Towards Stability Analysis of Data Transport Mechanisms: a Fluid Model and an Application

Gayane Vardoyan[*]

gvardoyan@cs.umass.edu

C.V. Hollot[†]

hollot@ecs.umass.edu

Don Towsley[*]

towsley@cs.umass.edu

[*]College of Information and Computer Sciences, [†]Department of Electrical and Computer Engineering
University of Massachusetts, Amherst

Abstract

The Transmission Control Protocol (TCP) utilizes congestion avoidance and control mechanisms as a preventive measure against congestive collapse and as an adaptive measure in the presence of changing network conditions. The set of available congestion control algorithms is diverse, and while many have been studied from empirical and simulation perspectives, there is a notable lack of analytical work for some variants. To gain more insight into the dynamics of these algorithms, we: (1) propose a general modeling scheme consisting of a set of functional differential equations of retarded type (RFDEs) and of the congestion window as a function of time; (2) apply this scheme to TCP Reno and demonstrate its equivalence to a previous, well known model for TCP Reno; (3) show an application of the new framework to the widely-deployed congestion control algorithm TCP CUBIC, for which analytical models are few and limited; and (4) validate the model using simulations. Our modeling framework yields a fluid model for TCP CUBIC. From a theoretical analysis of this model, we discover that TCP CUBIC is *locally uniformly asymptotically stable* – a property of the algorithm previously unknown.

1 Introduction

TCP carries most of the traffic on the Internet. One of its important functions is to perform end-to-end congestion control to alleviate congestion in the Internet and to provide fair bandwidth sharing among different flows. To date, many different congestion control algorithms (variants) have been developed, among which are Reno, Vegas, STCP [1], CUBIC [2], H-TCP [3], and BBR [4]. Stability is an imperative property for any dynamical system. The stability of several of these variants including Reno, Vegas, and STCP has been extensively and carefully studied, however, little is known about the stability properties of more recent variants such as CUBIC and H-TCP. These latter variants have typically been studied through simulation and experimentation, neither of which are adequate to make careful statements about stability. As we will observe, for some variants this deficiency is due to the lack of a modeling framework with which to develop appropriate models that are amenable to a formal stability analysis. The goals of this paper are to point out deficiencies in the previous framework used to study variants such as Reno that make it unsuitable to study a variant such as CUBIC, and then to present a new framework and apply it to the analysis of CUBIC. Our choice of CUBIC is because it is a popular variant that is the default in the Linux distribution.

The traditional approach for modeling a congestion control algorithm's behavior is to derive a differential equation (DE) for its congestion window (*cwnd*) or sending rate as a function of time. Such DEs typically include the algorithm's increase and decrease rules, as well as loss probability functions, for example, to incorporate an active queue management (AQM) policy.

This method is highly effective for modeling certain types of controllers, such as TCP Reno and STCP, whose *cwnd* update rules are very simple (*e.g.* Reno's *cwnd* grows by one every round trip and decreases by half upon congestion detection). However, this approach reaches its limitations when presented with a controller whose *cwnd* update functions are complex, thereby making it difficult or impossible to write a DE for the *cwnd* or sending rate directly. For example, CUBIC's increase update rule is a function of *time since last loss* and of the *congestion window size immediately before loss.* Moreover, in the case of CUBIC, the steady-state value of *cwnd* lies at the saddle point of the window function, which obstructs the stability analysis of this point of interest.

To overcome the impediments of the traditional approach, we develop a novel framework that exploits the fact that all *cwnd*- and rate-based controllers that utilize packet loss information[1] to make changes to the *cwnd* or rate have two variables in common: the value of *cwnd* (rate) immediately before loss and the time elapsed since last loss. As a consequence, one can derive a set of two DEs: the first for describing the maximum *cwnd* (rate) as a function of time, and the second for describing the duration of congestion epochs. This is a relatively easy task, compared to deriving a DE for *cwnd* (rate) of a complex algorithm directly. The advantage of such a model is that it offers tremendous versatility since it does not define *cwnd* or rate functions within the set of DEs, with the latter being identical for many controllers. Note that the proposed model is applicable not only to TCP-based congestion controllers, but also to UDT [5] and QCN [6].

In this work, we use simulation to validate our analytical models. As NS3 [7] does not natively support CUBIC, and existing implementations of this protocol have known problems, we introduce a lightweight simulation framework that is easily programmed to switch between different congestion control variants. We use this framework to validate the DE model and observe that the average *cwnd* predicted by both are in close agreement. As system parameters are varied, the simulation and the DE model agree on whether the system is stable. For TCP CUBIC, we observe that instability can be introduced by deviating the initial conditions too far from their fixed-point values. While our analysis states that CUBIC is locally asymptotically stable, these simulations complement the theory by demonstrating that CUBIC is *not* globally stable.

The contributions of this work are as follows:
- a new modeling framework applicable to a diverse set of algorithms,
- an application of this model to CUBIC, and a stability analysis of this algorithm,
- validation of this model with simulation; the simulation is of independent interest separate from this paper (a description of the framework is provided in the Appendix).

The rest of this paper is organized as follows: we discuss related work in Section 2. We introduce the modeling framework in Section 3.1 and apply it to TCP Reno. In Section 3.2, we show that the new framework is equivalent to the one presented in [8]. In Section 4.1, we apply the new modeling scheme to TCP CUBIC. In the remainder of Section 4, we perform a careful stability analysis of CUBIC and present a convergence result. In Section 5, we validate the new model and the stability result for CUBIC using simulations. We draw conclusions in Section 6.

2 Background

There exist a number of analytical studies for modeling TCP and analyzing its stability. In [8], Misra *et al.* derive a fluid model for a set of TCP Reno flows and show an application to a

---

[1]Note that this includes not only ACK-based algorithms, but also packet marking schemes as in ECN (Explicit Congestion Notification). From this point forward, we refer to such schemes as "loss-based".

Term	Definition
C	per-flow capacity
τ	link delay
$W_{\max}(t)$	the size of the *cwnd* immediately before loss
$s(t)$	the time elapsed since loss
$W(t)$	the *cwnd* as a function of time
$p(t)$	a probability of loss function

Table 1: Term definitions.

networked setting where RED (Random Early Detection) is the AQM policy. Kelly proposed an optimization-based framework for studying and designing congestion control algorithms in [9], where STCP was an output. In [10], Srikant presented a simple analysis of Jacobson's TCP congestion control algorithm. In [11], Hollot *et al.* analyze the stability of TCP with an AQM system implementing RED.

Huang *et al.* develop and analyze the stability of a general nonlinear model of TCP in [12], focusing on HighSpeed, Scalable, and Standard TCP for comparisons of relative stability. The authors rely on functions $f(w)$ and $g(w)$, which are additive and multiplicative parameters, respectively, and are both functions of the current congestion window size. Our model contrasts from these examples in that rather than modeling the congestion window directly, we instead model two interdependent variables (maximum *cwnd* and time between losses) that in turn determine the evolution of the window. This new method presents a window of opportunity for modeling complex, nonlinear transport algorithms for which it is not possible to write a DE for *cwnd* directly or whose $f(w)$ and $g(w)$ functions cannot be written in closed form.

Bao *et al.* propose Markov chain models for average steady-state TCP CUBIC throughput, in a wireless environment [13]. In [14], Poojary *et al.* derive an expression for average *cwnd* of a single CUBIC flow under random losses. In contrast to [13] and [14], the model we present in this work for CUBIC provides insight into both the transient and steady-state behavior of the algorithm. Moreover, we utilize Lyapunov stability theory to prove that CUBIC is locally asymptotically stable independent of link delay and other system parameters (the parameters only affect the region of stability). This result is one of the main contributions of this work.

3 The Model

In this section, we present the new model, which is the focus of this work. As a proof of concept, we apply this model to TCP Reno and show that it is mathematically equivalent to the well-known DE model originally presented in [8]. We note that while the two models are equivalent, they make use of different types of information, which is essential for developing a fluid model for TCP CUBIC presented in Section 4.

In the analysis that follows, we will use the notation $f \equiv f(t)$ to represent a function or variable that is not time-delayed. Similarly, we will use $f_T \equiv f(t - T)$ to represent a function or variable that is delayed by an amount of time T. We will also use $\dot{f} = df(t)/dt$ to represent a function or variable f differentiated with respect to time.

3.1 The New Model

Table 1 presents some useful definitions. The main idea behind the model is the following: instead of deriving a DE for the *cwnd* function $W(t)$ directly, which is specific to a data transport algorithm, we instead derive DEs for $W_{\max}(t)$ – the size of the *cwnd* immediately before loss,
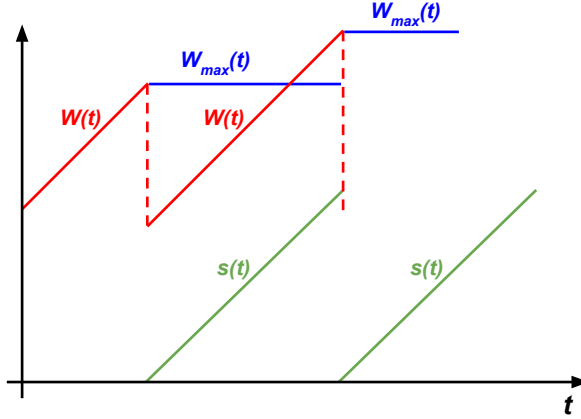
Figure 1: $W(t)$, $W_{\max}(t)$, and $s(t)$ for TCP Reno.

and $s(t)$ – the amount of time elapsed since last loss, which are variables common to all loss-based algorithms. Since $W(t)$, is a function of $W_{\max}(t)$ and $s(t)$, it is completely determined by their DEs. The result is the following model[2]:

$$\frac{dW_{\max}(t)}{dt} = -(W_{\max}(t) - W(t))\frac{W(t-\tau)}{\tau}p(t-\tau)$$
$$\frac{ds(t)}{dt} = 1 - s(t)\frac{W(t-\tau)}{\tau}p(t-\tau)$$
(1)

Above, $p(t-\tau)$ is a loss probability function. The expression $W(t-\tau)p(t-\tau)/\tau$ describes the rate of packet loss, which is delayed by τ because loss occurs at the congestion point, not at the source. The first DE describes the behavior of $W_{\max}$, which takes the value of $W(t)$ right before a loss. At the time of loss, if $W_{\max}(t) > W(t)$, then $W_{\max}$ decreases by the amount $W_{\max}(t) - W(t)$; otherwise, it increases by the same amount. The second DE describes the evolution of the time since last loss $s(t)$, which grows by one unit and is reset to zero upon loss. This system can be adapted to a rate-based scheme in terms of maximum rate and time since last rate decrease, simply by dividing each DE by τ. Since we will be describing applications of this model to TCP Reno and CUBIC, which are both *cwnd*-based, we use (1) in the interest of the paper.

Figure 1 illustrates $W_{\max}(t)$, $s(t)$, and $W(t)$ for TCP Reno. To adapt model (1) to TCP Reno, we define Reno's *cwnd* as a function of $W_{\max}(t)$ and $s(t)$. At the time of loss, $W(t) = W_{\max}(t)$ is halved. This becomes the initial value of $W(t)$ in the new congestion epoch. $W(t)$ then increases by one segment for every round-trip time, so the total increase is $s(t)/\tau$ after $s(t)$ time has elapsed since the last loss. Hence,

$$W(t) = \frac{W_{\max}(t)}{2} + \frac{s(t)}{\tau}.$$
(2)

Then the fluid model for Reno is (1) combined with (2).

The loss probability function can be customized according to the specific characteristics of a given system, such as queue size and AQM policy. For simplicity, we use the following function in all subsequent models:

$$p(t) = \max\left(1 - \frac{C\tau}{W(t)}, 0\right).$$
(3)

---

[2]Note that $W(t)$ must be either derived explicitly, for example as in (2) for TCP Reno or given in the definition of the controller, as in (7) for CUBIC.

This function is presented in [10] as an approximation of the M/M/1/B drop probability when the buffer size $B \to \infty$.

Note that model (1) does not specify $W(t)$, and therein lies the versatility of this scheme. For a given *cwnd*-based transport algorithm, the modeler need only to substitute a function describing the evolution of *cwnd* over time, as we did for Reno. We demonstrate this technique again with CUBIC in Section 4. This property of the model is useful both for analyzing existing algorithms and examining the stability of new ones.

3.2   Model Equivalence for TCP Reno

Consider the well-established model for TCP Reno's *cwnd* from [8] (equation (4) to be precise):

$$\frac{dW(t)}{dt} = \frac{1}{\tau} - \frac{W(t)}{2} \frac{W(t - \tau)}{\tau} p(t - \tau). \tag{4}$$

We assume τ to be constant for simplicity, even though the round-trip time in [8] varies in time as a function of both the propagation and queueing delays.

We now show that the two models (*i.e.* the model represented by (1), (2) and the model represented by (4)) are mathematically equivalent. Differentiating (2) with respect to t, we have:

$$\dot{W} = \frac{\dot{W}_{\max}}{2} + \frac{\dot{s}}{\tau}. \tag{5}$$

Substituting (1) into (5) yields

$$\dot{W} = \frac{1}{\tau} \left(1 - s \frac{W_\tau}{\tau} p_\tau \right) + \frac{1}{2} \left((W - W_{\max}) \frac{W_\tau}{\tau} p_\tau \right). \tag{6}$$

From (2), we know that $W_{\max} = 2(W - s/\tau)$. Substituting this expression for $W_{\max}$ into (6) and simplifying yields

$$\begin{aligned}
\dot{W} &= \frac{1}{\tau} \left(1 - s \frac{W_\tau}{\tau} p_\tau \right) + \frac{1}{2} \left(\left(W - 2 \left(W - \frac{s}{\tau} \right) \right) \frac{W_\tau}{\tau} p_\tau \right) \\
&= \frac{1}{\tau} - s \frac{W_\tau}{\tau^2} p_\tau - \frac{W}{2} \frac{W_\tau}{\tau} p_\tau + s \frac{W_\tau}{\tau^2} p_\tau \\
&= \frac{1}{\tau} - \frac{W}{2} \frac{W_\tau}{\tau} p_\tau.
\end{aligned}$$

The last line corresponds to equation (4) and completes our proof of the equivalence of the models. When used with Reno, model (1) can be linearized and used to derive a transfer function. The latter can be analyzed to yield system parameter-dependent conditions for Reno's stability. This analysis is similar to the one presented in [10].

4   Analysis of TCP CUBIC

In this section, we perform a local stability analysis of TCP CUBIC. To do so, we first create a fluid model for this congestion control algorithm using the framework introduced in the previous section. Then, we show that the system has a unique fixed point and prove the existence and uniqueness of a solution. Next, we show that the linearization method yields inconclusive results when applied to the model, and are thus motivated to use Lyapunov's direct method to prove the stability of the system. First, we introduce a Lyapunov function candidate and since the

system is time-delayed, use Razumikhin's Theorem to show that the candidate is suitable and that stability holds in a neighborhood of the fixed point of the system. A consequence of the failed linearization is that we will not prove exponential stability for CUBIC, but we can still show asymptotic and Lyapunov stability. Finally, we derive convergence results on the system's solution.

4.1 TCP CUBIC Fluid Model

TCP CUBIC's congestion window function is defined in terms of the time since last loss $s(t)$ and maximum value of *cwnd* immediately before the last loss $W_{\max}(t)$ [2]:

$$W(t) = c\left(s(t) - \sqrt[3]{\frac{W_{\max}(t)b}{c}}\right)^3 + W_{\max}(t) \tag{7}$$

where b is a multiplicative decrease factor and c is a scaling factor. Figure 2 illustrates the evolution of CUBIC's *cwnd* over time. The opaque red curves represent behavior in steady state: the window is concave until a loss occurs at CUBIC's fixed-point value of *cwnd*, $\hat{W}$. The light red curves describe *cwnd* behavior if a loss does not occur: the window becomes convex, also known as CUBIC's probing phase. The fluid model for CUBIC is then simply (1) coupled with (7), with (3) as the loss probability function. Prior to the development of (1), we attempted to develop a fluid model by first computing the equilibrium point for CUBIC, but this exercise gave a value of s at (7)'s saddle point and consequently, a confounding linearization of $dW/dt = 0$. Further attempts at deriving dW/dt, taking into account the time-dependencies $s(t)$ and $W_{\max}(t)$, resulted in a highly complex DE involving both $W_{\max}(t)$, $s(t)$, and their derivatives. Even obtaining the fixed points of this DE would be highly cumbersome, compared to obtaining the fixed point of (1).
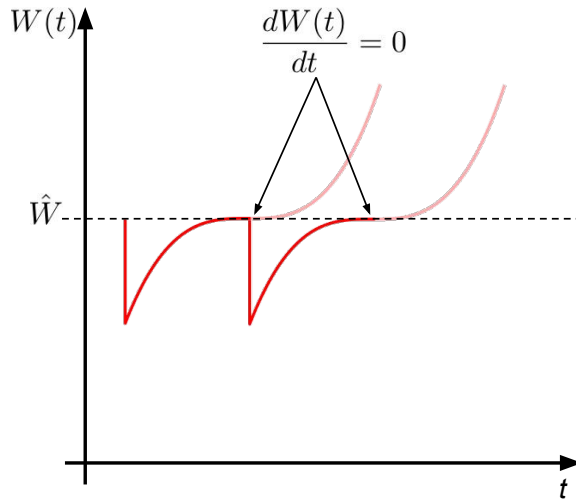

Figure 2: CUBIC's saddle point causes $dW(t)/dt$ to evaluate to zero at the fixed point of the system.

4.2 Fixed Point Analysis

Let $\hat{W}_{\max}$, $\hat{s}$, $\hat{W}$, and $\hat{p}$ represent the fixed point values of $W_{\max}(t)$, $s(t)$, $W(t)$, and $p(t)$, respectively. Using the fact that in steady state, $W(t) = W(t-\tau) = \hat{W}$ and $p(t) = p(t-\tau) = \hat{p}$,

system (1) becomes

$$-(\hat{W}_{\max} - \hat{W})\frac{\hat{W}}{\tau}\hat{p} = 0, \tag{8}$$

$$1 - \hat{s}\frac{\hat{W}}{\tau}\hat{p} = 0. \tag{9}$$

From (9), we see that

$$\hat{s} = \frac{\tau}{\hat{W}\hat{p}}. \tag{10}$$

It is clear that $\hat{W}$ and $\hat{p}$ do not equal zero in steady state. Using this information, along with (8), we conclude that $\hat{W}_{\max} = \hat{W}$. In steady state, (7) becomes

$$\hat{W} = c\left(\hat{s} - \sqrt[3]{\frac{\hat{W}b}{c}}\right)^3 + \hat{W}.$$

This equation yields

$$\hat{s} = \sqrt[3]{\frac{\hat{W}b}{c}}. \tag{11}$$

Combining (10) and (11), we have

$$\frac{\tau}{\hat{W}\hat{p}} = \sqrt[3]{\frac{\hat{W}b}{c}}$$

where $\hat{p} = 1 - C\tau/\hat{W}$ (since $\hat{p} > 0$ in steady state). Substitution yields

$$\hat{W}(\hat{W} - C\tau)^3 = \frac{\tau^3 c}{b},$$

which can be solved for $\hat{W}$ as a function of solely the system parameters c, b, C, and τ. This value can be used either with (10) or (11) to obtain a value for $\hat{s}$ solely as a function of the system parameters. This concludes the fixed point analysis. An interesting comparison is $\hat{W}$ as a function of $\hat{p}$ for Reno and CUBIC. Model (4) yields

$$\hat{W}_{Reno} = \sqrt{\frac{2}{\hat{p}}}, \text{ while } \hat{W}_{CUBIC} = \sqrt[4]{\frac{\tau^3 c}{\hat{p}^3 b}}.$$

In other words, whereas throughput under Reno depends on loss probability as $\mathcal{O}(\hat{p}^{-1/2})$, CUBIC exhibits a $\hat{p}^{-3/4}$ dependence.

4.3   Change of Variables

To simplify stability analysis, we perform a change of variables so that the fixed point of the system is located at the origin. To accomplish this, define $\mathbf{x}$ as follows:

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} W_{\max}(t) - \hat{W}_{\max} \\ s(t) - \hat{s} \end{bmatrix} = \begin{bmatrix} W_{\max}(t) - \hat{W} \\ s(t) - \hat{s} \end{bmatrix}$$

where the last equality follows because $\hat{W}_{\max} = \hat{W}$. Also, define $\Psi(t)$ and $\tilde{p}(t)$ as follows:

$$\Psi(t) = c\left(x_2(t) + \hat{s} - \sqrt[3]{\frac{b(x_1(t) + \hat{W})}{c}}\right)^3 + x_1(t) + \hat{W},$$

$$\tilde{p}(t) = \max\left(1 - \frac{C\tau}{\Psi(t)}, 0\right).$$

Then the new system is:

$$\dot{x}_1 = \left(\Psi - x_1 - \hat{W}\right)\frac{\Psi_\tau}{\tau}\tilde{p}_\tau,$$
$$\dot{x}_2 = 1 - (x_2 + \hat{s})\frac{\Psi_\tau}{\tau}\tilde{p}_\tau. \tag{12}$$

Note that Ψ_τ and $\tilde{p}_\tau$ are functions of $x_1(t - \tau) \equiv x_{1_\tau}$ and $x_2(t - \tau) \equiv x_{2_\tau}$. It is easy to verify that $\mathbf{x}^* = [x_1\ x_2\ x_{1_\tau}\ x_{2_\tau}]^T = \mathbf{0}$ is a fixed point of the new system.

Claim 4.1. $\mathbf{x}^* = \mathbf{0}$ *is a fixed point of the new system.*

Proof: *At* $\mathbf{x}^*$*, we have:*

$$\hat{\Psi} = c\left(\hat{s} - \sqrt[3]{\frac{b\hat{W}}{c}}\right)^3 + \hat{W}$$

$$\dot{x}_1 = \left(\Psi - \hat{W}\right)\frac{\Psi_\tau}{\tau}\tilde{p}_\tau \tag{13}$$

$$\dot{x}_2 = 1 - \hat{s}\frac{\Psi_\tau}{\tau}\tilde{p}_\tau \tag{14}$$

From the fixed point analysis of the original system, recall that $\hat{s} = \sqrt[3]{b\hat{W}/c}$*, which yields* $\hat{\Psi} = \hat{W}$*. Plugging this into equation (13), we get* $\dot{x}_1 = 0$*. Similarly, plugging in* $\hat{\Psi} = \hat{W}$ *into (14), we have:*

$$\dot{x}_2 = 1 - \hat{s}\frac{\hat{W}}{\tau}\hat{p}$$

From the fixed point analysis of the original system, recall that $\hat{s} = \tau/(\hat{W}\hat{p})$*. Therefore,* $\dot{x}_2 = 0$ *as well, and the proof is complete.* $\square$

We can analyze the stability of the system (12) at the origin, which is equivalent to analyzing the stability of the original system (1) at the equilibrium values $\hat{W}_{\max}$ and $\hat{s}$. The CUBIC representation in (12) forms the basis for our subsequent analyses.

4.4 Existence and Uniqueness of Solution

We state the existence and uniqueness theorem at it appears in [15]:

Theorem 4.1 (Theorem 1.2 from *Stability of Time-Delay Systems*)**.** *Suppose that* Ω *is an open set in* $\mathbb{R} \times \mathcal{C}$ *(where* $\mathcal{C}$ *is the set of* $\mathbb{R}^n$*-valued continuous functions on* $[-\tau, 0]$*),* $f : \Omega \to \mathbb{R}^n$ *is continuous, and* $f(t, \phi)$ *is Lipschitzian in* ϕ *in each compact set in* Ω*, that is, for each given compact set* $\Omega_0 \subset \Omega$*, there exists a constant* L *such that*

$$||f(t, \phi_1) - f(t, \phi_2)|| \leq L||\phi_1 - \phi_2||$$

for any $(t, \phi_1) \in \Omega_0$ *and* $(t, \phi_2) \in \Omega_0$*. If* $(t_0, \phi) \in \Omega$*, then there exists a unique solution of* $\dot{x}(t) = f(t, x_t)$ *through* (t_0, ϕ)*.*

To prove existence and uniqueness for our system, it is sufficient to show that $\dot{x}_1$ and $\dot{x}_2$ are continuously differentiable functions in some neighborhood of the fixed point. We assume that this neighborhood is small enough so that $\tilde{p}_\tau > 0$. Then the system becomes:

$$\dot{x}_1 = \left(\Psi - x_1 - \hat{W} \right) \frac{(\Psi_\tau - C\tau)}{\tau},$$

$$\dot{x}_2 = 1 - (x_2 + \hat{s}) \frac{(\Psi_\tau - C\tau)}{\tau}.$$

$$\text{Let } F = \frac{b(x_1 + \hat{W})}{c} \text{ and } \Phi = x_2 + \hat{s} - \sqrt[3]{F}.$$

Following are the partial derivatives of $\dot{x}_1$:

$$\frac{\partial \dot{x}_1}{\partial x_1} = -\frac{b}{\tau} \Phi^2 F^{-2/3} (\Psi_\tau - C\tau),$$

$$\frac{\partial \dot{x}_1}{\partial x_2} = \frac{3c}{\tau} \Phi^2 (\Psi_\tau - C\tau),$$

$$\frac{\partial \dot{x}_1}{\partial x_{1_\tau}} = \frac{(\Psi - x_1 - \hat{W})}{\tau} \left(-b\Phi_\tau^2 F_\tau^{-2/3} + 1 \right),$$

$$\frac{\partial \dot{x}_1}{\partial x_{2_\tau}} = \frac{3c}{\tau} (\Psi - x_1 - \hat{W}) \Phi_\tau^2.$$

These partials provide the first restriction to the region where stability is being analyzed. Specifically, the term $F^{-2/3}$ indicates that x_1 and x_{1_τ} should be restricted to an interval $[-\rho\hat{W}, \rho\hat{W}]$, $0 < \rho < 1$. Next, we look at the partial derivatives of $\dot{x}_2$:

$$\frac{\partial \dot{x}_2}{\partial x_1} = 0, \quad \frac{\partial \dot{x}_2}{\partial x_2} = -\frac{(\Psi_\tau - C\tau)}{\tau},$$

$$\frac{\partial \dot{x}_2}{\partial x_{1_\tau}} = -\frac{(x_2 + \hat{s})}{\tau} \left(-b\Phi_\tau^2 F_\tau^{-2/3} + 1 \right),$$

$$\frac{\partial \dot{x}_2}{\partial x_{2_\tau}} = -\frac{3c}{\tau} (x_2 + \hat{s}) \Phi_\tau^2.$$

Under the restriction stated above, these partials are also continuous, and hence, we have local Lipschitz continuity – the requirement for existence and uniqueness.

4.5 Stability Analysis

In general, the linearization of (12) and (3) about $\mathbf{x} = \mathbf{x}^* = \mathbf{0}$ is

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \frac{1}{\hat{s}} A_0 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \frac{\hat{s}}{\tau} A_1 \begin{bmatrix} x_{1_\tau} \\ x_{2_\tau} \end{bmatrix}, \text{ where}$$

$$A_0 = \begin{bmatrix} \frac{\partial \Psi}{\partial x_1} - 1 & \frac{\partial \Psi}{\partial x_2} \\ 0 & -1 \end{bmatrix} \Bigg|_{\mathbf{x}=\mathbf{x}^*} \text{ and } A_1 = \begin{bmatrix} 0 & 0 \\ \frac{\partial \Psi_\tau}{\partial x_{1_\tau}} & \frac{\partial \Psi_\tau}{\partial x_{2_\tau}} \end{bmatrix} \Bigg|_{\mathbf{x}=\mathbf{x}^*}.$$

For CUBIC, $\partial \Psi / \partial x_1|_{\mathbf{x}=\mathbf{x}^*} = 1$ and $\partial \Psi / \partial x_2|_{\mathbf{x}=\mathbf{x}^*} = 0$, so that $\dot{x}_1 = 0$. This means that linearization has failed; *i.e.*, stability of the linearized system cannot be generalized to local stability of the nonlinear system. The key cause of this problem is the fact that the fixed point value of x_2, 0, is the saddle point of the function Ψ (or equivalently, $\hat{s}$ is the saddle point of $W(t)$). Figure

2 illustrates this phenomenon. This causes all first-order partial derivatives of $\dot{x}_1$ to evaluate to zero at $\mathbf{x}^* = \mathbf{0}$. Hence, in order to incorporate a local contribution from $\dot{x}_1$ in the analysis, it is necessary to expand $\dot{x}_1$ further. Specifically, a third-order Taylor Series expansion is necessary, since all second-order terms also evaluate to zero at the origin. The expanded system looks as follows:

$$\dot{x}_1 = -\alpha x_1^3 + \beta x_1^2 x_2 - \gamma x_1 x_2^2 + \delta x_2^3 + h_1$$
$$\dot{x}_2 = -\frac{1}{\hat{s}}x_2 - \frac{\hat{s}}{\tau}x_{1_\tau} + h_2 \tag{15}$$

$$\text{where } \alpha = \frac{b^3}{27c^2\hat{s}^7}, \ \beta = \frac{b^2}{3c\hat{s}^5}, \ \gamma = \frac{b}{\hat{s}^3}, \text{ and } \delta = \frac{c}{\hat{s}}$$

and h_1 and h_2 are higher-order terms of $\dot{x}_1$ and $\dot{x}_2$, respectively. To analyze the stability of (15), we use the Lyapunov-Razumikhin Theorem, the statement of which is given below as it appears in [15]. For the purpose of this theorem, we introduce some notation. Let $\mathcal{C} = \mathcal{C}([-\tau, 0], \mathbb{R}^n)$ be the set of continuous functions mapping the interval $[-\tau, 0]$ to $\mathbb{R}^n$, where τ is the maximum delay of a system. For any $A > 0$ and any continuous function of time $\psi \in \mathcal{C}([t_0 - \tau, t_0 + A], \mathbb{R}^n)$, and $t_0 \leq t \leq t_0 + A$, let $\psi_t \in \mathcal{C}$ be a segment of the function ψ defined as $\psi_t(\theta) = \psi(t + \theta)$, $-\tau \leq \theta \leq 0$. The general form of a retarded functional differential equation is

$$\dot{x}(t) = f(t, x_t) \tag{16}$$

Below, $\mathbb{R}_+$ is the set of positive real numbers, and $\bar{\mathbb{S}}$ is the closure of the set $\mathbb{S}$.

Theorem 4.2 (Lyapunov-Razumikhin Theorem). *Suppose $f : \mathbb{R} \times \mathcal{C} \to \mathbb{R}^n$ takes $\mathbb{R} \times$ (bounded sets of $\mathcal{C}$) into bounded sets of $\mathbb{R}^n$, and $u, v, w : \bar{\mathbb{R}}_+ \to \bar{\mathbb{R}}_+$ are continuous nondecreasing functions, $u(s)$ and $v(s)$ are positive for $s > 0$, and $u(0) = v(0) = 0$, v strictly increasing. If there exists a continuously differentiable function $V : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}$ such that*

$$u(||x||) \leq V(t, x) \leq v(||x||), \quad \text{for } t \in \mathbb{R} \text{ and } x \in \mathbb{R}^n, \tag{17}$$

$w(s) > 0$ for $s > 0$, and there exists a continuous nondecreasing function $p(s) > s$ for $s > 0$ such that

$$\dot{V}(t, x(t)) \leq -w(||x(t)||) \tag{18}$$
$$\text{if } V(t + \theta, x(t + \theta)) \leq p(V(t, x(t))) \tag{19}$$

for $\theta \in [-\tau, 0]$, then the system (16) is uniformly asymptotically stable. If in addition $\lim_{s \to \infty} u(s) = \infty$, then the system (16) is globally uniformly asymptotically stable.

Note that in this work, we will only prove *local* stability for CUBIC. Therefore, our goal is to show that we can find a function V for which all conditions specified in the theorem are valid locally, *i.e.*, in a sufficiently small neighborhood around the fixed point.

A popular choice of Lyapunov function is the quadratic candidate, *i.e.* a function of the form

$$Z(\mathbf{x}) = \mathbf{x}^T P \mathbf{x}, \text{ where } P = \begin{bmatrix} p_1 & p_2 \\ p_2 & p_4 \end{bmatrix} \tag{20}$$

is positive definite. Not surprisingly, the quadratic form Z, which is a sufficient form in working with linear dynamic systems, proves unsuitable. To understand why, consider the time derivative of (20) along solutions to (15):

$$\dot{Z} = 2\dot{x}_1(p_1 x_1 + p_2 x_2) + 2\dot{x}_2(p_2 x_1 + p_4 x_2). \tag{21}$$

The first term above is quartic in x_1 and x_2 (because $\dot{x}_1$ in (15) is cubic in x_1, x_2), but the second term is quadratic in x_1, x_2, and x_{1_τ}. In a small neighborhood of $\mathbf{x}^*$, the quadratic terms dominate; *i.e.*, (21) becomes

$$\dot{Z} = 2\left(-\frac{1}{\hat{s}}x_2 - \frac{\hat{s}}{\tau}x_{1_\tau}\right)(p_2 x_1 + p_4 x_2) + h.o.t.,$$

where *h.o.t.* denotes higher-order terms. We cannot guarantee negativity of these terms, even locally. The main problem with $\dot{Z}$ is that p_2 must be non-zero for $\dot{Z}$ to be negative definite, yet this is the same coefficient responsible for the cross term of x_1 and x_2 in (20), which prevents us from effectively bounding x_{1_τ} using condition (19) of Theorem 4.2. However, the failure of this quadratic Lyapunov function serves as an instructive example. Namely, we would like a Lyapunov candidate to have the following two properties: (i) it must prevent $\dot{x}_2$'s terms from dominating the Lyapunov derivative, and (ii) the cross terms of x_1 and x_2 in the Lyapunov function should be absent so that delayed terms (like x_{1_τ}) can be easily bounded with non-delayed versions (like x_1) using (19). With these motivations, consider the following Lyapunov-Razumikhin candidate:

$$V(\mathbf{x}) = \frac{d_1}{2}x_1^2 + \frac{d_4}{4}x_2^4 \tag{22}$$

where d_1 and d_4 are positive. In a subsequent discussion, we will specify the values of d_1 and d_4 in terms of system parameters. We will also show that V can be bounded by functions $v(||\mathbf{x}||) = \epsilon_0||\mathbf{x}||_2^2$ and $u(||\mathbf{x}||) = \epsilon_1||\mathbf{x}||_2^4$, for appropriate choices of constants ϵ_0 and ϵ_1, and that these functions satisfy all conditions specified in Theorem 4.2. V satisfies (ii), as necessary, and allows us to choose a convenient function $p(V(\mathbf{x}(t)))$ for (19) (note: we can write $V(\mathbf{x}(t))$ instead of $V(t, \mathbf{x}(t))$ because V is autonomous, *i.e.* it is not explicitly a function of time). Let $p > 1$ be a constant, which can be arbitrarily close to one. Then for (19), we can use $p(V(\mathbf{x}(t))) = pV(\mathbf{x}(t))$:

$$V(\mathbf{x}(t-\theta)) \leq pV(\mathbf{x}(t)), \text{ for } \theta \in [0, \tau].$$

Since there are no cross terms of x_{1_θ} and x_{2_θ} on the left-hand side of the above inequality, bounding the absolute values of these delayed variables individually is straightforward (and instrumental in the proofs that follow).

Now, consider the Lyapunov derivative:

$$\dot{V} = d_1 x_1 \dot{x}_1 + d_4 x_2^3 \dot{x}_2.$$

Note that both of the terms above are now *quartic* in either x_1, x_2, or both. However, $\dot{x}_2$ still contributes a term with x_{1_τ}, which poses a challenge in proving local stability. Indeed, at the core of the proof for $\dot{V}$'s negativity is managing the x_{1_τ} term, as well as proving that h_1 and h_2 in (15), which are higher-order in both the delayed and non-delayed variables, are also higher-order in *only* the non-delayed variables x_1 and x_2.

The focus of the next discussion is the term that contains x_{1_τ}. Substituting the expanded system (15) into $\dot{V}$ and rearranging terms, we have

$$\dot{V} = d_1 x_1\left(-\alpha x_1^3 + \beta x_1^2 x_2 - \gamma x_1 x_2^2 + \delta x_2^3\right) + d_4 x_2^3\left(-\frac{1}{\hat{s}}x_2 - \frac{\hat{s}}{\tau}x_{1_\tau}\right) + d_1 x_1 h_1 + d_4 x_2^3 h_2$$

$$= d_1\left(-\alpha x_1^4 + \beta x_1^3 x_2 - \gamma x_1^2 x_2^2 + \delta x_1 x_2^3\right) - \frac{d_4}{\hat{s}}x_2^4 - d_4\frac{\hat{s}}{\tau}x_2^3 x_{1_\tau} + d_1 x_1 h_1 + d_4 x_2^3 h_2$$

$$= d_1\left(-\alpha x_1^4 + \beta x_1^3 x_2 - \gamma x_1^2 x_2^2\right) - \frac{d_4}{\hat{s}}x_2^4 + d_1\delta x_1 x_2^3 - d_4\frac{\hat{s}}{\tau}x_2^3 x_{1_\tau} + d_1 x_1 h_1 + d_4 x_2^3 h_2$$

11

$$= \mathbf{y}^T Q \mathbf{y} + d_1 \delta x_1 x_2^3 - d_4 \frac{\hat{s}}{\tau} x_2^3 x_{1_\tau} + d_1 x_1 h_1 + d_4 x_2^3 h_2$$

$$\text{where } \mathbf{y} = \begin{bmatrix} x_1^2 \\ x_1 x_2 \\ x_2^2 \end{bmatrix} \text{ and } Q = \begin{bmatrix} -d_1 \alpha & d_1 \beta/2 & 0 \\ d_1 \beta/2 & -d_1 \gamma & 0 \\ 0 & 0 & -d_4/\hat{s} \end{bmatrix}$$

Recall the Mean Value Theorem.

Theorem 4.3 (Mean Value Theorem). *Let $f : [a, b] \to \mathbb{R}$ be a continuous function on the closed interval $[a, b]$, and differentiable on the open interval (a, b), where $a < b$. Then there exists some c in (a, b) such that*

$$f'(c) = \frac{f(b) - f(a)}{b - a}.$$

Let I be the interval $[t - \tau, t]$. Then by the MVT, there exists some $\theta \in (0, \tau)$ such that

$$\dot{x}_1(t - \theta) = \frac{x_1(t) - x_1(t - \tau)}{t - (t - \tau)} = \frac{x_1(t) - x_1(t - \tau)}{\tau},$$
$$x_1(t - \tau) = x_1(t) - \dot{x}_1(t - \theta)\tau, \ \theta \in (0, \tau),$$
$$\text{or } x_{1_\tau} = x_1 - \dot{x}_{1_\theta}\tau,$$

where $\dot{x}_{1_\theta} = \dot{x}_1(t - \theta)$. We would like to combine the terms $d_1 \delta x_1 x_2^3$ and $-d_4 \frac{\hat{s}}{\tau} x_2^3 x_{1_\tau}$ in $\dot{V}$ using the MVT. To do so, let $d_1 = 1/\delta = \hat{s}/c$ and $d_4 = \tau/\hat{s}$.

$$\dot{V} = \mathbf{y}^T Q \mathbf{y} + x_1 x_2^3 - x_2^3 x_{1_\tau} + d_1 x_1 h_1 + d_4 x_2^3 h_2$$
$$= \mathbf{y}^T Q \mathbf{y} + x_2^3(x_1 - x_{1_\tau}) + d_1 x_1 h_1 + d_4 x_2^3 h_2$$
$$= \mathbf{y}^T Q \mathbf{y} + x_2^3 \dot{x}_{1_\theta}\tau + d_1 x_1 h_1 + d_4 x_2^3 h_2$$

Note that the last three terms above all have dependencies on x_{1_τ} and x_{2_τ}. Our goal is to show that these terms are of higher order than $\mathbf{y}^T Q \mathbf{y}$ in variables x_1 and x_2 alone. Consider $\dot{x}_{1_\theta}$:

$$\dot{x}_{1_\theta} = \Phi_\theta^3 \frac{\Psi_{\theta+\tau}}{\tau} \tilde{p}_{\theta+\tau}$$

We would like to find an upper bound for $|\dot{x}_{1_\theta}|$ only in terms of x_{1_θ} and x_{2_θ}. Since $\theta \in (0, \tau)$, we can expand Φ_θ^3 about $\mathbf{v} = [x_{1_\theta} \ x_{2_\theta}] = [0 \ 0]$ (in other words, the fixed point implicitly includes all $x_1(t - \xi)$, $x_2(t - \xi)$, $0 \le \xi \le \tau$, not just $x_1(t)$, $x_2(t)$, $x_1(t - \tau)$, and $x_2(t - \tau)$). Note also that by performing this expansion, we are applying a two-variable Taylor series expansion to a four-variable function. Specifically, we can use a second-order expansion and bound $|\dot{x}_{1_\theta}|$ using only the remainder, which consists of third-order partial derivatives.

In the expressions that follow, we use F and Φ as defined in subsection 4.4. Also, let $\Gamma = \Psi_{\theta+\tau} \tilde{p}_{\theta+\tau}/\tau$. The zero-, first-, and second-order terms in the expansion of $\dot{x}_{1_\theta}$ are zero when evaluated at $\mathbf{v} = \mathbf{0}$. The third-order partial derivatives are:

$$\frac{\partial \dot{x}_{1_\theta}(\mathbf{v} = \mathbf{0})}{\partial x_{1_\theta}^3} = \frac{-2b^3}{9c^2} \Gamma \left[F_\theta^{-2} + 6\Phi_\theta F_\theta^{-7/3} + 5\Phi_\theta^2 F_\theta^{-8/3} \right],$$

$$\frac{\partial \dot{x}_{1_\theta}(\mathbf{v} = \mathbf{0})}{\partial x_{1_\theta}^2 x_{2_\theta}} = \frac{2b^2}{3c} \Gamma \left[F_\theta^{-4/3} + 2\Phi_\theta F_\theta^{-5/3} \right],$$

12

$$\frac{\partial \dot{x}_{1_\theta}(\mathbf{v}=\mathbf{0})}{\partial x_{1_\theta} x_{2_\theta}^2} = -2b\Gamma F_\theta^{-2/3}, \text{ and } \frac{\partial \dot{x}_{1_\theta}(\mathbf{v}=\mathbf{0})}{\partial x_{2_\theta}^3} = 6c\Gamma.$$

We will bound the absolute values of these partial derivatives and use the following proposition [16].

Proposition 4.1. *If a function f is of class C^{k+1} on an open convex set S and $|\partial^\alpha f(\mathbf{x})| \leq M$ for $\mathbf{x} \in S$ and $|\alpha| = k+1$, then the absolute value of the remainder $R_{\mathbf{a},k}(\mathbf{h})$ of the kth-order Taylor series expansion of f about the point $\mathbf{a}$ can be bounded as follows:*

$$|R_{\mathbf{a},k}(\mathbf{h})| \leq \frac{M}{(k+1)!}\|\mathbf{h}\|^{k+1}, \text{ where}$$

$$\|\mathbf{h}\| = |h_1| + |h_2| + \cdots + |h_n|$$

Above, ∂f^α is the generic $(k+1)$th-order partial derivative of f, and $|\alpha| = \alpha_1 + \alpha_2 + \cdots + \alpha_n$. In our case, $\mathbf{a} = \mathbf{0}$, and $\mathbf{h} = [x_{1_\theta} \ x_{2_\theta}]$.

In three of these partial derivatives, there are terms of the form $\left(\frac{b(x_{1_\theta}+\hat{W})}{c}\right)^{-l}$, where l is a positive rational number. Hence, we must bound x_{1_θ} in a region $[-\rho\hat{W}, \rho\hat{W}]$, where $0 < \rho < 1$. Assuming that x_{1_θ}, x_{2_θ}, $x_{1_{\theta+\tau}}$, and $x_{2_{\theta+\tau}}$ are constrained in an appropriately-chosen local region $[-r, r]$ around $\mathbf{0}$, there exists a constant M such that $|\partial^3 f| \leq M$. Then, using the proposition,

$$|\dot{x}_{1_\theta}| \leq \frac{M}{3!}(|x_{1_\theta}| + |x_{2_\theta}|)^3 \tag{23}$$

By Razumikhin's Theorem, we require that $\dot{V}(\mathbf{x}) \leq -w(\|\mathbf{x}\|)$ whenever $pV(\mathbf{x}(t)) \geq V(\mathbf{x}(t-\theta))$, $\theta \in (0, \tau)$, for an $\epsilon > 0$ and some constant $p > 1$. When $pV(\mathbf{x}(t)) \geq V(\mathbf{x}(t-\theta))$,

$$p\left(\frac{d_1}{2}x_1^2 + \frac{d_4}{4}x_2^4\right) \geq \frac{d_1}{2}x_{1_\theta}^2 + \frac{d_4}{4}x_{2_\theta}^4$$

$$p\left(d_1 x_1^2 + \frac{d_4}{2}x_2^4\right) \geq d_1 x_{1_\theta}^2$$

$$|x_{1_\theta}| \leq \sqrt{\frac{p}{d_1}}\sqrt{\left(d_1 x_1^2 + \frac{d_4}{2}x_2^4\right)} \leq \sqrt{\frac{p}{d_1}}\left(\sqrt{d_1 x_1^2} + \sqrt{\frac{d_4}{2}x_2^4}\right) \leq \sqrt{\frac{p}{d_1}}\left(\sqrt{d_1}|x_1| + \sqrt{\frac{d_4}{2}}x_2^2\right)$$

Similarly,

$$p\left(d_1 x_1^2 + \frac{d_4}{2}x_2^4\right) \geq \frac{d_4}{2}x_{2_\theta}^4$$

$$x_{2_\theta}^4 \leq \frac{2p}{d_4}\left(d_1 x_1^2 + \frac{d_4}{2}x_2^4\right)$$

$$|x_{2_\theta}| \leq \sqrt[4]{\frac{2p}{d_4}}\sqrt[4]{\left(d_1 x_1^2 + \frac{d_4}{2}x_2^4\right)} \leq \sqrt[4]{\frac{2p}{d_4}}\left(\sqrt[4]{d_1 x_1^2} + \sqrt[4]{\frac{d_4}{2}x_2^4}\right) \leq \sqrt[4]{\frac{2p}{d_4}}\left(\sqrt[4]{d_1}|x_1|^{1/2} + \sqrt[4]{\frac{d_4}{2}}|x_2|\right)$$

Substituting these results into (23), we have:

$$|\dot{x}_{1_\theta}| \leq \frac{M}{6}\left(\sqrt{\frac{p}{d_1}}\left(\sqrt{d_1}|x_1| + \sqrt{\frac{d_4}{2}}x_2^2\right) + \sqrt[4]{\frac{2p}{d_4}}\left(\sqrt[4]{d_1}|x_1|^{1/2} + \sqrt[4]{\frac{d_4}{2}}|x_2|\right)\right)^3$$

$$= \frac{M}{6}\left(\sqrt{p}|x_1| + \sqrt{\frac{pd_4}{2d_1}}x_2^2 + \sqrt[4]{\frac{2pd_1}{d_4}}|x_1|^{1/2} + \sqrt[4]{p}|x_2|\right)^3$$

The lowest-order term in the equation above is $(|x_1|^{1/2})^3 = |x_1|^{3/2}$. Hence, we see that the term $|x_2^3 \dot{x}_{1_\theta} \tau|$ can be bounded by a function of order at least 4.5. We can use a similar procedure to bound the remainders of $\dot{x}_1$ and $\dot{x}_2$, as we will demonstrate.

First, consider the higher-order terms in $\dot{x}_1$. Since x_{2_θ} depends on $\sqrt{|x_1|}$, we cannot simply use a third-order expansion of $\dot{x}_1$ and bound the remainder of fourth-order partials, because a consequence of this is that the remainder will have a term $\left(|x_1|^{1/2}\right)^4 = |x_1|^2$. Recall that in the Lyapunov derivative, $\dot{x}_1$ is being multiplied by x_1, so the resulting term will have an order of merely three. Using this logic, it is clear that we need an expansion of at least order six; this way, the lowest-order term in the remainder will be $(|x_1|^{1/2})^7 = |x_1|^{7/2}$, and $x_1|x_1|^{7/2}$ is order 4.5, which is sufficient. However, it is not enough to simply do a sixth-order expansion of $\dot{x}_1$: we must also ensure that any fourth-, fifth-, and sixth-order partial derivatives in the expansion in terms of x_1, x_2, x_{1_τ}, and x_{2_τ}, are of order 3.5 or more in terms of only x_1 and x_2 (so that when multiplied by x_1 in the Lyapunov derivative, we have terms of order at least 4.5).

Claim 4.2. *Except for the third-order terms, the sixth-order expansion of $\dot{x}_1$ in $[x_1 \ x_2 \ x_{1_\tau} \ x_{2_\tau}]$ is of combined power at least 3.5 in $[x_1 \ x_2]$.*
Proof: *Consider the un-expanded $\dot{x}_1$ and assume that $\tilde{p}_\tau > 0$ in the region where we are considering this function:*

$$\dot{x}_1 = \left(\Psi - x_1 - \hat{W}\right)\frac{\Psi_\tau}{\tau}\tilde{p}_\tau$$

$$= \frac{c}{\tau}\left(x_2 + \hat{s} - \sqrt[3]{\frac{b(x_1 + \hat{W})}{c}}\right)^3 \left(c\left(x_{2_\tau} + \hat{s} - \sqrt[3]{\frac{b(x_{1_\tau} + \hat{W})}{c}}\right)^3 + x_{1_\tau} + \hat{W} - C\tau\right)$$

Recall that when expanding this function about $[x_1 \ x_2 \ x_{1_\tau} \ x_{2_\tau}] = \mathbf{0}$, the zero-, first-, and second-order terms are zero. The fourth-order terms are:

$$\frac{b^4 x_1^4}{27c^3\hat{s}^{10}} + \frac{cx_2^3 x_{1_\tau}}{\tau} - \frac{2b^3 x_1^3 x_2}{9c^2\hat{s}^8} + \frac{b^2 x_1^2 x_2^2}{3c\hat{s}^6} - \frac{bx_1 x_2^2 x_{1_\tau}}{\hat{s}^2\tau} - \frac{b^3 x_1^3 x_{1_\tau}}{27c^2\hat{s}^6\tau} + \frac{b^2 x_1^2 x_2 x_{1_\tau}}{3c\hat{s}^4\tau}$$

We see that there are no terms that depend on x_{2_τ} above. The terms that contain x_{1_τ} are not problematic: we can take their absolute values and replace $|x_{1_\tau}|$ by an expression that depends on $|x_1|$ and x_2^2 using Razumikhin's Theorem, to obtain an upper-bound for these terms. We conclude that these terms have power at least four in $[x_1 \ x_2]$.

Next, consider the fifth-order terms in the expansion:

$$\frac{13b^4 x_1^4 x_2}{81c^3\hat{s}^{11}} - \frac{8b^5 x_1^5}{243c^4\hat{s}^{13}} - \frac{5b^3 x_1^3 x_2^2}{27c^2\hat{s}^9} + \frac{b^4 x_1^4 x_{1_\tau}}{27c^3\hat{s}^9\tau} - \frac{2b^3 x_1^3 x_2 x_{1_\tau}}{9c^2\hat{s}^7\tau} + \frac{b^2 x_1^2 x_2^2 x_{1_\tau}}{3c\hat{s}^5\tau}$$

Again, there are no terms that depend on x_{2_τ}. Terms that contain x_{2_τ} only begin to show up in the sixth-order partial derivatives evaluated at $\mathbf{0}$. However, these terms contain at most $x_{2_\tau}^3$, since taking the derivative of $\dot{x}_1$ with respect to x_{2_τ} four times yields zero. The rest of the variables in such a term is any cubic combination of x_1, x_2, and x_{1_τ}. Hence, the minimum combined power of such a term (after taking the absolute value and bounding using Razumikhin's) is $3 + 3(1/2) = 3 + 1.5 = 4.5 > 4$ in $[x_1 \ x_2]$.

Finally, we can use the proposition to bound the remainder. For some positive constant M_1,

$$|R_{\mathbf{0},6}| \leq \frac{M_1}{7!}(|x_1| + |x_2| + |x_{1_\tau}| + |x_{2_\tau}|)^7$$

After substituting the expressions for the upper-bounds of $|x_{1_\tau}|$ and $|x_{2_\tau}|$ using Razumikhin's Theorem, the lowest-order term will have $|x_1|^{7/2} = |x_1|^{3.5}$. $\square$

Next, we analyze the higher-order terms and remainder of $\dot{x}_2$. Recall that in the Lyapunov derivative, $\dot{x}_2$ is being multiplied by x_2^3. Hence, we require a second-order expansion of $\dot{x}_2$ about $[x_1 \ x_2 \ x_{1_\tau} \ x_{2_\tau}] = \mathbf{0}$, and we must ensure that the second-order terms are of combined power greater than one.

Claim 4.3. *Except for the first-order terms, the second-order expansion of $\dot{x}_2$ in $[x_1 \ x_2 \ x_{1_\tau} \ x_{2_\tau}]$ is of combined power at least 1.5 in $[x_1 \ x_2]$.*

Proof: *Consider the un-expanded $\dot{x}_2$ and assume that $\tilde{p}_\tau > 0$ in the region where we are considering this function:*

$$\dot{x}_2 = 1 - (x_2 + \hat{s})\frac{\Psi_\tau}{\tau}\tilde{p}_\tau$$

$$= 1 - \frac{(x_2 + \hat{s})}{\tau}\left(c\left(x_{2_\tau} + \hat{s} - \sqrt[3]{\frac{b(x_{1_\tau} + \hat{W})}{c}}\right)^3 + x_{1_\tau} + \hat{W} - C\tau\right)$$

Recall that when expanding this function about $[x_1 \ x_2 \ x_{1_\tau} \ x_{2_\tau}] = \mathbf{0}$, the zero-order term is zero. The second-order terms do not depend on x_{2_τ}. To see this, let $g = \dot{x}_2$ and consider all second-order partial derivatives of g with respect to x_{2_τ}:

$$g_{x_{2_\tau}x_{2_\tau}} = -\frac{(x_2 + \hat{s})}{\tau}(6c)\left(x_{2_\tau} + \hat{s} - \sqrt[3]{\frac{b(x_{1_\tau} + \hat{W})}{c}}\right) \rightarrow g_{x_{2_\tau}x_{2_\tau}}(\mathbf{0}) = 0$$

$$g_{x_2 x_{2_\tau}} = -\frac{3c}{\tau}\left(x_{2_\tau} + \hat{s} - \sqrt[3]{\frac{b(x_{1_\tau} + \hat{W})}{c}}\right)^2 \rightarrow g_{x_2 x_{2_\tau}}(\mathbf{0}) = 0$$

$$g_{x_{1_\tau}x_{2_\tau}} = \frac{(x_2 + \hat{s})}{\tau}(2b)\left(x_{2_\tau} + \hat{s} - \sqrt[3]{\frac{b(x_{1_\tau} + \hat{W})}{c}}\right)\left(\frac{b(x_{1_\tau} + \hat{W})}{c}\right)^{-2/3} \rightarrow g_{x_{1_\tau}x_{2_\tau}}(\mathbf{0}) = 0$$

$$g_{x_1 x_{2_\tau}} = 0$$

Therefore, the second-order terms have combined powers of at least two in $[x_1 \ x_2]$. Next, using the Proposition, we can bound the remainder: for some positive constant M_2,

$$|R_{\mathbf{0},2}| \leq \frac{M_2}{3!}(|x_1| + |x_2| + |x_{1_\tau}| + |x_{2_\tau}|)^3$$

The lowest-order term above is $|x_1|^{3/2} = |x_1|^{1.5}$. $\square$

Finally, we show that the sum of the terms of order four is negative by proving that Q in $\mathbf{y}^T Q \mathbf{y}$ is negative definite.

$$\mathbf{y}^T Q \mathbf{y} = \begin{bmatrix} x_1^2 & x_1 x_2 & x_2^2 \end{bmatrix}\begin{bmatrix} -d_1\alpha & d_1\beta/2 & 0 \\ d_1\beta/2 & -d_1\gamma & 0 \\ 0 & 0 & -d_4/\hat{s} \end{bmatrix}\begin{bmatrix} x_1^2 \\ x_1 x_2 \\ x_2^2 \end{bmatrix}$$

This first leading principal minor of Q, $-d_1\alpha$, is always negative, as needed. The second leading principal minor should be positive:

$$d_1^2\alpha\gamma - d_1^2\frac{\beta^2}{4} \overset{?}{>} 0$$

$$\alpha\gamma - \frac{\beta^2}{4} \overset{?}{>} 0$$

$$\left(\frac{b^3}{27c^2\hat{s}^7}\right)\left(\frac{b}{\hat{s}^3}\right) - \frac{1}{4}\left(\frac{b^2}{3c\hat{s}^5}\right)^2 \overset{?}{>} 0$$

$$\frac{b^4}{27c^2\hat{s}^{10}} - \frac{1}{4}\left(\frac{b^4}{9c^2\hat{s}^{10}}\right) \overset{?}{>} 0$$

$$\frac{1}{27} - \frac{1}{36} \overset{?}{>} 0 \checkmark$$

The third leading principal minor should be negative:

$$-d_1\alpha\left(d_1 d_4 \frac{\gamma}{\hat{s}}\right) - d_1\frac{\beta}{2}\left(d_1\frac{\beta}{2}\left(-\frac{d_4}{\hat{s}}\right)\right) \overset{?}{<} 0$$

$$-\alpha\left(\frac{\gamma}{\hat{s}}\right) - \frac{\beta}{2}\left(\frac{\beta}{2}\left(-\frac{1}{\hat{s}}\right)\right) \overset{?}{<} 0$$

$$-\alpha\gamma + \frac{\beta^2}{4} \overset{?}{<} 0$$

$$\alpha\gamma - \frac{\beta^2}{4} \overset{?}{>} 0$$

We see that this condition is equivalent to the previous one (for the second leading principal minor), and hence it is satisfied.

Finally, it remains to bound V and $\dot{V}$ with functions u, v, and w that satisfy all conditions specified in Theorem 4.2. We note that the arguments above are valid for $|x_1|$, $|x_2| < r$, where $0 < r < 1$. We keep this in mind for the following bounds. Recall that previously, we let $d_1 = \hat{s}/c$ and $d_4 = \tau/\hat{s}$. Then the exact form of the Lyapunov-Razumikhin function is

$$V(\mathbf{x}) = \frac{\hat{s}}{2c}x_1^2 + \frac{\tau}{4\hat{s}}x_2^4.$$

For all bounds, we can use any norm on $\mathbf{x}$, as long as we are being consistent. We choose the l_2-norm. Let

$$v(||\mathbf{x}||) = \epsilon_0||\mathbf{x}||_2^2 = \epsilon_0(x_1^2 + x_2^2),$$

where $\epsilon_0 = \max\left(\frac{\hat{s}}{2c}, \frac{\tau}{4\hat{s}}\right)$. Then $V(\mathbf{x}) \leq v(||\mathbf{x}||)$. Also, $v(||\mathbf{x}||)$ is strictly increasing, $v(||\mathbf{0}||) = 0$, and $v(||\mathbf{x}||)$ is positive for $||\mathbf{x}|| > 0$. Next, let

$$u(||\mathbf{x}||) = \epsilon_1||\mathbf{x}||_2^4 = \epsilon_1(x_1^2 + x_2^2)^2 = \epsilon_1(x_1^4 + 2x_1^2 x_2^2 + x_2^4),$$

where ϵ_1 is a positive constant of our choice. We will show that $V(\mathbf{x}) \geq u(||\mathbf{x}||)$, or equivalently, $V(\mathbf{x}) - u(||\mathbf{x}||) \geq 0$ for some choice of ϵ_1.

$$\begin{aligned}
V(\mathbf{x}) - u(||\mathbf{x}||) &= \frac{\hat{s}}{2c}x_1^2 + \frac{\tau}{4\hat{s}}x_2^4 - \epsilon_1(x_1^4 + 2x_1^2 x_2^2 + x_2^4) \\
&\geq \frac{\hat{s}}{2c}x_1^2 + \frac{\tau}{4\hat{s}}x_2^4 - \epsilon_1(x_1^2 + 2x_1^2 + x_2^4) \\
&= \left(\frac{\hat{s}}{2c} - 3\epsilon_1\right)x_1^2 + \left(\frac{\tau}{4\hat{s}} - \epsilon_1\right)x_2^4 \\
&\geq 0 \text{ for } \epsilon_1 < \min\left(\frac{\hat{s}}{6c}, \frac{\tau}{4\hat{s}}\right).
\end{aligned}$$

Also, $u(||\mathbf{x}||)$ is positive for $||\mathbf{x}|| > 0$ and $u(||\mathbf{0}||) = 0$. Hence, condition (17) is satisfied.

So far, we have shown that for a sufficiently small neighborhood of $\mathbf{x}^* = \mathbf{0}$, the Lyapunov-Razumikhin candidate in (22) has a negative definite derivative, *i.e.* $\dot{V}(\mathbf{x}) < 0$ for all $\mathbf{x} \neq \mathbf{0}$ in this neighborhood and $\dot{V}(\mathbf{x}) = 0$ if $\mathbf{x} = \mathbf{0}$. Next, we show that $\dot{V}$ is bounded by a suitable function w as specified in Theorem 4.2. Recall that we have shown that

$$\dot{V} = d_1 \left(-\alpha x_1^4 + \beta x_1^3 x_2 - \gamma x_1^2 x_2^2 \right) - \frac{d_4}{\hat{s}} x_2^4 + h.o.t.$$

We can express the lower-order terms in matrix form, as follows:

$$\dot{V} = - \begin{bmatrix} x_1^2 & \sqrt{2} x_1 x_2 & x_2^2 \end{bmatrix} \begin{bmatrix} d_1 \alpha & -\frac{d_1 \beta}{2\sqrt{2}} & 0 \\ -\frac{d_1 \beta}{2\sqrt{2}} & \frac{d_1 \gamma}{2} & 0 \\ 0 & 0 & d_4/\hat{s} \end{bmatrix} \begin{bmatrix} x_1^2 \\ \sqrt{2} x_1 x_2 \\ x_2^2 \end{bmatrix} + h.o.t.$$

Let's call the matrix above $\tilde{Q}$. Since we have previously shown that the sum of the lower-order terms in $\dot{V}$ is a negative definite function, it follows that $\tilde{Q}$ is positive definite, so all of its eigenvalues are strictly positive. Since $\tilde{Q}$ is a real and symmetric matrix, its Rayleigh quotient is bounded below by $\lambda_{\min}[\tilde{Q}]$. Hence,

$$\dot{V} \leq -\lambda_{\min}[\tilde{Q}] \left\| \begin{bmatrix} x_1^2 \\ \sqrt{2} x_1 x_2 \\ x_2^2 \end{bmatrix} \right\|_2^2 + h.o.t.$$

$$= -\lambda_{\min}[\tilde{Q}] (x_1^4 + 2 x_1^2 x_2^2 + x_2^4) + h.o.t.$$

$$= -\lambda_{\min}[\tilde{Q}] ||\mathbf{x}||_2^4 + h.o.t.$$

Previously, we showed that the higher-order terms have orders of at least 4.5. Then for $|x_1|$, $|x_2|$ small enough, there exists a positive constant K such that

$$h.o.t. \leq K ||\mathbf{x}||_2^4.$$

Claim 4.4.

$$\text{Let } f(x_1, x_2) = \frac{h.o.t.}{||\mathbf{x}||_2^4}. \text{ Then } \lim_{(x_1, x_2) \to (0,0)} \frac{h.o.t.}{||\mathbf{x}||_2^4} = 0.$$

Proof: *To prove our claim, we will show that for every $\epsilon > 0$, there exists a $\delta > 0$ so that whenever $0 < \sqrt{x_1^2 + x_2^2} < \delta$, $|f(x_1, x_2)| < \epsilon$. The higher-order terms are a sum of terms that have format $c_0 x_1^{c_1} x_2^{c_2}$, where c_0 is either a constant or a function of $x_{1_{\theta+\tau}}$ and $x_{2_{\theta+\tau}}$, as in the case of the higher-order terms that arise from the $x_2^3 \dot{x}_{1_\theta} \tau$ term of $\dot{V}$. We showed that in all cases, $|c_0|$ is bounded above by a positive constant. In addition, we showed that $c_1 + c_2 \geq 4.5$. Let n be the number of higher-order terms (note: n is finite). Then we can write $|f(x_1, x_2)|$ as follows:*

$$|f(x_1, x_2)| = \frac{1}{||\mathbf{x}||_2^4} \left| \sum_{i=1}^{n} c_0^{(i)} x_1^{c_1^{(i)}} x_2^{c_2^{(i)}} \right|,$$

where the superscript (i) corresponds to the ith higher-order term. Clearly,

$$|f(x_1, x_2)| \leq \frac{1}{||\mathbf{x}||_2^4} \sum_{i=1}^{n} \left| c_0^{(i)} x_1^{c_1^{(i)}} x_2^{c_2^{(i)}} \right|.$$

Consider any term $|c_0 x_1^{c_1} x_2^{c_2}|$. Factor out any combination $|x_1|^a |x_2|^b$ such that $a + b = 4$. We know that

$$|x_1|, |x_2| \leq \sqrt{x_1^2 + x_2^2}.$$

Then

$$|x_1|^a |x_2|^b \leq \left(\sqrt{x_1^2 + x_2^2} \right)^4 = ||\mathbf{x}||_2^4,$$

$$|c_0 x_1^{c_1} x_2^{c_2}| \leq |c_0 g(x_1, x_2)| ||\mathbf{x}||_2^4$$

where $g(x_1, x_2)$ is defined s.t. $g(x_1, x_2) x_1^a x_2^b = x_1^{c_1} x_2^{c_2}$. Hence,

$$|f(x_1, x_2)| \leq \frac{1}{||\mathbf{x}||_2^4} \sum_{i=1}^{n} \left| c_0^{(i)} g^{(i)}(x_1, x_2) \right| ||\mathbf{x}||_2^4 = \sum_{i=1}^{n} \left| c_0^{(i)} g^{(i)}(x_1, x_2) \right|.$$

Each function $g(x_1, x_2)$ necessarily has the form

$$g(x_1, x_2) = x_1^{l_1} x_2^{l_2}, \ \ l_1, l_2 \geq 0, \ \ l_1 + l_2 \geq \frac{1}{2}.$$

Hence, we can bound the absolute value of each of these functions by a function of δ.

$$|x_{1/2}| \leq \sqrt{x_1^2 + x_2^2} < \delta,$$

$$|x_{1/2}|^{l_{1/2}} \leq \left(\sqrt{x_1^2 + x_2^2} \right)^{l_{1/2}} < \delta^{l_{1/2}},$$

$$|g(x_1, x_2)| \leq \left(\sqrt{x_1^2 + x_2^2} \right)^{l_1 + l_2} < \delta^{l_1 + l_2}.$$

This gives us

$$|f(x_1, x_2)| \leq \sum_{i=1}^{n} \left| c_0^{(i)} \right| \delta^{l_1^{(i)} + l_2^{(i)}}.$$

We would like the sum above to be less than a given $\epsilon > 0$. We can always find a $\delta > 0$ small enough to make this happen. $\qquad \square$

By Claim 4.4, we can always find a K small enough by restricting x_1 and x_2 into a smaller neighborhood around $\mathbf{0}$. Hence, we can find a $K < \lambda_{\min}[\tilde{Q}]$, which would give us

$$\dot{V} \leq -(\lambda_{\min}[\tilde{Q}] - K) ||\mathbf{x}||_2^4$$

where $\lambda_{\min}[\tilde{Q}] - K > 0$. By inspection, we have found a $w(||\mathbf{x}||)$ that satisfies condition (18) under (19). In addition, $w(||\mathbf{x}||)) > 0$ when $||\mathbf{x}|| > 0$, as necessary. Finally, $\lim_{||\mathbf{x}|| \to \infty} u(||\mathbf{x}||) = \infty$. By Theorem 4.2, we have shown that the function that drives CUBIC's *cwnd*, (7), is locally uniformly asymptotically stable.

Convergence

Using the Lyapunov-Razumikhin function and its derivative, it is possible to explicitly demonstrate the convergence of trajectories to the fixed point. In the analysis below, we assume that $t_0 = 0$. Recall that

$$V(\mathbf{x}) \leq \epsilon_0 ||\mathbf{x}||_2^2$$
$$\rightarrow V^2(\mathbf{x}) \leq \epsilon_0^2 ||\mathbf{x}||_2^4.$$

This gives us

$$\dot{V} \leq -\frac{(\lambda_{\min}[\tilde{Q}] - K)}{\epsilon_0^2} V^2$$
$$\rightarrow \frac{\dot{V}}{V^2} \leq -\frac{(\lambda_{\min}[\tilde{Q}] - K)}{\epsilon_0^2}.$$

We note that

$$\frac{d}{dt}\left(-\frac{1}{V}\right) = \frac{\dot{V}}{V^2} \leq -\frac{(\lambda_{\min}[\tilde{Q}] - K)}{\epsilon_0^2}, \text{ so}$$
$$\frac{d}{dt}\left(\frac{1}{V}\right) \geq \frac{(\lambda_{\min}[\tilde{Q}] - K)}{\epsilon_0^2}.$$

Then, solving the differential inequality,

$$\int_0^t \frac{d}{ds}\left(\frac{1}{V}\right) ds = \frac{1}{V(t)} - \frac{1}{V(0)} \geq \int_0^t \frac{(\lambda_{\min}[\tilde{Q}] - K)}{\epsilon_0^2} ds = \frac{(\lambda_{\min}[\tilde{Q}] - K)}{\epsilon_0^2} t,$$
$$\frac{1}{V(t)} \geq \frac{(\lambda_{\min}[\tilde{Q}] - K)}{\epsilon_0^2} t + \frac{1}{V(0)},$$
$$V(t) \leq \frac{1}{\frac{(\lambda_{\min}[\tilde{Q}] - K)}{\epsilon_0^2} t + \frac{1}{V(0)}}.$$

Since $V(t) \geq \epsilon_1 ||\mathbf{x}||_2^4$,

$$||\mathbf{x}||_2^4 \leq \frac{1}{\frac{\epsilon_1(\lambda_{\min}[\tilde{Q}] - K)}{\epsilon_0^2} t + \frac{\epsilon_1}{V(0)}}. \tag{24}$$

We can simplify this bound using the definition of uniform stability:

Definition 4.1 (Definition 1.1 from Stability of Time-Delay Systems)**.** *For the system described by $\dot{x} = f(t, x_t)$, the trivial solution $x(t) = 0$ is said to be stable if for any $t_0 \in \mathbb{R}$ and any $\epsilon > 0$, there exists a $\delta(t_0, \epsilon) > 0$ such that $||x_{t_0}||_c < \delta$ implies $||x(t)|| < \epsilon$ for $t > t_0$. It is said to be uniformly stable if it is stable and $\delta(t_0, \epsilon)$ can be chosen independently of t_0. It is uniformly asymptotically stable if it is uniformly stable and there exists a $\delta_a > 0$ such that for any $\eta > 0$, there exists a $T(\delta_a, \eta)$, such that $||x_{t_0}||_c < \delta_a$ implies $||x(t)|| < \eta$ for $t \geq t_0 + T$ and $t_0 \in \mathbb{R}$.*

Above, $||\phi||_c = \max_{a \leq \xi \leq b} ||\phi(\xi)||$ for $\phi \in \mathcal{C}[a, b]$ and $x_{t_0} = \phi$ or $x(t_0 + \theta) = \phi(\theta)$, $-\tau \leq \theta \leq 0$. From (24), we have that

$$||\mathbf{x}||_2^4 \leq \frac{1}{\frac{\epsilon_1(\lambda_{\min}[\tilde{Q}] - K)}{\epsilon_0^2} t + \frac{\epsilon_1}{V(0)}} \leq \frac{1}{\frac{\epsilon_1}{V(0)}} = \frac{V(0)}{\epsilon_1} \leq \frac{\epsilon_0 ||\mathbf{x}(t_0)||_2^2}{\epsilon_1} < \frac{\epsilon_0 \delta^2}{\epsilon_1}.$$

We would like

$$||\mathbf{x}||_2 < \epsilon, \text{ or}$$
$$||\mathbf{x}||_2^4 < \epsilon^4.$$

So let

$$\frac{\epsilon_0 \delta^2}{\epsilon_1} < \epsilon^4$$
$$\rightarrow \delta < \epsilon^2 \sqrt{\frac{\epsilon_1}{\epsilon_0}}.$$

This bound on δ provides a measure on the basin of attraction of the fixed point of a system using the system's parameters. *I.e.*, it indicates how close the initial conditions must be to the fixed point in order to guarantee stability. One possible way to apply this bound is in the implementation of TCP: one could specify the initial slow start threshold to be close to $\hat{W}$, so that when CUBIC's congestion avoidance phase begins, the systems is more likely to settle into its stable state.

Summary

For the system described by (12), the following properties hold:

(a) The system has a unique fixed point $\mathbf{x}^* = \mathbf{0}$.

(b) The system has a unique solution in a neighborhood of this fixed point.

(c) The fixed point is locally uniformly asymptotically stable for $\mathbf{x}$ small enough and in addition,

   (i) x_1 and x_{1_τ} are constrained to $[-\rho\hat{W}, \rho\hat{W}]$, $0 < \rho < 1$.
   (ii) $|x_1|, |x_2| < 1$.

(d) The solution is bounded according to (24) for $|x_1|$ and $|x_2|$ small enough.

5   Simulations

We use simulations to validate model (12) and its stability analysis for TCP CUBIC. Our simulation framework treats loss as a non-homogenous Poisson process and generates new loss events based on a user-defined probability of loss model. A detailed description of the framework is provided in the Appendix. An advantage of using this framework for validating the DEs over, for example, NS3, is that we can observe the behavior of solely the congestion avoidance phase of an algorithm, which allows us to more easily verify the theoretical analysis of the controller's stability. Moreover, as we observe from simulations of the DEs, an algorithm's stability can be highly sensitive to the initial conditions specified at the beginning of the congestion avoidance phase. The initial conditions are values of $W_{\max}(0)$ and $s(0)$ for all flows, and we can control them more easily with our simulation framework. This can be especially useful when testing the region of stability for a given system.

Figure 3 compares the average *cwnd* generated by the Non-Homogeneous Poisson Loss (NHPL) simulations against the average value of *cwnd* generated by the DEs. The fixed-point value of *cwnd*, $\hat{W}$, is also shown (albeit sometimes entirely hidden by the DE curve because of fast convergence). All flows in this figure have a per-flow capacity of 1 Gbps, while the round-trip time is varied (these combinations of C and τ are sufficient to generate a diverse set of
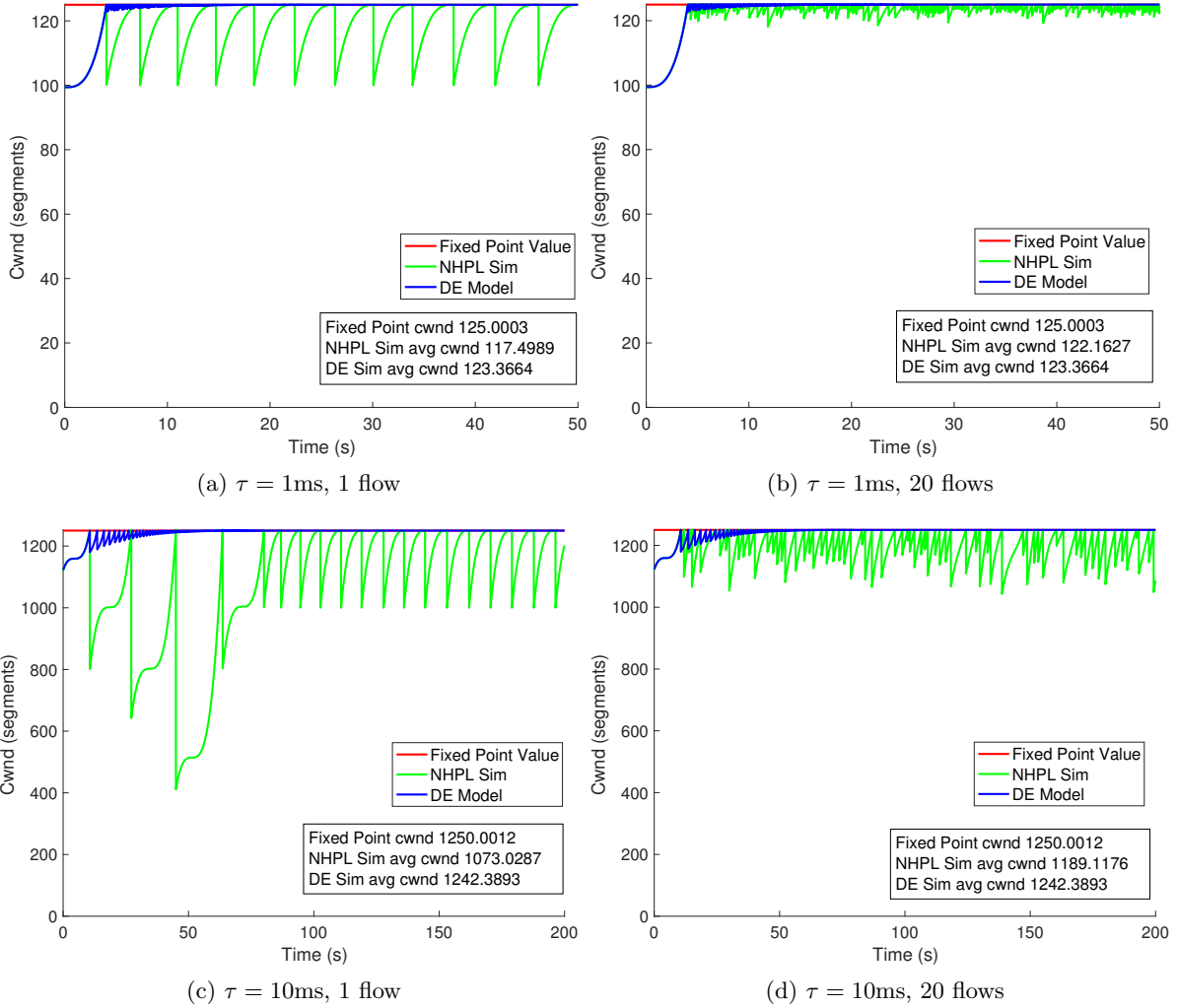
Figure 3: Comparison of average *cwnd* (computed post-transient phase) generated by NHPL simulations against steady-state *cwnd* generated by model (12) for TCP CUBIC. Also shown is the fixed-point value of *cwnd*. Per-flow capacity $C = 1$ Gbps.

behaviors). All flows have $b = 0.2$ and $c = 0.4$ (the default values used in Linux implementations of CUBIC).

Panel 3(a) shows a single stable flow with $\tau = 1$ms. The transient response of both simulations is clearly visible, and we observe that they reach steady-state within a similar period. Not shown in this panel is the value of $\hat{s} \approx 4$s. By observing the time between losses in the NHPL simulation, we see that there is a close agreement. Panel 3(b) shows the same experiment, but with 20 flows. As expected, the average value of *cwnd* from the NHPL simulation approaches $\hat{W}$ as the number of flows is increased. Panels 3(c) and 3(d) show one and 20 flows, respectively, for $\tau = 10$ms. The initial conditions (values of $s(0)$ and $W_{\max}(0)$) are deliberately far enough from the fixed point to demonstrate a more dramatic transient response from both simulations. Figure 4 shows two examples of 100ms flows: in (a), there is a single flow that is stable, while the initial conditions in (b) cause instability for 20 flows in both the DEs and NHPL simulation.

Figure 5 illustrates the transient and steady-state responses of a flow with $C = 100$ Mbps and $\tau = 10$ms, as well $||\mathbf{x}||_2$ as it compares to the convergence bound (24). Observe that $||\mathbf{x}||_2$ is always below the bound and approaches zero as the flow reaches steady state. The bound
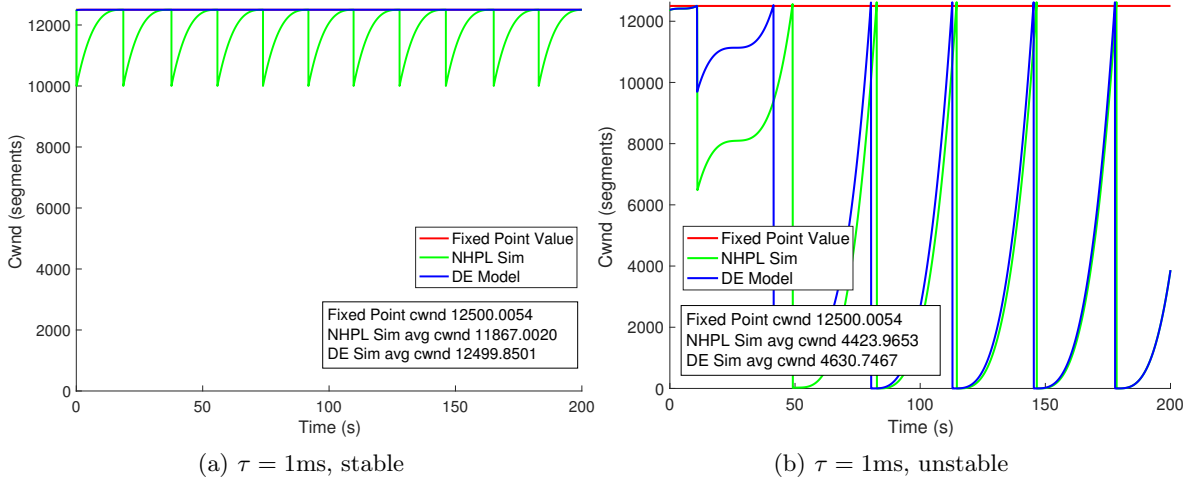
| (a) $\tau = 1$ms, stable | (b) $\tau = 1$ms, unstable |

Figure 4: The impact of initial conditions on stability. For both (a) and (b), $C = 1$ Gbps, $\tau = 100$ms. In (a), there is one flow whose initial conditions $W(0)$ and $s(0)$ are very close to the fixed point values $\hat{W}$ and $\hat{s}$, respectively. Both the NHPL simulation and the model exhibit stability. In (b), there are 20 flows whose initial conditions are set too far from the fixed point values, destabilizing the flows in both the NHPL simulation and the DE system.
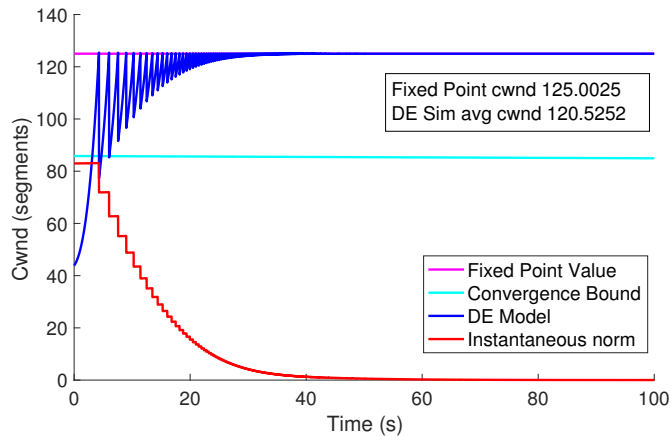

Figure 5: Convergence for CUBIC. At the top is the *cwnd* generated by DEs as it converges to the fixed point value of *cwnd*. Below these two curves is a comparison of $||\mathbf{x}||_2$ against the analytical bound in (24). $C = 100$ Mbps, $\tau = 10$ms.

appears flat in this example because for this system, $V(t_0)$ dominates in the denominator. We observe this phenomenon for many systems; this implies that the initial conditions are crucial for a flow's stability.

6   Conclusion

The main contribution of this work is a novel and versatile fluid model for *cwnd*- and rate-based data transport algorithms. The model is structured so that the differential equations are not dependent on the specific window or rate function of a controller. As a result, this framework offers opportunities to model and analyze the stability of a diverse set of controllers whose window or rate functions may not be linear and whose increase and decrease rules may not be given

in explicit form. We apply this model to two different algorithms: TCP Reno and CUBIC. For the former, we prove that the new model is equivalent to a well-established model for Reno. For the latter, the new model succeeds where traditional methods of modeling *cwnd* are ineffective. We go on to analyze the fluid model for CUBIC and discover that for a given probability of loss model, its window is locally uniformly asymptotically stable. We derive a convergence bound on the solution of the system as a function of the system parameters. Simulations of the model support our theoretical results. As a future direction, we plan to validate the model against a packet-based simulation, as well as analyze the model using alternate loss probability functions.

7 Appendix

We introduce a method of simulating the evolution of a congestion window given $W(t)$ – *cwnd* as a function of time, and $\lambda(t)$ – loss rate as a function of time. We first describe the procedure for generating loss events given arbitrary $W(t)$ and $\lambda(t)$. We then consider a specific loss model and discuss the workarounds necessary when dealing with capacity constraints and time delays. The final result is an algorithm whose pseudocode we present in detail. Finally, we illustrate the operation of the algorithm using an example *cwnd* trajectory.

7.1 Generating Loss Events

We would like to generate inter-loss times given a loss rate function $\lambda(t)$. In order to do so, we apply the Inverse Transform Method on the Poisson distribution, described in the following proposition.

Proposition 7.1. *Suppose a loss event occurs at time t_0. The time to the next loss is given by T where*

$$\int_{t_0}^{t_0+T} \lambda(t)dt = -\ln u,$$

where u is randomly generated from the uniform distribution $U(0,1)$.

Proof. Note that $\lambda(t)$ denotes a Non-Homogeneous Poisson Process, where the number of events between s and t, $N_s(t)$ has a Poisson distribution with parameter $m_s(t) = \int_s^t \lambda(\tau)d\tau$,

$$P(N_s(t) = k) = \frac{m_s(t)^k}{k!}e^{-m_s(t)}.$$

We can then write the CDF of the time from t_0 to the next loss as

$$F_{X_{t_0}}(T) = 1 - P(N_{t_0}(T) = 0) = P(N_{t_0}(T) > 0)$$

$$= 1 - \exp\Big(-\int_{t_0}^{t_0+T} \lambda(t)dt\Big).$$

Note that a CDF can be seen as a random variable with uniform distribution $U(0,1)$, and can be sampled by generating uniform random numbers (this is known as Inverse Transform Sampling). Therefore, inter-loss time samples can be generated as $T = F_{X_{t_0}}^{-1}(u)$. From the above equation we obtain

$$\int_{t_0}^{t_0+T} \lambda(t)dt = -\ln(1-u) \equiv -\ln u,$$

where the last equivalence follows from the fact that if u is uniformly distributed between 0 and 1, so is $1 - u$. ∎

7.2 Delays and Capacity Constraints

In TCP (and most other data transport protocols), the loss rate is a function of the sending rate $W(t)/\tau$ and of a probability of loss model $p(t)$:

$$\lambda(t) = \frac{W(t)p(t)}{\tau}. \tag{25}$$

Therefore, in order to obtain a sample of the time until next loss, the following equation can be solved for T:

$$\frac{1}{\tau} \int_{t_0}^{t_0+T} W(t)p(t)dt = -\ln(u). \tag{26}$$

Note that $W(t)$ and $p(t)$ are viewed from the perspective of the congestion point (*e.g.* a router) where the loss is being generated. Therefore, whenever a loss occurs, the subsequent reduction in the window size (multiplicative decrease) is not reflected in $W(t)$ until after a delay of approximately τ seconds. This is illustrated in Figure 6, which shows an example trajectory of the *cwnd* function. Each time a loss i occurs at time l_i at a congestion point, a corresponding loss indication is reflected in $W(t)$ at time $T_i = l_i + \tau$. The caveat of using (26) to compute T is that $W(t)$ may have changed sometime in the time interval $[t_0, t_0 + T]$ (which can happen if a loss indication is scheduled in this interval; we call this a *pending loss indication* (PLI)). In such a case, the solution is to project the current $W(t)$ until the next loss indication, update $W(t)$ to a new function, and use this new function to generate a new loss event. Once a new loss event is generated, the process may need to repeat until we either produce a loss event that takes place before the next PLI or until we run out of PLIs.

Another complication may arise with certain probability of loss models. For example, in this work we consider the following model:

$$p(t) = \left(1 - \frac{C\tau}{W(t)}\right)^+.$$

As a consequence, $\lambda(t) = 0$ whenever $W(t) < C\tau$. This is depicted in Figure 6, where losses only occur when $W(t) \geq C\tau$. In order to obtain an analytical solution for T during the ith loss event, we can first compute T_{BDP}, the time at which $W(t)$ reaches $C\tau$, or the bandwidth-delay product (BDP). Then, let $t_0 = \max(T_{BDP}, l_i) - T_{i-1}$, where T_{i-1} is the time of the most recent loss indication and l_i is the time of the most recent loss event at the congestion point.

Another feature of the simulation framework is the ability to generate multiple parallel flows. This feature is especially important for validating models that use a system of differential equations to characterize the behavior of congestion control algorithms. The output of such models (*e.g.* *cwnd*) usually describes the behavior of the average flow in a large population of flows. Indeed, in Section 5, we note that the average *cwnd* size from simulation results matches closer to the steady-state value of the DE models as we increase the number of flows in the simulation.

When multiple flows are involved, T_{BDP} is the time at which the sum of their congestion windows reaches the BDP, and l_i is the time at which the most recent loss (across all flows) occurred. We must compute t_0 for each flow, which is given by

$$t_{0,f} = \max(T_{BDP}, l_i) - T_{i-1,f},$$

where $T_{i-1,f}$ is the most recent loss indication of flow f. T is then computed using the following equation:

$$\frac{1}{\tau} \sum_{f=1}^{N} \int_{t_{0,f}}^{t_{0,f}+T} W_f(t)p_f(t)dt = -\ln(u). \tag{27}$$
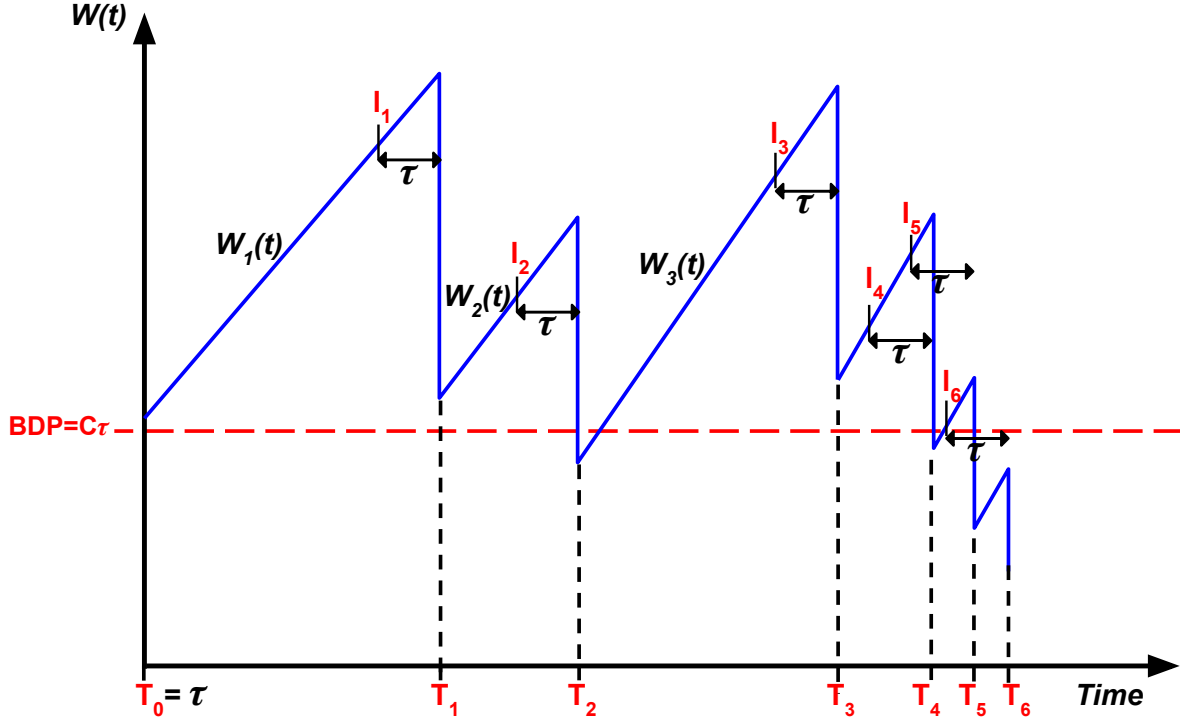
Figure 6: Example trajectory of Reno's congestion window. l_i is the time when loss occurs at the congestion point (*e.g.* router). T_i is the time of the ith loss indication.

Any time a new loss event is generated, we must also choose a flow that will suffer the loss. The flow is picked based on its congestion window size at the time the loss is scheduled to occur (flows with larger windows are more susceptible to suffer a loss).

7.3 Pseudocode

Loss generation can be described by the pseudocode in `GeneratePoiLoss`. This function is called from the main procedure each time a loss is occurring at the congestion point in a given interval. (So, for the example in Figure 6, `GeneratePoiLoss` would be called in the intervals containing the events l_i, $i \in \{1, \ldots, 6\}$.) The arguments of the function are as follows: $pendingLITs$ is a two-dimensional matrix whose first row is a list of pending loss indication times, and whose second row contains the corresponding flows that will suffer the losses. $LLIs$ is an array that keeps record of the last loss indication times of all flows. $GLLI$ is the most recent loss indication. T_l is the time of the most recent loss event. W_{loss} is an array containing the *cwnd* sizes of all flows immediately before their most recent loss events. $p(t)$ is a probability of loss function and τ is the round-trip time.

For the example in Figure 6, where there is only one flow, the procedure outlined in the pseudocode would do the following:

1. At time $t = 0$, a loss occurred at the congestion point (not shown in the figure), so a pending loss indication was scheduled for $T_0 = \tau$.

2. Also at the time of the loss (at $t = 0$), a new loss time was generated using `GeneratePoiLoss`. This loss time is l_1. Since l_1 occurs after the next pending loss indication (which is at T_0), the `while` loop in `GeneratePoiLoss` is triggered. We integrate the *cwnd* function from

25

---

function GENERATEPOILOSS($pendingLITs$, $LLIs$, $GLLI$, T_l, W_{loss}, $p(t)$, τ)

    $\triangleright$ LLT: last loss time at congestion point

    $\triangleright$ $pendingLITs$: a list of pending loss indication times and corresponding flows

    Initialization:

    $GNPLI \leftarrow pendingLITs.nextLossTime$ $\triangleright$ next (global) pending loss indication time

    $LF \leftarrow pendingLITs.nextFlow$ $\triangleright$ the corresponding flow of the next loss event

    $T_{BDP} \leftarrow$ time when sum of $cwnd$'s reaches BDP

    $t_0 \leftarrow \max(T_{BDP}, T_l)$

    $lossTime \leftarrow$ COMPUTET($LLIs, GLLI, W_{loss}, t_0, \tau, p(t)$)

    while $lossTime \geq GNPLI$ **do**

        $\triangleright$ next loss occurs after $GNPLI$, so:

        $\triangleright$ (1) determine the duration of the current congestion epoch for flow LF:

        $I \leftarrow GNPLI - LLIs(f)$

        $\triangleright$ (2) the window function is changed at $GNPLI$, and we are looking at a new

        $\triangleright$ congestion epoch, so update relevant variables

        $W_{loss} \leftarrow W_{LF}(I)$ $\triangleright$ get the W_{loss} value of the next congestion epoch for flow LF

        $GLLI \leftarrow GNPLI$

        $LLIs(LF) \leftarrow GLLI$

        if $NPLI.isEmpty$ **then**

            $NPLI \leftarrow \infty$

        else

            $GNPLI \leftarrow pendingLITs.nextLossTime$

            $LF \leftarrow pendingLITs.nextFlow$

        end if

        $\triangleright$ (3) generate a new loss event at congestion point

        Recompute T_{BDP}

        $t_0 \leftarrow \max(T_{BDP}, GLLI)$

        $lossTime \leftarrow$ COMPUTET($LLIs, GLLI, W_{loss}, t_0, \tau, p(t)$)

    end while

    $\triangleright$ schedule the next loss indication event

    $pendingLITs.add(lossTime + \tau)$

    return ($lossTime, pendingLITs$)

end function

function COMPUTET($LLIs$, $GLLI$, W_{loss}, t_0, τ, $p(t)$)

    $u \leftarrow rand()$ $\triangleright$ generate a number from uniform distr.

    construct $W_f(t)$, $\forall f \in \{1, \ldots, N\}$ using W_{loss}

    $t_{0,f} \leftarrow t_0 - LLIs(f)$, $\forall f \in \{1, \ldots, N\}$

    $\triangleright$ to generate the next loss interval:

    Use Equation (27) to compute T, keep only real, positive roots

    $lossTime \leftarrow GLLI + t_0 + T$

end function

$t = 0$ to $T_0 = \tau$, compute a new W_{loss} (which is the size of the window right before T_0), and feed these values as parameters to `computeT`. The latter function computes the next loss arrival time; this is a new value of l_1. We compare this new l_1 to the next pending loss indication time (which in this case is ∞ since no other pending loss indications have been scheduled after T_0). Since $l_1 < \infty$, we exit the loop and have a new loss time of l_1 and pending loss indication $T_1 = l_1 + \tau$.

3. The main procedure iterates until it reaches the interval containing l_1, at which point `GeneratePoiLoss` is called. The latter function generates l_2, and since l_2 occurs after the next pending loss indication time (which is T_1), we re-generate l_2 using the same procedure as for l_1.

4. The main procedure continues until it reaches the interval containing T_1, at which point the loss indication is processed (the window is halved and there is a new W_{loss}).

5. Loss events l_2 and l_3 and pending loss indications T_2 and T_3 are processed similarly.

6. At loss event l_4, a new loss time l_5 is generated. Since it appears before T_4, we simply schedule a pending loss at T_5 (no need to go through the `while` loop in `GeneratePoiLoss` as we did for the other losses).

7. At loss event l_5, l_6 is generated, but it occurs after the pending loss indication at T_4, which has not been processed yet. Hence, the `while` loop is triggered.

8 Acknowledgment

References

[1] T. Kelly, "Scalable TCP: Improving performance in highspeed wide area networks," *ACM SIGCOMM computer communication Review*, 2003.

[2] S. Ha, I. Rhee, and L. Xu, "CUBIC: a New TCP-Friendly High-Speed TCP Variant," *ACM SIGOPS Operating Systems Review*, 2008.

[3] D. Leith and R. Shorten, "H-TCP: TCP for high-speed and long-distance networks," in *Proceedings of PFLDnet*, 2004.

[4] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "BBR: Congestion-based congestion control," *Queue*, 2016.

[5] Y. Gu and R. L. Grossman, "UDT: UDP-based Data Transfer for High-speed Wide Area Networks," *Computer Networks*, 2007.

[6] R. Pan, B. Prabhakar, and A. Laxmikantha, "QCN: Quantized Congestion Notification," *IEEE802*, 2007.

[7] NS-3 Development Team. NS-3 Network Simulator, https://www.nsnam.org/, Accessed: 2018-01-10.

[8] V. Misra, W.-B. Gong, and D. Towsley, "Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED," *SIGCOMM Comput. Commun. Rev.*, 2000.

[9] F. P. Kelly, A. K. Maulloo, and D. K. Tan, "Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability," *Journal of the Operational Research society*, 1998.

[10] R. Srikant, *The Mathematics of Internet Congestion Control.* Springer Science & Business Media, 2012.

[11] C. Hollot, V. Misra, D. Towsley, and W.-B. Gong, "A Control Theoretic Analysis of RED," in *INFOCOM 2001.*, 2001.

[12] X. Huang, L. Chuang, and R. Fengyuan, "Generalized Modeling and Stability Analysis of Highspeed TCP and Scalable TCP," *IEICE transactions on communications*, 2006.

[13] W. Bao, V. Wong, and V. Leung, "A Model for Steady State Throughput of TCP CUBIC," in *GLOBECOM 2010.*

[14] S. Poojary and V. Sharma, "An Asymptotic Approximation of TCP CUBIC," *arXiv preprint arXiv:1510.08496*, 2015.

[15] K. Gu, J. Chen, and V. L. Kharitonov, *Stability of Time-Delay Systems.* Springer Science & Business Media, 2003.

[16] G. Folland, "Higher-Order Derivatives and Taylor's Formula in Several Variables," https://sites.math.washington.edu/~folland/Math425/taylor2.pdf, 2005, Accessed: 2018-01-10.