one with a non-zero packet drop rate for all flows. In those scenarios the use of FQ scheduling would have been sufficient to ensure that the stable state would have a zero packet drop rate. However in scenarios where the bandwidth available to a flow changes

[4]This can be recovered from Equation (3) by noting that for small drop rates, $p \approx \frac{1.5n^2}{T^2B^2}$.

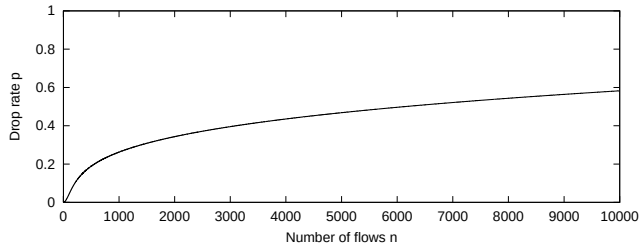

Figure 5: The packet drop rate p for TCP, with n senders ($B = 5000$ packets per second, $T = 0.1$ seconds, and $t_{RTO} = 0.2$s).

in a way that is not predictable in advance, greedy users can have an incentive to maintain a nonzero packet drop rate in an environment with either FIFO or FQ scheduling.

For a FIFO environment, we assume that the rate R of the non-adaptive cross traffic traffic varies independently from one round-trip time to the next, being R_1 with probability p_1 and R_2 with probability $p_2 = 1 - p_1$. We assume that $R_1 \leq R_2$, so that state 1 is the good state for the greedy flow. For a FQ environment, we could model a similar situation by varying the number of cross-traffic flows from one round-trip time to the next. These scenarios of unpredictable bandwidth could be thought of as corresponding to a scenario with varying bandwidth on wireless links, erratic higher-priority traffic, or a changing number of competing flows.

For simplicity, we assume a single greedy sender with linear utility and cost functions. The greedy source will choose a sending rate to maximize

$$r(1 - \mathbb{E}(d)) - \alpha r \mathbb{E}(d) = r\left(1 - (1+\alpha)\left(p_1 d_1 + p_2 d_2\right)\right).$$

where $d_1 = \min\left(0, 1 - \frac{1}{r+R_1}\right)$ is the loss rate when $R = R_1$ and similarly for d_2.

The greedy source's behaviour is equivalent to maximizing the function f, where

$$f(r) = \begin{cases} r & \text{if } r \leq 1 - R_2; \\ g(r) & \text{if } 1 - R_2 < r \leq 1 - R_1; \\ h(r) & \text{if } r > 1 - R_1. \end{cases}$$

The functions $g(r)$ and $h(r)$ are defined by

$$g(r) = r\left(1 - p_2(1+\alpha)\left(1 - \frac{1}{r+R_2}\right)\right), \qquad (9)$$

$$h(r) = r\left(1 - (1+\alpha)\left(p_1\left(1 - \frac{1}{r+R_1}\right) + p_2\left(1 - \frac{1}{r+R_2}\right)\right)\right),$$

These three regions correspond to no congestion, congestion only in state 2, and congestion in both states, respectively.

Let $\tilde{r}$ be the value of r that maximizes $g(r)$ in the second region, with congestion only in state 2, and $\bar{r}$ be the maximum of $h(r)$ in the third region.[5]

The maximum value of $f(r)$ in Equation 4 is then given by $\max\{1 - R_2, f(\tilde{r}), f(\bar{r})\}$. The value $\hat{r}$ that maximizes $f(r)$ is given by whichever of $1 - R_2, \tilde{r}, \bar{r}$ achieves this maximum.

Differentiating h and g in Equation 9 with respect to r gives

$$\begin{aligned} g'(r) &= (1+\alpha)\tfrac{p_2 R_2}{(R_2+r)^2} - (1 - p_2(1+\alpha)) \\ h'(r) &= (1+\alpha)\left(\tfrac{p_1 R_1}{(R_1+r)^2} + \tfrac{p_2 R_2}{(R_2+r)^2}\right) - \alpha. \end{aligned}$$

The value $\tilde{r}$, which maximizes $g(r)$ in the second region, is thus

$$\tilde{r} = \begin{cases} 1 - R_1 & \text{if } \alpha < \frac{1}{p_2} - 1 \text{ or } \alpha < A; \\ 1 - R_2 & \text{if } \alpha \geq \frac{1}{p_2} - 1 \text{ and } \alpha \geq B; \\ \sqrt{\frac{(1+\alpha)p_2 R_2}{(1+\alpha)p_2 - 1}} - R_2 & \text{if } \alpha \geq \frac{1}{p_2} - 1 \text{ and } A \leq \alpha \leq B. \end{cases}$$

for

$$A = \frac{1}{(1 + R_2 - R_1)^2 - R_2}\left((1 + R_2 - R_1)^2\left(1 - \frac{1}{p_2}\right) - R_2\right),$$

$$B = \frac{R_2}{1 - R_2}.$$

The value $\hat{r}$ can be found numerically by determining the root of $h'(r) = 0$.

Figures 6 to 11 display the optimal sending rates and the resulting loss rates for a variety of scenarios, and were computed as above. These figures show that for some scenarios, the unpredictability of bandwidth results in a significant steady-state packet drop rate with greedy senders.

5 Dumb senders in a cyclic topology.

We now consider scenarios with flows that traverse multiple congested links, and the dead packet ratios that arise for different links in the topology.

This section considers a cyclic network as in Figure 12. The network consists of $2K$ links of bandwidth B with M flows of rate R entering at any given point and traversing K links. We assume a packet drop rate p on all links in the cycle, Section 5.1 gives the analysis of the goodput and dead packet ratios, and the following sections give simulation results.

[5] The value $\tilde{r}$ exists as $g(r)$ is continuous on the closed bounded interval $[1 - R_2, 1 - R_1]$. The value $\hat{r}$ exists as $h(r)$ is a concave function on $[1 - R_1, \infty)$.

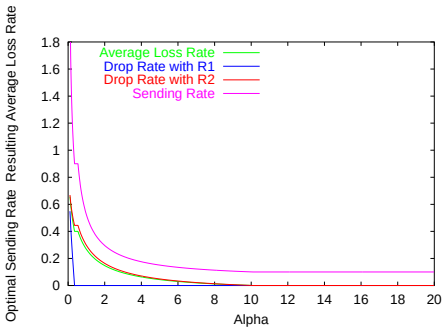

Figure 6: Sending rate and loss rate vs α for $R_1 = 0.1$, $R_2 = 0.9$, $p_1 = 0.1$, and $p_2 = 0.9$

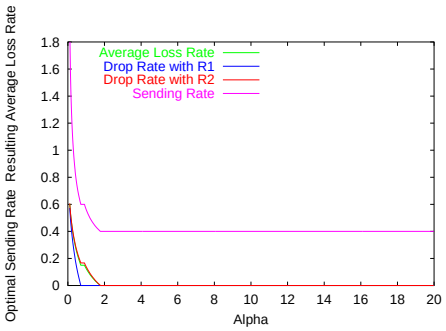

Figure 7: Sending rate and loss rate vs α for $R_1 = 0.4$, $R_2 = 0.6$, $p_1 = 0.1$, and $p_2 = 0.9$

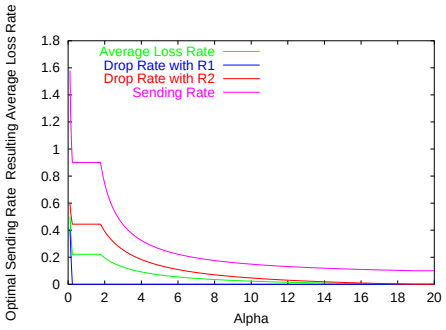

Figure 8: Sending rate and loss rate vs α for $R_1 = 0.1$, $R_2 = 0.9$, $p_1 = 0.5$, and $p_2 = 0.5$

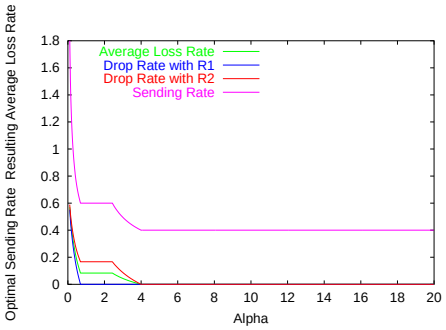

Figure 9: Sending rate and loss rate vs α for $R_1 = 0.4$, $R_2 = 0.6$, $p_1 = 0.5$, and $p_2 = 0.5$

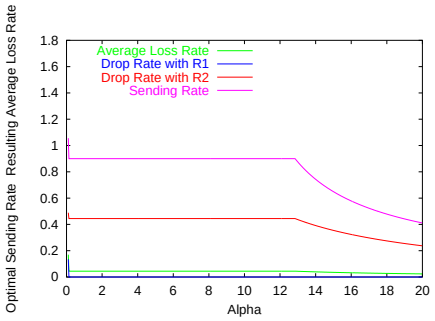

Figure 10: Sending rate and loss rate vs α for $R_1 = 0.1$, $R_2 = 0.9$, $p_1 = 0.9$, and $p_2 = 0.1$

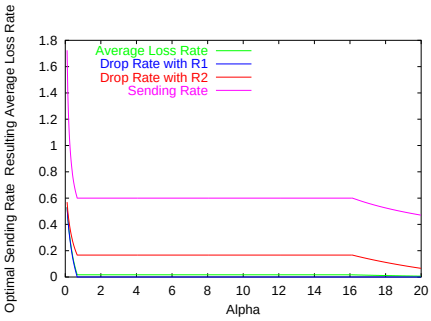

Figure 11: Sending rate and loss rate vs α for $R_1 = 0.4$, $R_2 = 0.6$, $p_1 = 0.9$, and $p_2 = 0.1$

5.1 Analytic Model

For the analysis in this section, we assume that all links in the cycle have a packet loss rate of p, experienced by each packet arriving at the link. Then for a given flow the rate at the i^{th} link is

$$R(1-p)^i.$$

Hence summing over all flows traversing a given link

$$B = MR\sum_{i=1}^{K}(1-p)^i = MR(1-p)\left(\frac{1-(1-p)^K}{p}\right). \tag{10}$$

Solving numerically for p allows the loss rate to be predicted. The simulations later in this section show that the predicted link loss rate corresponds quite closely to the link loss rate in simulations with CBR flows.

Let $\mu = \frac{MR}{B}$, so that μ gives the arrival rate of a set of M flows as a fraction of the link bandwidth. Then the aggregate network goodput measured in packets reaching their network destination is given by

$$\frac{t\sum_{i=1}^{2K} MR(1-p)^K}{s} = \frac{2tK\mu B(1-p)^K}{s}, \tag{11}$$

where t is the time over which the goodput is measured and s is the packet size of the packets in each flow.

Similarly the total load in packets over time t is given by

$$\frac{2KMRt}{s} = \frac{2KB\mu t}{s}. \tag{12}$$

Calculating the loss rate p from Equation 10 allows the total goodput to be calculated.

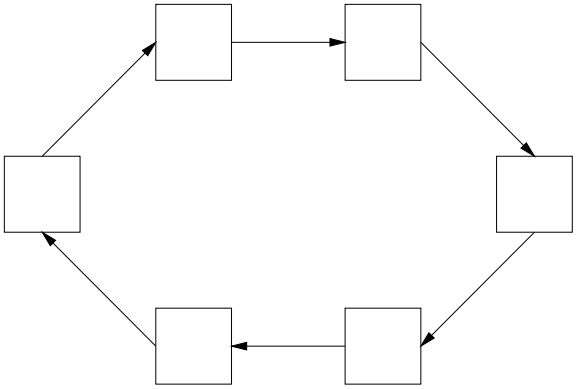

Figure 12: A cyclic network (K=3).

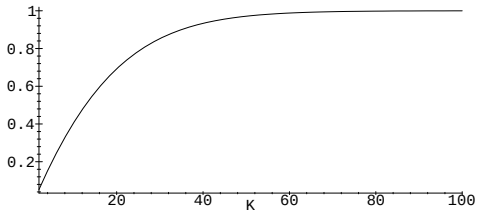

Figure 13: Dead packet ratio as a function of K, for fixed $p = 0.1$.

5.1.1 The dead packet ratio

Once we know the packet drop rate p for the links in the FIFO cyclic topology, we can determine the dead packet ratio for each link. Our results below show that for $K = 1$ the dead packet ratio is zero, and for $K = 2$ the dead packet ratio is $\frac{p}{2-p}$. If the link loss rate p remains fixed, then the dead packet ratio approaches 1 as K increases.

For $K = 1$ the dead packet ratio is zero, since no flow travels multiple congested links. We also compute the dead packet ratio for other values of K, given a fixed value for p for all links in the cycle. Assume that each link L carries flows that are on their ith link of the cycle, for i from 1 to K. The departure rate from the output queue is

$$\sum_{i=1}^{K} x(1-p)^i.$$

The rate on the link consisting of traffic that will be dropped downstream is

$$\sum_{i=1}^{K-1} x(1-p)^i(1-(1-p)^{K-i}).$$

Thus the dead packet ration for a link on the cycle is

$$\frac{1-(1-p)^{K-1}(1+p(K-1))}{1-(1-p)^K}. \qquad (13)$$

This is shown in Figure 13 for $p = 0.1$. For $K = 2$ the dead packet ratio is given by

$$\frac{p}{2-p}.$$

5.2 Simulation Method

We conducted simulations of a variety of scenarios to test the validity of the theoretical model described above. The approach was to simulate a topology in which the links had bandwidth 8 Mbps and delay 20 ms. Each flow was assumed to be consistent with a CBR voice call and so had bandwidth 80 Kbps in packets of size 200 bytes.[6] M such CBR sources were attached to each of the $2K$ nodes and the respective sinks were placed K nodes clockwise from each source. The CBR sources were instructed to start generating traffic at start times chosen uniformly between 0.1 and 0.12 seconds.

Each source packet's arrival time was randomized using a displacement jitter model[7]. In this jitter model, suppose that T_i is the time at which the i^{th} packet of a source is sent. Then for CBR traffic with packet interval t

$$T_i = i.t + D_i$$

where D_i is the random jitter on each packet. In the simulations here, D_i was a sequence of identical independent random variables with uniform distribution on $[0, d]$ with $d = 20ms$. This produced inter-arrival times

$$T_{i+1} - T_i = t + D_{i+1} - D_i$$

The usage of such a displacement jitter model more correctly models network- and machine-scheduling-induced jitter for CBR streams such as voice over IP where the sending rate is ultimately capped by the availability of speech samples.

Simulations were run for a period of 40 seconds, with the first 10 seconds of data discarded to remove startup transients. For each configuration $K \in \{2, 3, 4, 5, 6\}$, the network was simulated for loads of

$$\frac{KMR}{B} = K.\mu = i,$$

where i varied from 0.5 to 2.5 in 0.1 increments. The loads were generated by varying the number of flows M entering the network at a given node.

[6] This is consistent with 8 KHz 8-bit samples and 20 ms frames, giving a payload of 160 bytes with 40 bytes of packet header overhead.

[7] This differs from the standard ns-2 CBR random_ model in which the inter-arrival time is set to $t + t * X_i$ where X_i is distributed uniformly on $[-0.5, 0.5]$.

In order to remove phase effects caused by simulation artifacts, a small quantity of reverse-path traffic was added to each link. This was done by using small one-hop TCP connections along the reverse path of each link, resulting in a small number of TCP acknowledgement packets on the forward path.

For simulations using adaptive RED queue management, with dumb sources sending at rate R, the goodput and dead packet ratios can be computed, and the simulation results match the predictions from analysis.

For simulations with Drop Tail queue management, however, due to synchronization effects, the simulation results failed to match the analysis. In the simulations, one flow would be favored, and as a result overall goodput was higher than predicted. We decided not to pursue the vagaries of traffic dynamics with Drop Tail queue management in further detail.

5.3 A comparison of predicted and simulation results

Figures 14 and 15 present graphs of load (measured in $\mu = \frac{MR}{B}$) verses loss rate. We note that when $\mu = 1/K$, the network is fully loaded with no packet losses. Each point represents a queue's loss rate over the last 30 seconds of the simulations. The line and cross marks represent the predicted loss rate solving Equation 10 numerically. The dots from the simulations are hard to see because they overlap the line with the analytical results. Thus, the analysis predicts the loss rates seen in simulations almost exactly.

Figures 16 and 17 shows the aggregate throughput as a function of the aggregate offered load over the last 30 seconds of the simulations. The maximum possible aggregate throughput over that period is 300,000 packets. Each dot represents the total packets received over the last 30 seconds in simulations. The line and cross marks represent the predicted number of packets received by solving Equation 10 numerically and then substituting p into Equation 11. Again the analysis predicts the simulation results with almost no error. The results show how the aggregate goodput suffers for the cyclic topology as the offered load exceeds the available bandwidth.

5.4 Conclusion

The main conclusions from this section are the following:

- The analysis provides a good model for the goodput in a cyclic topology, given RED queue management.
- Analytic modelling of packet loss rates in simple scenarios requires care, as the behaviour of some queuing disciplines (especially DropTail) does not match the models. The addition of reverse-path traffic and other "realistic" background traffic can help to remove some simulation artifacts but not necessarily all.
- In the cyclic scenario, the aggregate goodput can decrease sharply as the offered load increases, particularly for larger values of K.

6 Greedy senders in a cyclic topology

Suppose now that there are many greedy senders on a cyclic topology with FIFO scheduling at the routers. As in the previous section, we consider a cyclic network in which there are $2K$ nodes connected with links of bandwidth B. At each node in the cycle there are M sources sending at rate x to sources that are K nodes clockwise downstream. By symmetry the drop rate at each link is p. Summing over each flow arriving at a link,

$$M \sum_{i=1}^{K} x(1-p)^i = B,$$

and thus

$$Mx(1-p)\left(\frac{1-(1-p)^K}{p}\right) = B. \qquad (14)$$

For the greedy senders, each source gets utility $U(y)$, where $U(y)$ is a concave function of the received rate y, and experiences a cost of α for each unit of transmitted data dropped. Then the source will attempt to solve the following optimization for the sending rate x:

maximise	$U\left(x\left(1-\tilde{p}\right)\right) - \alpha\tilde{p}x$
subject to	$x \geq 0$
where	$\tilde{p}$ is the end-to-end drop rate.

To simplify the analysis, we assume that each source does not consider the change in its end-to-end drop rate caused by altering its own sending rate; i.e. $\frac{\partial p_r}{\partial x_r} = 0$. This assumption would correspond to a network with a large number of independent sources. The sources will set their rates such that

$$(1-\tilde{p})U'\left(x\left(1-\tilde{p}\right)\right) - \alpha\tilde{p} = 0.$$

If $U(y)$ is a concave function such that $U'(y) \to \infty$ as $y \to \infty$ and $U'(y) \to 0$ as $y \to 0$, then such a rate x will exist.

6.1 Drop rates for log utility functions

Taking $U(y) = \log(y)$, then $U'(y) = \frac{1}{y}$ and so the optimizing source will send at rate $\tilde{x}$ satisfying

$$\frac{(1-\tilde{p})}{\tilde{x}(1-\tilde{p})} - \alpha\tilde{p} = 0.$$

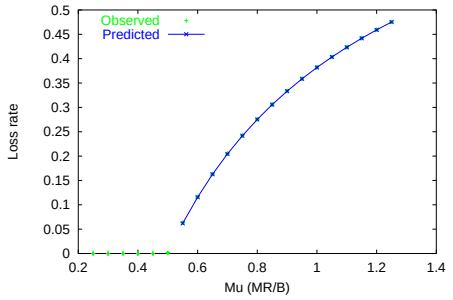

Figure 14: Load vs loss rates for K=2.

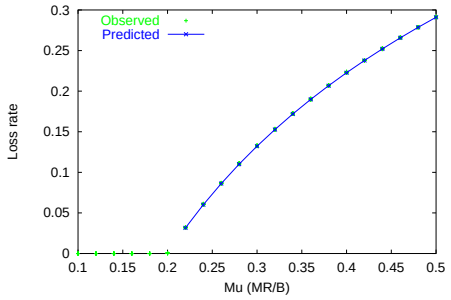

Figure 15: Load vs loss rates for K=5.

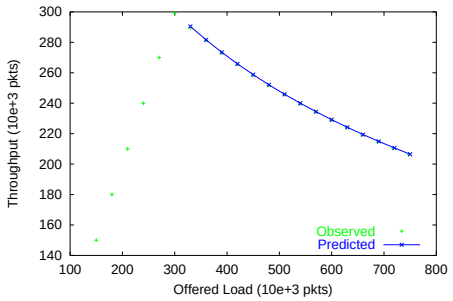

Figure 16: Offered load vs goodput for K=2.

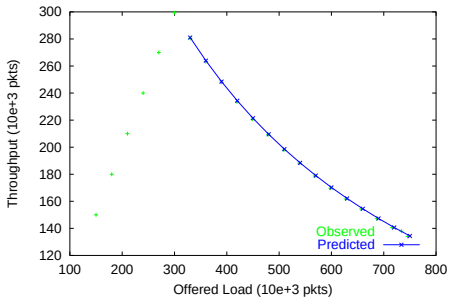

Figure 17: Offered load vs goodput for K=5.

Thus the sources send at

$$\tilde{x} = \frac{1}{\alpha\tilde{p}}. \tag{15}$$

For this cyclic topology, $\tilde{p} = 1 - (1-p)^K$, and so by substituting into Equation 14, the link drop rates are given by

$$p = \frac{1}{(\alpha B/M) + 1}. \tag{16}$$

Notice that this expression (somewhat surprisingly) does not depend on the number K of links that flows traverse. However, using Equation 13, the dead packet ratio is dependent on K and is given by

$$1 - \frac{KM(\alpha B)^{K-1}}{(\alpha B + M)^K - (\alpha B)^K} \tag{17}$$

6.2 Drop rates for polynomial utility functions

Taking $U(y) = -y^{-m}$ for $m > 0$, then $U'(y) = my^{-(m+1)}$ and so an optimizing source will send at rate $\tilde{x}$ satisfying

$$\frac{m}{(x(1-\tilde{p}))^{m+1}} = \frac{\alpha\tilde{p}}{1-\tilde{p}}.$$

Thus the sources will send at

$$\tilde{x} = \left(\frac{m}{\alpha\tilde{p}(1-\tilde{p})^m}\right)^{\frac{1}{m+1}}.$$

For this cyclic topology, $\tilde{p} = 1 - (1-p)^K$, and so, substituting into Equation 14,

$$B = \frac{Mp(1-(1-p)^K)}{p}\left(\frac{m}{\alpha(1-p)^{Km}(1-(1-p)^K)}\right)^{\frac{1}{m+1}}.$$

The link drop rates can be determined using numerical methods to solve this implicit equation for p. The results from Section 5.1.1 can be used to determine the dead packet ratio from p and K.

6.3 Drop rates with Fair Queueing

For this perfectly symmetric topology with greedy users, we show that the use of FQ scheduling would be sufficient to reduce the dead packet ratio on the congested links to zero.

In this topology, the use of FQ instead of FIFO scheduling would mean that instead of each flow experiencing the same drop rate p on each link in the cycle, instead each flow would receive the same per-flow bandwidth share on each link in the cycle. We note that this symmetry of per-flow bandwidth share depends on the underlying symmetry in the number of flows entering at each link, and in the underlying symmetry of the user behaviour. This symmetry of per-flow bandwidth share means that even if the greedy flows had an incentive to send more than this share, this would only result in packets dropped from each greedy flow at its first link in the cycle; no packets would be dropped from those flows on subsequent links in the cycle. Thus, with FQ, the dead packet ratio would be zero in this scenario even with greedy senders.

7 Dumb flows in a railroad topology, with both FQ and FIFO

This section shows scenarios with dumb (non-adaptive) flows in a railroad topology, as shown in Figure 18. We consider the goodput and the dead packet ratio with FQ as well as with FIFO scheduling. It has always been easy to construct scenarios where FQ gives better goodput than FIFO [2]. In this section we explore a wider range of topologies than the few simple topologies considered in [2]. We show that it is easy to construct scenarios where goodput is better with FIFO than with FQ scheduling, as well as the scenarios where goodput is better with FQ. This section is a first step at understanding when the use of FQ scheduling will significantly reduce the dangers of congestion collapse, when the use of FQ will increase the dangers of congestion collapse, and when the use of FQ will make little difference one way or another.

Consider a network with the topology shown in Figure 18, where each of the two congested links has bandwidth B. Each flow has unit bandwidth, with m and n cross-flows and i multi-hop flows.

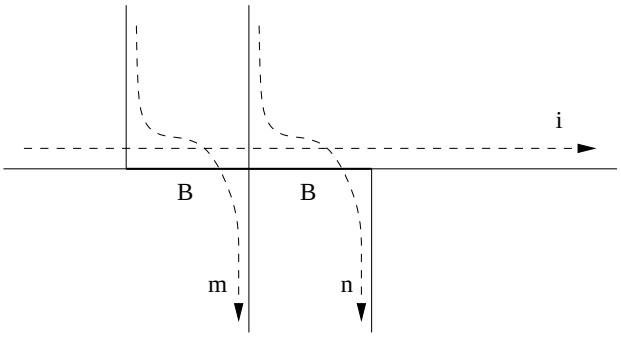

Figure 18: A multi-hop topology

We assume that the demand at each of the two shared links exceeds its capacity. In particular, we only consider cases where $i + m \geq B$, and where n is sufficiently large to result in congestion. With either FQ or FIFO scheduling, the loss rate p_1 at the first congested link is

$$p_1 = 1 - \frac{B}{i+m}. \tag{18}$$

As we show below, the total goodput for this scenario is the same with FQ or FIFO scheduling. (The total goodput is the throughput B in the second congested link, plus the fraction of the link bandwidth on the first congested link used by the m cross-flows.) However, the dead packet ratio for the first link depends on the scheduling discipline. The dead packet ratio for the first link is determined by the fraction of link bandwidth on the second congested link given to the i multi-hop flows.

7.1 All flows the same size

In this section, we use the assumption that all flows have unit bandwidth. We consider the dead packet ratio at the first congested link first with FQ scheduling, and then with FIFO scheduling.

7.1.1 FQ scheduling

In this section we assume that the routers use Fair Queueing scheduling. Because we assume that the demand at each congested link exceeds its capacity, on leaving the first congested link each flow receives $\frac{B}{i+m}$ while on leaving the second congested link each flow receives $\frac{B}{i+n}$. If

$$\frac{B}{i+m} > \frac{B}{i+n}, \tag{19}$$

that is, $n > m$, then there is wasted bandwidth at the first congested link.

Given $n > m$, the total network goodput G is

$$G_{FQ} = B + \frac{mB}{i+m}, \tag{20}$$

with an offered load of $i + m + n$.

With Fair Queueing, the throughput dropped from each long flow at the second congested link is $\frac{B}{i+m} - \frac{B}{i+n}$. Thus with FQ the dead packet ratio $D_{1,FQ}$ at the first congested link is

$$D_{1,FQ} = \frac{i}{(i+m)} - \frac{i}{i+n}.$$

The dead packet ratio is shown in Figure 19 for $i = 10$, for a range of values for m and n. As one might expect, the dead packet ratio is highest when m is small and n is large. Figure 20 shows the dead packet ratio for a different value for the bottleneck link bandwidth B, where the dead packet ratio is small both with FIFO and with FQ.

7.1.2 FIFO scheduling

Suppose that instead the routers use FIFO scheduling. The network goodput is

$$G_{FIFO} = B + (1 - p_1)m = B + \frac{mB}{i+m}, \tag{21}$$

with an offered load of $i + m + n$.

With FIFO scheduling, the second congested link has a total arrival rate $\frac{iB}{i+m} + n$, for a packet drop rate p_2 of

$$p_2 = 1 - \frac{1}{\frac{i}{i+m} + \frac{n}{B}}.$$

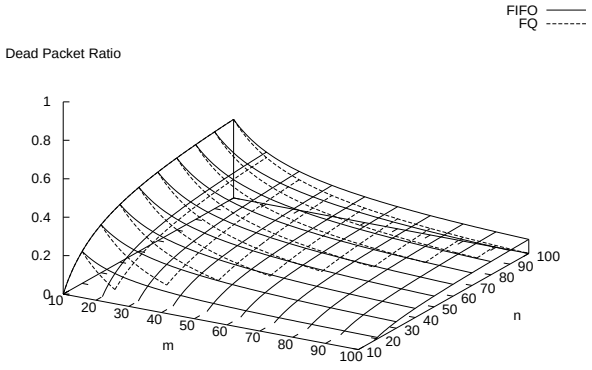

Figure 19: Dead packet ratio for the first congested link, for $i = 10$, $B = 20$.

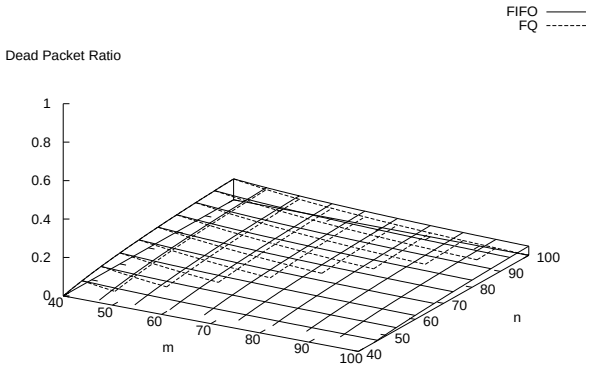

Figure 20: Dead packet ratio for the first congested link, for $i = 10$, $B = 50$.

Thus with FIFO the dead packet ratio at the first congested link is

$$D_{1,FIFO} = p_2 \frac{i}{(i+m)} = \frac{i}{i+m} - \frac{i}{i + \frac{n(i+m)}{B}}.$$

Observe that for $i + m > B$ (that is, assuming saturation at first link), with FIFO the dead packet ratio, $D_{1,FIFO}$, is always at least that for FQ, $D_{1,FQ}$. This is shown in Figure 19 for $i = 10$ and $B = 20$. As Figure 19 shows, for $m = 10$ the long and short flows have the same arrival rate at the second congested link, and the dead packet ratio at the first congested link is the same with FQ as with FIFO. In contrast, for $m > 10$ the long flows each have a smaller arrival rate at the second congested link than the short flows, and therefore receive less of the packet drops with FQ. Thus, for these scenarios, the dead packet ratio at the first congested link is higher with FIFO than with FQ.

7.2 Flows of different sizes

In this section we consider the scenario where the n short flows at the second congested link are each of size $1/k$, for $k \geq 1$, while the other flows are all still of unit size. Given the condition that both shared links remain congested, this change in the size of the n short flows does not change the overall goodput. However, it does change the dead packet ratio on the first congested link, by changing the bandwidth allocated to the i long flows on the second congested link.

With FQ, the dead packet ratio at the first congested link is

$$D_{1,FQ} = \begin{cases} \frac{i}{i+m} - \frac{i}{i+n} & \text{if } \frac{1}{k} > \frac{B}{i+n} \\ \frac{i}{i+m} - \frac{i}{i+n} + \frac{1}{i}\left(\frac{B}{i+n} - \frac{1}{k}\right) & \text{if } \frac{1}{k} \leq \frac{B}{i+n} \end{cases}$$

In the first case, flows at the second congested link receive less than $1/k$ bandwidth, while in the second case flows at the second congested link receive at least $1/k$ each.

With FIFO, given a non-zero drop rate at the second congested link (that is, $\frac{n}{k} > \frac{mB}{i+m}$), we have

$$p_2 = 1 - \frac{1}{\frac{i}{i+m} + \frac{n}{kB}}.$$

Thus with FIFO the dead packet ratio at the first congested link is now

$$D_{1,FIFO} = p_2 \frac{i}{(i+m)} = \left(1 - \frac{1}{\frac{i}{i+m} + \frac{n}{kB}}\right) \frac{i}{i+m}$$

$$= \frac{i}{i+m} - \frac{i}{i + \frac{n(i+m)}{kB}}.$$

Figure 21 compares the dead packet ratio at the first congested link with FIFO and with FQ scheduling. When k is large, the long flows are dropped more heavily at the second congested link under FQ than under FIFO, and in this case the dead packet ratio D_1 is higher with FQ. In particular, if $\frac{i+n}{B} > k > \frac{i+m}{B}$, then the dead packet ratio D_1 is higher with FQ than with FIFO, and if $k < \frac{i+m}{B}$, then the dead packet ratio D_1 is higher with FIFO. (Note that we already have the condition that $\frac{i+m}{B} \geq 1$.) As an example, Figure 21 shows that for $i = 10$, $B = 20$, $k = 2$, and $n \geq 30$, the dead packet ratio is higher with FQ than with FIFO for $m < 30$, and higher with FIFO for $m > 30$.

7.3 Adding another congested link

Now we add a third congested link where the n flows of size $1/k$ compete with r flows each of size $1/j$, with the assumption that all three shared links are fully utilized. The addition of the third congested link does not change the dead packet ratio D_1 at the first congested link. However, it does change the total goodput, with the total goodput now depending on

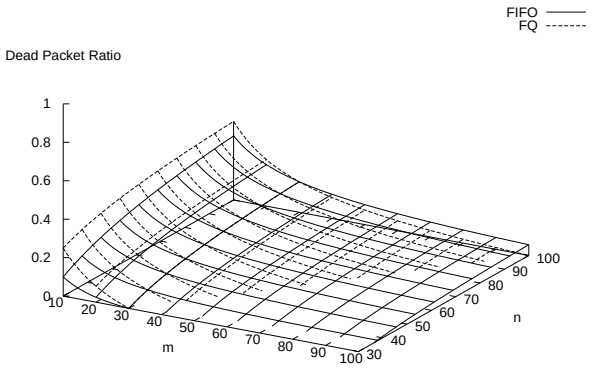

Figure 21: Dead packet ratio for the first congested link, for $i = 10$, $B = 20$, $k = 2$.

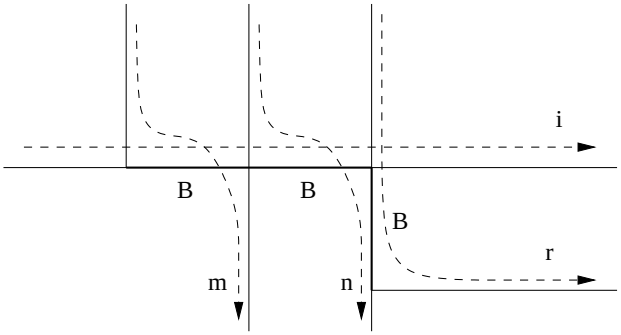

Figure 22: A multi-hop topology with three congested links.

D_1. We show that for some parameters the total goodput in this scenario is higher with FIFO, while for other parameters the total goodput is higher with FQ.

The total goodput now consists of the goodput from the third congested link plus the goodput from the first congested link. Thus with FQ, the total goodput in this case is:

$$G_{FQ} = 2B - BD_{1,FQ},$$

while with FIFO, the total goodput is:

$$G_{FIFO} = 2B - BD_{1,FIFO}$$

Thus, if $k > \frac{i+m}{B}$, then the total goodput is higher with FIFO than with FQ, while for $k < \frac{i+m}{B}$, the goodput will be higher with FQ.

8 Greedy flows in a railroad topology

In the previous section, the dead packet ratio and the overall goodput were explored in scenarios with dumb flows sending at fixed rates, regardless of the packet drop rates experienced by the flows. In Section 3 we showed that drop-tolerant greedy senders can send with high equilibrium packet drop rates in scenarios with FIFO scheduling, or in scenarios with FQ with an unpredictable per-flow bandwidth. Thus, it seems likely that one could have high equilibrium packet drop rates with drop-tolerant greedy senders in general FIFO topologies, or in general FQ topologies with an unpredictable per-flow bandwidth. This section is a first step at exploring the dead packet ratio and the overall goodput in scenarios like those in the previous section, but with greedy instead of with dumb flows.

The ultimate goal would be to consider both predictable and unpredictable per-flow bandwidth, and with both FIFO and FQ scheduling. However, the analysis in this section only considers the case with long-lived flows, with a predictable per-flow bandwidth. For a scenario with greedy flows and predictable bandwidth, the use of FQ should be sufficient to make the stable state be one with zero packet drops. However, for a scenario with greedy flows and unpredictable per-flow bandwidth, the ability of FQ to reduce the steady-state packet drop rate might vary from one scenario to another.

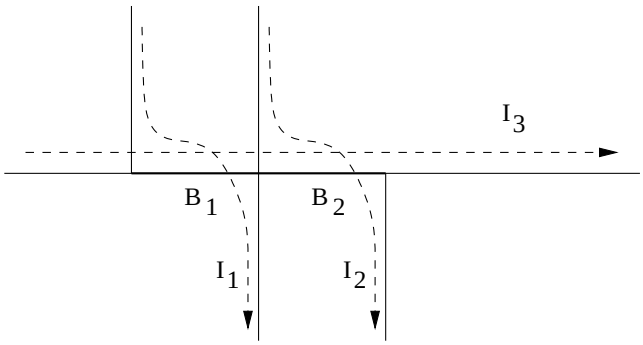

Figure 23: A multi-hop topology with greedy senders

Consider a network in which there are two congested links with greedy flows competing for bandwidth as shown in Figure 23. The index sets I_1, I_2, I_3 represent the routes from left-top to left-bottom, right-top to right-bottom, and left to right respectively. Let p_i be the end-to-end drop rate for flows in index set i, and $U(y)$ be the utility function for each flow as a function of the received rate y.

Using the results of Section 2.1 the equilibrium point will be such that the rates satisfy

$$\begin{aligned}
U'(x_r(1-p_1)) &= \frac{\partial C_r}{\partial x_r}\frac{1}{1-p_1-x_r\frac{\partial p_1}{\partial x_r}} \quad \text{For } r \in I_1;\\
U'(x_r(1-p_2)) &= \frac{\partial C_r}{\partial x_r}\frac{1}{1-p_2-x_r\frac{\partial p_2}{\partial x_r}} \quad \text{For } r \in I_2;\\
U'(x_r(1-p_3)) &= \frac{\partial C_r}{\partial x_r}\frac{1}{1-p_3-x_r\frac{\partial p_3}{\partial x_r}} \quad \text{For } r \in I_3.
\end{aligned} \tag{22}$$

where the drop rates are given by

$$p_1 = \begin{cases} 0 & \text{If } \sum_{r\in I_1\cup I_2} x_r \le B_1; \\ 1 - \frac{B_1}{\sum_{r\in I_1\cup I_2} x_r} & \text{If } \sum_{r\in I_1\cup I_2} x_r > B_1; \end{cases} \quad (23)$$

$$p_2 = \begin{cases} 0 & \text{If } A(x) \le B_2; \\ 1 - \frac{B_2}{\sum_{r\in I_2} x_r + \sum_{r\in I_3}(1-p_1)x_r} & \text{If } A(x) > B_2; \end{cases}$$
$$\text{where } A(x) = \sum_{r\in I_2} x_r + \sum_{r\in I_3}(1-p_1)x_r, \quad (24)$$

$$p_3 = 1 - (1-p_1)(1-p_2). \quad (25)$$

Such a model becomes difficult to analyse in full generality, so we will assume that each greedy sender is a small enough amount of aggregate traffic that $\frac{\partial p_r}{\partial x_r} = 0$ holds and that the cost function is given by $C(x) = \alpha x_r p_r$. Equation (22) becomes

$$\begin{array}{rcll} U'(x_r(1-p_1)) & = & \frac{\alpha p_1}{1-p_1} & \text{For } r \in I_1; \\ U'(x_r(1-p_2)) & = & \frac{\alpha p_2}{1-p_2} & \text{For } r \in I_2; \\ U'(x_r(1-p_3)) & = & \frac{\alpha p_3}{1-p_3} & \text{For } r \in I_3. \end{array} \quad (26)$$

Notice that the dead packet ratio for this topology is given by

$$\frac{\left(\sum_{r\in I_3} x_r(1-p_1)\right) p_2}{B_1}. \quad (27)$$

8.1 Log utility

With a log utility function $U(y) = log(y)$ of the received rate y,the equilibrium point will be such that

$$\begin{array}{rcll} x_r & = & \frac{1}{\alpha p_1} & \text{For } r \in I_1; \\ x_r & = & \frac{1}{\alpha p_2} & \text{For } r \in I_2; \\ x_r & = & \frac{1}{\alpha(1-(1-p_1)(1-p_2))} & \text{For } r \in I_3. \end{array} \quad (28)$$

8.2 Polynomial utility functions

With a polynomial utility functions of the form $U(y) = -y^{-m}$ with $m > 0$ where y is the received rate the equilibrium point will be such that

$$\begin{array}{rcll} x_r & = & \left(\frac{m}{\alpha p_1(1-p_1)^m}\right)^{\frac{1}{m+1}} & \text{For } r \in I_1; \\ x_r & = & \left(\frac{m}{\alpha p_2(1-p_2)^m}\right)^{\frac{1}{m+1}} & \text{For } r \in I_2; \\ x_r & = & \left(\frac{m}{\alpha(1-(1-p_1)(1-p_2))((1-p_1)(1-p_2))^m}\right)^{\frac{1}{m+1}} & \text{For } r \in I_3. \end{array} \quad (29)$$

9 Related Work

System equilibrium and resource allocations have been studied extensively as network optimisation problems in which load is *transmissive* [4, 5, 6] ; for example in an ECN network the load applied on a route is felt at all links that form the route. The study of congestion collapse is inherently the study of a non-transmissive system. This paper also considers scenarios in which a greedy flow may change its sending rate in response taking into account its own impact on the drop rate it experiences.

Bonald and Massoulie [1] consider how network resource allocation (or fairness) affects network congestion at the level of flow arrivals and departures. They consider a dynamic population of short-lived transfers, and explore the conditions under which response times of transfers remain finite. The paper shows that for arbitrary network topologies and a broad class of fair bandwidth allocations, response times remain bounded if and only if the load offered to each link is less than one. In contrast simple examples are given with class-based scheduling, such as fair queueing, where the number of flows in progress can grow unbounded while the average arriving load on each link is strictly, and sometimes greatly, less than one.

10 Conclusions

In the first half of this paper, we have explored the equilibrium packet loss rates that can arise in a range of scenarios, and in the second half we have explored the potential for loss of overall goodput as a result of this equilibrium packet loss rate. In particular, the first half of the paper has demonstrated the following:

- The equilibrium sending rates and packet loss rates for scenarios with n greedy users sharing a FIFO link are given in Section 3, for a range of values for the utility and cost functions. For example, for greedy users sharing a FIFO link the equilibrium packet loss rate can approach 1 as n approaches ∞.

- The results in Section 3 of a high equilibrium packet loss rate with greedy users applies only to the scenario with FIFO scheduling. The use of FQ scheduling would be sufficient, given n long-lived flows with unlimited demand, to ensure a zero packet loss rate in those scenarios. In Section 4 we added bursty, non-adaptive cross traffic, either of one flow or of many coordinated small flows, to give the greedy users an incentive to maintain a nonzero packet loss rate even in an environment with FQ scheduling. In these scenarios with bursty, non-adaptive cross traffic, the equilibrium packet loss rate depends on the cost functions of the users, but with loss-tolerant greedy users the equilibrium loss rates can be high even with FQ scheduling.

- The overall result of the first half of the paper is that it is possible to have high equilibrium loss rates with greedy senders sharing a FIFO link, or with greedy senders sharing an FQ link with bursty cross-traffic.

With greedy senders, the equilibrium loss rates depend heavily on the utility functions and cost functions of the end users.

In the second half of the paper we consider scenarios with many senders with high sending rates and high loss rates, in topologies with multiple congested links, and consider the goodput of the congested links (where the goodput includes only those packets that will not be dropped on other congested links downstream). The results include the following:

- Section 5 considers aggressive senders in a cyclic network with FIFO scheduling, where each flow traverses K links in a cycle, and evaluates the link goodput using both analysis and simulations. In this scenario, given a fixed equilibrium packet drop rate at each link, the link goodput approaches 0 as K approaches ∞. (We note that the per-flow end-to-end packet loss rate also approaches 100% in this case.)

- Section 7 considers aggressive senders in a railroad topology, and compares goodput with FIFO and with FQ scheduling. In addition to the known simple scenarios where the goodput is higher with FQ scheduling, this section gives scenarios where the goodput is higher with FIFO scheduling. For the topologies in Sections 7.2 and 7.3, adjusting simple parameters like the number or size of competing flows controls whether the goodput is higher with FQ scheduling or with FIFO scheduling.

We would note that this paper does not pretend to consider realistic topologies, realistic models of competing bursty traffic, or realistic cost and utility functions for greedy senders. Thus, this work can not be used to make predictions about packet loss rates or goodput levels in realistic scenarios. We see this work as a small step towards understanding the potential loss of goodput from high steady-state packet drop rates in topologies larger and more complex than the simple topologies of a single congested link or a single string of congested links.

This paper is motivated in part by a concern about the potential for congestion collapse, or of a simple loss of goodput, in the presence of greedy, loss-tolerant traffic. The point is not that there is anything wrong with scenarios with greedy traffic, or with FQ instead of FIFO scheduling. The point is simply that even with greedy traffic, or even with a topology with FQ scheduling, it is important for senders to avoid persistent high packet drop rates.

This work leaves open the analysis of the effects of sources which have non-concave utility functions. It should be noted that this is still an open issue in the case of transmissive load systems. It should also be noted that despite considerable effort this work has not been able to provide a generalized calculation method for determining system equilibriums in general topologies with greedy sources. Methods for analyzing transmissive load systems (e.g. an ECN network) given in [4, 6, 5] are able to make more progress on calculating system equilibriums.

References

[1] T. Bonald and L. Massoulie. Impact of Fairness on Internet Performance. In *ACM Sigmetrics 2001*, Cambridge, Massachusetts, 2001.

[2] S. Floyd and K. Fall. Promoting the Use of End-to-End Congestion Control. *IEEE/ACM Transactions on Networking*, August 1999.

[3] G. Hardin. The Tragedy of the Commons. *Science*, 162:1243–1248, 1968. `http://dieoff.org/page95.htm`.

[4] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan. Rate Control in Communication Networks: Shadow Prices, Proportional Fairness and Stability. *Journal of the Operational Research Society*, 49:237–252, 1998.

[5] S. Kunniyur and R. Srikant. End-to-End Congestion Control Schemes: Utility Functions, Random Losses and ECN Marks. In *IEEE INFOCOM 2000*, Tel Aviv, Israel, March 2000.

[6] S. H. Low and D. E. Lapsley. Optimization Flow Control, I: Basic Algorithm and Convergence. *IEEE/ACM Transactions on Networking*, 7(6):861–875, December 1999.

[7] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP throughput: A simple model and its empirical validation. In *ACM SIGCOMM 98*, Vancouver, British Columbia, 1998.