Understanding CHOKe

Ao Tang Jiantao Wang Steven H. Low
California Institute of Technology, Pasadena, USA
{aotang@, jiantao@cds., slow@}caltech.edu

***Abstract*— A recently proposed active queue management, CHOKe, is stateless, simple to implement, yet surprisingly effective in protecting TCP from UDP flows. As UDP rate increases, even though the number of UDP packets in the queue rises, its bandwidth share eventually drops to zero, in stark contrast to the behavior of a regular FIFO buffer. We derive a detailed model of CHOKe that accurately predicts this, and other behaviors of CHOKe, and validate the model with simulations. Its key features are the incorporation of the feedback equilibrium of TCP with dropping probability and the spatial characteristics of the queueing process. CHOKe produces a "leaky buffer" where packets can be dropped as they move toward the head of the queue. This leads to a spatially non-uniform distribution of packets and their velocity, and makes it possible for a flow to simultaneously maintain a large number of packets in the queue and receive a vanishingly small bandwidth share. This is the main mechanism through which CHOKe protects TCP from UDP flows.**

I. Introduction

TCP is believed to be largely responsible for preventing congestion collapse while the Internet has undergone dramatic growth in the last decade. Indeed, numerous measurements have consistently shown that more than 90% of traffic on the current Internet are still TCP packets, which, fortunately, are congestion controlled. Without a proper incentive structure, however, this state of affair is fragile and can be disrupted by the growing number of non-rate-adaptive (e.g., UDP-based) applications that can monopolize network bandwidth to the detriment of rate-adaptive applications. This has motivated several active queue management schemes, e.g., [6], [3], [4], [13], [8], [10], [2], that aim at penalizing aggressive flows and ensuring fairness. The scheme, CHOKe, of [10] is particularly interesting in that it does not require any state information and yet can provide a minimum bandwidth share to TCP flows. In this paper, we provide an analytical model of CHOKe that explains its spatial characteristics.

The basic idea of CHOKe is explained in the following quote from [10]:

> When a packet arrives at a congested router, CHOKe draws a packet at random from the FIFO (first-in-first-out) buffer and compares it with the arriving packet. If they both belong to the same flow, then they are both dropped; else the randomly chosen packet is left intact and the arriving packet is admitted into the buffer with a probability that depends on the level of congestion (this probability is computed exactly as in RED).

Appeared in Proceedings of IEEE Infocom, San Francisco, CA, April 2003.

The surprising feature of this extremely simple scheme is that it can bound the bandwidth share of UDP flows regardless of their arrival rate. In fact, as the arrival rate of UDP packets increases without bound, their bandwidth share approaches zero! An intuitive explanation is provided in [10]: "the FIFO buffer is more likely to have packets belonging to a misbehaving flow and hence these packets are more likely to be chosen for comparison. Further, packets belonging to a misbehaving flow arrive more numerously and are more likely to trigger comparisons." As a result, aggressive flows are penalized. This however does not explain why a flow that maintains a much larger number of packets in the queue does not receive a larger share of bandwidth, as in the case of a regular FIFO buffer. It turns out that a precise understanding of this phenomenon requires a detailed analysis of the queue dynamics, the key feature of our model. A simple model of CHOKe is also presented in [10] that assumes a Poisson packet arrival process and exponential service time. The Poisson assumption is critical in order to use the PASTA (Poisson Arrival Sees Time Averages) property to compute drop probabilities. This model however ignores both the feedback equilibrium of the TCP/CHOKe system and the spatial characteristics of the queue.

Here, we adopt a deterministic fluid model that explicitly models both features (Section II). Our model predicts, and simulations confirm, that as UDP rate becomes large, not only does the total number of UDP packets in the queue increase, more importantly, the *spatial* distribution of UDP packets becomes more and more concentrated near the tail of the queue, and drops rapidly to zero toward the head of the queue. Hence even though the total number of UDP packets in the queue is large, all of them will be dropped before they advance to the head. As a result the UDP bandwidth share drops to zero, in stark contrast to a non-leaky FIFO buffer where UDP bandwidth shares would approach 1 as its input rate increases without bound.

Our current model is too complex to be solved analytically. We outline a numerical solution (Section II-C), and compare our numerical results with detailed ns-2 simulations. They match accurately not only with average behavior of CHOKe as reported in [10], but also with much finer spatial characteristics of the queueing process.

The technique presented here should be applicable to analyzing the queueing process in other types of leaky buffer.

II. Model

In general, one can choose more than one packet from the queue, compare all of them with the incoming packet, and drop those from the same flow. This will improve CHOKe's performance, especially when there are multiple unresponsive sources

[10]. Here, we focus on the modeling of a single drop candidate packet. The analysis can be extended to the case of multiple drop candidates.

The general setup for our model and our simulations is shown in Figure 1. We focus on the single bottleneck FIFO buffer at

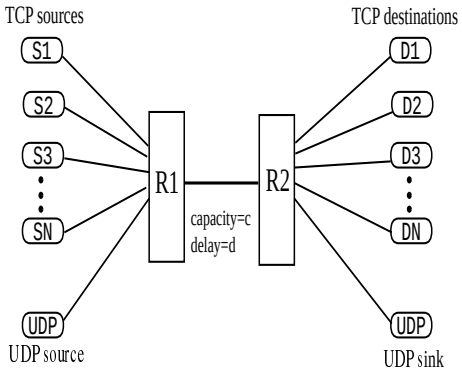


Fig. 1. Network topology

router R1 where packets are queued and drained at a rate of c packets per second. The buffer is shared by N identical TCP flows and a single UDP flow. All TCP flows have a common round trip propagation delay of d seconds. We assume the system is stable and model its *equilibrium* behavior.

A. Notations

Quantities (rate, backlog, dropping probability, etc) associated with the UDP flow are indexed by 0. Those associated with TCP flows are indexed by $i = 1, \ldots, N$. Since the TCP sources are identical, these quantities all have the same value, and hence we will refer to flow 1 as the generic TCP flow. These are *equilibrium* quantities which we assume exist.

We collect here the definitions of all the variables and some of their obvious properties:

b_i: packet backlog from flow i, $i = 0, 1$.

b: total backlog; $b = b_0 + b_1 N$.

r: Congestion based dropping probability. The spatial properties of CHOKe are insensitive to the specific algorithm, such as RED, to compute this probability, as long as it is the same for all flows. In general, $r = g(b, \tau)$ for some function g as a function of aggregate backlog b and queueing delay τ.

h_i: The probability that an incoming packet of flow i, $i = 0, 1$, is dropped by CHOKe:

$$h_i \quad = \quad \frac{b_i}{b}$$

p_i: overall probability that a packet of flow i, $i = 0, 1$, is dropped before it gets through, either by CHOKe or RED:

$$p_i \quad = \quad 2h_i + r - rh_i \tag{1}$$

The explanation of (1) is provided below.

x_i: source rate of flow i, $i = 0, 1$. The spatial properties of CHOKe are insensitive to the specific TCP algorithm, such as Reno or Vegas. In general, $x_1 = f(p_1, \tau)$ for some function f as a function of overall loss probability p_1 and queueing delay τ at equilibrium.

τ: common round-trip queueing delay. Round-trip time is $d + \tau$.

It is important to keep in mind that x_0 is the only independent variable; all other variables listed above are functions of x_0, though this is not made explicit in the notations.

B. TCP/CHOKe model

A packet may be dropped, either on arrival due to CHOKe or congestion (e.g., according to RED), or after it has been admitted into the queue when a future arrival from the same flow triggers a comparison. Let p_i be the probability that a packet from flow i is eventually dropped. To see why p_i is related to CHOKe and RED dropping probabilities according to (1), note that every arrival from flow i can trigger either 0 packet loss from the buffer, 1 packet loss due to RED, or 2 packet losses due to CHOKe. These events happen with respective probabilities of $(1 - h_i)(1 - r)$, $(1 - h_i)r$, and h_i. Hence, each arrival to the buffer is accompanied by an average packet loss of

$$2h_i + (1 - h_i)r + 0 \cdot (1 - h_i)(1 - r)$$

Hence we take the overall loss probability p_i to be the packet loss rate $2h_i + (1 - h_i)r$. We now derive this probability from another perspective.

Consider a packet of flow i that eventually goes through the queue without being dropped. The probability that it is not dropped on arrival is $(1 - r)(1 - h_i)$. Once it enters the queue, it takes τ time to go through it. In this time period, there are on average τx_i packets from flow i that arrive at the queue. The probability that this packet is not chosen for comparison is

$$\left(1 - \frac{1}{b}\right)^{\tau x_i}$$

Hence, the overall probability that a packet of flow i survives the queue is

$$1 - p_i \quad = \quad (1 - r)(1 - h_i)\left(1 - \frac{1}{b}\right)^{\tau x_i} \tag{2}$$

A simple interpretation of a leaky buffer is as follows: x_i is the source rate of flow i and $x_i(1 - r)(1 - h_i)$ is the rate at which flow i enters the queue after CHOKe and congestion-based dropping. This flow splits into two flows: one eventually exits the queue and the other is dropped inside the queue by CHOKe. The rate of the former flow is flow i's *throughput* $x_i(1 - p_i)$ and the rate of the latter flow is its *leak rate* $x_i h_i$, so that they sum to the input rate $x_i(1 - r)(1 - h_i)$. Since the link is fully utilized, the flow throughputs sum to link capacity:

$$x_0(1 - p_0) + Nx_1(1 - p_1) \quad = \quad c$$

This completes the description of the model. In summary, the independent variable is UDP rate x_0. The ten dependent variables of the model are:

- backlogs b_i of flow i, $i = 0, 1$; total backlog $b = b_0 + Nb_1$.

- congestion based dropping probability r, CHOKe dropping probabilities h_i, and overall dropping probabilities p_i, $i = 0, 1$.
- TCP rate x_1 and queueing delay τ.

The relations among these variables define our model. For ease of reference, we reproduce these ten equations here:

$$p_i = 2h_i + r - rh_i, \quad i = 0, 1 \tag{3}$$

$$p_i = 1 - (1-r)(1-h_i)\left(1 - \frac{1}{b}\right)^{\tau x_i}, \quad i = 0, 1 \tag{4}$$

$$h_i = \frac{b_i}{b}, \quad i = 0, 1 \tag{5}$$

$$b = b_0 + Nb_1 \tag{6}$$

$$c = x_0(1-p_0) + Nx_1(1-p_1) \quad \text{(full utilization)} \tag{7}$$

$$x_1 = f(p_1, \tau) \quad \text{(TCP)} \tag{8}$$

$$r = g(b, \tau) \quad \text{(e.g. RED)} \tag{9}$$

Let $z = (p_0, p_1, h_0, h_1, b_0, b_1, b, x_1, r, \tau)$ denote the ten dependent variables. Then the above equations (3)–(9) can be expressed as

$$F(z, x_0) = 0 \tag{10}$$

This can be regarded as implicitly defining z as a function of x_0. We assume:

A1: Given any $x_0 \geq 0$, there is a unique solution $z(x_0)$ that satisfies (10).

C. Numerical solution of TCP/CHOKe model

The set of nonlinear equations (3)–(9) that models the TCP/CHOKe system can be solved numerically. The solution can then be used in the differential equation model described below to solve for spatial properties of the leaky buffer under CHOKe; see Section III.

The nonlinear equation (10) can be solved by minimizing the quadratic cost:

$$\min_z \; J(z) := F(z, x_0)^T W F(z, x_0)$$

with an appropriate choice of positive diagonal weighting matrix W. A solution z^* of TCP/CHOKe satisfies $J(z^*) = \min_z J(z) = 0$.

Matlab is used to implement the above procedure. The weighting matrix is chosen such that each component in the vector Wz is in the range [10 100] near the fixed point. A direct search method [7] for multidimensional unconstrained nonlinear minimization implemented in matlab is used for this optimization problem. The search algorithm is stopped when $J(z)$ is smaller than 0.05. The solution is accurately validated with ns-2 simulations in section IV.

III. Spatial characteristics

A. Spatial distribution and packet velocity

If a packet cannot be dropped once it has been admitted into a FIFO queue, then, clearly, the queueing delay τ and bandwidth share μ_i are

$$\tau = \frac{b}{c} \quad \text{and} \quad \mu_i = \frac{b_i}{b} \tag{11}$$

For a leaky buffer where a packet can be dropped while it advances toward the head of the queue, (11) no longer holds, and the queueing delay and bandwidth share depend critically on the spatial characteristics of the queue. The key to their understanding is the spatial distribution of packets in the queue and the flow rate (velocity at which packets move through the queue) at different positions in the queue. We now define these two quantities and relate them to the variables previously defined.

Let $y \in [0, b]$ denote a position in the queue, with $y = 0$ being the tail and $y = b$ the head of the queue. In a leaky buffer, the queueing delay of a packet that *eventually* exits the queue is no longer the backlog it sees on arrival divided by the link capacity. This is because it advances toward the head both when packets in front of it exit the queue and when they are dropped by CHOKe. To model this dynamics, define $v(y)$ as the velocity at which the packet at position y moves toward the head of the queue:

$$v(y) = \frac{dy}{dt}$$

For instance, the velocity at the head of the queue equals the link capacity, $v(b) = c$. Then, the queueing delay τ is given in terms of $v(y)$ as

$$\tau = \int_0^\tau dt = \int_0^b \frac{1}{v(y)} dy \tag{12}$$

More generally, define, for $x_0 \geq 0$, $\tau(y)$ by

$$\tau(y) = \int_0^y \frac{1}{v(s)}\, ds \tag{13}$$

which can be interpreted as the time for a packet to reach position y from position 0. Clearly, $\tau(b) = \tau$.

Let $\rho_i(y)$ be the probability that the packet at position y belongs to flow i, $i = 0, 1$. As usual, we have

$$\rho_0(y) + N\rho_1(y) = 1, \quad \text{for all } y \in [0, b] \tag{14}$$

The average number of flow i packets in the entire backlog satisfies

$$b_i = \int_0^b \rho_i(y) dy \tag{15}$$

More importantly, the bandwidth share μ_i is the probability that the head of the queue is occupied by a packet from flow i:

$$\mu_i = \rho_i(b) \tag{16}$$

Note that if the queue is not leaky, then the spatial distribution of packets will be uniform, $\rho_i(y)$ being independent of position y:

$$\rho_i(y) = \rho_i(b) \quad \text{for all } y \in [0, b]$$

This, together with (15), implies the bandwidth share in (11), i.e., the bandwidth share depends only on the total number of flow i packets in the queue. When the queue is leaky, however,

the spatial distribution can be highly non-uniform. The bandwidth share $\rho_i(b)$ of flow i depends on the spatial distribution of packets only at the *head* of the queue and does not depend directly on the distribution at other positions or the total number of flow i packets, in stark contrast to the case of non-leaky buffer. This is the underlying reason why UDP packets can occupy almost half of the queue, yet receiving very small bandwidth share: when UDP rate is high, $\rho_0(y)$ decreases rapidly from $y = 0$ to $y = b$ with $\rho_0(b) \simeq 0$; see Section III-C.

We have completed the definition of spatial distribution $\rho_i(y)$ and velocity $v(y)$ of packets in the queue. We now derive an ordinary differential (ODE) equation model of these quantities.

B. ODE model of $\rho_i(y)$ and $v(y)$

We will derive an ODE model for $\rho_0(y)$ and $v(y)$; $\rho_1(y)$ can be obtained from (14).

Consider a small volume $v(y)dt$ of the (one-dimensional fluid) queue at position y. The amount of fluid (packets) in this volume that belongs to flow i is $\rho_i(y)v(y)dt$, $i = 0, 1$. For instance, $\rho_i(0)v(0)dt$, $i = 0, 1$, is the amount of fluid that arrives at the tail of the queue, packets that are not dropped by CHOKe or congestion based dropping on arrival and admitted into the buffer. Hence

$$\rho_i(0)v(0) \quad = \quad x_i(1-r)(1-h_i), \quad i = 0, 1 \qquad (17)$$

Another boundary condition is the packet velocity at the head of the queue mentioned above:

$$v(b) \quad = \quad c \qquad (18)$$

Suppose the small volume $\rho_i(0)v(0)dt$ of fluid (our "tagged packet") arrives at the buffer at time 0, and reaches position y at time $\tau(y)$. During this period $[0, \tau(y)]$, there are $x_i\tau(y)$ packet arrivals from flow i, and each of these arrivals triggers a comparison. The tagged packet is selected for comparison with probability $1/b$ each time. We model this by saying that the fluid is thinned by a factor $(1 - 1/b)^{x_i\tau(y)}$ when it reaches position y at time $\tau(y)$. Thus

$$\rho_i(0)v(0)\left(1 - \frac{1}{b}\right)^{x_i\tau(y)} \quad = \quad \rho_i(y)v(y) \qquad (19)$$

Note that this is the same argument that leads to (2).

Taking logarithm on both sides and using (13) to eliminate $\tau(y)$, we have

$$\ln(\rho_i(y)v(y)) \quad = \quad \ln(\rho_i(0)v(0)) + \beta x_i \int_0^y \frac{1}{v(s)}\, ds$$

where

$$\beta \quad := \quad \ln\left(1 - \frac{1}{b}\right)$$

Differentiating both sides with respect to y, we get

$$\beta\frac{x_0}{v(y)} \quad = \quad \frac{\rho_0'(y)}{\rho_0(y)} + \frac{v'(y)}{v(y)} \qquad (20)$$

$$\beta\frac{x_1}{v(y)} \quad = \quad \frac{\rho_1'(y)}{\rho_1(y)} + \frac{v'(y)}{v(y)} \qquad (21)$$

Now (20) $\times\, \rho_0(y) + (21) \times N \times \rho_1(y)$ yields

$$v'(y) \quad = \quad \beta\left(\rho_0(y)x_0 + (1 - \rho_0(y))x_1\right) \qquad (22)$$

where we have used (14). Substituting (22) into (20), we get

$$\rho_0'(y) = \beta(x_0 - x_1)\,\rho_0(y)(1 - \rho_0(y))\,\frac{1}{v(y)} \qquad (23)$$

Hence the spatial distribution $\rho_i(y)$ and packet velocity $v(y)$ is given by the two-dimensional system of nonlinear differential equations (22)–(23) with boundary conditions (17)–(18). Since the right-hand sides of (22)–(23) are continuously differentiable in (v, ρ_0), there exists a unique solution in its interval of existence [11].

We make an important remark. Given x_0, quantities such as τ, b, b_i are uniquely determined by (10) by assumption A1. At the same time $v(y)$ and $\rho_0(y)$ are uniquely determined by the differential equations (22)–(23) with boundary conditions (17)–(18). The relations (12) and (15) between these two sets of quantities are not necessarily true a priori. Even though they seem reasonable based on their physical interpretation, they nonetheless remain a postulation:

A2: Relations (12) and (15) hold.

Note that (14) holds without assumption and defines $\rho_1(y)$.

C. Structural properties

In this subsection, we prove some structural properties of the velocity $v(y)$ and spatial distribution $\rho_0(y)$. They are illustrated in Figure 2.

Theorem 1. *For all $x_0 \geq 0$, packet velocity $v(y)$ is a convex and strictly decreasing function with $v(0) = (x_0(1 - h_0) + Nx_1(1 - h_1))(1 - r)$ and $v(b) = c$. It is linear if and only if $x_0 = x_1$.*

Proof. Using (14), (22) can be rewritten as

$$v'(y) \quad = \quad \beta(\rho_0(y)x_0 + N\rho_1(y)x_1) \quad < \quad 0$$

where $\beta = \ln(1 - 1/b) < 0$. Hence $v(y)$ is strictly decreasing.

Differentiating again and using $\rho_0'(y) + N\rho_1'(y) = 0$ from (14), we have

$$\begin{aligned} v''(y) \quad &= \quad \beta(\rho_0'(y)x_0 + x_1\rho_1'(y)N) \\ &= \quad \beta(x_0 - x_1)\rho_0'(y) \end{aligned}$$

From (23), we have

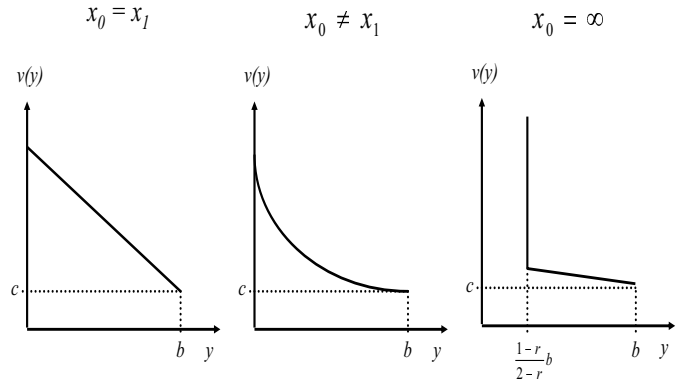
$$v''(y) \quad = \quad \beta^2(x_0 - x_1)^2\rho_0(y)\rho_1(y)\frac{N}{v(y)} \quad \geq \quad 0$$

with equality if and only if $x_0 = x_1$. Hence $v(y)$ is convex and is linear if and only if $x_0 = x_1$.

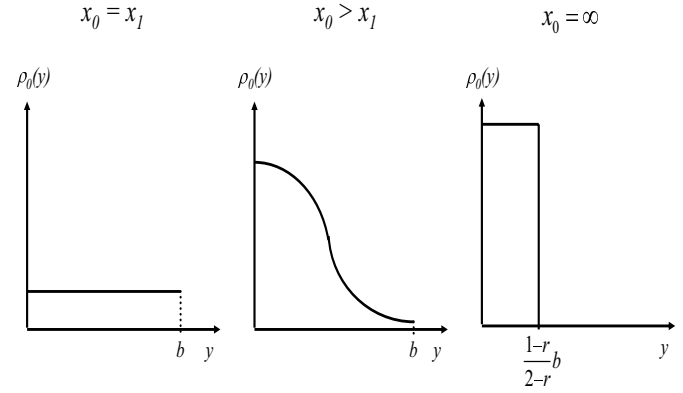
The boundary values of $v(y)$ follows from (17) (sum over i) and (18). □

Given x_0, define

$$\rho^* \quad := \quad \frac{1}{3}\left(1 - \frac{x_1}{x_0 - x_1}\right)$$



(a) Velocity $v(y)$



(b) Spatial distribution $\rho_0(y)$

Fig. 2. Illustration of Theorems 1, 2, 4, 6 and 7

and the inflexion point y^* by

$$\rho_0(y^*) \quad = \quad \rho^*$$

Theorem 2. *Suppose $x_0 > x_1$. Then $\rho_0(y)$ is a strictly decreasing function, with*

$$\rho_0(0) = \frac{x_0(1-h_0)}{x_0(1-h_0) + Nx_1(1-h_1)}$$

Moreover,

- *if $\rho_0(0) \leq \rho^*$, then $\rho_0(y)$ is convex.*
- *if $\rho_0(b) \geq \rho^*$, then $\rho_0(y)$ is concave.*
- *if $\rho_0(b) < \rho^* < \rho_0(0)$, then $\rho_0(y)$ is concave for $y \leq y^*$ and convex for $y \geq y^*$.*

Proof. From (23), we have

$$\rho_0'(y) \quad = \quad \beta(x_0 - x_1)\rho_0(y)(1-\rho_0(y))\frac{1}{v(y)}$$

Since $\beta < 0$ and $x_0 \geq x_1$, $\rho_0'(y) \leq 0$, i.e., $\rho_0(y)$ is a decreasing function. The value of $\rho_0(0)$ follows from (17) and Theorem 1.

Differentiating (22), we have after some algebra

$$\rho_0''(y) = \frac{\beta^2(x_0 - x_1)\rho_0(y)(1-\rho_0(y))}{v^2(y)} \cdot ((1 - 3\rho_0(y))(x_0 - x_1) \; - \; x_1)$$

There are three possible cases:

- $\rho_0(0) \leq \rho^*$: $(1-3\rho_0(y))(x_0-x_1) - x_1 \geq 0$ and $\rho_0''(y) \geq 0$. In this case $\rho(y)$ is convex decreasing.
- $\rho_0(b) \geq \rho^*$: $(1-3\rho_0(y))(x_0-x_1) - x_1 \leq 0$ and $\rho_0''(y) \leq 0$. In this case $\rho(y)$ is concave decreasing.
- $\rho_0(0) > y^* > \rho_0(b)$: Then $\rho_0''(y) \leq 0$ for $y \leq y^*$ and $\rho_0''(y) \geq 0$ for $y \geq y^*$. Hence $\rho(y)$ is strictly concave decreasing before the inflexion point y^* and strictly convex decreasing after.

□

Theorems 1 and 2 are illustrated in Figure 2. The figure also shows the asymptotic properties of $v(y)$ and $\rho_0(y)$, to which we now turn.

D. Asymptotic properties

We will prove that $v(y)$ and $\rho_0(y)$ take the form shown in the right-hand column of Figure 2 asymptotically as $x_0 \to \infty$.

We make more assumptions:

A3: Given any $x_0 \geq 0$, the solution $z(x_0)$ of (10) is continuous in x_0 and that $\lim_{x_0 \to \infty} z(x_0)$ exists. Denote $\lim_{x_0 \to \infty} z(x_0)$ by $z^\infty = (p_0^\infty, p_1^\infty, \ldots)$.

A4: The pointwise limits of $v(y)$ and $\rho_0(y)$ as $x_0 \to \infty$, denoted by $v^\infty(y)$ and $\rho_0^\infty(y)$, exist. Moreover, relations (12), (13) and (15) are satisfied in the limit with $z(x_0)$, $v(y)$ and $\rho_0(y)$ replaced by $z^\infty(x_0)$, $v^\infty(y)$ and $\rho_0^\infty(y)$ respectively.

A5: The TCP (equilibrium) algorithm $f(p_1, \tau)$ is continuous in its arguments. Moreover, $x_1 = f(p_1, \tau) < \infty$ when $p_1 > 0$.

A6: The congestion based dropping $g(b, \tau)$ is continuous in its arguments. Moreover, $g(b, \tau) \to 1$ as $b \to \infty$.

Conditions A5 and A6 are nonrestrictive: A5 says that the TCP rate is finite if there is any loss, and A6 says that if backlog grows without bound then eventually all incoming packets will be dropped. Note that we are not assuming that the limit functions $v^\infty(y)$ and $\rho_0^\infty(y)$ satisfy the ODEs of Section III-B.

We start with a result that says that regardless of the UDP rate x_0, every flow, including UDP flow, occupies less than half of the queue. This implies that asymptotically as $x_0 \to \infty$, congestion based dropping probability $r < 1$ and queueing delay $\tau > 0$. These properties are used later to prove the asymptotic UDP throughput and the asymptotic spatial properties of the leaky buffer of CHOKe.

Theorem 3. 1) *For all $x_0 \geq 0$, $b_i \leq b/2$, $i = 0, 1$.*

2) *As $x_0 \to \infty$*
   a) *$h_0^\infty = (1-r)/(2-r)$ and $h_1^\infty = 1/N(2-r)$.*
   b) *$r^\infty < 1$.*
   c) *$x_1^\infty < \infty$ and $\tau^\infty > 0$.*
   d) *$b_i^\infty < \infty, i = 0, 1$*

Proof.

1) From (3), $2h_i + (1-h_i)r = p_i \le 1$ and hence using (5), we have

$$\frac{b_i}{b} \;=\; h_i \quad \le \quad \frac{1-r}{2-r}$$

The right-hand side is a decreasing function for $0 \le r \le 1$ with a maximum value of $1/2$ at $r = 0$.

2) Since $x_0(1-\rho_0) \le c$ for all x_0, p_0 must tend to 1 as $x_0 \to \infty$. Hence $2h_0 + (1-h_0)r = p_0 \to 1$. Then $h_0 \to (1-r)/(2-r)$, or $b_0 \to b(1-r)/(2-r)$. Since $b = b_0 + Nb_1$, we have $h_1 \to 1/N(2-r)$.
Suppose $r^\infty = 1$. From (3), $2h_i + (1-h_i)r = p_i \le 1$ and hence, for $i = 0, 1$,

$$r \quad \le \quad \frac{1-2h_i}{1-h_i} \;\le\; 1$$

with equalities if and only if $h_i = 0$ for $i = 0, 1$. But this contradicts that $h_0 + Nh_1 = 1$. Hence $r^\infty < 1$.
Consider TCP flow $i = 1$. Since $h_1^\infty = 1/(N(2 - r^\infty)) > 0$ from part 2(a), (4) implies that $p_1^\infty > 0$. By assumption A5, $x_1^\infty = f(p_1^\infty, \tau^\infty) < \infty$. Now suppose for the sake of contradiction that $\tau^\infty = 0$. Then (4) implies

$$1 - p_1^\infty \quad = \quad (1-r^\infty)(1-h_1^\infty)$$

but (3) implies

$$1 - p_1^\infty \quad = \quad (1-r^\infty)(1-h_1^\infty) - h_1^\infty$$

yielding $h_1^\infty = 0$, contradicting that $h_1^\infty = 1/(N(2 - r^\infty)) > 0$. Hence $\tau^\infty > 0$.
Finally, if $b_i^\infty = \infty$, then by assumption A6, we have $r^\infty = \lim_{x_0\to\infty} g(b_i, \tau) = 1$, contradicting (b).

□

We next show that the UDP throughput vanishes as x_0 grows without bound. This result is also obtained independently in [9] and in [14], using different methods. Recall (13) that for $x_0 \ge 0$

$$\tau(y) \quad = \quad \int_0^y \frac{ds}{v(s)}.$$

Theorem 4. *As $x_0 \to \infty$, $x_0(1-p_0) = \rho_0(b)c \to 0$.*

Proof. From (23), we have

$$\frac{\rho_0'(y)}{\rho_0(y)(1-\rho_0(y))} \quad = \quad \beta(x_0 - x_1)\,\frac{1}{v(y)}$$

where $\beta = \ln(1 - 1/b)$. Integrating both sides from 0 to y, we get

$$\ln\frac{\rho_0(y)}{1-\rho_0(y)} - \ln\frac{\rho_0(0)}{1-\rho_0(0)} \quad = \quad \beta(x_0 - x_1)\tau(y)$$

Hence

$$\rho_0(y) \quad = \quad \frac{ae^{\beta(x_0-x_1)\tau(y)}}{1 + ae^{\beta(x_0-x_1)\tau(y)}} \qquad (24)$$

where

$$a \quad = \quad \frac{\rho_0(0)}{1-\rho_0(0)} \qquad (25)$$

UDP throughput share $x_0(1-p_0)/c$ is $\rho_0(y)$ evaluated at $y = b$ where $\tau(y) = \tau$. From Theorem 3(c), $\lim_{x_0\to\infty}\tau(b) = \tau^\infty > 0$, and hence by Lemma 5 below, $\lim_{x_0\to\infty}\rho_0(b) = 0$. □

The following lemma implies that, asymptotically as $x_0 \to \infty$, not only does the throughput of UDP $x_0(1-p_0) \to 0$, moreover, all UDP packets are dropped before the first point where $\tau(y)$ is nonzero.

Lemma 5. *If $\lim_{x_0\to\infty}\tau(y) > 0$, with y possibly a function of x_0, then*

$$\lim_{x_0\to\infty}\rho_0(y) \quad = \quad \lim_{x_0\to\infty}\frac{ae^{\beta(x_0-x_1)\tau(y)}}{1+ae^{\beta(x_0-x_1)\tau(y)}} \quad = \quad 0$$

Proof. We will show that the numerator of (24) tends to 0 as $x_0 \to \infty$. From Theorem 2, we have

$$\rho_0(0) = \frac{x_0(1-h_0)}{x_0(1-h_0) + Nx_1(1-h_1)} \qquad (26)$$

and hence from (25), we have

$$a \quad = \quad \frac{x_0(1-h_0)}{Nx_1(1-h_1)}$$

The numerator is then

$$\frac{x_0(1-h_0)}{Nx_1(1-h_1)} \cdot e^{\beta(x_0-x_1)\tau(y)}$$

From Theorem 3, as $x_0 \to \infty$, $h_i^\infty < 1$, $i = 0, 1$, and $x_1^\infty < \infty$. Moreover, from Theorem 3(d), $b^\infty < \infty$ and hence $\beta^\infty = \ln(1 - 1/b^\infty) < 0$. Then, if $\lim_{x_0}\tau(y) > 0$, it can be shown that a grows at most linearly to ∞, while $e^{\beta(x_0-x_1)\tau(y)}$ tends at least exponentially to 0. Hence the numerator $ae^{\beta(x_0-x_1)\tau} \to 0$. □

The next result says that the asymptotic spatial distribution $\rho_0^\infty(y)$ of UDP takes the form shown in Figure 2.

Theorem 6. *Let $y^* := b^\infty(1-r^\infty)/(2-r^\infty)$. Then*

$$\rho_0^\infty(y) \quad = \quad \begin{cases} 1, & 0 \le y < y^* \\ 0, & y^* < y \le b^\infty \end{cases}$$

Proof. From Theorem 4, we know $\rho_0^\infty(b) = 0$. Hence we can define

$$\hat{y} \quad := \quad \inf\{\, y \mid \rho_0^\infty(y) = 0 \,\}$$

Let y' be any point with $\rho_0^\infty(y') = 1$. One exists because, from proof of Lemma 5, we know $\rho_0^\infty(0) = 1$ (see (26)). Consider the midpoint y'' between y' and $\hat{y}$, $y'' = (y' + \hat{y})/2$. It suffices to prove that (i) either $\rho_0^\infty(y'') = 1$ or $y'' = \hat{y}$, and (ii) $\hat{y} = y^*$.

(i) Suppose $\rho_0^\infty(y'') < 1$. We need to show that $y'' = \hat{y}$. Since $\rho_0(y)$ is decreasing y for any finite $x_0 \ge x_1$ (Theorem 2), its limit $\rho_0^\infty(y)$ is nonincreasing in y. Hence, $y'' \le \hat{y}$. Suppose for the sake of contradiction that $\delta := \hat{y} - y'' > 0$.

We first prove that $v^\infty(y'') < \infty$. Consider for any finite $x_0 \geq 0$ the quantity $\rho_1(y)v(y)$, the TCP flow rate at position y'' in the queue. Using (19) and (17), we have

$$\rho_1(y'')v(y'') \;\leq\; \rho_1(0)v(0) \;=\; x_1(1-r)(1-h_1)$$

Taking limit as $x_0 \to \infty$, we have

$$\rho_1(y'')v(y'') \;\leq\; x_1^\infty(1-r^\infty)(1-h_1^\infty)$$

which is bounded by Theorem 3. Note that $\rho_0^\infty(y'') < 1$ by definition of y''. Since $\rho_0(y'') + N\rho_1(y'') = 1$, we have (taking limit of the corresponding finite-x_0 expression as $x_0 \to \infty$)

$$\rho_0^\infty(y'') \;=\; \frac{\rho_0^\infty(y'')v^\infty(y'')}{\rho_0^\infty(y'')v^\infty(y'') + N\rho_1^\infty(y'')v^\infty(y'')} \;<\; 1$$

Hence, $\rho_1^\infty(y'')v^\infty(y'') < \infty$ implies $\rho_0^\infty(y'')v^\infty(y'') < \infty$. This in turn implies that $v^\infty(y'') < \infty$ since $\rho_0^\infty(y'') > 0$ by definition of y'' (otherwise, $y'' = \hat{y}$ and we are done).

Now define y''' be the midpoint of y'' and $\hat{y}$, $y''' = (y'' + \hat{y})/2$. Then $\rho_0^\infty(y''') > 0$ by definition of $\hat{y}$. We now show that $v^\infty(y'') < \infty$ implies $\rho_0^\infty(y''') = 0$, a contradiction.

From Theorem 1, $v(y)$ is strictly decreasing in y and hence $\tau(y)$ defined by (13) satisfies

$$\begin{aligned}
\tau(y''') &= \int_0^{y'''} \frac{ds}{v(s)} \\
&\geq \int_{y''}^{y'''} \frac{ds}{v(s)} \\
&> \int_{y''}^{y'''} \frac{ds}{v(y'')} \\
&= \frac{\delta}{2v(y'')}
\end{aligned}$$

Taking limit as $x_0 \to \infty$, we have

$$\tau^\infty(y''') \;\geq\; \frac{\delta}{2v^\infty(y'')} \;>\; 0$$

where the last inequality follows from $v^\infty(y'') < \infty$. Hence $\rho_0^\infty(y''') = 0$ by Lemma 5. But this contradicts the definition of $\hat{y}$ since $y''' < \hat{y}$. Hence we must have $y'' = \hat{y}$.

(ii) The above shows that $\rho_0^\infty(y)$ takes the form shown in Figure 2. Then, by (15) and Theorem 3, we have (using assumption A4)

$$\frac{1-r^\infty}{2-r^\infty}\, b^\infty \;=\; \int_0^{\hat{y}} dy \;=\; \hat{y}$$

This completes the proof. □

The next result proves the shape of $v^\infty(y)$.

Theorem 7. *Let $y^* = b^\infty(1-r^\infty)/(2-r^\infty)$. Then*

$$v^\infty(y) \;=\; \begin{cases} \infty, & 0 \leq y < y^* \\ c - \beta^\infty x_1^\infty\,(b^\infty - y), & y^* < y \leq b^\infty \end{cases}$$

Proof. We first prove for $y < y^*$ and then for $y > y^*$.

Assume there exists $y' < y^*$ such that $v^\infty(y') < \infty$. Consider $y'' = (y' + y^*)/2$. Since $v^\infty(y)$ is nonincreasing in y, we have $v^\infty(y) \leq v^\infty(y') < \infty$ for any $y' \leq y \leq y''$. Hence, using assumption A4, we have

$$\begin{aligned}
\tau^\infty(y'') &= \int_0^{y''} \frac{ds}{v^\infty(s)} \\
&\geq \int_{y'}^{y''} \frac{ds}{v^\infty(s)} \\
&\geq \frac{y'' - y'}{v^\infty(y')} \;>\; 0
\end{aligned}$$

Then Lemma 5 implies that $\rho_0^\infty(y'') = 0$. Since $y'' < y^*$, this contradicts theorem 6. So for $y < y^*$, $v^\infty(y) = \infty$.

For $y > y^*$, we prove the theorem in 4 steps.

Step 1: $v^\infty(y) < \infty$ for all $y > y^*$.

Using (19) and (17), and taking limit as $x_0 \to \infty$, we have

$$\rho_1^\infty(y)v^\infty(y) \;\leq\; \rho_1^\infty(0)v^\infty(0) \;=\; x_1^\infty(1-r^\infty)(1-h_1^\infty)$$

But by Theorem 6, $\rho_1^\infty(y) = (1-\rho_0^\infty(y))/N = 1/N$ for $y > y^*$, and by Theorem 3, the right-hand side is finite. Hence $v^\infty(y) < \infty$ for all $y > y^*$.

Step 2: $\tau^\infty(y) > 0$ for $y > y^*$.

Fix $y > y^*$. Since $v(y)$ is strictly decreasing (Theorem 1) we have, for each finite $x_0 \geq 0$,

$$\begin{aligned}
\tau(y) &= \int_0^{y} \frac{ds}{v(s)} \\
&> \int_{\frac{y^*+y}{2}}^{y} \frac{ds}{v(s)} \\
&> \frac{y - y^*}{2v\left(\frac{y^*+y}{2}\right)}
\end{aligned}$$

Taking limit as $x_0 \to \infty$, we have for $y > y^*$

$$\tau^\infty(y) \;\geq\; \frac{y - y^*}{2v^\infty\left(\frac{y^*+y}{2}\right)} \;>\; 0$$

Step 3: There exists an integrable function $\overline{H}^{x_0}(y)$ such that, for all $x_0 \geq 0$ and $y > y^*$,

$$H^{x_0}(y) \;:=\; x_0\rho_0(y) + x_1(1-\rho_0(y)) \;\leq\; \overline{H}^{x_0}(y)$$

From (24) we have

$$x_0\rho_0(y) \;=\; \frac{x_0 a e^{\beta(x_0-x_1)\tau(y)}}{1 + a e^{\beta(x_0-x_1)\tau(y)}}$$

Since $\tau^\infty(y) > 0$ for all $y > y^*$, the proof of Lemma 5 shows that, as $x_0 \to \infty$, a grows at most logarithmically to ∞ while $e^{\beta(x_0-x_1)\tau(y)}$ grows at least exponentially to 0. Hence the numerator $x_0 a e^{\beta(x_0-x_1)\tau(y)}$ tends to 0 as $x_0 \to \infty$. Moreover,

$$m(y) \;:=\; \max_{x_0 \geq 0}\; x_0 a e^{\beta(x_0-x_1)\tau(y)}$$

is finite. Hence, for x_0 sufficiently large (so that $a > 0$), we have

$$x_0\rho_0(y) \;\leq\; m(y)$$

Note that $m(y)$ is independent of x_0. Hence if we define $\overline{H}^{x_0}(y) = m(y) + x_1$, then $H^{x_0}(y) \le \overline{H}^{x_0}(y)$. Since x_1 is finite for all x_0 (Theorem 3), $\overline{H}^{x_0}(y)$ is integrable over (y^*, b).[1]

Step 4: $v^\infty(y) = c - \beta^\infty x_1^\infty\ (b^\infty - y)$ for $y > y^*$.

Fixed $y > y^*$. From (22), we have, for each $x_0 \ge 0$,

$$\begin{aligned} v(y) &= c - \beta \int_y^b (\rho_0(s)x_0 + (1-\rho_0(s))x_1)ds \\ &= c - \beta \int_y^b H^{x_0}(s)ds \end{aligned}$$

Taking limit as $x_0 \to \infty$, we have

$$v^\infty(y) = c - \beta \lim_{x_0\to\infty} \int_y^b H^{x_0}(s)ds$$

From Step 3, $H^{x_0}(s)$ is upper bounded by an integrable function and converges pointwise to $H^\infty(y) = x_1^\infty$ as $x_0 \to \infty$. Hence Legesgue convergence theorem applies (see [12, p.229], [1, pp. 215]):

$$\lim_{x_0\to\infty} \int_y^b H^{x_0}(s)ds = \int_y^{b^\infty} H^\infty(s)ds = x_1^\infty\ (b^\infty - y)$$

Hence, $v^\infty(y) = c - \beta^\infty x_1^\infty\ (b^\infty - y)$ for $y > y^*$.

This completes the proof. □

We summarize these structural properties. First, when x_0 is large, the spatial distribution $\rho_0(y)$ decreases rapidly toward the head of the queue. This means that most of the UDP packets are dropped before they reach the head. It is therefore possible to simultaneously maintain a large number of packets (concentrating near the tail) and receive a small bandwidth share, in stark contrast to the behavior of a non-leaky FIFO buffer. Indeed, as x_0 grows without bound, UDP share drops to 0. Second, the packet velocity is infinite before the position y^* because UDP packets are being dropped at an infinite rate until y^*.

IV. Simulation results

We implemented a CHOKe module in ns-2 version 2.1b9 and have conducted extensive simulations using the network shown in Figure 1 to study the equilibrium behavior of CHOKe. There is a single bottleneck link from router R1 to router R2 shared by N TCP sources and one UDP source. The UDP source sends data at constant rate (CBR). For all our simulations, the link capacity is fixed at $c = 1$Mbps and the round trip propagation delay is $d = 100$ms. Packet size is 1KB. The simulation time is 200–300 seconds.

We use RED+CHOKe as the queue management with RED parameters: min_th $\underline{b} = 20$ packets, max_th $\overline{b} = 520$ packets, $p_{max} = 0.5$. This defines the function g in our model (9). For the TCP function in (8), we use ([5]):

$$p_1 = \frac{2}{2 + x_1^2(d+\tau)^2} \tag{27}$$

[1]To be precise, $b = b^{x_0}$ is a function of x_0. To make the domain of integration independent of x_0, extend $H^{x_0}(y)$ and $\overline{H}^{x_0}(y)$ to $(y^*, \max_{x_0} b^{x_0})$ by defining them to be zero for y not in (y^*, b^{x_0}).

We vary UDP sending rate x_0 from 12.5pkts/s to 1250 pkts/s and the number N of TCP flows from 12 to 64 to observe their effect on the equilibrium behavior of CHOKe. We measure, and compare with our numerical solutions, the following quantities

1) aggregate queue size b
2) UDP bandwidth share $\mu_0 = \rho_0(b)$
3) TCP throughput $x_1(1-p_1) = \mu_1 c = \rho_1(b)c$
4) spatial distribution $\rho_0(y)$ of UDP packets

The results illustrate both the equilibrium behavior of CHOKe and the accuracy of our analytical model. They show the ability of CHOKe to protect TCP flows and agree with those aggregate measurements of [10]. Moreover, they exhibit the fine structure of queue dynamics, and confirm the spatial character of our CHOKe model.

We next discuss these results in detail.

A. Effect of UDP rate x_0

First we study the effect of UDP sending rate on queue size and bandwidth allocation. The number of TCP sources is fixed at $N = 32$. We vary the UDP sending rate x_0 from $0.1\times$ link capacity to $10\times$ link capacity. The results are in Figure 3.

The aggregate queue length b steadily increases as UDP rate x_0 rises. UDP bandwidth share $\mu_0 = \rho_0(b)$ rises, peaks, and then drops to less than 5% as x_0 increases from $0.1c$ to $10c$, while the total TCP throughput follows an opposite trend eventually exceeding 95% of the capacity. We have shown both TCP throughput – the total number of TCP packets, from all N flows, transmitted by the link during each simulation – and TCP goodput – sum of largest sequence numbers in each simulation. TCP goodput does not count retransmitted packets and is hence lower than TCP throughput. These results match closely those obtained in [10], and with our analytical model.

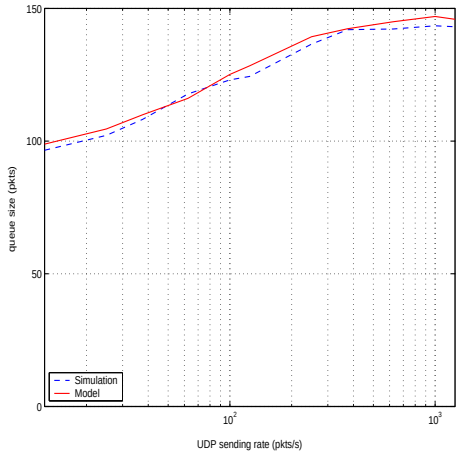
Figure 4 shows the spatial distribution $\rho_0(y)$ of UDP packets (compare with Figure 2(b)). To get the packet distribution from each simulation (x_0 value), we took $J = 3000$ snapshots of the queue every 100ms for 300 seconds. From the J sample queue sizes b^j, we first calculated the average $b_{avg} := \sum_j b^j/J$. The distribution was estimated over this range $[0, b_{avg}]$, as follows. For each $y \in [0, b_{avg}]$, the sample distribution is calculated as

$$\rho(y) = \frac{1}{J}\sum_j \mathbf{1}_j(y)$$

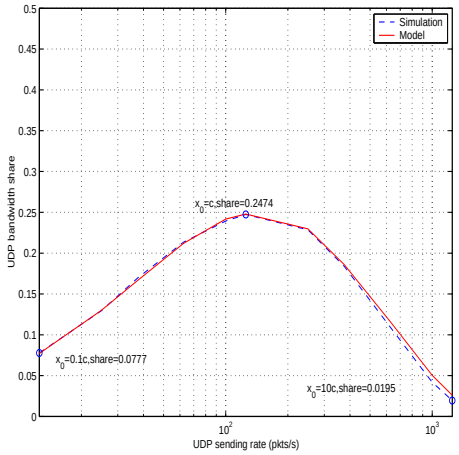
where $\mathbf{1}_j(y)$ is 1 if the packet in position $\lfloor y b^j/b_{avg} \rfloor$ of the jth snapshot is UDP, and 0 otherwise. This distribution is plotted in Figure 4, together with the numerical solution of our CHOKe model.

When $x_0 = 0.1c$ (Figure 4(a)), UDP packets are distributed roughly uniformly in the queue, with probability close to 0.08 at each position. As a result, its bandwidth share is roughly 10%. As x_0 increase, $\rho_0(y)$ concentrates more and more near the tail of the queue and drops rapidly toward the head, as predicted by Theorems 2 and 6.

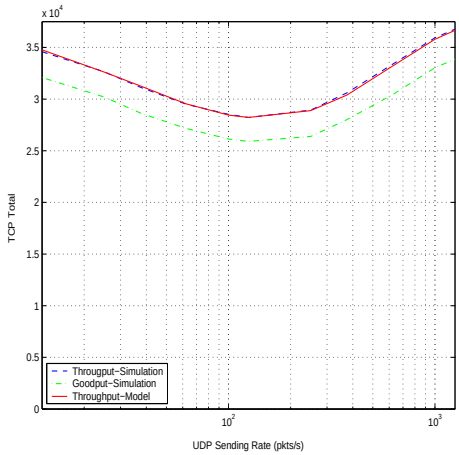
Also marked in Figure 3(b), are the UDP bandwidth shares corresponding to UDP rates in Figure 4. As expected the UDP bandwidth shares in 3(b) are equal to $\rho_0(b)$ in Figure 4. When



(a) Queue size b

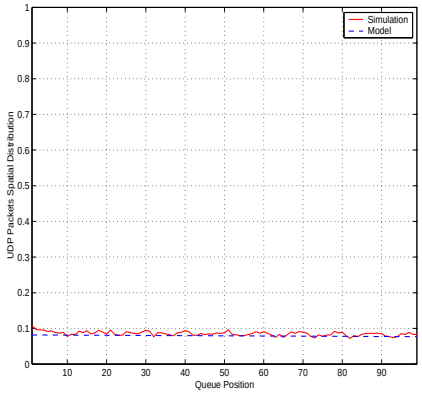


(b) UDP bandwidth share μ_0

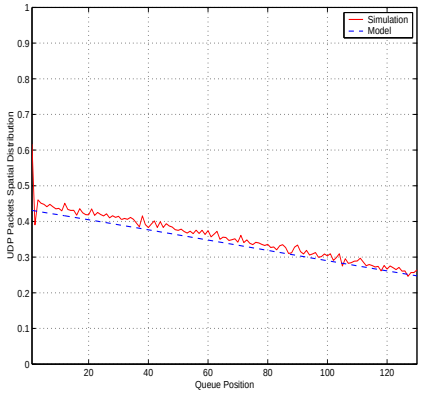


(c) Total TCP throughput $\mu_1 c$

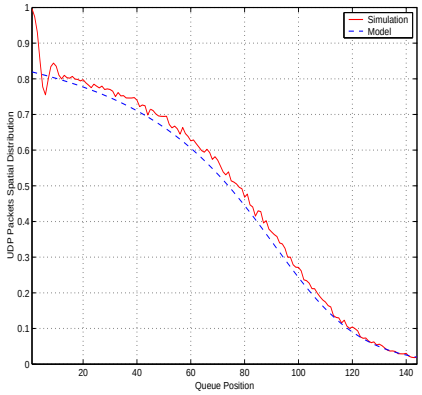
Fig. 3. Effect of UDP rate x_0 on queue size and bandwidth allocation. $N = 32$, $x_0 = 0.1c$ to $10c$, $c = 1$Mbps, simulation duration = 300sec.



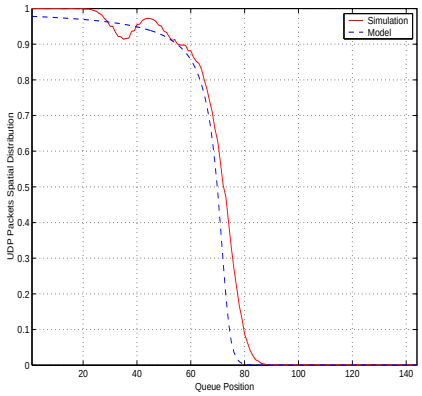
(a) UDP rate $x_0 = 0.1c$



(b) UDP rate $x_0 = c$



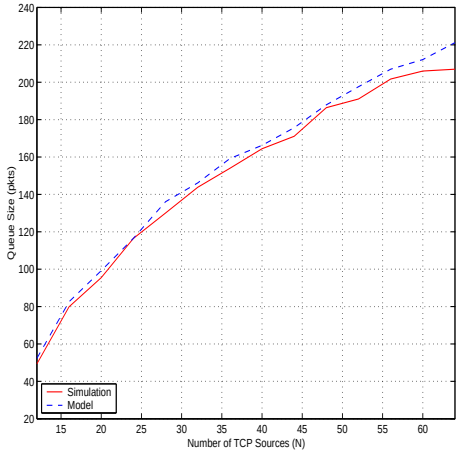
(c) UDP rate $x_0 = 10c$



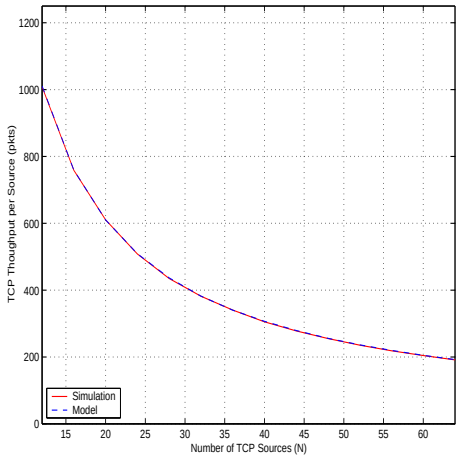
$x_0 > 10c$, even though roughly half of the queue is occupied by UDP packets, almost all of them are dropped before they reach the head of the queue!

B. *Effect of number N of TCP flows*

Figure 5 shows the effect of N on aggregate queue size b and on per-flow TCP throughput $\mu_1 c = \rho_1(b)c$.



(a) Queue size b



(b) TCP throughput per flow $\mu_1 c$

Fig. 5. Effect of number N of TCP flows on aggregate queue b and per-flow TCP throughput $\mu_1 c$. $N = 12 - 64$, $x_0 = 1250$ pkts/s, $c = 125$pkts/s, simulation duration = 300s

Not surprisingly, the queue size increases and per-flow TCP throughput decreases with N as the queue becomes more congested. Again, the simulation and analytical results match very well, further validating our model.

V. Discussion

Our model captures well the equilibrium behavior of CHOKe under the assumption that the queue size b is between $\underline{b}$ and $\overline{b}$. This holds if c is sufficiently small or N is sufficiently large. A sample queue size is shown in Figure 6. It may not hold for small N.

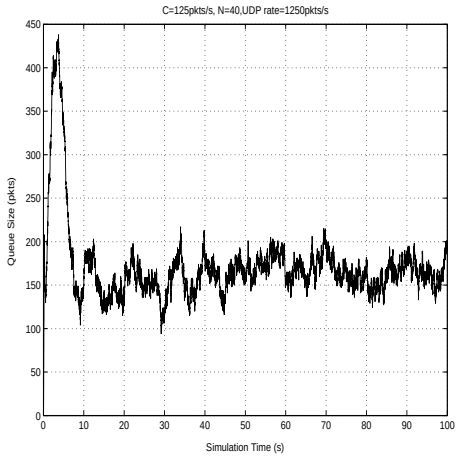


Fig. 6. Queue Size at N=40

With smaller N, each TCP source gets a larger bandwidth share, and a larger TCP sending rate requires a lower dropping probability. However, when CHOKe is active, it imposes a lower bound on the dropping probability: from (1)

$$\begin{aligned} p_i &= 2h_i + (1-h_i)r \\ &\geq 2h_i \geq \frac{1}{N} \end{aligned} \tag{28}$$

where the last inequality follows from (5) and the fact that UDP packets occupy at most half of the queue (Theorem 3(1)).

We can estimate the minimum N with which CHOKe is always active. Approximate the TCP function in (27) by

$$p_1 = \frac{2}{x_1^2(d+\tau)^2}$$

Combining with (28) to get

$$\frac{2N}{x_1^2(d+\tau)^2} \geq 1$$

When UDP sending rate is large, TCP flows take almost all the bandwidth, so

$$x_1 \approx \frac{c}{N}$$

Around $\underline{b}$, queueing delay is roughly

$$\tau \approx \frac{\underline{b}}{c}$$

Putting all these together, the minimum number of TCP flows required for CHOKe to remain active is roughly

$$N \geq \sqrt[3]{\frac{(cd+\underline{b})^2}{2}}$$

Using the same parameters as in the last section, we can estimate the minimal N to be 8.43. When $N = 8$, queue size oscillates around $\underline{b}$ = 20 packets, constantly turning CHOKe on and off, as shown in Figure 7 (compare with Figure 6).

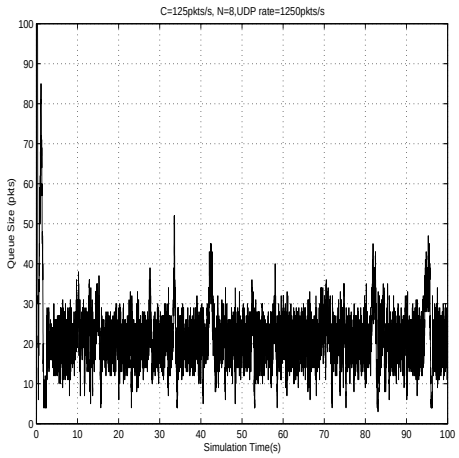


Fig. 7. Queue Size at N=8

The same phenomenon is observed when c increases (with fixed N). The lower bound on dropping probability when CHOKe is active, $p_1 \geq 1/N$, eventually prevents TCP flows from making full use of the available capacity. An interesting positive effect is that the queue length is controlled to stay around $\underline{b}$.

When there are more than one unresponsive UDP flows, CHOKe may not be very efficient in protecting TCP traffic. The UDP flows may take a larger bandwidth share at high sending rate.

VI. Conclusions

CHOKe is completely stateless and extremely simple to implement, yet surprisingly effective in bounding the bandwidth share of UDP flows. As shown in the simulations of [10], as UDP source rate increases, its bandwidth share eventually drops to zero, exactly opposite to how a regular FIFO buffer behaves. To understand this precisely requires a careful study of the queueing process under CHOKe.

In this paper, we have developed a model of CHOKe. Its key features are the incorporation of the feedback equilibrium of TCP and a detailed modeling of the queue dynamics. We have introduced the concepts of spatial distribution and velocity of packets in the queue. We prove structural and asymptotic properties of these quantities that make it possible for UDP to simultaneously maintain a large number of packets in the queue and receive a vanishingly small bandwidth share, the mechanism through which CHOKe protects TCP flows.

Finally, we remark that CHOKe algorithm may be constantly turned on and off when the link capacity is high or the number of TCP sources is small. This can prevent TCP flows from making full use of available capacity but regulate the queue size to around $\underline{b}$.

References

[1] Patrick Billingsley. *Probability and Measure*. John Wiley & Sons, 2 edition, 1986.

[2] W. Feng, K G. Shin, D. Kandlur, and D. Saha. Stochastic Fair Blue: A queue management algorithm for enforcing fairness. In *Proceedings of INFOCOM*, April 2001.

[3] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Trans. on Networking*, 1(4):397–413, August 1993. `ftp://ftp.ee.lbl.gov/papers/early.ps.gz`.

[4] Dong Lin and Robert Morris. Dynamics of random early detection. In *Proceedings of SIGCOMM'97*, pages 127–137, September 1997. `http://www.acm.org/sigcomm/sigcomm97/papers/p078.ps`.

[5] Steven H. Low. A duality model of TCP and queue management algorithms. *IEEE/ACM Trans. on Networking, to appear*, October 2003. `http://netlab.caltech.edu`.

[6] P. McKenny. Stochastic fairness queueing. In *Proceedings of Infocom*, pages 733–740, 1990.

[7] J. A. Nelder and R. Mead. A simplex method for function minimization. *Comput. J.*, pages 308–313, 1965.

[8] T. J. Ott, T. V. Lakshman, and L. Wong. SRED: Stabilized RED. In *Proceedings of IEEE Infocom'99*, March 1999. `ftp://ftp.bellcore.com/pub/tjo/SRED.ps`.

[9] Rong Pan, Chandra Nair, Brian Yang, and Balaji Prabhakar. Packet dropping mechanisms: some examples and analysis. In *Proc. of 38th Annual Allerton Conference on Communication, Control, and Computing*, October 2001.

[10] Rong Pan, Balaji Prabhakar, and Konstantinos Psounis. CHOKe: a stateless AQM scheme for approximating fair bandwidth allocation. In *Proceedings of IEEE Infocom*, March 2000.

[11] Lawrence Perko. *Differential equations and dynamical systems*. Springer, 3 edition, 2001.

[12] H. L. Royden. *Real analysis*. MacMillan, 1968.

[13] I. Stoica, S. Shenker, and H. Zhang. Core-stateless fair queueing: achieving approximately fair bandwidth allocations in high speed networks. In *Proceedings of ACM Sigcomm*, 1998.

[14] Jiantao Wang, Ao Tang, and Steven H. Low. Maximum and asymptotic UDP throughput under CHOKe. Submitted to ACM Sigmetrics, `http://netlab.caltech.edu`, November 2002.