A Model for Analyzing the Roles of Network and User Behavior in Congestion Control

Robert Morris
MIT Laboratory for Computer Science
rtm@lcs.mit.edu

Y.C. Tay
National University of Singapore
tay@acm.org

May 26, 2003

Abstract

Despite the Web-driven exponential growth in traffic volume, the Internet has not suffered a congestion collapse. Much credit for this robustness goes to TCP's congestion control mechanisms. However, TCP only controls the flow in a connection, whereas the number of connections is controlled by users. We present a model for analyzing the interaction between network and user behavior, and their complementary roles in Internet congestion control.

1 Introduction

Congestion Collapse in the Internet refers to a situation where some links suffer from overload but carry little useful traffic. By "useful", we mean packets that are not later discarded by router buffer management, and that are not retransmissions of packets that are received.

Congestion collapse was previously observed [Nag84, Jac88] and, when the Internet grew exponentially with the Web, there were fears that it might happen again [Met95, FF99]. In fact, it did not.

Most Internet traffic is carried by TCP connections, so much credit for the Internet's robustness goes to TCP's mechanisms for controlling connection throughput when congestion is detected.

However, TCP alone cannot account for the full story. To see this, let k be the number of connections on a link and suppose each connection carries traffic at rate b.

One measure of link congestion is **utilization** $u = kb/B$, where B is the link bandwidth. Some of this utilization is wasted by packets that are lost later or retransmitted, so let $b_{\text{useful}}(< b)$ be the useful per-connection throughput. Then the **useful utilization** is $u' = kb_{\text{useful}}/B$.

Whereas one expects u to increase monotonically with k, u' may eventually decrease (as when, say, the link is choked by retransmissions [Nag84]). This difference between u and u' is illustrated in Figure 1.

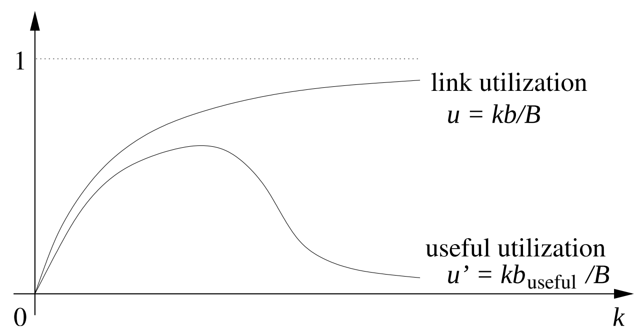

Figure 1: **As a link gets congested, the link utilization u continues to grow, but the useful utilization u' begins to fall.**

Now, TCP controls b, but not k. To understand congestion, we therefore need to examine how congestion affects k.

Given the prominence of Web traffic, one major factor in congestion dynamics is user behavior. When a user perceives congestion, she may

(U1) be inclined to click on fewer hyperlinks and

(U2) abort some slow connections by clicking "Stop", "Reload" or a hyperlink.

Such behavior is a form of congestion-induced **user backoff**: (U1) reduces the click rate on hyperlinks and (U2) cuts short the traffic demand.

However, in addition to the above-mentioned bandwidth wastage caused by packet loss and retransmissions, we also have wastage from aborted connections. Thus, (U2) depresses further the u' curve in Fig. 1, and adds to the danger of a congestion collapse.

2 TCP and user behavior

We now examine how TCP and user behavior, separately and together, determine network congestion. As "network" is a

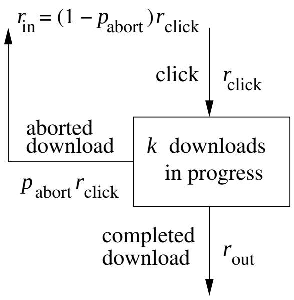

Figure 2: **Flow of downloads (one click generates one download). $r_{\rm click}$ is click rate, $p_{\rm abort}$ is the proportion of downloads that are aborted and $r_{\rm out}$ is the rate at which (nonaborted) downloads terminate.**

vague term, we focus on a single link (say, the one connecting an ISP's backbone to a network of customers). Since most Internet flows are generated by the Web, we focus on the latter.

When a user clicks on a hyperlink, she may initiate parallel TCP connections, each of which may carry pipelined HTTP requests. We aggregate these requests and their response into a **download**, so that each click initiates one download — see Fig. 2.

Note that packet flow in a download is regulated partially and indirectly through the TCP connections that it contains. The flows are also regulated by the browser and HTTP variants. For example, when a user aborts a download, the browser may close all related TCP connections.

Let $r_{\rm click}$ be the click rate on hyperlinks (including URLs typed in by the user) and $p_{\rm abort}$ the proportion of downloads that are aborted, and suppose there are $r_{\rm out}$ completed downloads per unit of time. Since each download (completed or not) takes some time, let k be the number of downloads that are in progress.

Here and in the following, the variables ($r_{\rm click}$, k, etc.) are averages, unless otherwise specified.

The rate of completed downloads, $r_{\rm out}$, corresponds to the useful utilization u' in Fig. 1 and has a similar shape. Fig. 3 illustrates the benefits of congestion control by showing how $r_{\rm out}$ varies with k if the downloads use (i) TCP without congestion control [Pos81] and (ii) TCP with congestion control [APS99].

Now let $r_{\rm in} = (1 - p_{\rm abort}) r_{\rm click}$, where

$$r_{\rm click} = n_{\rm click} r_{\rm session}, \qquad (1)$$

$n_{\rm click}$ is the number of clicks in a session, and $r_{\rm session}$ is the arrival rate of sessions. In steady state, the flows in Fig. 2 reach equilibrium when $r_{\rm in} = r_{\rm out}$.

To see the role of congestion control, assume first that users do not react to congestion, so $p_{\rm abort} = 0$ and $n_{\rm click}$ is constant

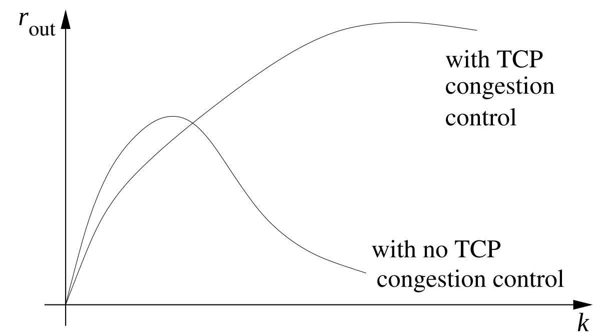

Figure 3: **$r_{\rm out}$ is the rate of completion of downloads. It is similar in shape to the u' curve in Fig. 1. With TCP congestion control, the curve stretches higher and further out.**

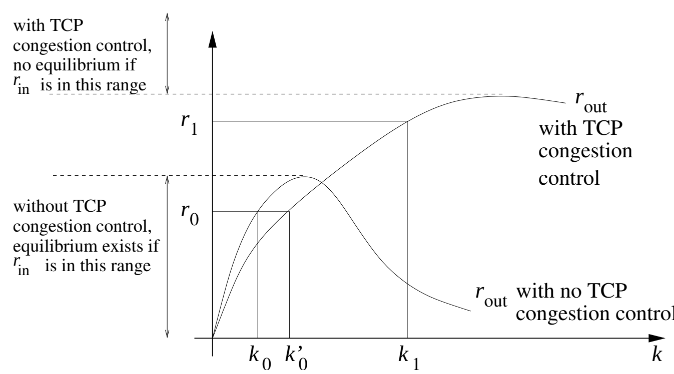

Figure 4: **If users do not react to congestion, $r_{\rm in}$ is a flat line. If $r_{\rm in} = r_0$, there is equilibrium at k_0 and $k_0' (> k_0)$ for TCP without and with congestion control, respectively. If $r_{\rm in} = r_1$ and there is no TCP congestion control, then there is no equilibrium (i.e. the link buffer overflows, the fraction of traffic that gets through is indeterminate, etc.). Even with TCP congestion control, there is no equilibrium if $r_{\rm in} (= r_{\rm click})$ is sufficiently high.**

(U1, U2). If $r_{\rm session}$ is constant, then $r_{\rm in} (= r_{\rm click})$ is a flat line when plotted against k.

(Note that congestion slows down web retrievals and reduce the click rate per user, but the total click rate can be constant. To draw an analogy: for a queue with arrival rate λ, a reduction in service rate will increase the waiting and service times, but the steady state departure rate is still λ.)

Since TCP congestion control raises the $r_{\rm out}$ curve in Fig. 3, a flat $r_{\rm in}$ line can intersect the $r_{\rm out}$ curve — thus providing equilibrium ($r_{\rm in} = r_{\rm out}$) — for higher values of $r_{\rm click}$. This is illustrated with $r_{\rm in} = r_1$ in Fig. 4.

If users react to congestion and $p_{\rm abort} \neq 0$, the abortions reduce the $r_{\rm out}$ curves in Fig. 3. This is illustrated in Fig. 5 for $r_{\rm out}$ without TCP congestion control.

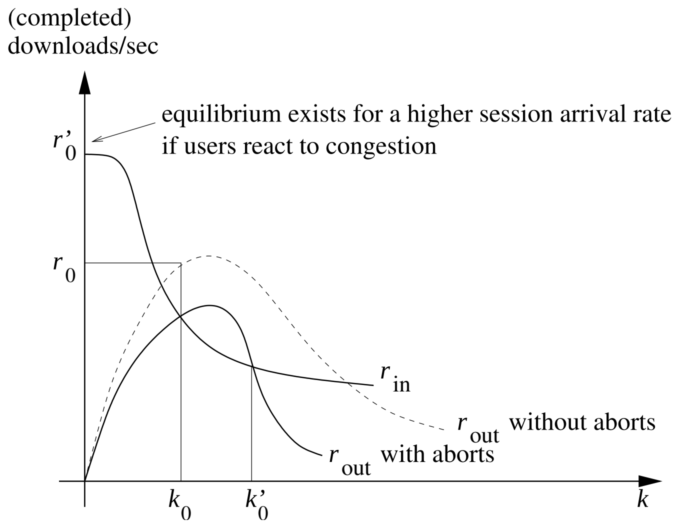

Figure 5: **If users react to congestion by aborting downloads, the $r_{\rm out}$ curve is lowered. The y-intercept for $r_{\rm in}$ is approximately $c_1 r_{\rm session}$, where $r_{\rm session}$ is the session arrival rate and c_1 is the number of downloads per session when $k = 1$. For the same equilibrium point k_0, user backoff makes a higher $r_{\rm session} (= r'_0 > r_0)$ possible even without TCP congestion control. The equilibrium at k_0 is stable; the one at k'_0 is unstable.**

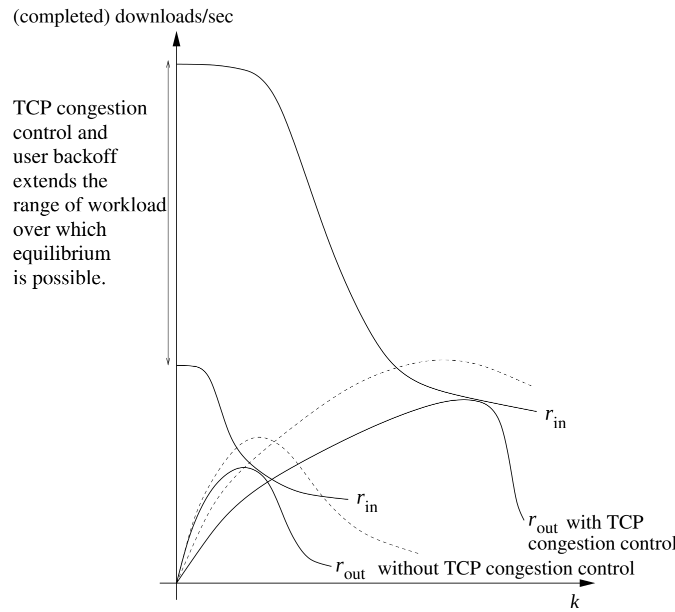

Figure 6: **With user backoff, the sloping $r_{\rm in}$ curve means session arrival rates can vary over a bigger range (compared to Fig. 4, where $r_{\rm in}$ is flat) without congestion collapse. This is partly because TCP congestion control pushes the $r_{\rm out}$ curve further to the right. Nonetheless, for a high enough $r_{\rm in}$ curve, collapse can still occur.**

As for $r_{\rm in}$, congestion may increase the click rate (since users may abort one download and initiate another) or decrease it, since users backoff (U1). Nonetheless, as $p_{\rm abort}$ increases with congestion (U2), we will see (Section 3.2) that, if $r_{\rm session}$ is the arrival rate of sessions, then

$$\text{for constant } r_{\rm session}, r_{\rm in} \text{ decreases as } k \text{ increases.} \quad (2)$$

Thus, $r_{\rm in}$ is no longer just a flat line — it slopes downwards, as in Fig. 5.

When k is small, the y-intercept in Fig. 5 is $r_{\rm in} \approx r_{\rm click} \approx c_1 r_{\rm session}$, where c_1 is the number of clicks per session when $k = 1$. We thus see from Fig. 5 that, even without TCP congestion control, user backoff makes equilibrium possible at a higher session arrival rate, thus acting as a form of congestion control itself.

Although the $r_{\rm in}$ and $r_{\rm out}$ curves may intersect at multiple places, only the first equilibrium point is useful. To see this, consider a small positive ε. At $k = k_0 - \varepsilon$ in Fig. 5, we have $r_{\rm in} > r_{\rm out}$, so (from Fig. 2) k increases and moves towards k_0. Conversely, at $k = k_0 + \varepsilon$, we have $r_{\rm in} < r_{\rm out}$, so k decreases and again moves towards k_0. In this way, k hovers around a stable equilibrium at k_0 as $r_{\rm in}$ and $r_{\rm out}$ fluctuate (over the course of a day, say).

The same argument shows that k is decreasing at $k = k'_0 - \varepsilon$ in Fig. 5, and increasing at $k = k'_0 + \varepsilon$, thus making the equilibrium at k'_0 unstable.

There may be stable equilibrium points for larger $k(> k'_0)$, but the rate of download completion $r_{\rm out}$ would be so low that these resemble the stable collapse observed by Nagle. Although users continue to react to such a collapse by backing off (U1, U2), k is large and new sessions continue to arrive. In this state, some TCP connections may have multiple retransmission timeouts and eventually abort (`RST`), thus increasing $p_{\rm abort}$ and moving both curves. Or, session arrivals may drop (e.g. after midnight), thus moving the $r_{\rm in}$ curve to the left. In any case, the equilibrium eventually becomes unstable, there is a net drain on k in Fig. 2, and the link returns to a stable equilibrium with a high $r_{\rm out}$.

Since TCP congestion control pushes $r_{\rm out}$ further to the right, Fig. 6 shows that equilibrium is now possible for even higher session arrival rates (compared to Fig. 5). In other words, with user backoff, this system can withstand a greater range of workload without slipping into congestion collapse.

Nonetheless, for a high enough $r_{\rm in}$ curve, collapse can still occur.

3 Network and user behavior models

We have modeled the system as an interaction between two curves: $r_{\rm in}$ that describes user behavior and $r_{\rm out}$ that describes network (browser/HTTP/TCP) behavior. We now present the behavior models that provide the underlying equa-

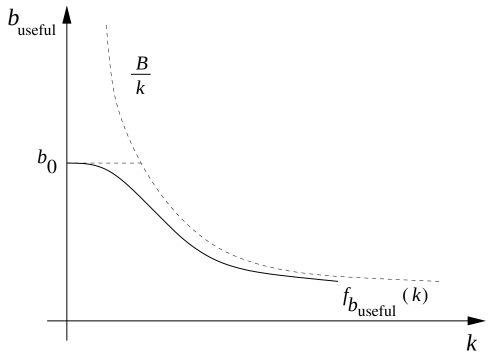

Figure 7: **The useful per-download throughput b_{useful} is determined by the application and TCP variants and bounded by b_0 and B/k.**

tions for the two curves.

3.1 Network model

With k active downloads on a link of bandwidth B, each download can expect the available bandwidth to be B/k. However, the download may not be able to use all of B/k if k is small. Instead, it may be limited (by, say, a modem link) to a maximum bandwidth of b_0. Hence the per-download useful throughput is

$$b_{\text{useful}} = f_{b_{\text{useful}}}(k), \tag{3}$$

bounded by $\min(b_0, \frac{B}{k})$, where $f_{b_{\text{useful}}}$ is determined by the application and TCP variants. This is illustrated in Fig. 7.

Thus, if S_{download} is the size of a download and users do not abort, then

$$r_{\text{out}} = \frac{kb_{\text{useful}}}{S_{\text{download}}} = \frac{k}{S_{\text{download}}} f_{b_{\text{useful}}}(k); \tag{4}$$

this is the equation for the curves in Fig. 3.

Since download abortion is initiated by the user, the TCP-determined function $f_{b_{\text{useful}}}$ does not take into account these abortions. (Indeed, TCP should, ideally, ensure that $\lim_{k\to\infty} kf_{b_{\text{useful}}}(k) = B$.) To describe the lower r_{out} curve in Fig. 5, we therefore need to refine the flow diagram in Fig. 2 for user behavior.

3.2 User model: Open system

We present here an open model of user arrivals, i.e. sessions are started by users at some constant rate r_{session}. (Later, we will extend this to a closed model where r_{session} is not constant, but varies as the number of active users changes.)

If we consider a user to be in a **wait** state while a download is in progress, we can split the state in Fig. 2 into a **wait-abort** state for aborted downloads and a **wait-complete** state for

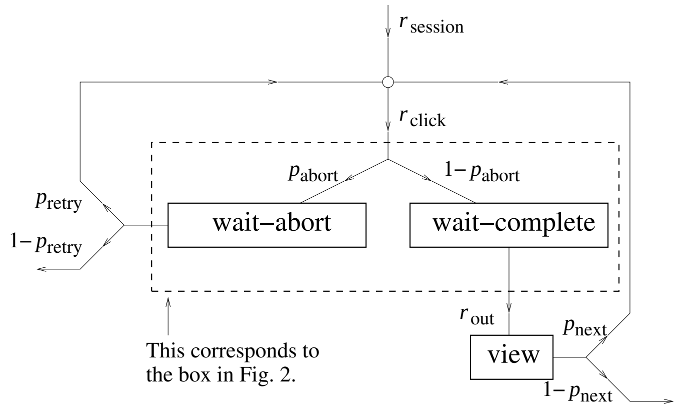

Figure 8: **User session model that is an expansion of Fig. 2. Here, $p_{\text{abort}} = \frac{\#(aborted\ downloads)}{\#(downloads)}$, p_{retry} is the proportion of aborted downloads that are followed by another click in the session, and p_{next} is the proportion of completed downloads that are followed by another click in the session.**

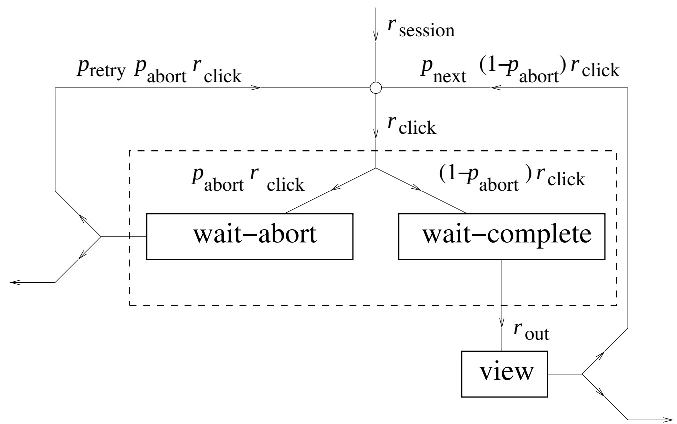

Figure 9: **Transition rates for Fig. 8 when viewed as a Markov state diagram.**

completed downloads. We can also follow up the latter with a **view** state for when a user is viewing a completed download. Since a user may click again after viewing, or after aborting a download, we get feedback flows. The result is the model of a user session shown in Fig. 8. Now let

$$p_{\text{abort}} = \frac{\text{number of aborted downloads}}{\text{number of downloads}},$$

let $1 - p_{\text{retry}}$ be the proportion of aborted downloads where the abortion ends the session, and let $1 - p_{\text{next}}$ be the proportion of completed downloads that are last in their sessions. We can then interpret Fig. 8 as a Markov state diagram with branching probabilities given by p_{abort}, p_{retry} and p_{next}. The steady state flow rates are then given by Fig. 9.

Recall that S_{download} is the size for a completed download, so the time spent in the **wait-complete** state is

$$t_{\text{download}} = \frac{S_{\text{download}}}{b_{\text{useful}}}. \qquad (5)$$

Now let T_{abort} be the time spent in the **wait-abort** state (also called the **frustration timeout** [ONA02]). One can, for example, model p_{abort} as

$$\begin{aligned} p_{\text{abort}} &= \text{Prob}(t_{\text{download}} > T_{\text{abort}}) \\ &= \text{Prob}(S_{\text{download}} > b_{\text{useful}} T_{\text{abort}}). \end{aligned}$$

Thus, p_{abort} increases as b_{useful} decreases; this and Fig. 7 imply that p_{abort} increases with k.

Zhang et al. recently observed, to their surprise, that Internet flow rates are strongly correlated with flow sizes [ZBPS02]. Their observation is consistent with the following user behavior: Suppose T_{abort} is approximately constant (5 to 10 seconds, say). Then T_{abort} has very weak correlation to the download bandwidth (b_{useful}) and the size of the aborted download ($b_{\text{useful}} T_{\text{abort}}$); on the other hand, the flow rate b_{useful} is obviously strongly correlated with the flow size $b_{\text{useful}} T_{\text{abort}}$. In other words, some of the correlation observed by Zhang et al. may have come from the aborted downloads.

By Little's Law, Fig. 9 gives

$$k = p_{\text{abort}} r_{\text{click}} T_{\text{abort}} + (1 - p_{\text{abort}}) r_{\text{click}} t_{\text{download}} \qquad (6)$$

so by Eqn. 5 and Eqn. 3,

$$r_{\text{click}} = \frac{k}{p_{\text{abort}} T_{\text{abort}} + (1 - p_{\text{abort}}) \frac{S_{\text{download}}}{f_{b_{\text{useful}}}(k)}}. \qquad (7)$$

In steady state, we have $r_{\text{out}} = (1 - p_{\text{abort}}) r_{\text{click}}$, so

$$r_{\text{out}} = \frac{k}{\frac{p_{\text{abort}}}{1 - p_{\text{abort}}} T_{\text{abort}} + \frac{S_{\text{download}}}{f_{b_{\text{useful}}}(k)}}. \qquad (8)$$

Note that if $p_{\text{abort}} = 0$, we get Eqn. 4. For small k, $p_{\text{abort}} \approx 0$ and $f_{b_{\text{useful}}}(k) \approx b_0$ (see Fig. 7). Therefore, $r_{\text{out}} \approx k b_0 / S_{\text{download}}$, i.e. r_{out} is approximately linear, for small k, as in Fig. 5. For large k, $p_{\text{abort}} \approx 1$ and $f_{b_{\text{useful}}}(k) \approx 0$, so $r_{\text{out}} \approx 0$ like in Fig. 5.

From Fig. 9, the click rate is given by

$$r_{\text{click}} = p_{\text{retry}} p_{\text{abort}} r_{\text{click}} + r_{\text{session}} + p_{\text{next}} (1 - p_{\text{abort}}) r_{\text{click}},$$

so

$$\begin{aligned} r_{\text{click}} &= \frac{r_{\text{session}}}{1 - p_{\text{click}}} \\ \text{where } p_{\text{click}} &= p_{\text{abort}} p_{\text{retry}} + (1 - p_{\text{abort}}) p_{\text{next}}; \end{aligned} \qquad (9)$$

here, p_{click} is the proportion of clicks that are followed by further clicks within their sessions. We thus get $n_{\text{click}} = 1/(1 - p_{\text{click}})$ from Eqn. 1.

Recall from Fig. 2 that $r_{\text{in}} = (1 - p_{\text{abort}}) r_{\text{click}}$. Hence,

$$\begin{aligned} r_{\text{in}} &= \frac{1 - p_{\text{abort}}}{1 - p_{\text{click}}} r_{\text{session}} \\ &= \frac{r_{\text{session}}}{1 - p_{\text{next}} + \frac{1 - p_{\text{retry}}}{\frac{1}{p_{\text{abort}}} - 1}} \end{aligned} \qquad (10)$$

This is the equation for the r_{in} curve in Fig. 5. When congestion is low, $p_{\text{abort}} \approx 0$ so from Eqn. 10,

$$r_{\text{in}} \approx \frac{r_{\text{session}}}{1 - p_{\text{next}}} \quad \text{for small } k.$$

(so $c_1 = 1/(1 - p_{\text{next}})$ in Fig. 5.) Without congestion, p_{next} is insensitive to k, so we get the flat segment of r_{in} in Figs. 5 and 6. As congestion increases, p_{next} and p_{retry} decrease while p_{abort} increases. The claim (2) now follows from Eqn. 10.

4 Model variations

The basic idea in our model is to extract two submodels from the single equilibrium $r_{\text{in}} = r_{\text{out}}$, and thus view steady state for a link as the interaction of user and network behavior.

In particular, if r_{session} changes (over 24 hours, say), the r_{in} curve (10) moves, but the r_{out} curve (8) remains fixed. We can thus view the r_{out} curve as the locus of equilibrium points as r_{session} changes.

On the other hand, changing the link bandwidth B will change the function $f_{b_{\text{useful}}}$ and thus r_{out} as well; however, the r_{in} curve will also move since, for the same k, a change in B will change the congestion perceived by the user.

If we assume that users perceive congestion through observing b_{useful}, then — through the monotonic relationship between b_{useful} and k in Fig. 7 — we can restate the model using r_{in} and r_{out} curves that are plotted with b_{useful} as the x-axis. In that case, the r_{in} curve does not change with B; users merely see a different b_{useful} and slide to a different part of the same r_{in} curve. However, the r_{out} curve (8) does change since, for the same b_{useful}, a different B means a different k. We can therefore view the r_{in} curve as the locus of equilibrium points as B changes.

Users also perceive congestion through observing t_{download}, so t_{download} could be used as the x-axis as well. Rather than focusing on completed downloads, we could use the offered load r_{click} for the y-axis instead.

We now present an alternative formulation of the basic two-curve idea in terms of r_{click} and t_{download}.

4.1 An alternative formulation

Since $t_{\text{download}} = S_{\text{download}}/b_{\text{useful}}$, where b_{useful} is as in Fig. 7, we have $t_{\text{download}} = f_d(k)$ for some function f_d determined by the application and TCP variants, as illustrated

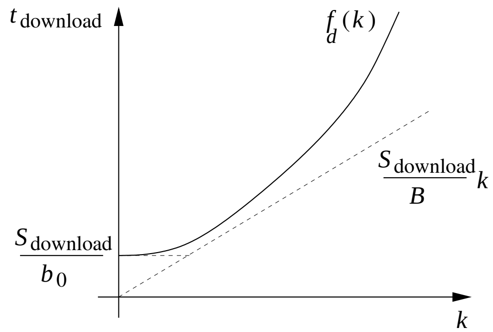

Figure 10: **Download delay t_{download} as a function $f_d(k)$ determined by the application and TCP.**

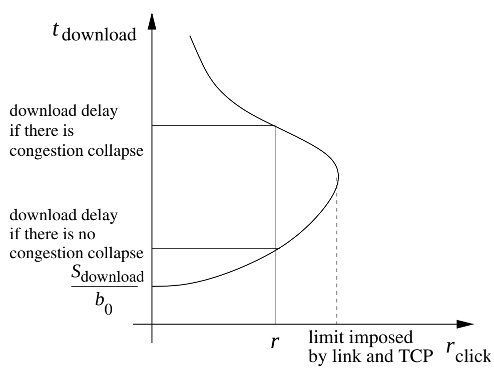

Figure 11: **In the network model, the click rate is limited by the link bandwidth B, TCP and the applications. For each click rate r before that limit, there are two possible delays t_{download}, depending on whether there is congestion collapse.**

in Fig. 10. Since f_d is monotonic, Eqn. 7 becomes

$$r_{\text{click}} = \frac{f_d^{-1}(t_{\text{download}})}{p_{\text{abort}}T_{\text{abort}} + (1 - p_{\text{abort}})t_{\text{download}}} . \quad (11)$$

This relationship is shown in Fig. 11, which corresponds to the network model in Fig. 3. It shows that, if we increase r_{click}, the link starts out uncongested, so $t_{\text{download}} \approx S_{\text{download}}/b_0$. As r_{click} approaches the limit that the link can support, congestion causes delay to grow exponentially. If there is congestion collapse, then the link can still support a small click rate, but at a higher delay.

Hence, for a given r_{click}, there are two possible t_{download} values: one before and one after collapse, as illustrated in Fig. 11.

If users experience delays and backoff, Eqn. 10 gives

$$r_{\text{click}} = \frac{r_{\text{session}}}{1 - p_{\text{next}} - p_{\text{retry}}p_{\text{abort}} + p_{\text{next}}p_{\text{abort}}} . \quad (12)$$

As the delay t_{download} increases, p_{abort} also increases, while p_{next} and p_{retry} decrease. At first, the click rate may increase as users abort downloads, but r_{click} eventually decreases. This is illustrated in Fig. 12.

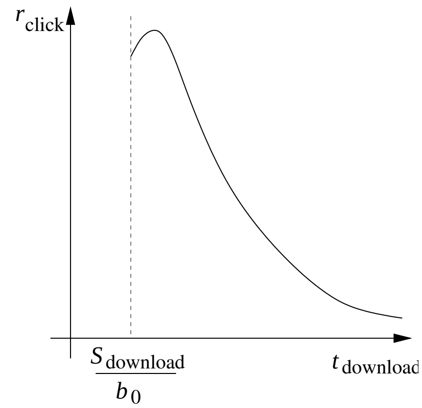

Figure 12: **In the user model, as the delay t_{download} increases, the click rate may increase at first (because of abortions), but r_{click} eventually decreases.**

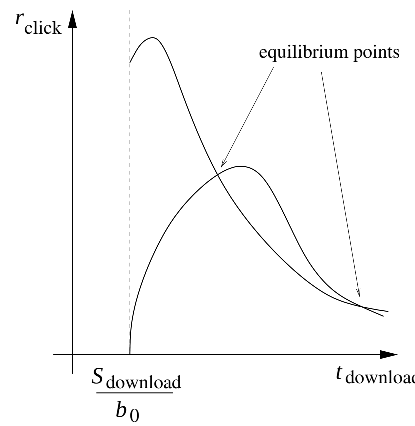

Figure 13: **The equivalent form of Fig. 6, in terms of t_{download} and r_{click}.**

If we now put this relationship from the user model together with the network model in Fig. 11, we get Fig. 13 — it shows how the network and user models interact to determine the equilibrium click rate. The arguments in Figs. 4, 5 and 6 about the roles of TCP congestion control and user backoff carry over to this equivalent formulation.

4.2 Persistent large flows

Users react to congestion only if they are monitoring the progress of a download, which may not be the case — a user may start downloading a large (e.g. MP3) file over a modem link, then go for dinner or to bed. Only a small fraction of Internet connections carry such large flows, but they dominate in terms of byte count [ZBPS02]. These large downloads can last for a long time, so with only TCP congestion control and no user backoff to regulate them, they can have a serious effect on performance.

To model this effect, we can add a class of downloads that

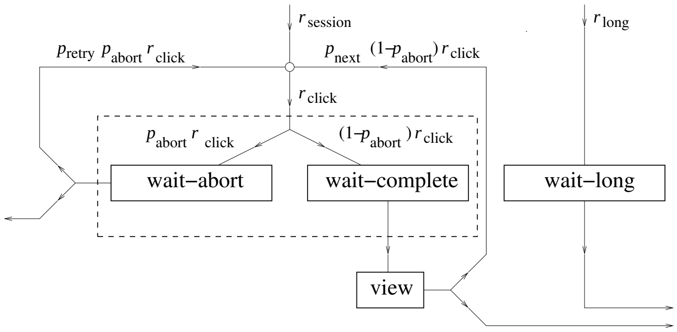

Figure 14: **Persistent large flows are modeled by adding a class to Fig. 9. Requests in this class arrive at a rate r_{long}, the size of each download is S_{long}, and they do not abort.**

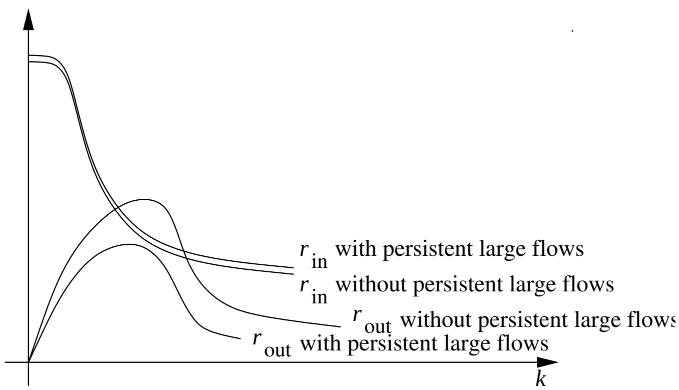

Figure 15: **Although persistent large flows may form a small percentage of TCP connections (so the effect on r_{in} is small), they can significantly lower the r_{out} curve, possibly causing congestion collapse (despite TCP congestion control).**

arrive at rate r_{long}, have size S_{long} each, and do not abort (regardless of congestion). This is illustrated in Fig. 14.

Now, instead of Eqn. 10, we have

$$r_{\text{in}} = \frac{r_{\text{session}}}{1 - p_{\text{next}} + \frac{1-p_{\text{retry}}}{\frac{1}{p_{\text{abort}}}-1}} + r_{\text{long}} \qquad (13)$$

Since large flows form a small fraction of Internet connections, we have $r_{\text{long}} \ll r_{\text{session}}$. Moreover, without $1 - p_{\text{retry}}$, $1 - p_{\text{next}}$ and p_{abort} to magnify its contribution, r_{long} makes a negligible difference to r_{in}, as illustrated in Fig. 15.

The impact on r_{out}, however, is significant. Instead of Eqn. 6, we now have

$$\begin{aligned} k = p_{\text{abort}} r_{\text{click}} T_{\text{abort}} \quad &+ \quad (1-p_{\text{abort}}) r_{\text{click}} \frac{S_{\text{download}}}{f_{b_{\text{useful}}}(k)} \\ &+ \quad r_{\text{long}} \frac{S_{\text{long}}}{f_{b_{\text{useful}}}(k)} \end{aligned}$$

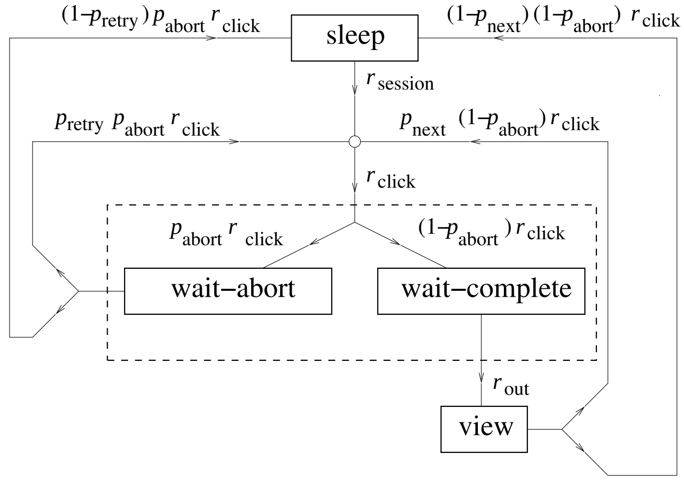

Figure 16: **Changing Fig. 9 into a closed model by adding a sleep state for users who are not active in a download-cum-view session.**

so

$$\begin{aligned} r_{\text{out}} &= (1-p_{\text{abort}}) r_{\text{click}} + r_{\text{long}} \\ &= \frac{k - r_{\text{long}} \frac{S_{\text{long}}}{f_{b_{\text{useful}}}(k)}}{\frac{p_{\text{abort}}}{1-p_{\text{abort}}} T_{\text{abort}} + \frac{S_{\text{download}}}{f_{b_{\text{useful}}}(k)}} + r_{\text{long}} \, . \qquad (14) \end{aligned}$$

Comparing Eqn. 14 and Eqn. 7, we see that — for the same value of k — r_{out} has an additional term r_{long} and a reduced term for the user-responsive flows. In effect, the persistent large flows reduce the bandwidth available to the others by occupying $r_{\text{long}} \frac{S_{\text{long}}}{f_{b_{\text{useful}}}(k)}$ of the k downloads. This can significantly lower the r_{out} curve, possibly causing a loss of equilibrium and resulting in congestion collapse, as illustrated in Fig. 15. This can happen even though the large flows are using TCP congestion control.

4.3 User model: closed system

So far, we have focused on an open model, in which sessions arrive at a constant rate (regardless of congestion). However, it may sometimes be more relevant (for an ISP, say) to consider a closed model, in which there is a constant number of users and r_{session} depends on how many of them are active. We therefore close the loops in Fig. 9 by introducing a **sleep** state for inactive users who are not in download-cum-view sessions, as shown in Fig. 16.

Let T_{view} be the time a user spends in the **view** state, and suppose there are k_{view} such users. As in the open model, r_{out} is given by Eqn. 8, so

$$k_{\text{view}} = r_{\text{out}} T_{\text{view}} = \frac{k T_{\text{view}}}{\frac{p_{\text{abort}}}{1-p_{\text{abort}}} T_{\text{abort}} + \frac{S_{\text{download}}}{f_{b_{\text{useful}}}(k)}} \, . \qquad (15)$$

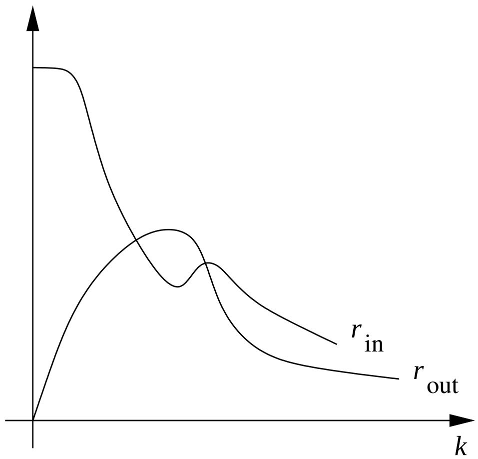

Figure 17: **For a closed model, r_{out} and r_{in} are given by Eqn. 8 and Eqn. 10. Heavy congestion can cause a drop in k_{view} (the number of users who are viewing their downloads), resulting in a local increase for r_{in}.**

Next, let T_{sleep} be the time a user spends in the **sleep** state, and suppose the user population has size N_{user}. Then

$$N_{\text{user}} = r_{\text{session}} T_{\text{sleep}} + k + k_{\text{view}},$$

so

$$r_{\text{session}} = \frac{1}{T_{\text{sleep}}}\big(N_{\text{user}} - k - k_{\text{view}}\big). \quad (16)$$

Substituting this into Eqn. 10 gives

$$\begin{aligned} r_{\text{in}} &= \frac{N_{\text{user}} - k - k_{\text{view}}}{T_{\text{sleep}}\big(1 - p_{\text{next}} + \frac{1-p_{\text{retry}}}{\frac{1}{p_{\text{abort}}}-1}\big)} & (17)\\ &= \frac{N_{\text{user}} - k - \frac{kT_{\text{view}}}{\frac{p_{\text{abort}}}{1-p_{\text{abort}}}T_{\text{abort}} + \frac{S_{\text{download}}}{f_{b_{\text{useful}}}(k)}}}{T_{\text{sleep}}\big(1 - p_{\text{next}} + \frac{1-p_{\text{retry}}}{\frac{1}{p_{\text{abort}}}-1}\big)} & (18) \end{aligned}$$

by Eqn. 15. Fig. 17 illustrates this new r_{in} equation. It shows that, while congestion is low, r_{in} behaves like in the open model, decreasing as k increases. However, when congestion becomes serious and p_{abort} approaches 1, the number of users in **view** state (k_{view}) drops — see Eqn. 15. By Eqn. 17, this can cause a local increase in r_{in}. This change in r_{in} does not affect the equilibrium and stability arguments in Figs. 4, 5 and 6, and they remain valid for the closed model.

5 Model analysis: bandwidth provisioning

We now illustrate how an analysis of the model can help us understand the issue of bandwidth provisioning.

If the link bandwidth B is well-provisioned to support user demand, we expect the throughput (rate of completed downloads) to increase as k increases. In other words, the r_{in}-vs-r_{out} equilibrium should occur at some k for which $\frac{dr_{\text{out}}}{dk} > 0$.

From Eqn. 8,

$$r_{\text{out}} T_{\text{abort}} = \big(\frac{1-p_{\text{abort}}}{p_{\text{abort}}}\big)\big(k - \frac{S_{\text{download}}}{b_{\text{useful}}} r_{\text{out}}\big). \quad (19)$$

Assuming the averages T_{abort} and S_{download} to be constants with respect to k, we get

$$\begin{aligned} &\big(p_{\text{abort}} T_{\text{abort}} + (1-p_{\text{abort}})\frac{S_{\text{download}}}{b_{\text{useful}}}\big)\frac{dr_{\text{out}}}{dk} \\ = \ &\big((1-p_{\text{abort}}) - \frac{k}{p_{\text{abort}}}\frac{dp_{\text{abort}}}{dk}\big)\big(1 - \frac{S_{\text{download}} r_{\text{out}}}{k b_{\text{useful}}}\big) \\ &+(1-p_{\text{abort}})\frac{S_{\text{download}} r_{\text{out}}}{k b_{\text{useful}}}\big(1 + \frac{k}{b_{\text{useful}}}\frac{db_{\text{useful}}}{dk}\big) \\ = \ &\big((1-p_{\text{abort}}) - \frac{k}{p_{\text{abort}}}\frac{dp_{\text{abort}}}{dk}\big)\frac{p_{\text{abort}} r_{\text{out}} T_{\text{abort}}}{k(1-p_{\text{abort}})} \\ &+(1-p_{\text{abort}})\frac{S_{\text{download}} r_{\text{out}}}{k b_{\text{useful}}}\big(1 + \frac{k}{b_{\text{useful}}}\frac{db_{\text{useful}}}{dk}\big) \end{aligned}$$

by Eqn. 19. Thus,

$$\frac{dr_{\text{out}}}{dk} \geq 0 \quad \text{if} \quad \frac{k}{p_{\text{abort}}}\frac{dp_{\text{abort}}}{dk} \leq 1 - p_{\text{abort}} \quad \text{and} \quad \frac{k}{b_{\text{useful}}}\frac{db_{\text{useful}}}{dk} \geq -1 \quad (20)$$

and

$$\frac{dr_{\text{out}}}{dk} < 0 \quad \text{if} \quad \frac{k}{p_{\text{abort}}}\frac{dp_{\text{abort}}}{dk} > 1 - p_{\text{abort}} \quad \text{and} \quad \frac{k}{b_{\text{useful}}}\frac{db_{\text{useful}}}{dk} < -1.$$

Now, for small Δx,

$$\frac{x}{y}\frac{dy}{dx} \approx \frac{\frac{\Delta y}{y}}{\frac{\Delta x}{x}},$$

so $\frac{x}{y}\frac{dy}{dx}$ measures the **elasticity** [WB76] of y with respect to x, i.e. the percentage change in y induced by a percentage change in x. We therefore refer to $\frac{k}{p_{\text{abort}}}\frac{dp_{\text{abort}}}{dk}$ as the **abort-elasticity** (for user behavior) and $\frac{k}{b_{\text{useful}}}\frac{db_{\text{useful}}}{dk}$ as the **bandwidth-elasticity** (for network behavior).

Hence, one can monitor the adequacy of B by checking the requirements for these two elasticities in Eqn. 20. If one of the two inequalities fail, then the equilibrium is approaching an undesirable operating zone (where $\frac{dr_{\text{out}}}{dk} < 0$).

6 Conclusion

The model in this report is a work in progress. Its basic ideas are the concept of user backoff (U1, U2) and the technique of splitting the equilibrium into two interacting submodels for the users and the network. Beyond these, we expect the model to evolve, possibly into a form very different from the one described here.

The evolution will be driven by measurements. The first step in this direction is to understand how user actions (click, abort, etc.) manifest themselves in client/server/router traces, so that these can be reverse-engineered to yield the user model (p_{retry}, T_{abort}, etc.); the common practice now of using various thresholds for clustering packet flows is inadequate for this purpose.

The clustering of trace events by user actions is also necessary for building an analytical model for a download. Although much work has been done in recent years on modeling TCP and HTTP behavior [HMTG01, JBKB03], we have not come across analytical models for downloads (i.e. the multiple TCP connections and HTTP requests associated with a single user action). Such models are needed to determine $f_{b_{\mathrm{useful}}}(k)$ in Fig. 7.

Our model analyzes traffic behavior by splitting downloads (wait-abort, wait-complete, long transfers). Measurements may show that further refinement is necessary.

The model here is *qualitative* — we see it as a tool for analyzing the interaction between the network and the users, a framework for understanding their combined behavior. For now, it is unclear whether it can evolve into a *quantitative* model that can predict performance and thus become useful as a tool for capacity planning (e.g. bandwidth provisioning for ISPs).

Acknowledgements
Many thanks to Dave Andersen, Arthur Berger, Kyle Jamieson, Jaeyeon Jung, Dina Katabi, Jinyang Li and Alex Yip for helpful discussions.

References

[APS99] M. Allman, V. Paxson, and W. Stevens. *TCP Congestion Control*. IETF, April 1999. RFC 2581.

[FF99] S. Floyd and K. Fall. Promoting the use of end-to-end congestion control in the Internet. *IEEE/ACM Transactions on Networking*, 7(4):458–472, 1999.

[HMTG01] C. V. Hollot, Vishal Misra, Donald F. Towsley, and Weibo Gong. A control theoretic analysis of RED. In *INFOCOM*, pages 1510–1519, April 2001.

[Jac88] V. Jacobson. Congestion avoidance and control. In *Symposium proceedings on Communications architectures and protocols*, pages 314–329. ACM Press, 1988.

[JBKB03] Jaeyeon Jung, Arthur W. Berger, C. Emre Koksal, and Hari Balakrishnan. *A Structural Model for HTTP Connection Arrivals*, May 2003. sumitted.

[Met95] B. Metcalf. Predicting the Internet's catastrophic collapse and ghost sites galore in 1996. *Infoworld*, December 1995.

[Nag84] J. Nagle. *Congestion Control in IP/TCP*. IETF, January 1984. RFC 896.

[ONA02] David P. Olshefski, Jason Nieh, and Dakshi Agrawal. Inferring client response time at the web server. In *Proceedings of the 2002 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 160–171. ACM Press, 2002.

[Pos81] J. Postel. *Transmission Control Protocol*. IETF, September 1981. RFC 793.

[WB76] A.C. Williams and R.A. Bhandiwad. A generating function approach to queueing network analysis of multiprogramming computers. *Networks*, 6(1):1–22, 1976.

[ZBPS02] Y. Zhang, L. Breslau, V. Paxson, and S. Shenker. On the characteristics and origins of internet flow rates. In *Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 309–322. ACM Press, 2002.