Telecommunication Systems 30:4, 351–385, 2005

Transport Protocols in the TCP Paradigm and their Performance

TEUNIS J. OTT ott@oak.njit.edu
Department of Computer Science, New Jersey Institute of Technology

Abstract. This paper defines a class of "TCP-like" transport protocols, called protocols in the TCP Paradigm. The name indicates the protocols are not TCP, but in some sense similar to it.

The class includes TCP and also Tom Kelly's "Scalable TCP", and more. Most of the protocols in the class require ECN in order to become implementable. Most even require a form of ECN that allows a high rate of marked packets.

The paper analyzes performance of protocols in the TCP Paradigm and indicates a subset that is likely to perform as well as, or quite possibly better than, Scalable TCP. Criteria are the ability to achieve high throughput, to maintain a steady flow if the ECN marking probability is constant, and to adapt quickly to a changing marking probability.

A main conclusion is that in order to benefit from protocols in this paradigm, "source behavior" (reaction of endpoints to marked and unmarked packets) and "router behavior" (how routers choose marking probabilities) must be investigated together.

Keywords: Internet, transport protocols, TCP, protocol analysis, performance, performance of transport protocols

1. Introduction

This is a re-write of the paper [46] which has existed on the web since 1999 and was presented in a workshop at ENS, Paris, Sept 2000, but which has not appeared in the open literature yet. It has been rewritten to reflect changes since 1999. Some new results have been added, in particular in Sections 8 and 9.

The paper [46] was a natural extention of the paper [50] which existed on the web since 1996 and was presented in an informal workshop of the IFIP Working Group WG7.3 during Performance, 96 in Lausanne, in Oct 1996, and also in DIMACS workshop in Nov 1996. Also that paper has not been published in the open literature, yet. It started an avalanche of papers of which [51] is the best known.

Consider a flow of packets in IP with per packet acknowledgement. Assume the system allows "Explicit Congestion Notification" (ECN, see e.g. [25,59]): When a router recognizes one of its buffers is getting close to congestion, it can set a "Congestion Indicator Bit" in packets flowing through this buffer. To avoid confusion with the similarly named bit in ATM with ABR, Floyd and Ramakrishnan call this the "Congestion Experienced" (CE) bit, or, more accurately, the CE Codepoint (two bit-pattern) (1,1). In this note we use that name (CE), but make different use of the "CE bit pattern" and simply call it the CE bit. Setting of the CE bit can be done probabilistically, with a probability p

that depends on the perceived level of congestion. The destination copies those CE bits into the ECN-Echo bit in acknowledgements. Thus, the source is informed of congestion. [25] and [59] discuss a number of implementation issues, such as location of the bit and what to do when there are delayed acknowledgements.

When the router sets the CE bit in a packet, we also say it marks the packet. The router can (for example) choose a state-dependent probability p and mark packets with probability p.

[25] and [59] repeatedly state the opinion that the source of traffic must react to a returning ECN-Echo bit that has been set (=1) in (almost) exactly the same way a TCP source reacts to discovering a "congestion event" that includes loss of at least one packet.

This is an opinion the author of this note does not share.

Among the advantages of marking instead of dropping (as in RED) are (i) that no re-transmission is needed, and unlike in case of drop there is no period of about 1 RTT of duplicate acknowledgements, with possible confusion about further dropped packets, and therefore (ii) any marking probability $0 \leq p \leq 1$ is acceptable. Drop probabilities have to be at most not much more than .1, or (for example) TCP stops functioning, see e.g. Appendix B in this paper. In addition, modern traffic endpoints (sources, destinations) have the ability to interprete the meaning of a stream of ECN-Echo bits quite carefully, and for example as a function of the type or class of service the packets belong to. The author of this note advocates that ECN-capable flows react to ECN-Echo bits in ways that still need to be defined and that may be quite different from the way a non ECN-capable flow reacts to dropped packets. In this paper we often use the terms "ECN savvy" and "ECN illiterate" instead of "ECN-capable" and "non ECN-capable".

Unfortunately, the authors of [25] and [59] chose an implementation of ECN that makes it impossible to signal congestion more than once every RTT: Once the "destination" receives a packet with the CE codepoint set, it sets the ECE bit in all packets in the opposite direction, until it gets a packet with the CWR bit set, i.e. for about 1 RTT. While this gives protection against lost packets with the ECE flag set, it significantly reduces the flexibility on the feedback mechanism, as will be shown at the end of Section 2.

Later in this paper a number of arguments will be given that indicate that having a wider range of possible marking probabilities improves controlability of the network.

This paper thus, among other items, proposes to modify the ECN implementation, enabling and even encouraging a higher marking probability for ECN savvy flows than drop probability for ECN ignorant flows.

In the simplest situation the intent is to create a situation where ECN savvy flows and ECN ignorant flows, following the same path through the network, can have strongly dissimilar marking probability, respectively drop probability, yet (for fairness) see similar or at least not too dissimilar performance. If the marking probability of ECN savvy flows is higher than the drop probability of ECN ignorant flows, and if the goal is to have the various flows obtain similar congestion windows, the ECN savvy flows must react differently to marking than the ECN ignorant flows react to drop. This limited use of ECN, if well-designed, should lead to better controlled traffic, in particular if the fraction of ECN savvy traffic increases.

In a multi-class situation (e.g. Differentiated Services) the goal may in fact be to achieve differentiation in the performances seen by different flows. In the discussion below we have one class of ECN ignorant flows and potentially multiple classes of ECN savvy flows.

Thus, a challenge is to create source behaviors (how sources react to drop or marking) and router behaviors (how routers decide which packets to drop or mark) that achieves the goal of appropriate (dis)similarity in performance encountered.

For the sake of simplicity of language we will often pretend that “router behavior” is defined simply in terms of state dependent drop- and marking probabilities, that is, a drop probability and a number of marking probabilities (one for every class of ECN savvy flows) that are functions of the current and recent congestion as seen by the router, e.g. as in RED functions of output buffer occupancies. Actual router behavior may be more complicated. For example, we will see that under certain circumstances routers may decide to alternate high and low levels of drop probability, instead of having a closer to constant drop probability. Routers also may decide to temporarily overshoot a target drop probability in order to force flows to quickly adapt to changing circumstances. These ideas are not further developed in this paper.

The marking probability for ECN-capable flows may depend on the type (type of service, priority class, etc.) of the packet. Thus, the router has a number of state dependent “signaling congestion” variables: The drop probability for ECN non-capable flows; the drop probability for ECN-capable flows (might be class dependent); and the marking probability for ECN-capable flows (might be class dependent). For example, for flows with a guaranteed rate (and that stay within their rate) it makes no sense to drop unless there is no choice, and it makes no sense to mark apart from as a warning signal that involuntary drop is imminent. Similarly, if in IP there is an option of distinguishing between “In-Rate” and “Out-of-Rate” packets (or “In-Profile” versus “Out-of-Profile” packets, using an “In-Rate” bit), “In-Rate” packets would be dropped only involuntarily, and would be marked only as a warning that involuntary drop is imminent. “Out-of-Rate” packets could be dropped as well as marked, with marking of course the preferred option. This implies that the signalling (e.g. carried by acknowledgements) from destination to source must separately signal markings of In-Rate and of Out-of-Rate packets. This router behavior is illustrated in Table 1.

This use of different marking and/or dropping probabilities for different types of flows is further illustrated in the Sections 2 and 3.

In the foreseeable future, ECN-capable routers would set drop probabilities for ECN non-capable flows in a way consistent with RED or SRED or some such mechanism. ECN capable flows would react to drop in essentially the same way as ECN non-capable flows. This is necessary, because for some time there would be ECN-capable as well as ECN non-capable routers. Some changes would be allowed for special classes (say flows paying a “premium” tariff). ECN-capable routers could set the

Table 1
Marking and dropping probabilities.

Class	Marking	Dropping
Non-ECN	0	$p_{n,d}$
ECN Class 0	$p_{0,m}$	$p_{0,d}$
$\vdots$	$\vdots$	$\vdots$
ECN Class L	$p_{L,m}$	$p_{L,d}$

marking probability for ECN-capable flows in just about any way, as long as Router Behavior and Flow Behavior are designed together. Some indication of the consideration going into Router Behavior can be found in the Sections 3 and 5.

Making sure that those new (marking) behaviors have been studied and implemented in (some) routers before many end-stations become ECN capable may very well be the only way to make an elegant transition to a new environment.

Once ECN is ubiquitous, at least in routers, (endstations may take longer!) the meaning of drop for ECN-capable flows in theory could still change. This would be a hard transition to make and might very well be impossible, because by that time there will be many ECN-capable endstations that can not make a synchronized change. By that time, the only way to achieve change might be by introducing new classes.

Thus, the advent of ECN is an opportunity to modify the congestion algorithms in IP. We should use this opportunity to study Router Behavior (whether and when to mark packets) and Source Behavior (how to react to marked and unmarked packets) as two aspects of the same problem. The advent of ECN temporarily gives us a clean slate that we can fill in with new mechanisms, using what we have learned in the past 20+ years, and taking into consideration the greater capabilities in modern endstations, and the much higher bandwidths in the modern Internet. Since this is likely the last opportunity to do a significant amount of redesign of the control mechanisms, it is important to use the opportunity well!

In much of this paper we do a “quasi-stationary” analysis of the performance of “TCP-like” flows under the assumption of constant marking probability p_{mark} (or constant drop probability p_{drop}) and obtain information about stationary distributions of congestion windows etc. To study “controllability” of flows under changing circumstances, we need to study transient behaviors. This is done in [48]. Some results from that paper are used in Section 9 to predict how fast a flow will adapt when a router changes the drop probability.

In future work we expect to use the insight gained in this paper, and in [47–49] and other work in progress, to propose router behaviors, i.e. ways for routers to decide whether and when to mark packets. That will include, but go beyond, simply setting state dependent marking probabilities.

Apart from small detours in the Sections 3 and 5 we do, in this note, not yet consider class dependent marking. In particular we do not yet consider separate marking policies for In-Rate and Out-of-Rate packets. Mechanisms where “entry-routers” or "edge-routers" mark packets as either In-Rate or Out-of-Rate, where other routers have different marking policies for such packets, and where the reaction of sources to marked packets depends on whether the packets were In-Rate or Out-of-Rate look like a very promising area of future research.

In Section 2 a general class of "source behaviors" will be defined. In Section 5 this class will be restricted to a class which still contains "classical" TCP as well as "Scalable TCP", see [32], and many of the behaviors described in [11]. In later sections, the performance of these behaviors is analyzed. For the time being, "Scalable TCP" comes off as an attractive candidate. There are, however, other candidates with at least in theory more attractive high-performance behavior.

It must be noted that Tom Kelly's "Scalable TCP", [32], is quite different from Robert Morris' "Scalable TCP", [45]: The first is on how TCP sources ought to react to dropped or marked packets, the second is on how routers ought to choose buffer sizes.

The most promising candidates for "source behavior" will be discussed in Section 11.

The analyses done in this note are examples of the analyses that should have been done before a newly designed ECN was finalized. There is no pretension of having achieved closure. The most important recommendation is that Router behavior and Source behavior must be designed together and must be class dependent, i.e. depend on the class of the packet.

2. The TCP Paradigm with general increases and decreases

Let us consider a Congestion Window based protocol where whenever an acknowledgement arrives at the source that acknowledges an unmarked data packet while the congestion window is W, then the congestion window increases by *incr(W)*, and when a marked packet is acknowledged the congestion window decreases by *decr(W)*. (Also, when a packet is marked while the W is small there will be a time-out, with time-outs probably increasing exponentially after repeated such markings. We do not consider such details in this note.) At this point it does not matter whether W is expressed in bytes or in MSSs or in some other entity. Assuming packets are marked with probability p, and that "locally in time" p is constant, the drift per packet in the congestion window while the congestion window is W is

$$\begin{aligned} E[W_{n+1} - W_n | W_n = W] &= \mathit{drift}(W, p) = (1 - p).\mathit{incr}(W) - p.\mathit{decr}(W) \\ &= p.\mathit{incr}(W).\left(\frac{1-p}{p} - \frac{\mathit{decr}(W)}{\mathit{incr}(W)}\right). \end{aligned} \tag{2.1}$$

Let us now consider the function

$$q(W) = \frac{\mathit{decr}(W)}{\mathit{incr}(W)}. \tag{2.2}$$

Assuming this function is reasonably smooth, and that p indeed is constant during a large number of packets sent, the congestion window will tend to spend most of its time at W values for which $q(W)$ is not too far from $\frac{1-p}{p}$.

Common sense indicates that higher p must indicate a lower target value for W. It therefore is highly desirable that $q(\cdot)$ is a strictly increasing function with

$$q(1) = 0, \quad \lim_{W \to \infty} q(W) = \infty. \tag{2.3}$$

In that case, if the marking probability p is constant, $q(W)$ will fluctuate around $\frac{1-p}{p}$. Thus we can predict the average window size: find $W(p)$ with

$$q(W(p)) = \frac{1-p}{p}. \tag{2.4}$$

Equation (2.3) ensures that (possibly with interpolation or rounding to an integer) there always is a solution to (2.4). The actual window size will tend to fluctuate around $W(p)$. We use $W(p)$ as approximation for the average. The functions $q(W)$ and $W(p)$ really are response surfaces of the sources to drop- or marking probabilities p.

It is useful to note here a connection with "fairness": In the situation of (2.3), if two different flows that react in the same way to packet markings encounter the same marking probability p, they will tend to have the same (average) congestion windows.

Note that if the function $q(\cdot)$ is almost constant over a long range of W values, we do not have fairness: if for a while $\frac{1-p}{p}$ happens to remain constant, equal to that $q(W)$ value, and two flows start with different W values in that range of W values, they will tend to keep their different congestion windows for a long time.

In the special case of TCP (without delayed acknowledgements, and dropping instead of marking) we have (congestion windows are now measured in MSSs):

$$incr(W) = \frac{1}{W}, decr(W) = \frac{W}{2}, \tag{2.5}$$

and thus

$$q(W) = \frac{W^2}{2}, \quad W(p) = \sqrt{\frac{2(1-p)}{p}}. \tag{2.6}$$

This is the basis for the celebrated square root formula, see e.g. [50]. With Delayed Acknowledgements a better formula is

$$W(p) = \sqrt{\frac{1-p}{p}} \tag{2.7}$$

and often $1 - p$ is approximated by 1, see [50, 51] for details and refinements.

The function $q(\cdot)$ defines a "response surface" of W to p. We see that we can first choose $q(\cdot)$ arbitrarily (subject to the monotonicity and (2.3)), and then still can choose *incr*($\cdot$) and *decr*($\cdot$) somewhat arbitrarily. Thus we have a choice of two modi operandi: we can choose *incr*($\cdot$) and *decr*($\cdot$) functions and find out what he resulting response surface is, or we can choose a response surface and then (with that degree of freedom gone) find *incr*($\cdot$) and *decr*($\cdot$) functions.

If we decide to first choose the response surface $q(W)$, (or $W(p)$), we can go one step further: after choosing this response surface we do not choose *incr*($\cdot$) and *decr*($\cdot$), but we let the source directly estimate the marking probability *p*, and adjust the congestion window *W* accordingly. This method has interesting consequences, that will be investigated in Sections 12 and 13. In fact, under certain circumstances it leads back to a system with *incr*($\cdot$) and *decr*($\cdot$) functions as above. Thus, it can be seen as a somewhat scientific way of choosing such functions.

For TCP, the desire to have multiplicative decrease governed the choice of $decr(\cdot)$. This led to the somewhat unfortunate result that the congestion window occasionally halves. For connections with a long round trip time that therefore need a high congestion window this is a problem: after a halving of the congestion window, the congestion window increases quite slowly. This has a number of unfortunate consequences. One of these is that TCP traffic is not very good "background" traffic for other streams.

Choosing both *decr*(W) and *incr*(W) much smaller (but leaving the quotient the same) has the advantage that with *p* constant a flow following that behavior has the same average *W* value but behaves much more smoothly. This can have the disadvantage that the flow reacts quite slowly to changing *p*. However, we will see that this disadvantage is greatly reduced, or eliminated, if ECN is used with the ability to effectively set the marking probability *p* very high.

It must be noted that not all versions of TCP are consistent in using multiplicative decrease. Many versions of TCP allow the window to be halved only once in a round trip time or only once per congestion episode.

It must also be noted that as long as the marking probability *p* is enabled to effectively take on high values, e.g. close to one or even equal to one, quite small values of *decr*(w) still effectively allow multiplicative decrease:

In the extreme we could use $decr(W) = d$ (MSSs) (with $0 < d < 1$) in a scheme with marking instead of dropping. In that case, setting $p = 1$ during one round trip time (or, in case of delayed acknowledgements during two round trip times) reduces all windows (for flows with that RTT!) to a fraction $(1 - d)$ of the value before that period (or possibly to 1 MSS or less, time-out counting as a congestion window of less than one MSS): if $.5 < d < 1$ more draconic than halving all windows! Thus, because a marked packet is not lost (with all the unpleasant consequences of losing a packet), it is possible to send a strong signal by marking many packets in a short period. Choosing $d = 1, decr(W) = 1$, has the additional effect that acknowledgement of a marked packet does not cause transmission of a new packet. Thus, the router can fairly accurately predict the consequences of marking a packet. With multiplicative decrease, the router needs to "know" the window size to predict the consequences of marking a packet.

Another interesting side-effect of choosing $decr(w) = d$ is that in some cases (repeated short periods of high p) it gives a bias in favor of flows with long RTT, thus possibly counteracting the usual bias against flows with long RTT. This is an aspect of router behavior that needs further study.

Of course, a larger value of $decr(w)$, for example $decr(w) = c_2\sqrt{w}$ (see Section 5) will decrease or eliminate the congestion window in an even shorter amount of time.

It must be noted that the current implemetation of ECN as in [59] makes it impossible to set the marking probability effectively higher than $p = \frac{1}{W}$, where of course W is the congestion window in MSSs.

3. Fairness and TCP friendliness

A modification of router-and endpoint behavior as above risks leading to a "less than TCP friendly" situation. The NSF report [19] states that insisting on "TCP Friendliness" is freezing further Internet protocol development and recommends to stop making a big point of TCP friendliness.

Of course, when TCP friendliness can be achieved at low cost, an attempt should be made. It is quite possible that the new behaviors advocated can in fact achieve some kind of TCP friendliness:

When multiple different *incr* and *decr* functions are used, and routers have "class dependent" dropping and marking policies, we can aim to maintain certain relationships between the "typical" congestion windows of different classes. For example, we can aim to maintain a specific ratio. That ratio can be 1: for example, there may be a state dependent drop probability p_{drop} for "best effort, ECN illiterate" TCP flows and a state dependent p_{mark} for "best effort, ECN savvy" TCP flows. In that case we want the drop policy (p_{drop} as function of the level of congestion) and the marking policy (p_{mark} as function of the level of congestion), and the *incr* and *decr* functions for the different flows, to be such that with plausible accuracy, most of the time, p_{drop} and p_{mark} are related in such a way that

$$W_{drop}(p_{drop}) = W_{mark}(p_{mark}), \tag{3.1}$$

where of course $W_{drop}(\cdot)$ is obtained as in (2.4) from $q_{drop}(W) = decr_{drop}(W)/incr_{drop}(W)$, similar for the case of marking. This makes it possible to compute (e.g.) p_{drop} as function of p_{mark-}, and p_{mark} we let depend in one of the usual ways on the perceived level of congestion in the router. An example is given in (5.10) in this paper. This mechanism was used in a few special cases in [40] and in [41].

Maintaining such a relationship between various drop- and marking probabilities will be sufficient to ensure relative fairness between the “ECN savvy” and “ECN illiterate” flows as long as the dropping and marking probabilities change only slowly (the “quasi-stationary” approach works). It is an interesting problem to find out what happens in case of a very dynamic situation.

4. Outside the TCP paradigm

In the previous section we had, as in TCP, $incr(\cdot)$ and $decr(\cdot)$ functions, and (in principle) a congestion window modification is made every time the source receives an acknowledgement, and the update uses the $incr(\cdot)$ and $decr(\cdot)$ functions. We call this situation the “TCP Paradigm”. There are of course more general mechanisms. For example, the source could count “marked” and “unmarked” acknowledgements, and every now and then (say once every Round Trip Time RTT) update the congestion window. Such mechanisms are outside the TCP Paradigm.

Mechanisms “on the boundary of” the TCP Paradigm are for example those discussed in Section 12, where a “response surface” $q(\cdot)$, $W(\cdot)$ as in (2.3), (2.4) has been chosen, an estimate $\bar{p}$ for the marking probability p is maintained, and the congestion window actually used is $W(\bar{p})$. Such a scheme was already referred to in Section 2. In Section 13 a very interesting such mechanism will be described, of which we then find that it can be implemented in two different ways: one mechanism inside the TCP Paradigm, the other outside. This can be seen as a “scientific” way of choosing $incr(\cdot), decr(\cdot)$.

More research is needed to check whether there are pairs of router behavior–endstation behavior outside the TCP Paradigm that are superior to all mechanisms inside the TCP Paradigm.

5. A special class of $incr(\cdot)$ and $decr(\cdot)$ functions

In the remainder of this note we restrict ourselves to $incr(\cdot)$ and $decr(\cdot)$ of the form

$$incr(w) = c_1 w^{\alpha}, decr(w) = c_2 w^{\beta}. \tag{5.1}$$

Similar functions have been studied in [11]. For these functions to make sense we obviously want $\alpha < 1, c_1 > 0$ and $\beta \leq 1, c_2 > 0$, and if $\beta = 1$ we clearly need $0 < c_2 < 1$.

In this Section and in the next few Sections we investigate the behavior of congestion control mechanisms based on $incr(\cdot)$ and $decr(\cdot)$ functions as above. In Section 11 we draw some conclusions on what values are promising.

Throughout most of this note we consider general α and β etc. Many of the results in this section can be extended to more general $incr(\cdot)$ and $decr(\cdot)$ functions. At this point that is an exercise of limited interest.

It must be noted that the combination ($\alpha = -1, \beta = 1$) is TCP, and is usually called "AIMD" (Additive Increase, Multiplicative Decrease), see e.g. [11]. The combination ($\alpha = 0, \beta = 1$) is called "MIMD", this is what Tom Kelly [32] with good reason calls "Scalable TCP". That may very well be the best combination. However, there are some other extremely interesting possibilities, see Section 11.

With the choice of $incr(\cdot)$ and $decr(\cdot)$ functions as above, from a mathematical point of view we are studying the process $W_{p,n}$ defined by:

Let the random variables $(\chi_{p,n})_{n=0}^{\infty}$ be iid, with

$$P\{\chi_{p,n} = \text{success}\} = 1 - p,\ P\{\chi_{p,n} = \text{failure}\} = p, \tag{5.2}$$

where of course $0 < p < 1 . \chi_{p,n} = $ success means that the n-th packet was unmarked, and $\chi_{p,n} =$ failure means that the n-th packet was marked.

Further, let the discrete time, continuous state space process $W_{p,n}$ ($n = 0, 1, 2, \ldots, 0 < W_{p,n} < \infty, 0 < p < 1$) be defined by

$$W_{p,n+1} = \begin{cases} W_{p,n} + c_1 W_{p,n}^{\alpha} & \text{if } \chi_{p,n} = \text{success}, \\ W_{p,n} - c_2 W_{p,n}^{\beta} & \text{if } \chi_{p,n} = \text{failure}, \end{cases} \tag{5.3}$$

To make sure of no problems at $W_{p,n} = 0$ if $\alpha > 0$ or $\beta < 1$ (or both), we add in those cases a lower bound $W_{p,n} \geq 1$. In those cases, if at some point (5.3) causes $W_{p,n+1} < 1$, $W_{p,n+1}$ is immediately (or after an ensuing time-out ends) reset to 1.

Henceforth we always write χ_n and W_n for $\chi_{p,n}$ and $W_{p,n}$. It must be remembered, in particular in Section 9, that the subscript p really is there.

With the choice of functions as in (5.1) we get the response surface

$$q(w) = \frac{decr(W)}{incr(W)} = \frac{c_2}{c_1} w^{\beta - \alpha}. \tag{5.4}$$

Thus, for packet marking probability p constant, and marking independent from packet to packet, when transporting a very big file the congestion window will tend to fluctuate around $w(p)$, defined as

$$w(p) = \left(\frac{c_1}{c_2} \quad \frac{1-p}{p} \right)^{\frac{1}{\beta - \alpha}}. \tag{5.5}$$

When p is quite close to zero, (5.5) becomes

$$w(p) = \left(\frac{c_1}{c_2 p} \right)^{\frac{1}{\beta - \alpha}}. \tag{5.6}$$

If we choose that "more marking signals more congestion" (if we choose $W(p)$ to be decreasing in p) we must have $\alpha < \beta \leq 1$. This achieves (2.3). It also means that for p small the lower bound following (5.3) is no more than a mathematical convenience.

While from a mathematical point of view we are studying the process W_n as in (5.3) we must never forget that the real objects of study are the protocols it is related to.

The main weakness of the process W_n as model for a transport protocol in the TCP paradigm is that it does not include consideration of the delay in the feedback (of 1 RTT).

In addition, practically, when the congestion window is small the protocols will have different behaviors than indicated by (5.3). Equation (5.3) is likely to mainly describe behavior for large W_n (the high performance situation).

In classical TCP we had $c_1 = 1, \alpha = -1, c_2, = \frac{1}{2}, \beta = 1$.

For the special class of functions considered in this section, the constraint (3.1) becomes more tractable. Let the "ECN savvy" flows use $\alpha_m, \beta_m, c_{1m}, c_{2,m}$ and let the "ECN illiterate" flows use $\alpha_d, \beta_d, c_{1,d}, c_{2,d}$. We can let p_{drop} depend on the perceived level of congestion, then let p_{mark} be the function of p_{drop} that satisfies

$$\left(\frac{c_{1,d}}{c_{2,d}}\frac{1-p_d}{p_d}\right)^{\frac{1}{\beta_d-\alpha_d}} = \left(\frac{c_{1,m}}{c_{2,m}}\frac{1-p_m}{p_m}\right)^{\frac{1}{\beta_m-\alpha_m}} \tag{5.7}$$

Slightly less accurate, but possibly more convenient, would be

$$\left(\frac{c_{1,d}}{c_{2,d}}\frac{1}{p_d}\right)^{\frac{1}{\beta_d-\alpha_d}} = \left(\frac{c_{1,m}}{c_{2,m}}\frac{1}{p_m}\right)^{\frac{1}{\beta_m-\alpha_m}} \tag{5.8}$$

(5.8) makes it attractive to consider the situation where $\beta_d - \alpha_d = \beta_m - \alpha_m$. In that case we could choose

$$p_m = \frac{c_{1,m}c_{2,d}}{c_{2,m}c_{1,d}}p_d. \tag{5.9}$$

However, in all likelihood $\beta_d - \alpha_d = 2$, and we will see that almost certainly in any intelligently chosen set-up $\beta_m - \alpha_m \leq 1$. So, (5.9) is just too simple to wish for.

From the point of view of implementability it may be preferable to make p_m the variable directly determined by the perceived level of congestion, and set p_d equal to

$$p_d = \frac{c_{1,d}}{c_{2,d}}\left(\frac{c_{2,m}}{c_{1,m}}p_m\right)^{\frac{\beta_d-\alpha_d}{\beta_m-\alpha_m}}, \tag{5.10}$$

with for example

$$\frac{c_{1,d}}{c_{2,d}}\left(\frac{c_{2,m}}{c_{1,m}}\right)^{\frac{\beta_d-\alpha_d}{\beta_m-\alpha_m}} < 1. \tag{5.11}$$

(5.11) guarantees that $p_d < 1$ even when $p_m = 1$. If perceived congestion grows beyond the point where $p_m = 1$ the router could further increase p_d. It must be remembered, though, that for TCP setting $p_d \geq .1$ essentially causes the TCP flow to become extremely erratic, and to almost disappear, see Appendix B.

Henceforth we have only one set of functions at a time.

In order to study behavior of the functions above with constant marking probability p, we study the evolution of W_n as a stochastic process. Using the same ideas as in [50] we get the following results:

Theorem 1. If

$$\alpha < \beta = 1, \quad c_1 > 0, \quad 0 < c_2 < 1, \tag{5.12}$$

then for $p \downarrow 0$ the process $(X(t))_{0 \leq t < \infty}$ defined by

$$X_p(t) = p\big(W_{\lfloor \frac{t}{p} \rfloor}\big)^{1-\alpha} \tag{5.13}$$

approximately behaves as the process $X(t)$ defined by: there is a Poisson Process with intensity 1. In-between the points of the Poisson Process,

$$\frac{d}{dt}X(t) = c_1(1-\alpha), \tag{5.14}$$

and in the points of the Poisson Process (say point τ) we have

$$X(\tau^+) = (1-c_2)^{1-\alpha}X(\tau^-). \tag{5.15}$$

For a discussion of why this result is “obvious”, see [47,50].

The reader will notice that the “theorem” above leaves open the question of exactly in what sense the processes $X_p(\cdot)$ become similar to the process $X(\cdot)$.

Weak convergence of the processes $X_p(.)$ to the process $X(.)$ (weak convergence of all finite dimensional distributions) has not been proven in this paper. It was conjectured by the author of this paper and has recently been proven by Jason Swanson [54]. The construction of the processes in [47,50] indicates the similarity is strong even for p not terribly small.

In Section 9 we will use the fact, not yet proven, that the relaxation times of the processes $X_p(\cdot)$ converge to the relaxation times of the process $X(\cdot)$, or at least have the same order of magnitude.

Finally, in [47] it is proven that if $\beta = 1$, then for all $-\infty < \alpha < 1 = \beta$ the stationary distributions of the processes X_p converge, for $p \downarrow 0$, to the stationary distribution of the process $X(t)$, and the rate of convergence is given. In the original version of [47] this result was proven only for $-\infty < \alpha \leq 0$. The extension to $0 < \alpha < 1$ was added in the modified version of [47]. Common sense and the rate of convergence results also indicate the approximation is quite good even if p is not really small. Of the two ways we want $X_p(\cdot)$ to be similar to $X(\cdot)$ the one that is usually hardest to prove, and the two that are of most importance in this paper thus have been proven.

It is convenient to define

$$c = (1 - c_2)^{(1-\alpha)} \tag{5.16}$$

(so that $0 < c < 1$) and to also introduce the process $Z(t)$ defined by

$$Z(t) = \frac{X(t)}{c_1(1-\alpha)} \tag{5.17}$$

for which of course

$$\frac{d}{dt} Z(t) = 1 \tag{5.18}$$

"in between" the points τ of the Poisson Process, and

$$Z(\tau^+) = cZ(\tau^-) \tag{5.19}$$

"in" the points τ of the Poisson Process. The stationary distribution of the process $Z(t)$ (and thereby of the process $X(t)$) was derived in [50]. It has the form

$$Z = \sum_{k=0}^{\infty} c^k E_k, \tag{5.20}$$

where $(E_K)_{k=0}^{\infty}$ are independent, identically distributed random variables, all exponentially distributed with expected value 1. The distribution of this infinite sum of random variables, including all its moments, was described in detail in [50]. For example, we have for all real μ

$$E[Z^\mu] = \Gamma(\mu + 1) \prod_{k=1}^{\infty} \left(\frac{1 - c^{\mu+k}}{1 - c^k} \right). \tag{5.21}$$

In particular (as is easier seen directly!)

$$E[Z] = \frac{1}{1-c}, \quad \mathit{Var}(Z) = \frac{\mathit{1}}{\mathit{1} - c^2}, \tag{5.22}$$

and therefore

$$\mathit{Coeff}\cdot\mathit{Var}(Z) = \frac{st.dev(Z)}{E[Z]} = \sqrt{\frac{1-c}{1+c}}. \tag{5.23}$$

Thus, we know that the stationary distribution of the congestion window approximately has the form

$$W = p^{-\frac{1}{1-\alpha}} (c_1(1-\alpha)Z)^{\frac{1}{1-\alpha}}. \tag{5.24}$$

Possibly a more interesting way of stating the same result is that $(\frac{p}{c_1})^{\frac{1}{1-\alpha}}W$ has a distribution approximately independent of p and c_1: it approximately has the form

$$\left(\frac{p}{c_1}\right)^{\frac{1}{1-\alpha}} W = ((1-\alpha)Z)^{\frac{1}{1-\alpha}} \tag{5.25}$$

That stationary distribution, including all its moments, therefore is explicitly known. Among other results, we have

$$E[W] \sim p^{-\frac{1}{1-\alpha}}(c_1(1-\alpha))^{\frac{1}{1-\alpha}} E[Z^{\frac{1}{1-\alpha}}], \tag{5.26}$$

$$\mathit{coeff}\cdot\mathit{var}(W) = \frac{st.dev(W)}{E[W]} \sim \mathit{coeff}\cdot\mathit{var}(Z^{\frac{1}{1-\alpha}}), \tag{5.27}$$

approximately independent of p and c_1.

This leads to a clean expression for *Coeff.Var*(W) only if $\alpha = -1$ or $\alpha = 0$. If $\alpha = 0$ we get

$$\mathit{Coeff}\cdot\mathit{Var}(W) \sim \sqrt{\frac{c_2}{2-c_2}}. \tag{5.28}$$

An expression for *Coeff.Var*(W) if $\alpha = -1$ can be found in [50].

It is an interesting excercise to compare the approximation to $E[W]$ in (5.26) with $W(p)$ as obtained from (2.4), (5.5) etc.

A lot is known about the transient behavior of the process $Z(t)$ in (5.17), see [48]. In Section 9 some of this will be used to predict how fast the congestion window W_n reacts when the marking probability p changes (in the case $\beta = 1$).

For the case $\beta = 1$ we have provable limit results as above. For the case $0 \leq \beta < 1$ (and of course still $\alpha < \beta$) we get a different limiting process if $p \downarrow 0$.

With the rescaling as below, all jumps become (relatively) very small. In the limit for $p \downarrow 0$ we get continuous sample paths. Also, for the rescaled process the drift toward the likely center of the stationary distribution becomes linear in the distance from that center, and the "dispersion" becomes almost constant over a wide region: This is strong evidence (almost a mathematical proof!) that in the limit the process below behaves like an Ornstein–Uhlenbeck process. It must be noted that weak convergence has been proved for neither process nor stationary distribution. The evidence is strong enough to demand further research. In fact, in an industrial environment the evidence would be strong enough to base engineering decisions on.

Conjecture. If

$$\alpha < \beta < 1, \quad c_1 > 0, \quad c_2 > 0, \tag{5.29}$$

then for $p \downarrow 0$ the process $(X_p(t))_{0 \le t < \infty}$ defined by

$$X_p(t) = p^{\nu_1} \left(W_{\lfloor \frac{t}{p^{\nu_2}} \rfloor} - \left(\frac{c_1(1-p)}{c_2 p} \right)^{\frac{1}{\beta - \alpha}} \right), \tag{5.30}$$

with

$$\nu_1 = \frac{(1+\beta)}{2(\beta - \alpha)}, \quad \nu_2 = \frac{1-\alpha}{\beta - \alpha}, \tag{5.31}$$

becomes the Ornstein–Uhlenbeck process with local drift

$$E[X(t+\Delta) - X(t)|X(t) = x] = -\Delta.x.(\beta - \alpha) c_1^{-\frac{1-\beta}{\beta-\alpha}} c_2^{\frac{1-\alpha}{\beta-\alpha}} + o(\Delta)(\Delta \downarrow 0), \tag{5.32}$$

and local dispersion

$$Var(X(t+\Delta)|X(t) = x] = \Delta.c_1^{\frac{2\beta}{\beta-\alpha}} c_2^{-\frac{2\alpha}{\beta-\alpha}} + o(\Delta)(\Delta \downarrow 0), \tag{5.33}$$

i.e. the stochastic process $X(t)$ defined by the stochastic differential equation

$$dX(t) = -\mu X(t) + \sigma dB(t) \tag{5.34}$$

with

$$\mu = (\beta - \alpha) c_1^{-\frac{1-\beta}{\beta-\alpha}} c_2^{\frac{1-\alpha}{\beta-\alpha}}, \sigma = c_1^{\frac{\beta}{\beta-\alpha}} c_2^{-\frac{\alpha}{\beta-\alpha}}, \tag{5.35}$$

and where $B(\cdot)$ is standard Brownian Motion. $\mu > 0$ is called the drift parameter and σ^2 is the dispersion.

As in the case of Theorem 1, we should define what we mean by "becomes".

Weak convergence of the processes $X_p(.)$ to the process $X(t)$ has not been proven in this paper. It was conjectured by the author of this paper and has recently been proven by Jason Swanson [54]. In this case there is a major question of the rate of convergence when $p \downarrow 0$: How small must p be before the limiting process becomes a good approximation? Berry-Esseen type results are likely to answer this question.

Convergence of relaxation times has not been proven, and will be used anyhow in Section 9. Intuitively this "must be true", and in fact the only requirement is that relaxation times of the processes X_p remain of the order of magnitude of the relaxation time of the limiting process.

In this case $\beta < 1$ there is no proof yet of convergence of stationary distributions. Intuitively, this "must be true", but a formal proof might be quite hard, and obtaining a rate of convergence result might be even harder.

A nice source of information on the Ornstein–Uhlenbeck process (5.34) is [23]. Among other things, it shows that the stationary distribution of the process $X(t)$ in (5.34) is the normal distribution with mean zero and variance

$$Var(X) = \frac{\sigma^2}{2\mu} = \frac{c_1^{\frac{1+\beta}{\beta-\alpha}} c_2^{-\frac{1+\alpha}{\beta-\alpha}}}{2(\beta-\alpha)}. \tag{5.36}$$

Our conjecture is that if $p \downarrow 0$ the stationary distributions of the processes $X_p(\cdot)$ converge to that same normal distribution.

Results about the autocorrelation functions and first passage times of the Ornstein–Uhlenbeck process can be used to obtain results for how fast window sizes adapt when the marking probability changes. Some of this will be done in Section 9. However, this method may not give the right results if the change in the expected congestion window size is very large compared with the standard deviation as obtained from (5.38) below. In that case the best way to obtain results for the rate of convergence after a change in the marking probability is by direct methods, as at the end of Section 2 (for increasing marking probability), and in Section 9 (for decreasing marking probability).

When one of the mathematical models is used to investigate rates of convergence (to a new stationary distribution), it must not be forgotten that the models used are based on "packet time": the result would be information on the number of packets transmitted before the new equilibrium is reached.

In order to translate results into clock time (number of RTTs) we have to divide the number of packets by the congestion window W.

Strong supporting evidence for the Ornstein–Uhlenbeck conjecture will be given in Appendix A.

The stationary distribution of $X(\cdot)$ immediately translates into a stationary distribution for W_n: As long as we use the model (5.34), i.e. as long as we disregard the delay in the feedback, we get: For $p \downarrow 0$, the stationary distribution of W_n has

$$E[W] \sim \left(\frac{c_1(1-p)}{c_2 p}\right)^{\frac{1}{\beta-\alpha}} \sim \left(\frac{c_1}{c_2 p}\right)^{\frac{1}{\beta-\alpha}}, \tag{5.37}$$

$$\text{st.dev}(W) \sim \frac{c_1^{\frac{1+\beta}{2(\beta-\alpha)}} c_2^{-\frac{1+\alpha}{2(\beta-\alpha)}}}{\sqrt{2(\beta-\alpha)}} p^{-\frac{1+\beta}{2(\beta-\alpha)}}, \tag{5.38}$$

$$coeff.var(W) = \frac{st.dev(W)}{E[W]} \sim \frac{c_1^{-\frac{1-\beta}{2(\beta-\alpha)}} c_2^{\frac{1-\alpha}{2(\beta-\alpha)}}}{\sqrt{2(\beta-\alpha)}} p^{\frac{1-\beta}{2(\beta-\alpha)}}. \tag{5.39}$$

(5.27) and (5.39) show that there is a certain charm to choosing $\beta = 1$: with that choice, and that choice only, the coefficient of variation of W becomes independent of p for $p \downarrow 0$ (i.e. when the congestion window is allowed to be very large). In fact, for that choice the distribution of $(\frac{p}{c_1})^{\frac{1}{1-\alpha}}\ W$ becomes independent of p and c_1 for p small. The non-dependence on p may seem no big deal, but non-dependence on p implies non dependence on the average value of W: scale invariance!

This doubtlessly is one of the reasons "Scalable TCP" chooses $\beta = 1$.

Any smaller value of β makes the (stationary) window size almost deterministic, equal to the expected value, when $p \downarrow 0$ (p small, but constant, and the average value of W becomes large). This is a nice property when p actually is constant, but is worrisome when p varies and we need quick convergence to a new equilibrium. However, we saw that as long as it is possible to mark multiple packets per RTT it is always possible to decrease windows geometrically fast. We also know that as long as $\alpha = 0$ (larger might be dangerous! see also Section 8) the congestion windows grow exponentially fast if p becomes zero or at least decreases by an order of magnitude.

6. The number of marked packets per round trip time

This section, and most further sections, require that W is expressed in MSSs and that practically all packets contain practically MSS data bytes, so that every RTT practically has W data packets.

This section studies the average number of marked packets per RTT strictly using the models $X(t)$ obtained from (5.13) etc ($\beta = 1$) or from (5.30) etc ($\beta < 1$).

If a flow has a marking probability of p per packet and a congestion window of W packets, it will on average have pW marked packets per Round Trip Time. In the remainder of this section we will use $E[W]$ or $W(p)$ instead of W. This number of marked packets per RTT is important, among other reasons because a large number of both marked packets and unmarked packets per RTT makes it possible to "gently" control the flows: It is possible, within one Round Trip Time, to signal a large change in the marking probability p, as well as to signal a fairly subtle change. In the situation of Section 5, when a flow has been in existence long enough, and p has been constant long enough to have reached stationarity, we have the following results:

Theorem 2. In the situation of Theorem 1 ($\beta = 1$), when a flow is in existence for a long time and the marking probability p is constant and close to zero, if the congestion window is the only effective limit on the number of outstanding packets, the flow has in average about

$$p^{-\frac{\alpha}{1-\alpha}}(c_1(1-\alpha))^{\frac{1}{1-\alpha}}E\big[Z^{\frac{1}{1-\alpha}}\big] \tag{6.1}$$

marked packets per Round Trip Time. The distribution of Z depends only on c_2 and on α, not on c_1 and p.

Using (5.6) instead of (5.26) we would have gotten the approximation of about

$$p^{\frac{\beta-\alpha-1}{\beta-\alpha}}\left(\frac{c_1}{c_2}\right)^{\frac{1}{\beta-\alpha}} = p^{-\frac{\alpha}{1-\alpha}}\left(\frac{c_1}{c_2}\right)^{\frac{1}{1-\alpha}} \tag{6.2}$$

marked packets per Round Trip Time.

Theorem 3. in the situation of the conjecture ($\beta < 1$), when a flow is in existence for a long time and the marking probability p is constant and close to zero, it has on average about

$$p^{\frac{\beta-\alpha-1}{\beta-\alpha}} \left(\frac{c_1}{c_2}\right)^{\frac{1}{\beta-\alpha}} \tag{6.3}$$

marked packets per Round Trip Time. In this case it makes no difference whether we use (5.6) or (5.37).

We see that in both cases there is a factor $p^{\frac{\beta-\alpha-1}{\beta-\alpha}}$. We see that it is highly desirable that

$$\beta - \alpha \leq 1. \tag{6.4}$$

Namely, in that case, and that case only, the number of marked packets per Round Trip Time will not go to zero when $p \downarrow 0$, at least as long as the flow is allowed very large windows (as large as the "response surfaces" permit). That way, the router can signal relatively subtle changes in desired rates. When the number of marked packets per Round Trip Time falls (significantly) below 1, it becomes hard or impossible for routers to signal a desired minor change in congestion window. Classical TCP is an extreme case, with only in the order of $\sqrt{p}$, i.e. much fewer than 1, "marked" packets per Round Trip Time (if the reader prefers it, we can restate this as about one "marked" packet per $p^{-\frac{1}{2}}$, i.e. many, RTTs).

(6.3) shows that if $\beta - \alpha = 1$ it is preferable to choose $\frac{c_1}{c_2}$ large.

7. The delay in the feedback

It has been observed before that the model (5.3) used in this paper is not suitable for investigating the effect of the delay of one Round Trip Time in the feedback. This is an important weakness, because it is well-known that delay in feedback can lead to oscillatory behavior and even non-stationarity. An example is given in [49]. The reason the model used here can not be used to investigate the effect of the delay in the feedback is that an arriving acknowledgement, whether it acknowledges a marked or unmarked packet, uses the current window size W and not the window size of one RTT ago. Thus, as long as the window size does not affect the marking probability, if one RTT ago the window was large we now get a more intense stream of acknowledgements, but with unaffected marking probabilities: The drift toward equilibrium is stronger (if we think in clock time), but the equilibrium distribution has not changed.

Similarly, if one RTT ago the window was small, the drift toward equilibrium is weaker, but the equilibrium distribution has not changed.

This changes of course when the window size affects the marking probability. In that case, it is natural to assume that higher window size tends to cause higher marking probability. In that case, a larger window size one RTT ago will tend to cause a stronger downward drift, even if the current window already is small (etc.) This situation obviously can cause oscillatory behavior.

Once the dependence of the marking probability p on the window size W has been quantified, it may be possible to use results in [49] to study the deviation from the results in this paper.

8. Smoothness under stationarity

The conjecture in Section 5 predicts that if $\beta < 1$ and p is small, once stationarity has been reached, the congestion window will remain most of the time within in the order of a few standard deviations (5.38) of the expected value (5.37). We saw that if p is small this standard deviation is small compared to the expected value, thus relatively W is not expected to vary significantly.

The conjecture also predicts that in the same situation, over an interval that contains a number of packets that is small compared with $p^{-\nu_2} = p^{-\frac{1-\alpha}{\beta-\alpha}}$ the variation (largest minus smallest) will be small compared with the standard deviation, and over an interval of in the order of $p^{-\nu 2}$ packets the variation will be at most of the order of a standard deviation. More on this topic can be found in Section 9.

In this section we investigate variability of the congestion window, in the situation of the conjecture, when p is small and stationarity has set in, over intervals of in the order of one RTT. A problem is that the number of packets is not quite known. Thus, we will investigate variability over an interval that contains in the order of

$$p^{-\frac{1}{\beta-\alpha}} \tag{8.1}$$

packets. It must be noted that if $\alpha < 0$ then $p^{-\nu_2}$ is large compared with $p^{-\frac{1}{\beta-\alpha}}$ and the additional work is not necessary (except may be for readers who do not trust the conjecture). If $\alpha = 0$ the two entities are of the same order of magnitude.

The results in this section also are of interest in the situation $\beta = 1$.

Loosely, we will call a sample path of the congestion window "very smooth" if under stationarity the increase of the congestion window over an interval that contains in the order of $p^{-\frac{1}{\beta-\alpha}}$ packets has a standard deviation that is small compared with the expected value of the congestion window.

We will do a thought experiment: Suppose the congestion window W has been "practically constant" over a period of say 2 RTTs. "Practically constant" means that the quotient of minimal value over maximal value is close to one. In that "practically constant" situation we compute the standard deviation of the increase of the window over the second Round Trip Time. If that standard deviation is small compared with the congestion window, the original assumption (of an almost constant W) is at least self-consistent.

Suppose the congestion window remains close to W. Suppose during the second RTT exactly W acknowledgements arrive, N of which are "failures" and $W - N$ of which are "successes". N has the Binomial (W, p) distribution, its expected value is pW and its variance is $p(1-p)W$. The total increase Incr over the second RTT thus is

$$\text{Incr} = (W-N)c_1W^{\alpha} - Nc_2W^{\beta} = c_1W^{1+\alpha} - N(c_1W^{\alpha} + c_2W^{\beta}). \tag{8.2}$$

Hence (given W)

$$E[\text{Incr}] = c_1W^{1+\alpha} - pW(c_1W^{\alpha} + c_2W^{\beta}), \tag{8.3}$$

$$\mathit{Variance}(\text{Incr}) = p(1-p)W\left(c_1W^{\alpha} + c_2W^{\beta}\right)^2, \tag{8.4}$$

and

$$\mathit{Std.Dev}(\text{Incr}) = \sqrt{(p(1-p)W)}(c_1W^{\alpha} + c_2W^{\beta}). \tag{8.5}$$

Thus,

$$\frac{\mathit{Std.Dev}(\text{Incr})}{W} = c_2\sqrt{(p(1-p))}W^{\beta-\frac{1}{2}}\left(1 + \frac{c_1}{c_2}W^{-(\beta-\alpha)}\right). \tag{8.6}$$

Now we choose for W the predicted value given p, from (5.5). This gives:

$$\text{Quotient} = p^{\frac{1}{2}\frac{1-(\alpha+\beta)}{\beta-\alpha}}(1-p)^{\frac{1}{2}\frac{(\alpha+\beta)-1}{\beta-\alpha}}c_1^{\frac{\beta-1/2}{\beta-\alpha}}c_2^{\frac{1/2-\alpha}{\beta-\alpha}}. \tag{8.7}$$

We see that a condition for sample paths to be "very smooth" if p is small and stationarity has been reached is that

$$\alpha + \beta < 1. \tag{8.8}$$

Clearly, with a more reasonable definition, in terms of the sample paths, instead of in terms of standard deviations, this result is inadequate: In case $\beta = 1, \alpha < 0$, the condition (8.8) is satisfied, but clearly in every RTT there is a high probability of an almost constant window, but also a small probability of a significant (downward) jump.

Of course, it also is likely that as long as $\beta < 1$ then, under stationarity, if p is small, even when $\alpha + \beta > 1$ the actual sample paths of W will vary, during one RTT, over a range small compared with W: If the deviation from the expected value becomes large compared with the standard deviation (even though still small compared with the expected value) the drift toward the expected value becomes quite strong. Hence, the requirement (8.8) is a "soft" requirement, and in particular is unnecessary if $\alpha < 0$ (in which case it is automatically satisfied anyhow).

It must be noted that if $\beta < 1$ and $\alpha < 0$ or even $\alpha = 0$ the condition (8.8) is superfluous: in the first place (as just observed) it is automatically satisfied, in the second place, in that situation, a RTT contains a number of packets small compared with, or in the worst case of the order of, $p^{-\nu_2}$ packets.

It is interesting to note that requiring that the expected increase in (8.2) is zero leads back to (5.5).

It is interesting to not only look at the distribution of the increase of W over W packets (or acknowledgements), but also at the expected total variation, i.e. the sum of the absolute values of the jump sizes.

This expected value is

$$(1-p)c_1 W^{\alpha+1} + pc_2 W^{\beta+1}. \tag{8.9}$$

Dividing (8.9) by W and setting W as in (5.5) gives

$$2c_1^{\frac{\beta}{\beta-\alpha}} c_2^{-\frac{\alpha}{\beta-\alpha}} (1-p)^{\frac{\beta}{\beta-\alpha}} p^{-\frac{\alpha}{\beta-\alpha}} \tag{8.10}$$

To make sure that the expected total variation over W packets is small compared with W when p is small and (5.5) holds we need that $\alpha < 0$. If $\alpha > 0$ the expected total variation over W packets during stationarity (and W assumed very smooth) becomes of the order of W. If $\alpha > 0$ the expected total variation becomes large compared with W. While not necessarily bad, that becomes a somewhat worrisome situation.

Thus, this section indicates that the condition

$$\alpha + \beta < 1 \tag{8.11}$$

is somewhat desirable, to keep the standard deviation of the increase of W over W packets small compared with W, and it indicates that

$$\alpha \leq 0 \tag{8.12}$$

is somewhat desirable, because it keeps the expected total variation of W over W packets in the order of W, or smaller.

If $\alpha = 0$ we see that smaller values of c_1 may be preferred.

9. Transient behavior, and relaxation times

In [48] an in principle complete characterization is given of the transient behavior of the process $Z(t)$ in (5.17). Some results will be quoted below.

These results can in principle be used to investigate issues like "how fast does the flow behavior change when the marking probability changes", or "what must a router do to quickly achieve a desired change in the behavior of certain flows".

This section makes a start with such work. An important fringe benefit is that this section illustrates the difference between "packet time" and "clock time".

The discussion is in terms of the relaxation times of the processes involved.

Among the results in [48] is that for any (possibly random, but non-negative) initial value $Z(0)$ we have

$$E[Z(t)] = \frac{1}{1-c} + \left(E[Z(0)] - \frac{1}{1-c}\right) e^{-t(1-c)}, \tag{9.1}$$

and

$$\begin{aligned} E[(Z(t))^2] &= \frac{2}{(1-c)(1-c^2)} \\ &\quad + \left(E[(Z(0))^2] - \frac{2}{(1-c)(1-c^2)}\right) e^{-t(1-c^2)} \\ &\quad + 2\left(E[Z(0)] - \frac{1}{1-c}\right)\left(\frac{1-e^{-ct(1-c)}}{c(1-c)}\right) e^{-t(1-c)}. \end{aligned} \tag{9.2}$$

These specific results are actually easily proven directly and the proof is left to the reader.

The result is that the process $Z(t)$, if it starts out not very far from the new equilibrium value, "loses its memory" in a small multiple of $\frac{1}{1-c}$ units of time.

We say that the process $Z(\cdot)$ has "relaxation time"

$$\frac{1}{1-c}. \tag{9.3}$$

Translating this back into results for the process W_n we see that it loses its memory in a small multiple of $\frac{1}{p(1-c)}$ packets. In other words, if $\beta = 1$ the process W_n approximately has relaxation time

$$\frac{1}{p(1-c)}\text{packets}. \tag{9.4}$$

We would like to express this in RTTs. A problem is that the number of packets per RTT (i.e. W) is random. The best we can do is: Using results from Section 5 we see that if $\beta = 1$, the process W_n has approximately

$$\frac{p^{\frac{\alpha}{1-\alpha}}}{(1-c)(c_1(1-\alpha))^{\frac{1}{1-\alpha}} E[Z^{\frac{1}{1-\alpha}}]} \tag{9.5}$$

units of $E[W]$ packets in its relaxation time. If we had used (5.6) instead of (5.26) we would have gotten that the process W_n approximately has

$$\frac{p^{\frac{\alpha}{1-\alpha}}}{1-c}\left(\frac{c_2}{c_1}\right)^{\frac{1}{1-\alpha}} \tag{9.6}$$

units of $W(p)$ packets in its relaxation time. In the case $\beta = 1$ the dominating factor for p small is $p^{\frac{\alpha}{1-\alpha}}$ and (roughly) the relaxation time is in the order of $p^{\frac{\alpha}{1-\alpha}}$ RTTs.

Next we study the relaxation times in case $\beta < 1$.

In [23] we see that if $X(t)$ is the Ornstein–Uhlenbeck process with drift parameter $\mu > 0$ and local dispersion σ^2 as in (5.34) etc, then ($s > 0, t > 0$)

$$E[X(t) \mid X(0) = x_o] = x_o e^{-\mu t}, \tag{9.7}$$

$$Cov(X(s), X(t) \mid X(0) = x_o) = \frac{\sigma^2}{2\mu}\left(e^{-\mu|t-s|} - e^{-\mu(t+s)}\right). \tag{9.8}$$

Thus, we see that this Ornstein–Uhlenbeck process looses its memory in a small multiple of $\frac{1}{\mu}$ units of time: the relaxation time is $\frac{1}{\mu}$, or

$$\frac{c_1^{\frac{1-\beta}{\beta-\alpha}} c_2^{-\frac{1-\alpha}{\beta-\alpha}}}{\beta - \alpha}. \tag{9.9}$$

Thus, if $\beta < 1$, the process W_n approximately has relaxation time

$$\frac{p^{-\frac{1-\alpha}{\beta-\alpha}} c_1^{\frac{1-\beta}{\beta-\alpha}} c_2^{-\frac{1-\alpha}{\beta-\alpha}}}{\beta - \alpha} \text{packets.} \tag{9.10}$$

Just as in the case $\beta = 1$, we see that if $\beta < 1$ and p small, the relaxation time of the process W_n contains about

$$\frac{p^{\frac{\alpha}{\beta-\alpha}} c_1^{-\frac{\beta}{\beta-\alpha}} c_2^{\frac{\alpha}{\beta-\alpha}}}{\beta - \alpha}. \tag{9.11}$$

units of $W(p)$ packets. In this case the number of packets per RTT is “almost constant” (has very low coefficient of variation), and the derivation is internally consistent.

The analysis above indicates that if we only consider the model W_n without delay in the control, then for $\beta = 1$ as well as $\beta < 1$ we would need to set $\alpha > 0$ in order to get a control scheme that reacts in less than an RTT to small variations in the marking probability p, and we would need to set $\alpha = 0$ to get a control scheme that reacts in just a few RTTs to a small change in p. If we set $\alpha < 0$ it takes many (or very many) RTTs to react to a small change in p. An example is the slow growth of the congestion window under TCP with no marked packets.

For changes in p that result in changes in $W(p)$ large compared with the new standard deviation, the results above probably are not usefull. As long as $\beta \geq 0$ increasing p to close to 1 leads to multiplicative (or faster) decrease (per RTT). As long as $\alpha \geq 0$ a drastic decrease in p (say to zero) leads to exponential growth in W.

Of course, in the situation with delay of 1 RTT in the feedback, a relaxation time small compared with RTT can be dangerous and can lead to oscillatory behavior if routers “unintelligently” adapt the marking probability to perceived congestion.

10. Achievable window sizes

TCP has $\beta - \alpha = 2$, thus roughly $w(p) = \sqrt{\frac{1}{p}}$. This has as effect that it is very hard indeed to get large congestion windows: to get $W = 1000$ p must be in the order of 10^{-6}. To make it possible to get large congestion windows, we need $\beta - \alpha$ much smaller, for example $\beta - \alpha \leq 1$.

In addition, choosing α small ($-\alpha$ large) has the consequence that it takes an enormous number of unmarked acknowledgements before a considerable increase in the window size is obtained.

Combining this with the results in Section 6 gives us a powerful incentive to choose $\beta - \alpha \leq 1$, preferably $\beta - \alpha < 1$, and to choose α much larger than under TCP: In order to have sufficiently many marked packets per RTT to enable the router to give fairly subtle signals, in order to enable the source to achieve large window, and to make it possible to reach that large window in a reasonable number of unmarked acknowledgements.

11. Desirable values for α and β

At this point we have sufficient information to discuss choices for α and β. We saw that the response function behaves like

$$w(p) = p^{-\frac{1}{\beta-\alpha}} \left(\frac{c_1}{c_2}\right)^{\frac{1}{\beta-\alpha}}, \tag{11.1}$$

that in equilibrium there are in the order of

$$pW(p) = p^{\frac{\beta-\alpha-1}{\beta-\alpha}} \left(\frac{c_1}{c_2}\right)^{\frac{1}{\beta-\alpha}} \tag{11.2}$$

marked packets per RTT, that the "relaxation time" in equilibrium is in the order of

$$p^{\frac{\alpha}{\beta-\alpha}} \frac{c_1^{-\frac{\beta}{\beta-\alpha}} c_2^{\frac{\beta}{\beta-\alpha}}}{\beta - \alpha} \text{RTTs}, \tag{11.3}$$

and that in order to allow "multiplicative increase" and relaxation times short compared with a RTT we need $\alpha \geq 0$. In addition, Section 8 shows that in order to keep the expected value of the total variation of W over a RTT in the order of W or smaller it is necessary to have

$$\alpha \leq 0, \tag{11.4}$$

though it is not clear how important this constraint on the total variation really is. Finally. Section 8 indicates that if we decide to try $\alpha > 0$ then it may be desirable to have

$$\alpha + \beta \leq 1 \tag{11.5}$$

or at least $\alpha + \beta$ not much larger than 1, in order to have some additional confidence that under p constant and small, under stationarity, the sample paths remain fairly smooth.

Not surprisingly, we have a number of contradictory constraints or desires.

The author of this paper confesses to an intuitive aversion against $\alpha > 0$, because of what may happen due to the delay in the feedback (not investigated in this paper). Giving in to this aversion, a recommendation is to use either "$\alpha = 0$" or "$\alpha < 0$ but quite close to 0". and of course in any case $\beta - \alpha \leq 1$, preferably $\beta - \alpha < 1$.

Tom Kelly's work indicates that the choice $\beta = 1, \alpha = 0$ has good performance, probably better than $\beta = 1$, any $\alpha < 0$.

Thus, the most sensible choices are

1. $\alpha = 0, \beta = 1$ (Scalable TCP).
2. $\alpha < 0 < \beta < 1$ with $\beta - \alpha < 1$. In this case we probably must have α close to zero to get a "smallish" relaxation time even when p is small. The constraint $\beta - \alpha < 1$ ensures a large number of marked packets per RTT when p becomes small.
3. $\alpha < 0 < \beta < 1$ with $\beta - \alpha = 1$. This is the limiting situation of the case above. In this case the number of marked packets per RTT remains bounded away from both zero and infinity when $p \downarrow 0$ (assuming large congestion windows are possible). In this case we probably prefer to choose $\frac{c_1}{c_2}$ "large".
4. $\alpha = 0$ and $0 < \beta < 1$ and c_1, c_2 "suitable". This combination, if it works, has all the best characteristics: Large number of marked packets per RTT if p is small, and a relaxation time in the order of a few RTTs (independent of $p(!?)$).

In addition to the choices above, there is the fascinating but possibly dangerous choice $0 < \alpha < \beta < 1$ with (preferably) $\alpha + \beta < 1$. In the objectives used in this paper it does great: large $W(p)$ for moderate p, many marked packets per RTT, relaxation time small compared with the RTT, faster than exponential increase when p is zero, and a reasonable guarantee of smooth sample paths if p is smallish.

It has the disadvantage of large total variation in the congestion window even when the congestion window is almost constant, and the consequences of delay in the feedback are thus far unpredictable.

The analyses that led to these conclusions, while sound from a common sense point of view, clearly have mathematical holes. Thus, the various possibilities must be investigated by further mathematical analysis, simulation if necessary, and absolutely by implementation in the laboratory and measurements in the laboratory.

Due to the research funding situation in the USA, the author of this paper has given up (hopefully temporarily) plans to implement in Linux schemes as above (for endstations as well as routers).

12. First estimating *p*

With more powerful endstations it is possible to have a more sophisticated algorithm. The main proposal in this section is to first choose a response function $q(\cdot)$ and then explicitly estimate p. The desired W is computed from the estimated p using (2.4). p could be estimated using exponential smoothing, but there may be problems doing this: Let

$$Zap(k) = \begin{cases} 0 & \text{if } \chi_k = \text{success} \\ 1 & \text{if } \chi_k = \text{failure} \end{cases} \tag{12.1}$$

(where χ_k is as in (5.2)), and let

$$\bar{p}_k = (1 - r)\bar{p}_{k-1} + rZap(k). \tag{12.2}$$

be the estimate for p. (12.2) has the disadvantage that when the estimate $\bar{p}$ is small compared with the smoothing parameter r, a single "zapped" (i.e. marked) packet increases $\bar{p}$ far too much. It is desirable to let r depend on $\bar{p}$, for example a well chosen positive constant times $\bar{p}$. However, this may lead to problems when $\bar{p}$ becomes extremely small. A comprehensive solution seems to be:

Choose a minimal value for $\bar{p}$. For example, choose a maximal acceptable value $W_{\max}$ for the congestion window W (say the receive window). From the chosen response surface $q(\cdot)$, compute p^* such, that $W_{\max} = W(p^*)$. Now choose $p_{\min}$ "appropriately small" (to be defined) compared with p^*. Choose a positive constant $c_3, 0 < c_3 < 1$, for example $c_3 = \frac{1}{8}$ or $\frac{1}{16}$. Now, instead of (12.2) use

$$\bar{p}_k = \max\left((1 - c_3\bar{p}_{k-1})\bar{p}_{k-1} + c_3\bar{p}_{k-1}Zap(k), p_{\min}\right). \tag{12.3}$$

This way, when $\bar{p}$ is small, it takes in the order of $(\log(1 + c_3))^{-1}$(log base 2) marked packets in relatively quick succession (much faster than probability $\bar{p}$ per packet) to double the value of $\bar{p}$.

Every time $\bar{p}$ has been recomputed, recompute W from

$$W = \min(W(\bar{p}), W_{\max}). \tag{12.4}$$

Thus, as long as $p_{\min} \leq \bar{p} \leq \bar{p}^*$, W remains at $W_{\max}$. When $\bar{p}$ increases above p^*, W decreases below $W_{\max}$. As long as $\bar{p}$ remains below p^*, randomly marked packets do not affect W. It seems to make sense to choose $p_{\min} = \frac{p^*}{2}$. In that situation about $(\log(1 + c_3))^{-1}$ marked packets in quick succession always start decreasing W.

13. Estimating *p* with an example response function

In this section we choose, as example, the response function

$$q(W) = \frac{W}{c_4}, \tag{13.1}$$

$$W(p) = \frac{c_4}{p}. \tag{13.2}$$

based on $\beta - \alpha = 1, c_4 = \frac{c_1}{c_2}$. α, β, c_1 and c_2 no longer have meaning by themselves. Window evolution is done as in Section 12, with parameter c_3.

Next we analyze the evolution of W in the domain where $W < W_{\max}$, $p^* < \bar{p}$. In other words, we always have

$$W_k = \frac{c_4}{\bar{p}_k}. \tag{13.3}$$

Since we have

$$\bar{p}_k = \begin{cases} (1 - c_3\bar{p}_{k-1})\bar{p}_{k-1} & \text{if} \quad Zap(k) = 0, \\ (1 + c_3 - c_3\bar{p}_{k-1})\bar{p}_{k-1} & \text{if} \quad Zap(k) = 1, \end{cases} \tag{13.4}$$

we also have

$$W_k - W_{k-1} = \begin{cases} \dfrac{c_3c_4}{1 - c_3\bar{p}_{k-1}} & \text{if} \quad Zap(k) = 0, \\ -\dfrac{c_3(1 - \bar{p}_{k-1})}{1 + c_3(1 - \bar{p}_{k-1})}W_{k-1} & \text{if} \quad Zap(k) = 1, \end{cases} \tag{13.5}$$

Thus, we see that for $p^* < \bar{p} << 1$ the evolution of W is as in Section 5 with $\alpha = 0, \beta = 1, c_1 = c_3c_4$, and $c_2 = \frac{c_3}{c_3+1}$.

Since $\beta - \alpha = 1$, $\frac{c_1}{c_2} = \frac{c_4}{c_3+1}$ is the desired number of marked packets per Round Trip Time (once stationarity has been reached). For every marked packet, the congestion window is decreased from W to $\frac{W}{c_3+1}$. If $c_3 = 1$ every marked packet halves the window. A less draconic choice is $c_2 = \frac{1}{8}$ or even $\frac{1}{16}$. There now are two trains of thought that can be used to set c_2 or c_3: the one based on how fast the estimate for p is changing when there are marked packets, and the one based directly on how fast the congestion window must change when packets are marked.

A similar analysis can be done with response functions other than (13.1) etc.

14. Router behavior

This note does not study router behavior. It is however possible to make some relevant observations that may be the start of a later serious study.

A router can estimate, for all its buffers, the number of active flows of class i that are using that buffer. This can be done, for example, by the methods described in [58].

Let N_i be the estimated number of active flows of class i. If the router also knows that all class i flows are ECN–capable, and that all sources of class i flows are using the "c_1, c_2, α, β" policy (with c_1 etc of course depending on i), it can for example set the marking probability $p^{(i)}$ for class i packets in that buffer in the order of

$$p^{(i)} \sim c_5.N_i^{\beta_i - \alpha_i}. \tag{14.1}$$

In (14.1) the constant c_5 can depend on the buffer occupation etc. We again see that the case $\beta_i - \alpha_i = 1$ has a certain charm: the dependence of the probability p on the estimated number N of flows is smoother than for TCP. A small error in the estimate N has less serious consequences. $\beta_i - \alpha_i < 1$ might be even better.

The router can always drastically reduce congestion windows by setting $p = 1$ for a significant fraction of a Round Trip Time. Since the router is marking, no packet loss ensues. It is desirable to do this only if the router can predict the effect of markings: If it does this "until the effect is noticeable", most congestion windows have been reduced to one MSS or less. This is one of the places where the delay of 1 RTT in the control is important and must be included in future work.

15. Conclusions

In this paper we study mechanisms in the Internet where Routers give feedback about their state of congestion to endstations (say sources) by dropping or marking (ECN, Explicit Congestion Notification) packets. We argue that Router Behavior (e.g. whether and when to mark packets) and Source Behavior (e.g. how to modify congestion windows in reaction to marked and unmarked packets) must be designed together. We argue that the advent of ECN is an opportunity, quite possibly the last opportunity, to modify the TCP feedback system (in the short term: give different interpretations to "drop" and "mark", and make the interpretation of "mark" dependent on the type of IP packet).

We discuss the general TCP Paradigm, where there are general $incr(\cdot)$ and $decr(\cdot)$ functions. We then restrict our attention to a smaller class of such schemes, where $incr(w) = c_1 w^\alpha$ and $decr(w) = c_2 w^\beta$. For these functions we predict performance, including the stationary behavior of congestion window sizes, as function of the marking probability p.

We observe that congestion control schemes in the TCP Paradigm should attain or approach the following ideals:

- *Fast Response*: when the marking probability p changes, the congestion window must quickly converge to the new equilibrium. In other words, we want "small relaxation times".
- *Smoothness*: when the marking probability p is constant, congestion windows must fluctuate in a narrow band around the desired equilibrium value (insofar advertised windows etc allow). In other words, we want the standard deviation of the window size W to be small if p is constant.

- *Sensitivity*: it must be possible for routers to signal small changes in the marking probability p. This means that as long as the congestion window is not small, neither the number of marked packets per congestion window nor the number of unmarked packets per congestion window must be small.
- *Non-Oscillatory Behavior*: We do not want the delay in the feedback to cause oscillatory behavior when p is constant. While oscillatory behavior has not been analyzed in this paper, intuitively it seems that while we like small relaxation times for the reason above, having relaxation times small compared with the RTT may be dangerous and needs study.

The analysis results in the following observations:

- Based on the number of marked packets per Round Trip Time we recommend $\beta - \alpha \leq 1$, preferably $\beta - \alpha < 1$.
- Based on the desire to enable exponential growth under low (zero) marking probability we recommend $\alpha \geq 0$. In addition, $\alpha = 0$ leads to relaxation times roughly independent of p, while $\alpha > 0$ leads, for p small, to relaxation times short compared with the RTT.
- Based on the desire to have a strong guarantee that for p small, when stationarity sets in, the sample paths of the congestion window becomes smooth, we recommend $\alpha \leq 0$. Possibly $\alpha > 0$ with $\alpha + \beta < 1$ might be manageable.
- Combining the last two items there is a strong incentive to choose $\alpha = 0$. Combined with the first item this means that $\beta < 1$ becomes very attractive.

As result, "Scalable TCP" with $\alpha = 0, \beta = 1$ comes out as a real contender. However, other combinations with $\beta - \alpha \leq 1$ and $\alpha \leq 0 < \beta < 1$ (in particular $\alpha = 0$) deserve more research and might end up being the preferred choices, see in particular Section 11.

We give an alternative way of thinking about source behaviors, where sources estimate marking (or drop) probabilities and react to these estimates, instead of to individual marked or dropped packets. We show that this alternative way of thinking can lead to an identical implementation (inside the TCP Paradigm). This may lead to a more scientific way of choosing source behaviors.

Appendices

A. The Ornstein-Uhlenbeck Approximation

In the situation of the conjecture,

$$W_{n+1} = \begin{cases} W_n + c_1 W_n^{\alpha} & \text{with probability } p, \\ W_n + c_2 W_n^{\beta} & \text{with probability } 1 - p, \end{cases} \tag{A.1}$$

with $\alpha < \beta < 1, c_1 > 0, c_2 > 0$. For the process

$$X(t) = p^{\nu_1} \left(W_{\lfloor \frac{t}{p^{\nu_2}} \rfloor} - \left(\frac{c_1(1-p)}{c_2 p} \right)^{\frac{1}{\beta-\alpha}} \right), \tag{A.2}$$

we therefore have:

$$\begin{aligned} &\frac{1}{p^{\nu_2}} E[X(t+p^{\nu_2}) - X(t) | X(t) = x] = p^{\nu_1-\nu_2} \Big(W_{\lfloor \frac{t}{p^{\nu_2}} + 1 \rfloor} - W_{\lfloor \frac{t}{p^{\nu_2}} \rfloor} \Big) \\ &= p^{\nu_1-\nu_2} \left((1-p)c_1 W^{\alpha}_{\lfloor \frac{t}{p_2^{\nu}} \rfloor} - pc_2 W^{\beta}_{\lfloor \frac{t}{p_2^{\nu}} \rfloor} \right) \\ &= p^{\nu_1-\nu_2} \left((1-p)c_1 \left(\left(\frac{c_1(1-p)}{c_2 p} \right)^{\frac{1}{\beta-\alpha}} + p^{-\nu_1} x \right)^{\alpha} - pc_2 \left(\left(\frac{c_1(1-p)}{c_2 p} \right)^{\frac{1}{\beta-\alpha}} + p^{-\nu_1} x \right) \right)^{\beta}. \end{aligned}$$

We initially guess that as long as $|x|$ is not very large compared with the "guessed" standard deviation of the process $X(t)$,

$$|p^{-\nu_1} x| << \left(\frac{c_1(1-p)}{c_2 p} \right)^{\frac{1}{\beta-\alpha}}, \tag{A.3}$$

and do binomial expansions of the inner expressions. Then later, when the standard deviation is computed or confirmed, we check that guess. (A.3) gives

$$\begin{aligned} &\frac{1}{p^{\nu_2}} E[X(t+p^{\nu_2}) - X(t) | X(t) = x] \\ &\sim p^{\nu_1-\nu_2} \left\{ (1-p)c_1 \left(\frac{c_1(1-p)}{c_2 p} \right)^{\frac{\alpha}{\beta-\alpha}} + (1-p)c_1\alpha \left(\frac{c_1(1-p)}{c_2 p} \right)^{\frac{\alpha-1}{\beta-\alpha}} .p^{-\nu_1} x \right. \\ &\left. - pc_2 \left(\frac{c_1(1-p)}{c_2 p} \right)^{\frac{\beta}{\beta-\alpha}} - pc_2\beta \left(\frac{c_1(1-p)}{c_2 p} \right)^{\frac{\beta-1}{\beta-\alpha}} .p^{-\nu_1} x \right\} \end{aligned}$$

The highest–order terms drop out and we get

$$\begin{aligned} &\frac{1}{p^{\nu_2}} E[X(t+p^{\nu_2}) - X(t) | X(t) = x] \\ &\sim -x(\beta-\alpha) c_1^{-\frac{1-\beta}{\beta-\alpha}} c_2^{\frac{1-\alpha}{\beta-\alpha}} (1-p)^{-\frac{1-\beta}{\beta-\alpha}} p^{\frac{1-\alpha}{\beta-\alpha} - \nu_2} \end{aligned}$$

We see that to get a "useful" (Ornstein–Uhlenbeck type) result for $p \downarrow 0$ we need

$$\nu_2 = \frac{1-\alpha}{\beta-\alpha}. \tag{A.4}$$

Repeating the process for second moments we see that the Ornstein–Uhlenbeck result holds as long as in addition_to (A.4) also

$$\nu_1 = \frac{1+\beta}{2(\beta-\alpha)}. \tag{A.5}$$

With (A.4) this yields the condition (5.31).

Since $\alpha < \beta$, for p small downward jumps in the process $X(\cdot)$ are (much) larger than upward jumps. By first approximation, the quotient of downward jump sizes and standard deviation of $X(\cdot)$ is

$$\left(\frac{\text{Jump}}{\text{St.Dev}}\right) = \sqrt{2(\beta-\alpha)}c_1^{-\frac{1-\beta}{2(\beta-\alpha)}} c_2^{\frac{1-\alpha}{2(\beta-\alpha)}} (1-p)^{\frac{\beta}{\beta-\alpha}} p^{\frac{1-\beta}{2(\beta-\alpha)}}, \tag{A.6}$$

which shows that for $p \downarrow 0$ the paths of the process $X(\cdot)$ become continuous. The smaller the expression in (A.6), the more "almost continuous" the paths of the process $X(\cdot)$.

It must be noted that since in the situation of the conjecture, $\alpha < \beta < 1$, and hence

$$\nu_2 = \frac{1-\alpha}{\beta-\alpha} > 1, \tag{A.7}$$

the speed-up of the process $X(t)$ compared with the process W_n is higher in the case $\alpha < \beta < 1$ than in the case $\alpha < \beta = 1$.

The "guess" (A.3) is proven to be correct by the same idea as used in (A.6):

$$|x| = O(\text{st.dev}(X)) = O(1) \ll p^{\nu_1}\left(\frac{c_1(1-p)}{c_2 p}\right)^{\frac{1}{\beta-\alpha}} = \left(\frac{c_1(1-p)}{c_2}\right)^{\frac{1}{\beta-\alpha}} p^{-\frac{1-\beta}{2(\beta-\alpha)}}.$$

B. Reno and NewReno with high drop probability

When the drop probability p increases, the performance of "classical" TCP, say Reno or NewReno, decreases and eventually deteriorates. There is of course no sharp boundary. The author of this paper uses, somewhat arbitrarily, a value of $p = .1$ beyond which he considers the performance to be unacceptable even for Telnet.

The "square root law" for TCP, used uncritically, predicts that if $p = .1$ then during congestion-avoidance periods the congestion window will fluctuate "around" $\sqrt{10} \sim 3.16$ MSSs, that is, much of the time is in the range of 1 to 7 MSSs.

If a packet is lost while the congestion window is 1 or 2 or 3 MSSs (and often when it is 4 MSSs) there will not be a third duplicate acknowledgement, and the flow goes into time–out. Thus, somewhere around $p = .1$ we will see that half or more of dropped packets (outside time–out) cause a time—out, and the situation gets worse at higher drop probabilities.

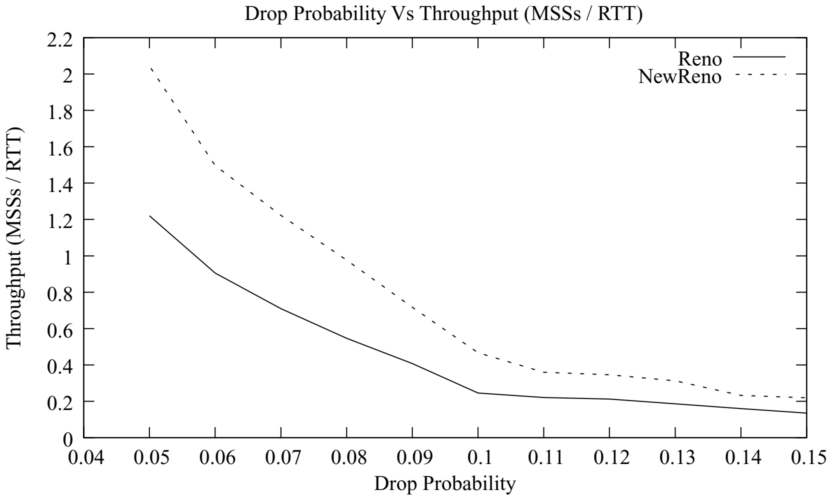

Figure 1. Throughputs for TCP Reno and NewReno.

In addition, simple arithmetic shows that by first approximation the average length of a time-out (assuming drop probability does not change, and there is independence) is around $\frac{1-p}{1-2p}$ RTTs (in fact a bit worse than this because it takes more than one RTT to recognize time–out). Thus, when p increases beyond .1, the fraction of drops that causes a time-out increases and at the same time the average duration of time–outs increases.

When the time–out is over the congestion window quickly returns to roughly half the value it had before the packet loss that caused the time–out, but then increases slower once the system is back in congestion avoidance.

Figure 1 shows the result of NS simulations in a network with a "nominal roundtrip time" (sum of propagation delays) RTT = 20 msec, and a negligible "serialization delay". The throughput is given for Reno as well as NewReno, as function of the drop probability p, and is given in MSSs/RTT (MSSs per 20 msec). The simulation is of course not for Telnet but for the situation of FTPing a large file.

In these simulations, the throughput of Reno dips below 1 MSS per RTT at a drop probability of about .06, while for NewReno it dips below that rate at a drop probability of about .08. At a drop probability of .1 both have pathetic performance: Considerably less than 1 packet per RTT. Thus, the system spends much of its time in time-out, and in for example Telnet the responsiveness to the human customer is unacceptable.

Acknowledgment

We thank Mr. Rahul B. Jain for the material in this Appendix.

References

[1] E. Altman, K. Avrachenkov and C. Barakat, A stochastic model for TCP/IP with stationary random loss, SIGCOMM'OO (2000) 231–242.

[2] E. Altman, K. Avrachenkov and C. Barakat, TCP network calculus: The case of large delay-bandwidth product, in: IEEE INFOCOM (New-York, USA, 2002).

[3] E. Altman, K. Avrachenkov, C. Barakat, A.A. Kherani and B.J. Prabhu, Analysis of scalable TCP, 7-th IEEE HSNMC (2004) 51 – 62.

[4] E. Altman, K. Avrachenkov and B. Prabhu, Fairness in MIMD congestion control algorithms, IEEE Infocom, Miami (2005) 13–17.

[5] E. Altman, K. Avrachenkov, A.A. Kherani and B.J. Prabhu, Analysis of scalable TCP in the presence of markovian losses, 3rd International workshop on protocols for fast long-distance networks, Lyon, France, February 3,4 (2005).

[6] E. Altman, K. Avrachenkov, A. Kherani and B. Prabhu, Performance analysis and stochastic stability of congestion control protocols, IEEE Infocom, Miami (2005) 13–17.

[7] E. Altman, C. Barakat and V.M. Ramos Ramos, Analysis of AIMD protocols over paths with variable delay, IEEE Infocom Hong-Kong (2004).

[8] E. Altman, T. Jimenez and D. Kofman. DPS queues with stationary ergodic service times and the performance of TCP in overload, IEEE INFOCOM Hong-Kong (2004).

[9] J. Barras, A. Misra and T.J. Ott Generalized TCP Congestion Avoidance and its effect on Bandwidth Sharing and Variability, in: *Proceedings Globecomm 2000*, (San Francisco, 2000).

[10] J. Baras, A. Misra and T.J. Ott, in: *Predicting Bottleneck Bandwidth Sharing by Generalized TCP Flows*, Computer Networks 40/4 (2002) 557–576.

[11] D. Bansal and H. Balakrishnan, Binomial congestion control algorithms, in: *Proceedings, INFOCOM* pp. 631–640 (2001).

[12] D. Bansal, H. Balakrishnan, S. Floyd and S. Shenker, Dynamic behavior of slowly-responsive congestion control algorithms, in: *Proceedings, SigComm* (2001).

[13] J. Baras, A. Misra and T.J. Ott, The window distribution of multiple TCPs with random loss queues, *Globecomm '99*, (1999).

[14] J. Baras, A. Misra and T.J. Ott, Using drop-biasing to stabilize the occupancy of random drop queues with TCP traffic, in: *Proceedings ICCS* (Singapore, 2000).

[15] Lawrence S. Brakmo and Larry L. Peterson, Performance problems in BSD4.4 TCP, in: *Proceedings of ACM SIGCOMM '95*, (1995).

[16] Lawrence S. Brakmo, Sean W. O'Malley and Larry L. Peterson, TCP vegas: New techniques for congestion detection and avoidance, in: *Proceedings of ACM SIGCOMM '94*, (1994).

[17] A. Budhiraja, F. Hernandez-Campos, V.G. Kulkarni and F.D. Smith, Stochastic differential equation for TCP window size: Analysis and experimental validation, in: *Probability in the Engineering and Informational Sciences*, p. 18 (2004).

[18] D. Chiu and R. Jain, Analysis of the increase/decrease algorithms for congestion avoidance in computer networks, Journal of Computer Networks and ISDN, (1989).

[19] Committee on Research Horizons in Networking, National Research Council, Looking over the Fence at Networks, A Neighbor's View Networking Research. National Academy Press, Washington, D.C., (2001).

[20] V. Dumas, F. Guillemin and P.H. Robert, A markovian analysis of additive-increase, multiplicative decrease (AIMD) algorithms. Adv. Appl. Prob. 34(1) (2002) 85–111.

[21] R. El Khoury and E. Altman, Analysis of scalable TCP. Available at www-sop.inria.fr/mistral/personnel/Eitan.Altman/fl-cont.html, (2004).

[22] W. Feng, D. Kandlur, G. Kang and D. Saha, Adaptive packet marking for providing differential services in the internet. ICNP (1998).

[23] S. Finch, Ornstein—uhlenbeck process. Availabe at pauillac.inria.fr/algo/csolve/ou.pdf, (2004).

[24] Sally Floyd, Connections with multiple congested gateways in packet-switched networks Part I: One way traffic, CCR 21(5) (1991) 30–47.

[25] S. Floyd, TCP and explicit congestion notification, ACM Computer Communications Review 21(5) (1994) 8–23.

[26] Sally Floyd, HighSpeed TCP for large congestion windows, IETF RFC 3649, (2003).

[27] Sally Floyd and Van Jacobson, Random early detection gateways for congestion avoidance, IEEE/ACM Transactions on Networking, (1993).

[28] F. Guillemin, P.H. Robert and B. Zwart, AIMD algorithms and exponential functionals, Ann. Appl. Prob. 14(1) (2004) 90–117.

[29] Van Jacobson, Congestion avoidance and control, in: *Proceedings of ACM SIGCOMM'88*, (1988).

[30] D. Katabi, M. Handley and C. Rohrs, Congestion control for high bandwidth-delay product networks, in: *Proceedings. ACM SIGCOMM'02*, (2002).

[31] C.T. Kelly, An ECN probe-based connection acceptance control, Available at www-lce.eng.cam.ac.uk/∼ctk21/papers/, (2002).

[32] C.T. Kelly, Scalable TCP: Improving performance in highspeed wide area networks, ACM SIGCOMM Computer Communication Review 32(2) (2003) 83–91.

[33] C.T. Kelly, Engineering Flow Controls for the Internet, PhD Thesis, Cambridge Univ, England, Available at http://www-lce.eng.cam.ac.uk/∼ctk21/papers/, (2004).

[34] F.P. Kelly, A. Maulloo and D. Tan, Rate control for communication networks: Shadow prices, Proportional fairness and stability, J. Oper. Res. Soc. (1998) 237–252.

[35] S. Kunniyur and S. Srikant, Analysis and design of an adaptive virtual queue (AVQ) algorithm for active queue management, in: *Proceedings of ACM Sigcomm'01* pp. 123–134, (2001).

[36] T.V. Lakshman and U. Madhow, The performance of TCP/IP for networks with high Bandwidth-Delay products and random loss, Trans of Netw, (1997).

[37] T.V. Lakshman, U. Madhow and B. Suter, Window-based error recovery and flow control with a slow acknowledgement channel: A study of TCP/IP performance, Infocom'97, (1997).

[38] Steven H. Low and R. Srikant, A mathematical framework for designing a low-loss, low-delay internet. Network and Spatial Economics 4(1) (2004) 75–102.

[39] A. Misra and T.J. Ott, Effect of exponential averaging on the variability of a RED queue, in: *Proceedings, ICC'01*, (2001) Helsinki.

[40] A. Misra and T.J. Ott, Jointly coordinating ECN and TCP for rapid adaptation to varying bandwidth, in: *Proceedings of MILCOM 2001*, Best Unclassified Paper in MILCOM, (2001).

[41] A. Misra and T.J. Ott, Performance sensitivity and fairness of ECN-aware Modified TCP, Journal of Performance Evaluation (PEVA) 53/3 (2003) 255–272.

[42] R. Marquez, E. Altman and S. Sole-Alvarez, Modeling TCP and high speed TCP: A nonliinear extension to AIMD mechanisms. 7-th IEEE HSNMC (2004) 132–143.

[43] M. Mathis, J. Semke, J. Mahdavi and T.J. Ott, The macroscopic behavior of the TCP congestion avoidance algorithm, Computer Communications Review 27(3) (1997) 67–82.

[44] A. Misra and T.J. Ott, The window distribution of idealized TCP congestion avoidance with variable packet loss, in: *Proceedings Infocom '99*, (1999), pp. 1564–1572.

[45] R. Morris, Scalable TCP congestion control. IEEE Infocom, (2000).

[46] T.J. Ott, ECN protocols and the TCP paradigm. Available at web.njit.edu/∼ott/Papers Also presented at the *Workshop on Modeling of Flow and Congestion Control Mechanisms*, Ecole Normale Superieure, Paris, Sept., (2000).

[47] T.J. Ott, Rate of convergence for the "Square root formula" in TCP. Available at web.njit.edu/∼ott/Papers Submitted for publication. On the same website there also is a "modified version" which contains stronger results for the case $0 < \alpha < 1$, (2004).

[48] T.J. Ott, The transient behavior of idealized TCP congestion avoidance, Work in progress, draft available on request, (2005).

[49] T.J. Ott, On the Ornstein-Uhlenbeck process with delayed feedback, Available at web.njit.edu/∼ott/Papers, (draft).

[50] T.J. Ott, J.H.B. Kemperman and M. Mathis, The stationary behavior of ideal TCP congestion avoidance. [47] This paper has circulated on the Web since August 1996 and has frequently been cited. See http://web.njit.edu/∼ott/Papers for a copy, (1996).

[51] J. Padhye, V. Firoiu, D. Towsley and J. Kurose, Modeling TCP throughput: A simple model and its empirical validation, *ACM SIGCOMM*, (1998).

[52] R. Srikant, The mathematics of internet congestion control Birkhauser, (2004).

[53] R. Srikant, Models and methods for analyzing internet congestion control algorithms, in: (eds.) *Lecture Notes in Control and Information Sciences (LCNCIS)*, C.T. Abdallah, J. Chiasson, and S. Tarbourich, Springer Verlag, (2004).

[54] J. Swanson, Private Communication, (2005).

[55] A. Kumar, Comparative performance analysis of versions of TCP in a local network with a lossy link, IEEE/ACM Transactions on Networking, (1998).

[56] A. Tang, J. Wang and S.H. Low, Understanding CHOKe: Throughgput and spatial characteristics. IEEE/ACM Transactions on Networking (ToN) 12(4) (2004) 694–707.

[57] F. Paganini, Z. Wang, J.C. Doyle and S.H. Low, Congestion control for high performance, stability, and fairness in general networks, To appear in *IEEE/ACM Transactions on Networking (ToN)*, (2005).

[58] T.J. Ott, T.V. Lakshman and L.H. Wong, SRED: Stabilized RED, in: *Proceedings of IEEE INFOCOM'99*, pp. 1346–1355 (1999).

[59] K.K. Ramakrishnan, F. Floyd and D. Black, The addition of explicit congestion control (ECN) to IP. IETF RFC 3168, (2001).

[60] W.R. Stevens, *TCP/IP Illustrated*, vol. 1. (Addison-Wesley, Reading MA, 1994).

[61] G.R. Wright and W.R. Stevens, *TCP/IP Illustrated*, vol. 2. (Addison-Wesley, Reading MA, 1995).