

Stability Analysis of RED Gateway with Multiple TCP Reno Connections

Xi Chen*, Siu-Chung Wong*, Chi K. Tse*, and Ljiljana Trajković†

*The Hong Kong Polytechnic University, Hong Kong

†Simon Fraser University, Vancouver, Canada

Abstract—It has been observed that a bottleneck Random Early Detection (RED) gateway becomes oscillatory when regulating a flow in multiple TCP connections. The stability boundary of the TCP-RED system depends on various network parameters, making the adjustment of the RED gateway a difficult task. Based on a fluid-flow model, we formulate analytical conditions that describe the stable boundary of the RED gateway depending on the number of TCP Reno connections. The proposed model accurately generates a stability boundary surface in a four dimensional space, which facilitates the adjustment of parameters for stable operation of the RED gateway. The accuracy of the analytical results has been verified using the *ns-2* network simulations.

I. INTRODUCTION

The TCP/IP protocol suite and the interconnected gateways provide a reliable way for sharing the flow traffic among connections in a bottleneck gateway. In this paper, we analyze the Additive Increase Multiplicative Decrease (AIMD) congestion avoidance algorithm based on TCP Tahoe [1] and TCP Reno [2] implementations. In a typical congestion avoidance, the window size w increases linearly by $1/w$ for each returned acknowledgment (additive increase). Hence, after successfully transmitted a window of packets, w accumulatively increases by 1 within one *round-trip-time* RTT. When there is indication of congestion, detected by receiving three duplicated acknowledgments of a particular packet or by an Explicit Congestion Notification (ECN) header information injected from a gateway and forwarded in the acknowledgment by the receiver, w is reduced to half its current value (multiplicative decrease).

Random Early Detection (RED) [3] gateways were introduced to provide early feedback to the TCP senders by marking (or dropping) packets before the buffer becomes full and, hence, preventing large packet loss. However, TCP-RED may exhibit instability and oscillatory problems [4]–[6], which degrade the transmission rate.

The TCP-RED system is a discrete-time system that can be simulated using the *ns-2* network simulator [7]. While the *ns-2* simulations provides detailed models, simulation may become prohibitive for networks with a large number of TCP flows. Numerical simulations using simplified discrete-time models may be used to simulate a small number of TCP connections [8]. These models, however, may not be easily scaled for a large number of TCP connections [9].

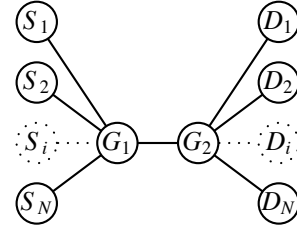


Fig. 1. A simple network with N TCP flows S_i to D_i , where $i = 1, 2, \dots, N$, passing through a common bottleneck link between gateways G_1 and G_2 .

In this paper, we propose a computationally efficient model for the TCP-RED gateway having a relatively large number of connections and derive closed-form stability conditions for the traffic flow.

We first consider a deterministic fluid-flow model where randomness is removed by averaging [4]. We then apply linearization to generate a simple analytical model. Although this linearized model does not capture all features of the original system, it correctly reflects the qualitative behavior of the system with respect to the locations of stability boundaries in the parameter space. This is verified by comparing the stability boundaries derived analytically from the linearized analytical model with those generated from the original fluid-flow model and by the *ns-2* simulations.

II. TCP/RED FLUID-FLOW MODEL

We consider a system of N TCP flows passing through a bottleneck RED gateway [5], as shown in Fig. 1. It is assumed that the system is operating in its desired stable condition [3]. The N long-live TCP connections are controlled by the ECN markings in the RED gateway. The ECN markings are controlled by the RED gateway to maintain an even distribution among the N TCP flows, forcing them to evenly share the bandwidth of the bottleneck link.

The dynamics of the queue can be described using the fluid-model [4]:

$$\frac{dw(t)}{dt} = \frac{1}{r(t)} - \frac{w(t)}{2} \frac{w(t-r(t))}{r(t-r(t))} p(t-r(t)) \quad (1)$$

$$\frac{dq(t)}{dt} = N \frac{w(t)}{r(t)} - C \quad (2)$$

$$\frac{dx(t)}{dt} = C \ln(1-\alpha)(x(t) - q(t)) \quad (3)$$

$$p_b(t) = \begin{cases} 0 & 0 \leq x(t) < X_{\min} \\ \frac{x(t) - X_{\min}}{X_{\max} - X_{\min}} p_{\max} & X_{\min} \leq x(t) \leq X_{\max} \\ p_{\max} + \frac{1-p_{\max}}{X_{\max}}(x(t) - X_{\max}) & X_{\max} < x(t) \leq 2X_{\max} \\ 1 & 2X_{\max} \leq x(t) \leq B \end{cases} \quad (4)$$

$$p(t) = \kappa p_b(t) \quad (5)$$

$$r(t) = \frac{q(t)}{C} + R_0, \quad (6)$$

where

$w(t)$ = averaged instantaneous window size (in packets) of the TCP sources

$r(t)$ = round trip time

$q(t)$ = averaged instantaneous queue length (in packets)

$x(t)$ = filtered queue length after removal of short bursts

$p(t)$ = marking probability

α = filter resolution ($0 < \alpha < 1$)

κ = a proportionality constant dependent on the implementation of the RED algorithm

X_{\max} = maximum threshold of $x(t)$

X_{\min} = minimum threshold of $x(t)$

p_{\max} = maximum threshold of $p(t)$

R_0 = propagation delay

C = bottleneck bandwidth in packets/second

B = maximum physical queue length.

The model has been analyzed using small-signal linearization and perturbation [10], [11]. Our goal is to find conditions when the TCP-RED system becomes unstable. We follow the framework described in [11] and derive the stability boundary for the TCP-RED system.

III. STABILITY BOUNDARY OF TCP-RED SYSTEM

A. Steady-State Solution and Target Queue Length

We assume that the TCP-RED system is stable under a certain control algorithm. In the steady-state, the system converges to an equilibrium point (q_o, r_o, w_o, p_o) , where

$$q_o = \frac{X_{\max} + X_{\min}}{\phi} \quad (7)$$

is the target queue length and

$$r_o = \frac{q_o}{C} + R_0 \quad (8)$$

$$w_o = \frac{C r_o}{N} \quad (9)$$

$$p_o = \kappa p_{\max} \frac{1 + (1 - \phi) \frac{X_{\min}}{X_{\max}}}{\phi \left(1 - \frac{X_{\min}}{X_{\max}}\right)} = 2 \left(\frac{N}{C r_o}\right)^2. \quad (10)$$

The constant ϕ (7) is usually chosen to be equal to 2 to achieve maximum symmetrical swing of the queue length and, hence, minimize the queue length saturation. The steady-state probability p_o (10) is a function of q_o and does not depend on ϕ , X_{\max} , and X_{\min} .

B. Linearization and Perturbation

We linearize (1), (2), and (5) at the equilibrium point and obtain

$$\begin{aligned} \delta w(t) &= \frac{-N}{r_o^2 C} [\delta w(t) + \delta w(t - r_o)] \\ &\quad - \frac{N}{r_o^2 C} [\delta q(t) + \delta q(t - r_o)] \\ &\quad - \frac{r_o C^2}{2N} \delta p(t - r_o) \\ \delta \dot{q}(t) &= \frac{N}{r_o} \delta w(t) - \frac{1}{r_o} \delta q(t) \\ \delta \dot{p}(t) &= C \ln(1 - \alpha) (\delta p(t) - \beta \delta q(t)), \end{aligned} \quad (11)$$

where

$$\begin{aligned} \delta w &= w - w_o \\ \delta q &= q - q_o \\ \delta p &= p - p_o \end{aligned} \quad (12)$$

are the perturbed variables at the equilibrium point and $\beta = \kappa p_{\max} / X_{\max} - X_{\min}$. The characteristic equation of (11) can be obtained by taking Laplace transform:

$$\begin{aligned} 0 &= \lambda^3 + \left(\frac{1}{r_o} + \frac{N}{r_o^2 C} - \alpha_1 C\right) \lambda^2 \\ &\quad + \left(\frac{2N}{r_o^3 C} - \frac{\alpha_1 C}{r_o} - \frac{\alpha_1 N}{r_o^2}\right) \lambda - \frac{2\alpha_1 N}{r_o^3} \\ &\quad + \left(\frac{N}{r_o^2 C} \lambda^2 - \frac{N\alpha_1}{r_o^2} \lambda - \frac{C^3 \alpha_1 \beta}{2N}\right) \exp(-\lambda r_o), \end{aligned} \quad (13)$$

where $\alpha_1 = \ln(1 - \alpha)$. The term $\exp(-\lambda r_o)$ (13) helps determine λ in an approximation to be used. The first-order Padé(0,1) approximation $\exp(-\lambda r_o) \approx \frac{1}{1 + \frac{\lambda r_o}{2}}$ proved inaccurate [11]. Hence, we use Padé(1,1) approximation

$$\exp(-\lambda r_o) \approx \frac{1 - \frac{\lambda r_o}{2}}{1 + \frac{\lambda r_o}{2}}. \quad (14)$$

Using (14), (13) becomes

$$\lambda^4 + a_1 \lambda^3 + a_2 \lambda^2 + a_3 \lambda + a_4 = 0, \quad (15)$$

where

$$\begin{cases} a_1 = \left(\frac{3}{K} - \alpha_1\right) C \\ a_2 = \left(\frac{6N}{K^3} + \frac{2}{K^2} - \frac{3\alpha_1}{K}\right) C^2 \\ a_3 = \left(\frac{4N}{K^4} - \frac{6N\alpha_1}{K^3} - \frac{2\alpha_1}{K^2} + \frac{2\alpha_1 N}{\Phi K^2}\right) C^3 \\ a_4 = -4\alpha_1 N \left(\frac{1}{K^4} + \frac{1}{\Phi K^3}\right) C^4 \\ K = C r_o \\ \Phi = X_{\max} - X_{\min}. \end{cases} \quad (16)$$

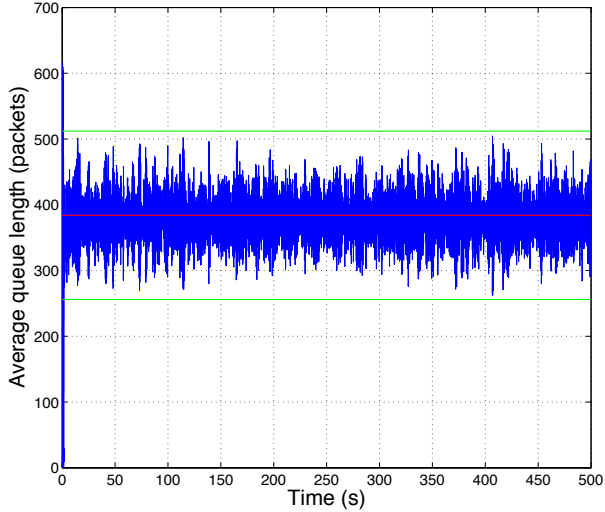


Fig. 2. $ns-2$ simulation results for $P_{\max} = 1/11.4$: $N = 256$, $C = 51.2$ Mbps, $\alpha = 0.01$, $q_o = 384$ packets, and $r_o = 64$ ms.

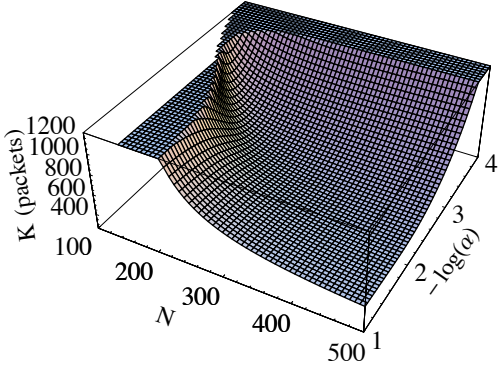


Fig. 3. Stability boundary surface for $\Phi = 256$ packets. Region below the surface is “stable” and region above is “unstable”.

C. Closed-Form Stability Condition

After applying the Routh-Hurwitz stability criterion, the approximate system (15) is locally stable if and only if the network parameters satisfy [11]:

$$\begin{aligned} & \left(\frac{4N}{K^2} - \frac{6N\alpha_1}{K} - 2\alpha_1 + \frac{2\alpha_1 N}{\Phi} \right) \cdot \\ & \left[\left(\frac{3}{K} - \alpha_1 \right) \left(\frac{6N}{K^2} + \frac{2}{K} - 3\alpha_1 \right) \right. \\ & \left. - \left(\frac{4N}{K^3} - \frac{6N\alpha_1}{K^2} - \frac{2\alpha_1}{K} + \frac{2\alpha_1 N}{\Phi K} \right) \right] \\ & + \left(\frac{3}{K} - \alpha_1 \right)^2 4\alpha_1 N \left(\frac{1}{K} + \frac{1}{\Phi} \right) > 0. \end{aligned} \quad (17)$$

The boundary surface (17) is 4-dimensional, involving independent variables N , α_1 , K , and Φ . The 3-dimensional plot of the stability boundary is shown in Fig. 3 for $\Phi = 256$. The area below and above the surface correspond to stable and unstable regions, respectively.

IV. VERIFICATION OF STABILITY BOUNDARIES

We have calculated stability boundaries using the analytical closed-form solution derived from linearizing the fluid-flow model with Padé(1,1) approximation. The results show that the stability boundary changes as system parameters vary. We use the $ns-2$ simulations to compare the original fluid-flow model (without linearization) and the closed-form analytical solution obtained from the linearized fluid-flow model with Padé(1,1) approximation. A typical $ns-2$ simulation run is shown in Fig. 2 for $P_{\max} = 1/11.4$: $N = 256$, $C = 51.2$ Mbps, $\alpha = 0.01$, $q_o = 384$ packets, and $r_o = 64$ ms. Details of other simulation waveforms are omitted due to space limitation. The validity of stability boundaries in various cross-sections corresponding to the 3-D surface shown in Fig. 3 was also verified using the $ns-2$ simulations.

A. Definition of Stability

The approximate system defined by the fluid-flow model is well behaved. The default statistical parameters used in the $ns-2$ simulations are replaced by their expected values. Hence, the stability boundary can be easily identified as in the case of a damped oscillatory system. We consider an $ns-2$ simulation *unstable* if the average queue length cannot be maintained within the boundary defined by X_{\min} and X_{\max} for a target average queue length equal to $(X_{\max} - X_{\min})/2$. The flow is also considered *unstable* if it fails to reach the target average queue length.

It is often observed that for small number of connections, the system may converge at the beginning and then suddenly diverge as the simulation progresses [12]. When the system encounters a sudden unstable oscillation, it becomes saturated (queue lengths above the X_{\max} and below the X_{\min}). The system is then controlled by the fast retransmit and fast recovery algorithms or, even worse, a transmission time-out followed by slow start [1]. It is, thus, important to specify the duration of simulation time when determining stability. The decision regarding a flow’s stability should be based on the occurrence of such instabilities *within the duration of simulation time*.

B. Cross-Sectional Views

Fig. 4 shows the cross-section of Fig. 3 for various α . It is evident that for large α , when the RED gate uses smaller integration time constant, the stochastic nature of individual TCP connections becomes apparent, resulting in larger discrepancy between the $ns-2$ simulation results and the fluid-flow model.

Fig. 5 shows the cross-section of Fig. 3 for various N . Analytical closed-form results and the $ns-2$ simulation results show very good match. When N is small, there is larger discrepancy between the fluid-flow model and the $ns-2$ simulation results (and the approximate closed-form analytical calculations) indicating that the fluid-flow model produces less accurate results, as shown in Fig. 6. Interestingly, the discrepancy is constant ($N = 30$) for this choice of parameters. Indeed, the fluid-flow model is accurate for large number of connections at the TCP gateway.

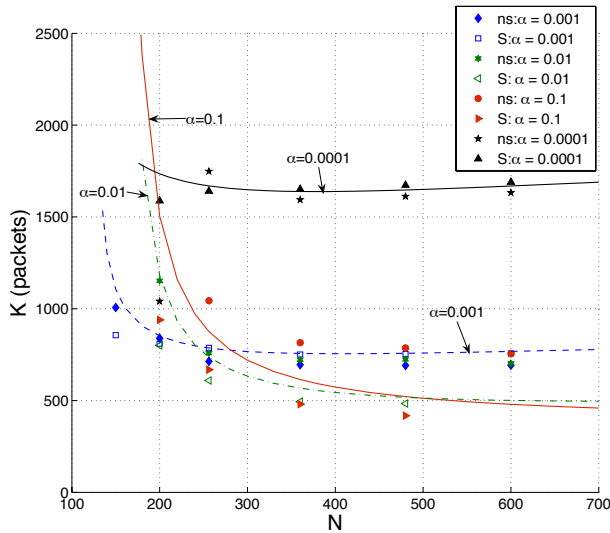


Fig. 4. Comparison of stability boundaries from closed-form solution based on Padé(1,1) linearization (solid and dashed curves) for various α , $\Phi = 256$ packets, and $q_0 = 384$ packets, corresponding to Fig. 3. Points labeled as “S” correspond to full numerical simulation based on the fluid-flow model, and points labeled as “ns-2” correspond to ns-2 simulations. Region below a curve is “stable” and above is “unstable”.

V. CONCLUSION

We have developed an analytical closed-form solution for finding the stability boundary of the TCP-RED system using the fluid-flow model. The solution is very accurate for multiple TCP connections. The simplicity of the solution allows easy and fast generation of the stability boundaries in the essential parameter space. In this paper, we have demonstrated a good match between the model and the ns-2 simulations. The model, which is based on averaging the flows of the multiple connections, becomes less accurate when the number of connections is small. In the case of small number of connections, a discrete-time model would prove to be advantageous [8].

REFERENCES

- [1] V. Jacobson, “Congestion avoidance and control,” *ACM Computer Communication Review*, vol. 18, no. 4, pp. 314–329, Aug. 1988.
- [2] V. Jacobson, “Modified TCP congestion avoidance algorithm,” <ftp://ftp.isi.edu/end2end/end2end-interest-1990.mail>, Apr. 1990.
- [3] S. Floyd and V. Jacobson, “Random early detection gateways for congestion avoidance,” *IEEE/ACM Trans. Networking*, vol. 1, no. 4, pp. 397–413, Aug. 1993.
- [4] 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,” in *Proc. ACM/SIGCOMM*, Stockholm, Sweden, Aug. 2000, pp. 151–160.
- [5] V. Firoiu and M. Borden, “A study of active queue management for congestion control,” in *Proc. IEEE INFOCOM*, Tel Aviv, Israel, Mar. 2000, vol. 3, pp. 1435–1444.
- [6] S. Floyd, R. Gummadi, and S. Shenker, “Adaptive RED: an algorithm for increasing the robustness of RED’s active queue management” [Online]. Available: <http://www.icir.org/floyd/red.html>.
- [7] The network simulator ns-2 [Online]. Available: <http://www.isi.edu/nsnam/ns>.
- [8] M. Liu, A. Marciello, M. di Bernardo, and Lj. Trajković, “Discontinuity-induced bifurcations in TCP/RED communication algorithms,” in *Proc. IEEE Int. Symp. Circ. Syst.*, Kos, Greece, May 2006, pp. 2629–2632.

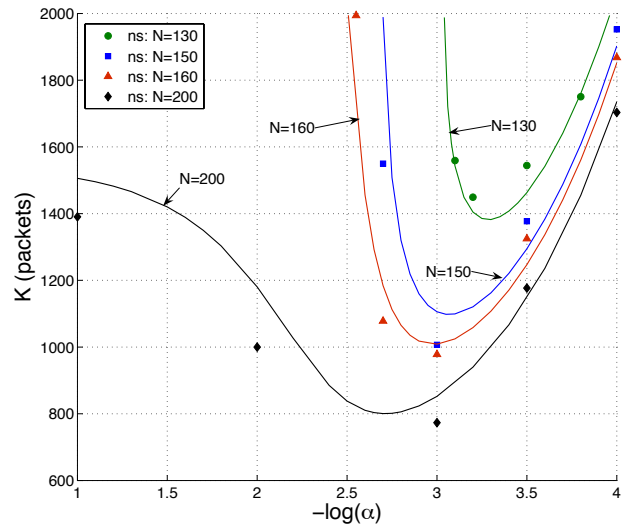


Fig. 5. Comparison of stability boundaries from closed-form solution based on Padé(1,1) linearization (solid curves) for various N and $\Phi = 256$ packets corresponding to Fig. 3. Points labeled as “ns-2” correspond to ns-2 simulations. Region below a curve is “stable” and above is “unstable”.

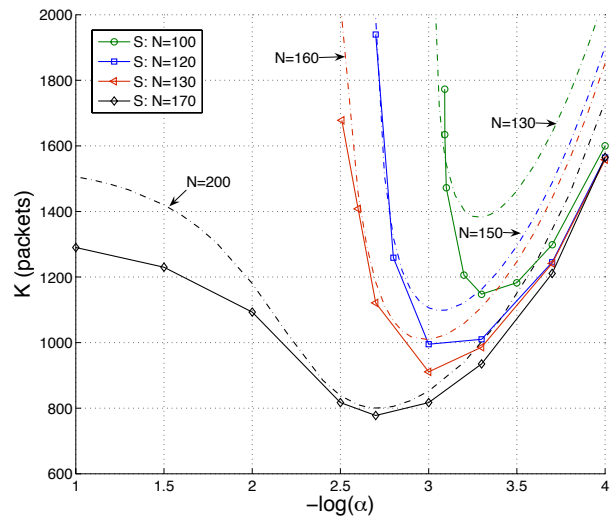


Fig. 6. Comparison of stability boundaries between the closed-form solution based on Padé(1,1) linearization (dashed curves) for various N and $\Phi = 256$ packets corresponding to Fig. 3. The values of N for the full simulations based on the fluid-flow model (lines labeled “S”) is intentionally adjusted to fit the ns-2 simulations of Fig. 5 and to show a constant offset of $N = 30$ of the fluid-flow model from the actual values given by ns-2 simulations. Region below a curve is “stable” and above is “unstable”.

- [9] P. Tinnakornsrisuphap and A. M. Makowski, “Limit behavior of ECN/RED gateways under a large number of TCP flows,” in *Proc. IEEE INFOCOM*, San Francisco, CA, USA, Mar. 2003, pp. 873–883.
- [10] C. V. Hollot, V. Misra, D. Towsley, and W. B. Gong, “A control theoretic analysis of RED,” in *Proc. IEEE INFOCOM*, Anchorage, AK, USA, Apr. 2001, vol. 3, pp. 1510–1519.
- [11] E. C. Park, H. Lim, K. J. Park, and C. H. Choi, “Analysis and design of the virtual rate control algorithm for stabilizing queues in TCP networks,” *Computer Networks*, vol. 44, no. 1, pp. 17–41, Jan. 2004.
- [12] S. C. Wong, C. K. Tse, and F. C. M. Lau, “Characteristic oscillation and intermittency in TCP-RED gateway,” in *Proc. Int. Symp. Nonlinear Theory and its Applications*, Bruges, Belgium, Oct. 2005, pp. 90–93.