The CSI Journal on
Computer Science and Engineering
Vol. 16, No. 1, 2018
Pages 20-31
Regular Paper

A Stochastic Model for Multi-Operational-Mode TCP in High-Speed Networks

Talal A. Edwan

Department of Computer Engineering, Princess Sumaya University for Technology, Amman, Jordan

Abstract

This paper presents a stochastic model for a multi-operational-mode TCP variant optimised to effectively utilise the bandwidth in high-speed-long-delay networks. Particularly, a model for a TCP variant that increases its rate as power function of the current congestion window, and uses multiplicative and subtractive decrease to reduce the congestion window upon packet loss. The power function is adapted dynamically using powers of {0.5, 1, 2} according to the level of congestion in the network. The proposed model can be generalised to other powers; it captures the dynamics of the congestion window size evolution and sending rate under the influence of random packet loss by providing a closed-form expression for both the congestion window size and the normalised sending rate. We show that depending on the increase/decrease rules adopted for the congestion window size evolution, the congestion window size can be modelled either as function of exponential random variable or a Markov chain. Simulation results validate the analytical model and show that the multiplicative decrease factor has more impact on the rate especially when a quadratic – instead of linear – increase rule is adopted under heavy packet loss rate. In addition to that, results show that the initial value of additive increase when a square-root rule is adopted has minimal effect on the average congestion window size.

Keywords: TCP, Congestion control, Markov chain, Performance modelling, Stochastic model, High-Speed networks.

1. Introduction

Since 1981 the widely used Transmission Control Protocol (TCP) went through a series of modifications to meet the changing requirements of computer networks. One of the major modifications appeared in 1988 [1], which defined the congestion avoidance and control. A relatively recent modifications to TCP were made to address the problem of under-utilisation of High-Speed Long-Delay links. In fact, the research community responded by a number of proposals [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12] to solve this problem. We proposed TCP-Gentle [13]; a High-Speed TCP congestion control algorithm that uses multi-mode approach, the algorithm estimates the pipe capacity based on round trip time measurements and uses this information to deduce the number of queued packets along the path. Using this information, TCP-Gentle adopts different strategies for increasing its congestion window and backing it off, particularly, if the queue length is zero it adopts a quadratic increase rule until a predefined congestion window threshold is reached, then it switches to linear increase rule. If packet loss happens it backs off multiplicatively. If the queue length is not zero but below a threshold, TCP-Gentle increases its congestion window according to a square-root rule and backs off when the number of queued packets reaches a predefined threshold which is much less than typical buffer size, hence the name gentle. If packet loss happens in this case, the congestion window is backed off by subtracting the excess amount of packets, that is, the estimated queue length. TCP-Gentle is capable of detecting competing greedy flows– typically TCP Reno– and falls back to traditional TCP Reno operation when that happens. The essence of detection dwells in the fact that when a greedy flow is competing with a TCP-Gentle flow, the queue estimate is always larger than the predefined queue length threshold no matter how many times TCP-Gentle backs off by subtracting the queue estimate from its congestion window, therefore, after several back-offs TCP-

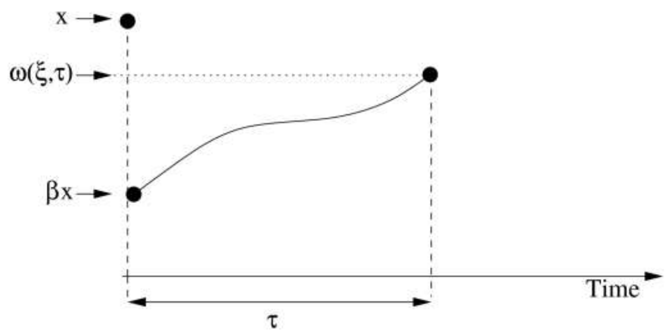
Gentle gives up and assumes the other flow is greedy, thus falls back to TCP-Reno operation.

There has been many attempts to stochastically model TCP and its congestion control algorithms to gain insight and have better understanding of performance issues. Focus was on deriving a closed-form expression for the sending rate as a function of packet loss probabilities. A simple stochastic model for TCP that takes into account the effect of packet loss and timeouts was presented in [14]. In [15] a mean-field and fluid model approach was used to assess the performance of TCP-CUBIC in cloud setting. In [3], [16], [17], [18] a Markov chain model was used to model the behaviour of TCP. In [19] a stochastic model was presented that approximates TCP throughput for several distributions of inter-loss rates. A queueing model approach was used in [20] to provide an expression for the throughputs of multiple TCP-CUBIC and TCP-NewReno connections when sharing the same link. A general theoretical framework for developing analytical models of TCP variants that account for finite buffer size, maximum window constraints, and parallel TCP streams operating in High-Bandwidth-Delay-Product networks was presented in [21], while a comprehensive summary of existing TCP analytical models and their classifications was provided in [22].

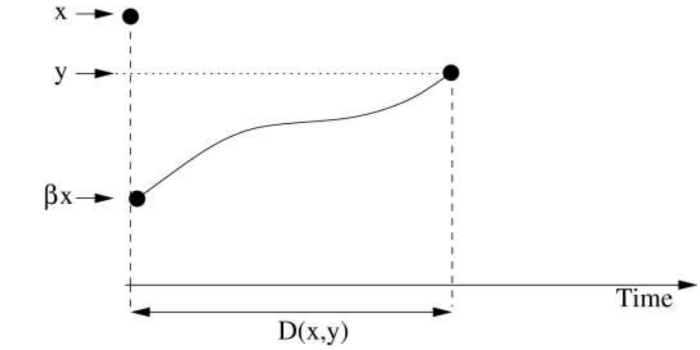
In this paper we provide a stochastic model for TCP-Gentle, we model the congestion window in the different modes of operation and obtain a closed-form expression for the rate. This allows us to study the effect of different parameters on TCP-Gentle's rate under the assumption of Poisson packet loss arrivals to mimic random wireless packet loss events. The contributions of our work are as follows:

1. Provide a model for a TCP variant that increases its rate as power function of the current congestion window, and uses multiplicative and subtractive decrease to reduce the congestion window upon packet loss.
2. Validate the general stochastic model for an example TCP variant (TCP-Gentle).
3. Provide a closed-form expression for TCP-Gentle's rate in each of its operational modes.
4. Analytical and Simulation results show that the initial value of additive increase used when TCP-Gentle is using its square-root rule to increase its congestion window has minimal effect on TCP-Gentles rate.
5. Analytical results show that the quadratic increase rule has slow response to packet loss compared to the linear increase rule when packet loss rates are high. This agrees with the classic TCP Slow Start algorithm [23].
6. Analytical results show that high values of the multiplicative decrease factor have substantial effect on increasing TCP-Gentles rate. The same result was achieved for TCP-CUBIC in literature [16].

The rest of the paper is organised as follows: in Section 2 we define the terminologies used in the system model. In Section 3 we derive an expression for both the congestion window and rate when TCP-Gentle uses its square-root rule to increase its congestion window, while in Section 4 and Section 5 we develop a Markov chain to model the dynamic behaviour of TCP-Gentle's congestion window and to derive an expression for the rate when TCP-Gentle uses its quadratic rule and linear rule, respectively, to increase its congestion window. A special case of the linear rule arises when TCP-Gentle falls back to TCP-Reno operation, this is analysed in Section 6. We discuss the model validation and simulation results in Section 7 and Section 8. Finally, we wrap up this paper with conclusions and discussion of future work in Section 9.



(a) Generic congestion window size evolution.



(b) Elapsed time since last congestion window reduction.

Fig. 1: Illustration of the defined terms.

2. Definition of Terms

TCP Gentle keeps track of the estimated queueing length along the end-to-end path as seen by the sender. The algorithm first estimates this queue length from the sender side, then reacts by entering different modes of operation depending on the estimated value of the queue. In each mode; TCP-Gentle defines a special increase and decrease rules for the congestion window that fits the level of congestion as deduced by the queue length estimate. The queue estimate is done as follows: the sender measures *RTT* base, the minimum *RTT* ever seen by the sender, and also estimates the minimum RTT in a sent congestion window (*cwnd*), RTT_{min}. The sender then estimates the queueing delay as $d_q = RTT_{min} - RTT_{base}$, and therefore the number of enqueued packets Q as $Q = d_q \times (cwnd/RTT_{min})$. Also, TCP-Gentle estimates the path capacity (C_p) as $C_p = RTT_{base} \times cwnd/RTT_{min}$. The value of estimated queueing delay is then compared to a predefined threshold Q_{max} and a decision on how to increase/decrease the congestion window size is made as follows:

$$\text{if } Q > Q_{max}:$$
$$\alpha = \alpha_{reno}$$
$$w(x,\tau) = \alpha\ \tau/RTT + \beta_r x$$
$$\text{if} 0 < Q \le Q_{max}:$$
$$w(\tau) = C_p + A\sqrt{\tau}$$
$$\text{if} Q = 0 \text{ and } \alpha < \alpha_{max}:$$
$$\alpha = \alpha_{gh}$$
$$w(x,\tau) = \alpha\ \tau^2/2\,RTT^2 + \beta_g x$$
$$\text{if} Q = 0 \text{ and } \alpha = \alpha_{max}:$$
$$w(x,\tau) = \alpha\ \tau/RTT + \beta_g x$$

where,

- x: congestion window size just before the last congestion window reduction.
- C_p: estimated path capacity.
- τ: elapsed time from the last congestion window reduction.
- $w(x,\tau)$: congestion window size as a function of x and τ.
- $A = \sqrt{(2\alpha_{gh}Q_{max})/(NP_{max}^{v}RTT)}$.
- Q_{max}, NP_{max}, and v: algorithm parameters set to fixed values.
- α: Additive increase factor.
- α_{max}, α_{gh}, and α_{reno}: different additive increase parameters ($\alpha_{max}=1$).
- β_r and β_g: multiplicative decrease parameters for the Reno and Gentle modes, respectively. Can be set to the same value (i.e. $\beta_r=0.5$).
- RTT: same as RTT_{min}.

Fig. 1a depicts a generic congestion window size evolution according to the previously defined terms, $\omega(\xi,\tau) \equiv w(x,\tau)$. We denote by $D(x,y)$ the elapsed time since the last congestion window reduction. This value can be derived from the congestion window formulae by setting the congestion window size equal to y and solving for τ. Fig.1b shows an illustration. Below are the derived values of $D(x, y)$, note that in the second case (i.e. $0 < Q \le Q_{max}$) the congestion window size is only function of y, since the starting point after a congestion window reduction is always the same, which is the estimated path capacity, C_p.

Depending on the operational mode, TCP-Gentle can have one of the rates, $\{y_q, y_\ell, y_s, y_r\}$ which are the average rates when the congestion window evolves according to the quadratic rule, linear rule, square-root rule, and Reno rule respectively. The switching between modes is random and depends on the level of congestion in the network. However, we note that it is possible for TCP-Gentle flow to remain in one mode, for example, in short-lived flow, TCP-Gentle can remain in the mode that uses the quadratic rule, and in case of long-lived flow in an uncongested network, TCP-Gentle adopts the square root rule in steady state. In the following sections, we formulate and determine each of the aforementioned average rates.

3. Modeling the Square-Root Rule

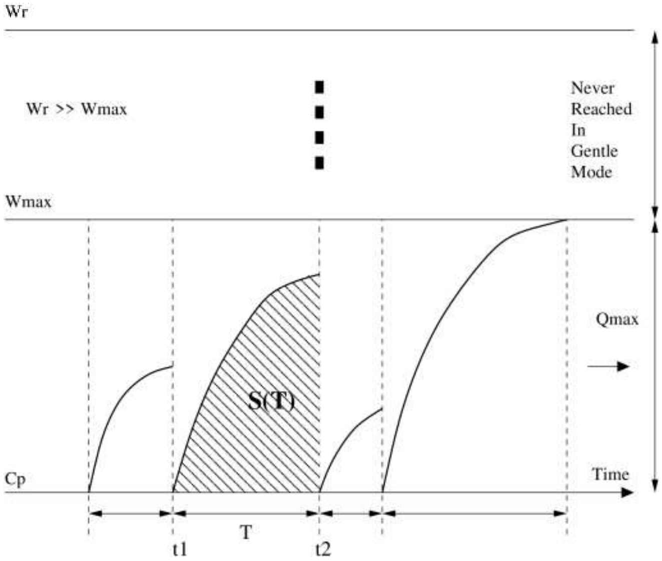


Fig. 2: Congestion window size evolution for the square-root rule.

When TCP-Gentle's queue estimate is in the interval (0, Q_{max}], the congestion window size evolves according to a square-root function of time (see Section 2). The congestion window backs off by subtracting the queue estimate from the congestion window, and this happens in two cases: (i) when a queue estimate threshold (Q_{max}) is reached the congestion window size is W_{max}, this is different from the traditional maximum congestion window reached by TCP-Reno, W_r , in fact it is much less than that value, see Fig.2. When W max is reached, TCP-Gentle reduces the congestion window by Q_{max}, where $Q_{max} = W_{max} - C_p$, thus the value of the congestion window becomes, C_p. (ii) when packet loss happens, TCP-Gentle reduces the congestion window by the current queue estimate, $q(t)$, where $q(t) = w(t) - C_p$, thus the value of the congestion window becomes again, C_p.

We assume that the path capacity (C_p) was estimated and is a constant value, further, we assume that back offs can happen deterministically when a threshold is reached or stochastically according to *Poisson* random packet loss (cases (i) and (ii) respectively). Considering a Poisson random packet loss, the inter-loss time interval can be regarded as an exponential random variable, which we denote by T, Fig. 2 illustrates the congestion window evolution under random packet loss, note that in the last epoch no packet loss happens and the back off is due to the congestion window reaching the threshold W_{max}. In general, we express the density of the exponential random variable T as:

$$f_T(t) = \frac{\lambda exp(-\lambda t)}{exp(-\lambda t_1) - exp(-\lambda t_2)},\ \ t_1 < T < t_2 \tag{1}$$

And since we have,

$$W = C_p + A\sqrt{T} \tag{2}$$

We can derive the distribution of the random variable W as follows:

$$\begin{aligned} F_W(w) &= P[W \leq w] = P\left[\left(C_p + A\sqrt{T}\right) \leq w\right] \\ &= P\left[T \leq \left(\frac{w-C_p}{A}\right)^2\right] \\ &= \frac{exp(-\lambda t_1) - exp\left(-\lambda\left(\frac{w-C_p}{A}\right)^2\right)}{exp(-\lambda t_1) - exp(-\lambda t_2)} \end{aligned} \tag{3}$$

And thus the expected value of W can be obtained by using the following formula for none-negative random variables:

$$\begin{aligned} E[W] &= \int_{C_p}^{W_{max}} P\left[W > w\right] dw \\ &= \int_{C_p}^{W_{max}} (1 - P[W \leq w])\, dw \end{aligned} \tag{4}$$

$$\begin{aligned} &= \int_{C_p}^{W_{max}} \left(1 - \left[\!\left[\frac{\exp(-\lambda t_1) - \exp\left(-\lambda\left(\frac{w-C_p}{A}\right)^2\right)}{\exp(-\lambda t_1) - \exp(-\lambda t_2)}\right]\!\right]\right) dw \\ &= \frac{\sqrt{\pi}\, \mathrm{A}\exp(\lambda(t_1+t_2))\mathrm{erf}\left(\frac{\sqrt{\lambda}(C_p - W_{max})}{A}\right)}{2\sqrt{\lambda}(\exp(\lambda t_1) - \exp(\lambda t_2))} \\ &\quad - \frac{2\sqrt{\lambda}\exp(\lambda t_1)(C_p - W_{max})}{2\sqrt{\lambda}(\exp(\lambda t_1) - \exp(\lambda t_2))} \end{aligned} \tag{5}$$

The rate can be obtained using a similar approach, particularly let the area under the congestion window curve be *S(T)*, then,

$$\begin{aligned} S(T) &= \int_0^T w\,(\tau)d\tau = \int_0^T \left(C_p + A\sqrt{\tau}\right) d\tau \\ &= \left[C_p\tau + \frac{2A}{3}\tau^{3/2}\right]_0^T = C_pT + \frac{2A}{3}T^{3/2} \end{aligned} \tag{6}$$

Now, we define the number of sent packets in the interval (0, *T*], *N (T)* as: *N (T) = S(T)/RTT* , thus, the rate, R_s, becomes:

$$R_s = \frac{N(T)}{T} = \frac{1}{RTT}\left[\!\left[C_p + \frac{2A}{3}\sqrt{T}\right]\!\right] \tag{7}$$

And the distribution of the rate is,

$$\begin{aligned} F_{Rs}(r_s) &= P[R_s \leq r_s] = P\left[\frac{1}{RTT}\left(C_p + \frac{2A}{3}\sqrt{T}\right) \leq r_s\right] \\ &= P\left[\!\left[T \leq \left(\frac{RTT\ r_s - C_p}{2\,A/3}\right)^2\right]\!\right] \\ &= \frac{\exp(-\lambda t_1) - \exp\left(-\lambda\left(\frac{RTT\ r_s - C_p}{2A/3}\right)^2\right)}{\exp(-\lambda t_1) - \exp(-\lambda t_2)} \end{aligned} \tag{8}$$

So, the expected value is:

$$\begin{aligned} \boldsymbol{y_s} &= \boldsymbol{E[R_s]} = \int_{C_p}^{W_{max}} \boldsymbol{P\left[R_s > r_s\right] dr_s} \\ &= \int_{C_p}^{W_{max}} (\boldsymbol{1 - P[R_s \leq r_s]})\, \boldsymbol{dr_s} \\ &= \frac{\sqrt{\pi}\boldsymbol{A exp(\lambda(t_1+t_2))}\mathrm{erf}\left(\frac{3\sqrt{\lambda}(C_p - W_{max})}{2A}\right)}{\boldsymbol{3RTT\sqrt{\lambda}(exp(\lambda t_1) - exp(\lambda t_2))}} \\ &\quad - \frac{\sqrt{\lambda}\boldsymbol{exp(\lambda t_1)(C_p - W_{max})}}{\boldsymbol{RTT\sqrt{\lambda}(exp(\lambda t_1) - exp(\lambda t_2))}} \end{aligned} \tag{9}$$

We can normalise the rate by dividing by the maximum rate, that is,

$$\tilde{y}_s = \frac{y_s}{W_{max}/RTT} \tag{10}$$

4. Modelling the Quadric Rule

It is possible that TCP-Gentle estimates zero queue (i.e. $q(t) = 0$), in this case the congestion window size evolves according to a quadratic rule by adapting an additive increase factor α until the value of α reaches a threshold (α_{max}) then it remains constant, that is, the rule becomes linear. Thus, if$\{q(t) = 0\} \cap \{\alpha < \alpha_{max}\}$ the congestion window evolves according to quadratic rule. Upon packet loss the congestion window is multiplicatively reduced by a factor of β_g.

We follow the quantisation approach mentioned in [16] and let W q be the maximum window size that can be achieved when TCP-Gentle is using the quadratic rule, we divide the interval $(0, W_q]$ into N equally-sized intervals; $((i-1)W_q/N, iW_q/N]$, where, ($i = 1, 2, 3, \ldots, N$). We let each subinterval have a representative point; a i, which is the midpoint of the subinterval, that is, $a_i = (i - 0.5)W_q/N$. If the congestion window size is in the i[th] interval, we regard the congestion window size to be a_i. Observe that we have a finite set of values $\{a_1, a_2, \ldots, a_N\}$ and that we map a continuous range of values in $(0, W_q]$ to this finite set of values, which we regard as states, thus, the congestion window can be in one of these *N* states. Now, we observe the state at each time instant *k* when there is a loss event (congestion window reduction), thus, the k th time instant – where, ($k=1, 2, \ldots$) – corresponds to the k^{th} loss event. We denote by x_k the congestion window size when the k th window reduction is *just about to happen*, thus x_k is in one of the *N* intervals and is mapped to, x_k□, where, x_k□ $\in \{a_1, a_2, \ldots, a_N\}$, so, when x_k□ $= a_i$, we say that the congestion window size is in the i^{th} state at time instant k. Also, we let x_k denote the state of the congestion window size at time instant *k*, thus, $x_k \in \{1, 2, \ldots, N\}$, in other words, if x_k□ $\in ((i-1)W_q/N, iW_q/N]$ then x_k□ $= a_i$ and $x_k = i$. Observe that x_k is a random process.

Let τ_k be a random variable that denotes the time duration between the k^{th} congestion window reduction and $(k+1)^{th}$ congestion window reduction, then we can write:

$$\begin{aligned} x_{k+1} = w(x_k, \tau_k) &= \frac{\alpha_{gh}\,\tau_k^2}{2RTT^2} + \beta_g x_k, \\ \tau_k &\leq D(x_k, W_q) \end{aligned} \tag{11}$$

Observe that x_{k+1} depends on x_k, and not on $x_{k-1}, x_{k-2}, \ldots, x_1$. That is,

$$P(x_{k+1}|x_k, x_{k-1}, \ldots, x_1) = P(x_{k+1}|x_k) \tag{12}$$

so,

$$P(\tilde{x}_{k+1}|\tilde{x}_k, \tilde{x}_{k-1}, \ldots, \tilde{x}_1) = P(\tilde{x}_{k+1}|\tilde{x}_k) \tag{13}$$

and therefore,

$$P(X_{k+1}|X_k, X_{k-1}, \ldots, X_1) = P(X_{k+1}|X_k) \tag{14}$$

Thus, the random process X_k is a *Markov Chain*.

4.1. Transition Probabilities

We are concerned with the stationary-one-step-transition probability from state Xk = i to state Xk+1 = j, which can be expressed as P [Xk+1 = j|Xk = i] ≡ P ij . We define Pij as:

$$P_{ij} = \begin{cases} 0 & \text{, if } j\,W_q/N < \beta_g a_i \\ P\left[\tau_{ij}^{min} < \tau_{loss} \leq \tau_{ij}^{max}\right], & \text{otherwise} \end{cases} \quad (15)$$

With reference to Fig. 3, we note that at the congestion event ($k+1$) the congestion window must be larger than $\beta_g\ a_i$, therefore, $P_{ij} = 0$. This is equivalent of saying, $jW_q /N < \beta_g\ a_i = \beta_g(i - 0.5)W/N$, which implies, $j < \beta_g\ (i - 0.5)$. Also, we note that, given that a loss happened at time instant (k) and the system was in state i, when the next loss happens at time instant ($k+1$) the congestion window size is in the interval $((j-1)W_q /N, jW_q /N]$, which means that the elapsed time since the previous loss to the next loss (the random variable τ_{loss}) is in the interval $(\tau^{min}_{ij}, \tau^{max}_{ij}]$ as illustrated in Fig 3. Therefore we can regard the probability that τ loss be in this interval as P_{ij}.

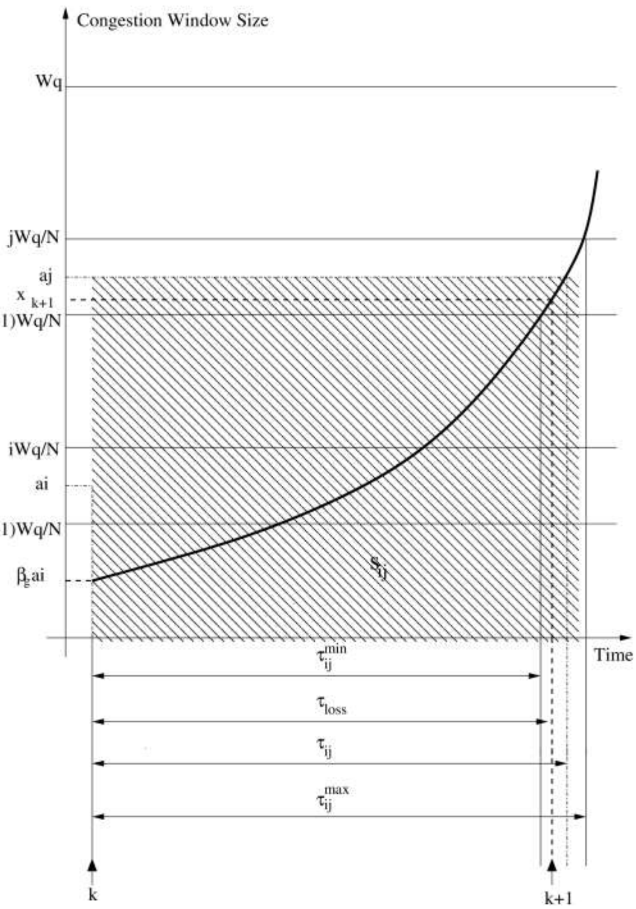


Fig. 3: Congestion window size evolution for the quadratic rule.

When TCP-Gentle is increasing its congestion window according to the quadratic rule; the elapsed time since the last congestion window reduction is given by $D(x, y) = [(y - \beta_g\, x)\ 2RTT/\alpha_{gh}]^{1/2}$, so,

$$\begin{aligned} \tau_{ij}^{min} &= D\left(a_i, (j-1)\,W_q/N\right) \\ &= \sqrt{\left((j-1)\,W_q/N - \beta_g a_i\right)\frac{2RTT^2}{\alpha_{gh}}} \end{aligned} \quad (16)$$

And,

$$\begin{aligned} \tau_{ij}^{max} &= D\left(a_i, j\,W_q/N\right) \\ &= \sqrt{\left(j\,W_q/N - \beta_g a_i\right)\frac{2RTT^2}{\alpha_{gh}}} \end{aligned} \quad (17)$$

Therefore we can write:

$$\begin{aligned} P_{ij} &= P\left[D\left(a_i, (j-1)\,W_q/N\right) < \tau_{loss} \leq D\left(a_i, j\,W_q/N\right)\right] \\ &= exp\left(-\lambda\, D\left(a_i, (j-1)\,W_q/N\right)\right) \\ &\quad - exp\left(-\lambda\, D\left(a_i, j\,W_q/N\right)\right) \end{aligned} \quad (18)$$

Where, $\Sigma_j\, P_{ij} = 1, j \geq \beta_g\ (i - 0.5)$, and $i \in \{1, 2, . . . , N\}$.

4.2. Steady-State Distribution & Rate Expression

Let $\{\pi_1, \pi_2, . . . , \pi_N\}$ be the steady-state probabilities of the Markov Chain, then we can determine these probabilities using $\Sigma_i\, \pi_i P_{ij} = \pi_j$, and $\Sigma_i\, \pi_i=1$, where, $j \in \{1, 2, . . . , N\}$. Furthermore, we know that the elapsed time τ_{ij} lays in the interval $(\tau_{ij}^{min}, \tau_{ij}^{max}]$, but if N is large; we can regard,

$$\tau_{ij} \approx D\left(a_i, (j-0.5)\,W_q/N\right) \quad (19)$$

The area s_{ij} can be calculated as:

$$\begin{aligned} s_{ij} &= \int_0^{\tau_{ij}} w\,(a_i, t)dt \\ &= \int_0^{\tau_{ij}} \left(\frac{\alpha_{gh}}{2RTT^2}\ t^2 + \beta_g a_i\right) dt \\ &= \left[\!\left[\frac{\alpha_{gh}}{6RTT^2}\ t^3 + \beta_g a_i t\right]\!\right]_0^{\tau_{ij}} \\ &= \frac{\alpha_{gh}}{6RTT^2}\tau_{ij}^3 + \beta_g a_i \tau_{ij} \end{aligned} \quad (20)$$

It can be shown that– see for example [16]– the rate can be expressed as:

$$y_q = \frac{1}{RTT}\ \frac{\sum_{i=1}^{N}\sum_{j=1}^{N}\pi_i P_{ij} s_{ij}}{\sum_{i=1}^{N}\sum_{j=1}^{N}\pi_i P_{ij}\tau_{ij}} \quad (21)$$

We can normalise the rate by dividing by the maximum achieved rate, which is in this case W_q/RTT, thus we can write the normalised rate $\tilde{y}_q$ as:

$$\tilde{y}_q = \frac{1}{W_q}\ \frac{\sum_{i=1}^{N}\sum_{j=1}^{N}\pi_i P_{ij} s_{ij}}{\sum_{i=1}^{N}\sum_{j=1}^{N}\pi_i P_{ij}\tau_{ij}} \quad (22)$$

5. Modeling the Linear Rule

When TCP-Gentle estimates zero queue (i.e. $q(t) = 0$),and the adaptive additive increase factor α reaches a threshold (α_{max}) then it remains constant, that is α is upper bounded by α_{max}.

When this happens the congestion window size evolves according to a linear function of time. Thus, if $\{q(t) = 0\} \cap \{\alpha = \alpha_{max}\}$ the congestion window evolves according to linear rule. Upon packet loss the congestion window is multiplicatively reduced by a factor of β_g.

Following the approach mentioned in the previous section, if in this case we let W_ℓ be the maximum window, then it is trivial to show that:

$$x_{k+1} = w(x_k, \tau_k) = \frac{\alpha_{max}}{RTT}\tau_k + \beta_g x_k, \quad \tau_k \le D(x_k, W_\ell) \tag{23}$$

And that x_{k+1} depends on x_k, and not on $x_{k-1}, x_{k-2}, \ldots, x_1$. That is,

$$P(x_{k+1}|x_k, x_{k-1}, \ldots, x_1) = P(x_{k+1}|x_k) \tag{24}$$

so,

$$P(\tilde{x}_{k+1}|\tilde{x}_k, \tilde{x}_{k-1}, \ldots, \tilde{x}_1) = P(\tilde{x}_{k+1}|\tilde{x}_k) \tag{25}$$

and therefore,

$$P(X_{k+1}|X_k, X_{k-1}, \ldots, X_1) = P(X_{k+1}|X_k) \tag{26}$$

Thus, the random process X_k is a *Markov Chain*.

5.1. Transition Probabilities

When TCP-Gentle is increasing its congestion window according to the linear rule; the elapsed time since the last congestion window reduction is given by $D(x, y) = (y - \beta_g x)\, RTT/\alpha_{max}$, so,

$$\tau_{ij}^{min} = D(a_i, (j-1)\, W_\ell/N) = \frac{RTT}{\alpha_{max}}\left((j-1)\, W_\ell/N - \beta_g a_i\right) \tag{27}$$

And,

$$\tau_{ij}^{max} = D(a_i, (j)\, W_\ell/N) = \frac{RTT}{\alpha_{max}}\left((j)\, W_\ell/N - \beta_g a_i\right) \tag{28}$$

Therefore we can write:

$$\begin{aligned} P_{ij} &= P[D(a_i, (j-1)\, W_\ell/N) < \tau_{loss} \le D(a_i, j\, W_\ell/N)] \\ &= exp\left(-\lambda\, D(a_i, (j-1)\, W_\ell/N)\right) \\ &\quad -exp\left(-\lambda\, D(a_i, j\, W_\ell/N)\right) \end{aligned} \tag{29}$$

Where, $\Sigma_j P_{ij} = 1, j \ge \beta_g (i - 0.5)$, and $i \in \{1, 2, \ldots, N\}$. And $P_{ij} = 0$, when $jW_\ell/N < \beta_g a_i$.

5.2. Steady-State Distribution & Rate Expression

Let $\{\pi_1, \pi_2, \ldots, \pi_N\}$ be the steady-state probabilities of the Markov Chain, then we can determine these probabilities using $\Sigma_i \pi_i P_{ij} = \pi_j$, and $\Sigma_i \pi_i = 1$, where, $j \in \{1, 2, \ldots, N\}$. Furthermore, we know that the elapsed time τ_{ij} lays in the interval $(\tau_{ij}^{min}, \tau_{ij}^{max}]$, but if N is large; we can regard,

$$\tau_{ij} \approx D(a_i, (j-0.5)\, W_\ell/N) \tag{30}$$

The area s_{ij} can be calculated as:

$$\begin{aligned} s_{ij} &= \int_0^{\tau_{ij}} w\,(a_i, t)dt \\ &= \int_0^{\tau_{ij}} \left(\frac{\alpha_{max}}{RTT}\, t + \beta_g a_i\right) dt \\ &= \left[\!\left[\frac{\alpha_{max}}{2RTT}\, t^2 + \beta_g a_i t\right]\!\right]_0^{\tau_{ij}} \\ &= \frac{\alpha_{max}}{2RTT}\tau_{ij}^2 + \beta_g a_i \tau_{ij} \end{aligned} \tag{31}$$

Now, the rate can be expressed as:

$$y_\ell = \frac{1}{RTT}\, \frac{\sum_{i=1}^N \sum_{j=1}^N \pi_i P_{ij} s_{ij}}{\sum_{i=1}^N \sum_{j=1}^N \pi_i P_{ij} \tau_{ij}} \tag{32}$$

And the normalised rate, $\tilde{y}_\ell$, can be obtained by dividing by W_ℓ/RTT.

$$\tilde{y}_\ell = \frac{1}{W_\ell}\, \frac{\sum_{i=1}^N \sum_{j=1}^N \pi_i P_{ij} s_{ij}}{\sum_{i=1}^N \sum_{j=1}^N \pi_i P_{ij} \tau_{ij}} \tag{33}$$

6. Modeling the RENO Rule

The Reno rule is a special case of the linear rule, particularly; instead of α_{max} we have $\alpha_{reno} = 1$ and instead of β_g we have $\beta_r = 0.5$. The elapsed time since the last congestion window reduction is thus given by $D(x, y) = (y - \beta_r x)\, RTT/\alpha_{reno}$. Similarly:

$$\tau_{ij} \approx D(a_i, (j-0.5)\, W_r/N) \tag{34}$$

$$\begin{aligned} P_{ij} &= P[D(a_i, (j-1)\, W_r/N) < \tau_{loss} \le D(a_i, j\, W_r/N)] \\ &= exp\left(-\lambda\, D(a_i, (j-1)\, W_r/N)\right) \\ &\quad -exp\left(-\lambda\, D(a_i, j\, W_r/N)\right) \end{aligned} \tag{35}$$

Where, $\Sigma_j P_{ij} = 1, j \ge \beta_r (i - 0.5)$, and $i \in \{1, 2, \ldots, N\}$. And $P_{ij} = 0$, when $jW_r/N < \beta_r a_i$. And the rate is,

$$y_r = \frac{1}{RTT}\, \frac{\sum_{i=1}^N \sum_{j=1}^N \pi_i P_{ij} s_{ij}}{\sum_{i=1}^N \sum_{j=1}^N \pi_i P_{ij} \tau_{ij}} \tag{36}$$

Normalising the rate by dividing by W_r/RTT, where W_r is the maximum congestion window size that can achieved when TCP-Gentle is increasing its congestion window according to the Reno rule.

$$\tilde{y}_r = \frac{1}{W_r}\, \frac{\sum_{i=1}^N \sum_{j=1}^N \pi_i P_{ij} s_{ij}}{\sum_{i=1}^N \sum_{j=1}^N \pi_i P_{ij} \tau_{ij}} \tag{37}$$

7. Simulation Results

In order to validate the analytical results, we used the statistical language/environment "R" [24] to generate the required random variables, and to create a discrete event simulator to simulate the Markov chain. The aims of the simulation are:

1. to study the effect of the parameter α_{gh} on the average congestion window size when TCP-Gentle is increasing its congestion window according to the square-root rule.
2. to study the effect of the parameter α max on the rate when TCP-Gentle is increasing its congestion window according to the quadratic rule.
3. to compare the analytical-steady-state probabilities with the steady-state probabilities obtained from simulation.
4. to study the effect of the multiplicative decrease factor on the average rate under different packet loss rates.

We first layout the set of parameters used in the simulation; unless otherwise stated, TCP-Gentle's parameters are set to the values listed in Tab. 1.
The parameters NP_{max} and v are used to damp the additive increase factor when TCP-Gentle operates in the gentle mode (i.e. increases its congestion window according to a square-root rule), these parameters are tunable and explained in more detail in [13].

7.1. Square-Root Rule

We studied the effect of the parameter α_{gh} on the average congestion window size when TCP-Gentle's uses the square-root rule to increase its congestion window. The algorithm achieves the square-root-shape by using this parameter as an initial additive increase value in this mode, TCP-Gentle then damps this value each round trip time until a steady-state value is reached. We studied the effect of this initial value by comparing the average window size obtained from the analytical model with that obtained from simulation over a range of α_{gh} values from 1 to 100 in increments of 5. The average window size was obtained by generating a random variable according to the distribution in Eq.3. The results are depicted in Fig. 4 when the number of generated samples (n) is 2000 and in Fig. 5 when the number of samples is 10^6. Obviously, when the number of samples is increased the analytical and experimental average values of the congestion window size converge to the same values, however we observe that the effect of the initial value of (α_{gh}) is minimal over most of the values in the range, this can be seen from the small slope of the curves, we observe that this agrees with damping effect that TCP-Gentle uses in this mode.

Tab. 1: TCP-Gentle Parameters

Parameter	Value/Formula
α_{gh}	2 [pkts]
α_{max}	5 [pkts]
β_g	0.5
β_r	0.5
C_p	200 [pkts]
Q_{max}	10 [pkts]
NP_{max}	2
v	2
W_{max}	$C_p + Q_{max}$
W_q	$\alpha_{max} / 2\alpha_{gh}$
W_ℓ	C_p
A	$A = \sqrt{(2\alpha_{gh}Q_{max})/(NP_{max}^{v}RTT)}$

7.2. Steady-State Probabilities

We developed a discrete event simulator to simulate the Markov chains obtained in Section 4 and Section 5. From the simulation we obtained both the steady-state probabilities and the normalised rate, then compared the steady-state probabilities with theoretical values obtained from solving the analytical equations of the chain. Fig.6 illustrates the root-mean-square (RMS) error when the number of states (N) is 500 and 1000 for the quadratic rule case. Fig.7 shows the case for the linear rule. Two points can be noted: first, the RMS error decreases as the simulation time increases, this behaviour is generic in Markov chains, since the more time we run the simulation the more the relative frequencies will represent the probabilities. Second, when the number of states increases, the RMS error decreases, the reason behind this is that the quantisation error becomes smaller as N gets larger, hence, the probabilities obtained by the analytical model and the simulation model get close to each other.

7.2. TCP Gentle's Rate

Building on the results of the previous subsection, we trusted our analytical model and relied on it to study the effect of α_{max} and β – the value of β this mode is the same as β_g – under different packet loss rates when TCP- Gentle is using the

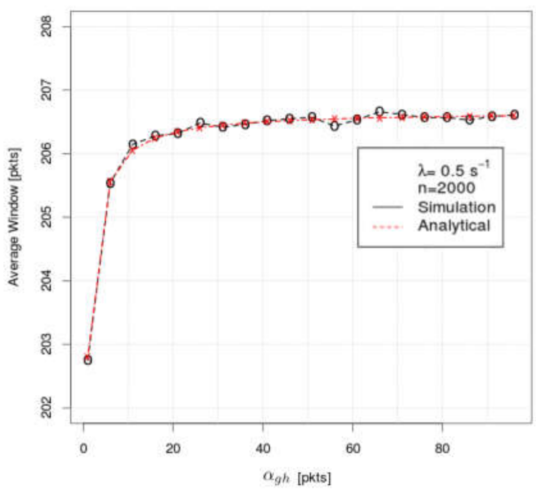


Fig. 4: Average congestion window – small number of samples.

quadratic rule. We used three values of α_{max}: 5, 10, and 20. For each value; we plotted the normalised rate versus a range of β values (0 – 0.95 in increments of 0.05). The results are illustrated in Fig. 8-10. A general observation is that a large value of β results in larger rate, however, a subtle – or or less obvious – observation is that when α max increases, W_q increases – see Tab. 1 – thus the range to which the congestion window can rise, also increases, now, under heavy packet loss, low β and after the congestion window size is reduced, it takes longer to increase the congestion window size using the quadratic rule to the same value just before the loss happened. An extreme case is depicted in Fig. 10 when the packet loss rate $\lambda = 1\ s^{-1}$.

When TCP-Gentle uses the linear rule to increase its congestion window, α_{max} has effect on the rate of additive increase, but not on W_ℓ. We note that despite the fact that $W_\ell > W_q$ see Tab. I – the linear increase rule results in larger rate under heavy packet loss as can be seen in Fig.11.

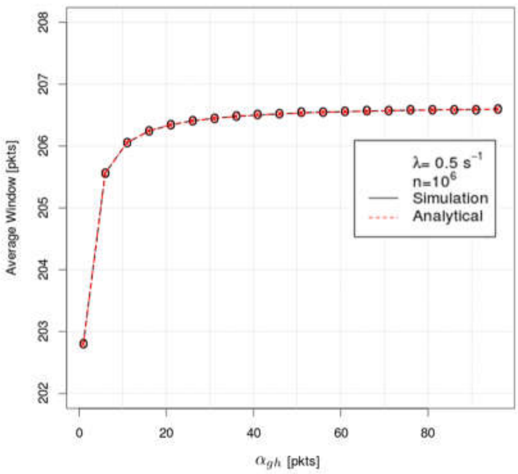


Fig. 5: Average congestion window – large number of samples.

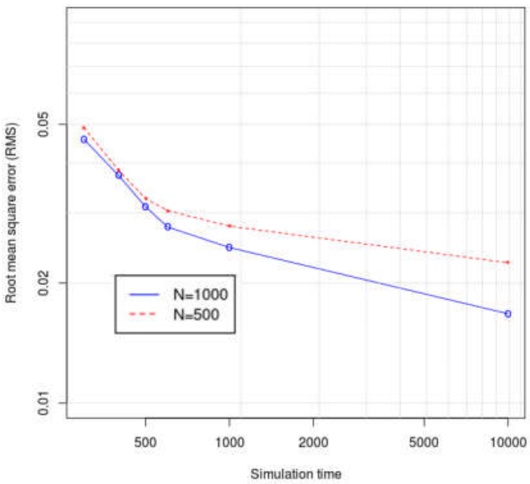


Fig. 6: Root-Mean-Square error of steady-state probabilities for the quadratic rule.

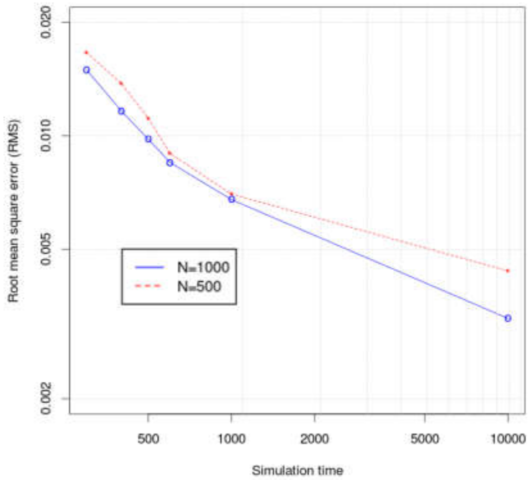


Fig. 7: Root-Mean-Square error of steady-state probabilities for the linear rule.

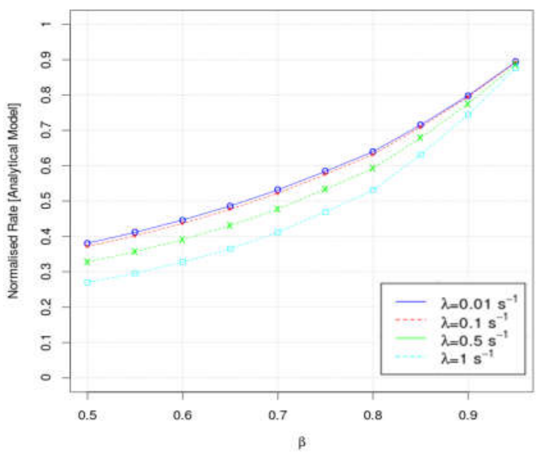


Fig. 8: Normalised rate: quadratic rule (α_{max} = 5 [pkts]).

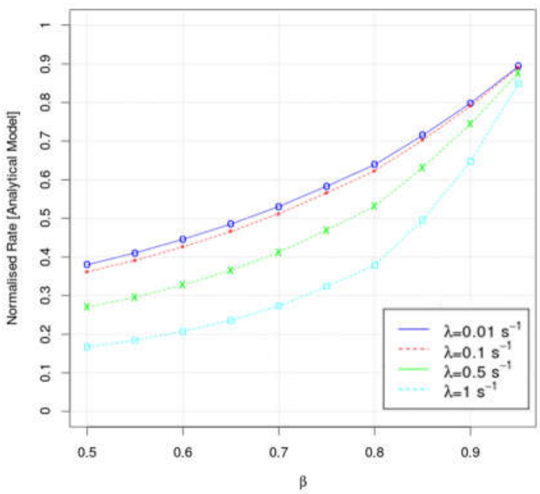


Fig. 9: Normalised rate: quadratic rule (α_{max} = 10 [pkts]).

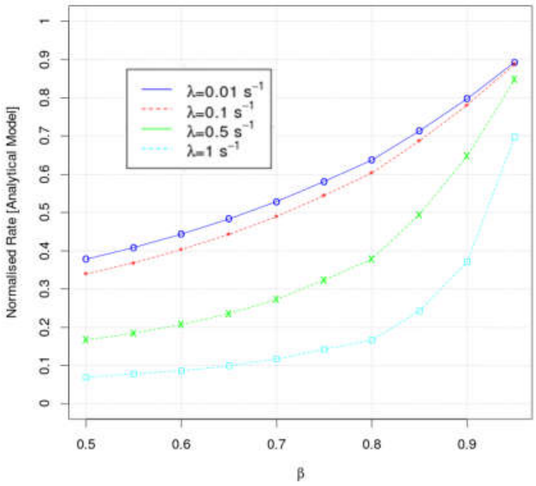


Fig. 10: Normalised rate: quadratic rule (α_{max} = 20 [pkts]).

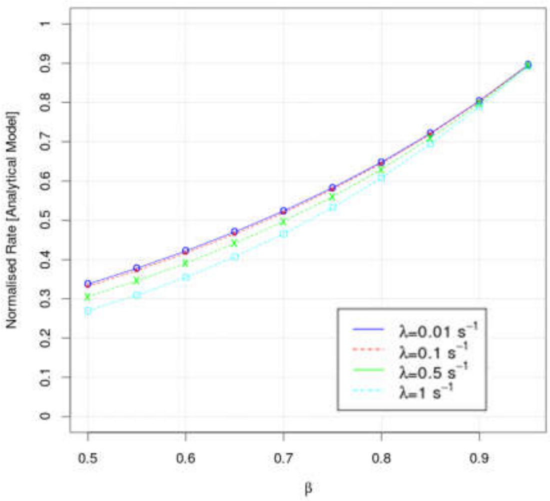


Fig. 11: Normalised rate: quadratic rule (α_{max} = 20 [pkts]).

8. Performance Evaluation of TCP Gentle

We implemented TCP-Gentle and used `TCP/Linux` patch for `ns2` simulator [25]. The following set of TCP-Gentle's parameters was used: $\alpha_{gh} = 2$, $\alpha_{max} = 20$, $NP_{max} = 4.5$, $v = 2$, and the same queue size threshold in TCP-YeAH (80 packets), to be able to compare TCP-Gentle and TCP- YeAH under the same conditions. The choice of 80 packets gives reasonable performance for TCP-YeAH in a high-speed long-delay environment [12].

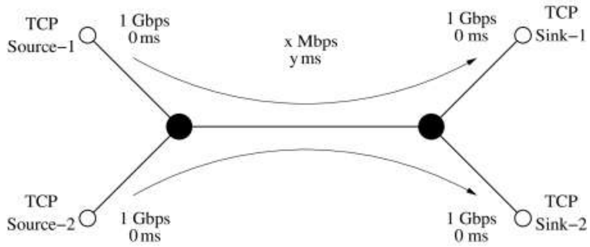


Fig. 12: Simulation topology.

We used the simple topology illustrated in Fig. 12 and let two TCP flows use the same bottleneck link which has a First Come First Serve (FCFS) queue policy. Two different bottleneck capacities were used: 100 Mbs and 300 Mbps in most of the experiments. Other parameters are set as follows: fixed packet size of 1000 bytes, fixed buffer size of 5% of BDP, and a total propagation delay of 46 ms – roughly an RTT of 100 ms.

8.1 Friendliness to TCP-NewReno

We compared TCP-Gentle's congestion control algorithm to the legacy TCP congestion control (TCP NewReno). Initially, we selected a suitable metric for the study. Jain's fairness index indicates whether all flows receive a fair share or not (when the index is equal to one it means that all flows equally share the resource). Or if "*k*" out of "*n*" flows

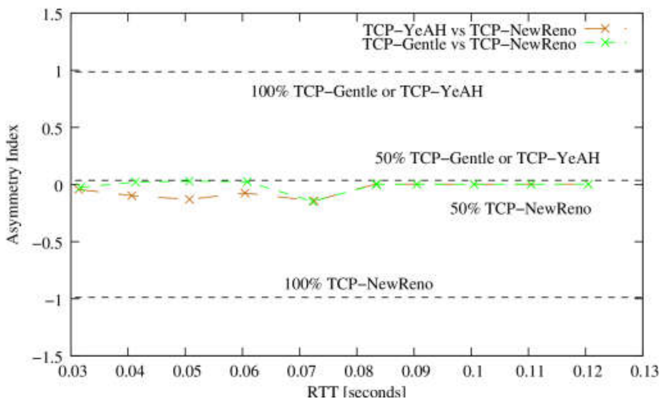


Fig. 13: Asymmetry index.

equally share the resource and the rest have zero allocation, the index is "*k/n*", thus the worst case is (i.e. one flow utilise the resource and the rest of flows have zero allocation) "*1/n*". However, Jain's index does not give information about the amount of share for each flow. Alternatively, we found that the *Asymmetry Index* [26], gives more information about the amount of share of each competing flow for our two-flow study. The Asymmetry Index is defined as: $A = (X_1-X_2)/(X_1+X_2)$, where X_i is the average throughput and $i = \{1, 2\}$ is the number of flows. The range of the index is [-1,1], with zero indicating a fair share.

We used two flows and varied the bottleneck link propagation delay in the interval [10 ms, 60 ms] in increments of 5 ms. For each run; one flow uses TCP-Gentle or TCP-YeAH and the other flow uses TCP-NewReno. As it can be seen in Fig. 13, TCP-Gentle additive increase rule is more friendly to TCP-NewReno, this is indicated by the approximately zero values of the Asymmetry index.

8.2 Response Function

In order to test TCP-Gentle in a real environment, we emulated a high-speed long-delay link, and tested the proposed algorithm on it. Fig. 14 depicts the topology of the test bed, it consists of four Linux boxes, one of which is used as a router running dummynet, the other three PCs running `Iperf` (two TCP senders and a receiver). All PCs have two processors each is Intel(R) Pentium(R) 4 CPU clock frequency 3.20GHz and 1GB RAM. The network interface cards used are all 1 Gigabit Ethernet. All PCs have TCP-Gentle enabled on the TCP senders. We used `tcpprobe` module to probe the congestion window and the slow start threshold values. A bottleneck capacity of 100 Mbps and a packet size of 1500 bytes were used in all of our experiments in this test bed, in addition to that we changed the queue size threshold (i.e. Q_{max}) to 50 packets and disabled slow start algorithm which is implemented separately in the Linux kernel, so we can compare TCP-CUBIC, TCP-YeAH, and TCP-Gentle under the same conditions.
We studied the response of TCP-CUBIC, TCP-YeAH, and TCP-Gentle to a range of random packet loss probabilities. In our experiments, we considered empirical Response Function. In the setup shown in Fig. 14 we used a single flow running one of the algorithms, a bottleneck capacity of 100 Mbps, a RTT of 100 ms and used the uniform packet-loss

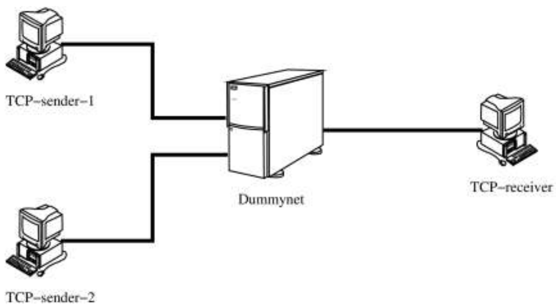


Fig. 14: Test bed topology.

generator provided by `dummynet`. The range of packet loss probabilities was $[10^{-08}, 10^{-01}]$ and the duration of run for each experiment was 5 minutes. For each probability in this range the average throughput was reported.

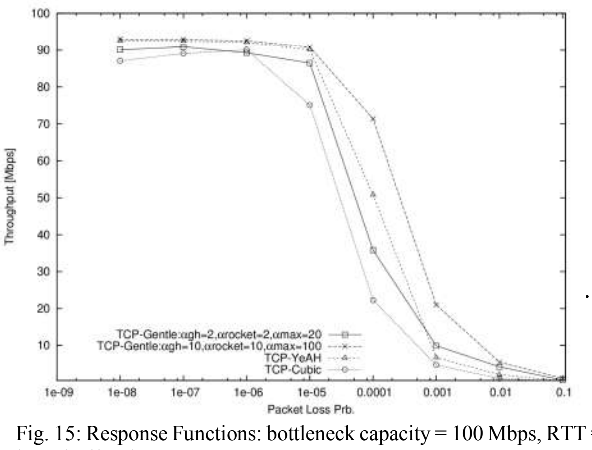


Fig. 15: Response Functions: bottleneck capacity = 100 Mbps, RTT = 100 ms, large buffer size.

Fig. 15 depicts a plot of the average throughput versus packet loss probability for the algorithms. The response of TCP-Gentle is very close to that of TCP-YeAH. When we tried different values for TCP-Gentle parameters, we noticed a better response, this is indicated by increase in throughput at higher loss rates, however the price for that is indeed more aggressiveness in the additive increase rule. Since our aim is to reduce network load as much as possible, we would trade a tolerable decrease in responsiveness (response to packet loss) for reducing network load.

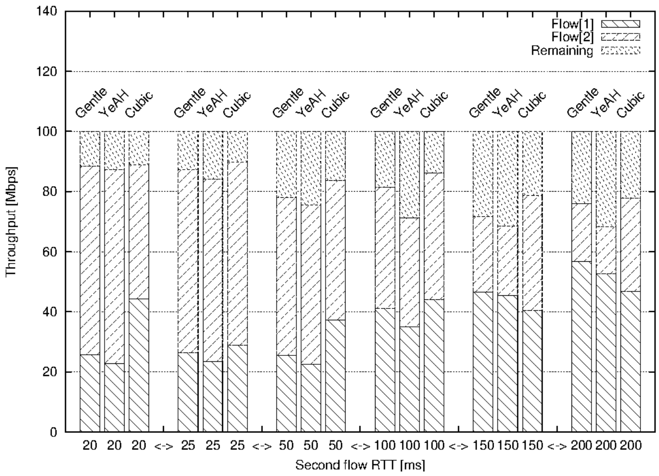


Fig. 16: Algorithm Fairness & RTT Unfairness.

8.3 Intra-Protocol Fairness & RTT Unfairness

Intra-protocol fairness is the fairness between flows running the same algorithm. We studied the intra-protocol fairness of TCP-Gentle compared to TCP-YeAH and TCP-CUBIC. The experiments consists of two flows running the same algorithm and competing on the same bottleneck which has a capacity of 100 Mbps. We also studied RTT unfairness by varying the RTT of the second flow while fixing the RTT of the first flow. The RTT of the first flow was fixed at 100 ms, while the RTT of the second flow was varied in steps of 5 ms, as follows: 20 ms, 25 ms, 50 ms, 100 ms, 150 ms, 200 ms, respectively for each experiment. Note that when the second flow has a RTT of 100ms, both flows have the same RTT, thus the experiment can be used to study intra-protocol fairness. The average throughput after multiple runs was reported.

Referring to Fig. 16, all algorithms underutilised the pipe capacity, this is indicated by the upper shaded areas. This is due to time-outs. TCP-YeAH and TCP-Gentle exhibited clear RTT unfairness as the RTT of the second flow was varied, we also noted that the unfairness increases as the as the difference in RTT increases, however, TCP-Gentle outperformed TCP-YeAH, we blame the multiplicative increase rule of TCP-YeAH for that. TCP-CUBIC was very immune against RTTunfairness this is due to its increase rule being not a function of RTT, instead, it is a function of real time, in fact the algorithm was optimised to achieve this goal. When both flows had the same RTT, TCP-Gentle performed better than TCP-YeAH and its performance was similar to that of TCP-

CUBIC. We noted that TCP-YeAH substantially underutilised the pipe capacity compared to the other two algorithms in almost all of the experiments.

9. Conclusions

In this paper we derived a stochastic model for a TCP variant that increases its rate as power function of the current congestion window, and uses multiplicative and subtractive decrease to reduce the congestion window upon packet loss. The power of the function changes dynamically according the the level of congestion in the network. We used TCP-Gentle as example of this TCP variant and analysed its main parameters under random packet loss to mimics packet loss in wireless links. The stochastic model captures the dynamics of the different modes of operation of TCP-Gentle under this type of packet loss. We have shown that when TCP-Gentle uses its quadratic or linear rules to increase its congestion window and backs off multiplicatively upon packet loss, a Markov chain models the dynamics of the congestion window. However, when TCP-Gentle uses its square-root rule to increase its congestion window and backs off subtractively by the queue estimate upon packet loss or when the threshold (W_{max}) is reached, a random variable – whose distribution was derived – models the dynamics of the congestion window.

The model was validated by means of simulation, then was used to study the effect of TCP-Gentle parameters (α_{gh} , α_{max} , and β) on the rate under several levels of packet loss rates. Results showed that the initial value (α_{gh}) that TCP-Gentle uses at the beginning of its damping phase has minimal effect on the average congestion window size when the square-root rule is adopted. On the other hand, large values of α max reduce the average rate of TCP-Gentle when it is operating its quadratic congestion window increase rule. Finally, the multiplicative decrease factor β has substantial effect on TCP-Gentle's rate, particularly, we found that large values of β result in large rates. Our future work aims at studying the effect of multiple flows by using a network model, and modelling the random switching between TCP-Gentle's modes for the single-flow case.

References

[1] V. Jacobson, "Congestion avoidance and control," SIGCOMM Comput.Commun. Rev., vol. 18, no. 4, pp. 314–329, Aug. 1988. [Online]. Available:http://doi.acm.org/10.1145/52325.52356

[2] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "Bbr: Congestion-based congestion control," Queue, vol. 14, no. 5, pp. 50:20–50:53, Oct. 2016. [Online]. Available:http://doi.acm.org/10.1145/3012426.3022184

[3] S. Liu, T. Başar, and R. Srikant, "TCP-Illinois: a loss and delay-based congestion control algorithm for high-speed networks," in valuetools '06: Proceedings of the 1 st international conference on Performance evaluation methodolgies and tools. New York, NY, USA: ACM, 2006, p. 55.

[4] D. Leith and R. Shorten, "H-tcp: Tcp for high-speed and long-distance networks," in in Proc. PFLDnet, Argonne, 2004. [Online]. Available:http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.3.7816

[5] S. Ha, I. Rhee, and L. Xu, "CUBIC: a new TCP-friendly High-Apeed TCP variant," SIGOPS Oper. Syst. Rev., vol. 42, no. 5, pp. 64–74, 2008.

[6] S. Floyd, "HighSpeed TCP for Large Congestion Windows," Dec. 2003. [Online]. Available:http://www.ietf.org/rfc/rfc3649.txt

[7] T. Kelly, "Scalable TCP: Improving Performance in Highspeed Wide Area Networks," Apr. 2003. [Online]. Available:http://www.deneholme.net/tom/scalable/

[8] R. King, R. G. Baraniuk, and R. H. Riedi, "TCP-Africa: an adaptive and fair rapid increase rule for scalable TCP." in INFOCOM. IEEE, 2005, pp. 1838–1848. [Online]. Available:http://dblp.unitrier.de/db/conf/infocom/infocom2005.html#KingBR05

[9] S. H. David X. Wei, Jin Cheng; Low and H. Sanjay, "FAST TCP: motivation, architecture, algorithms, performance," IEEE/ACM Trans. Netw., vol. 14, no. 6, pp. 1246–1259, 2006.

[10] L. S. Brakmo, S. W. O'Malley, and L. L. Peterson, "TCP Vegas: new techniques for congestion detection and avoidance," SIGCOMM Comput. Commun. Rev., vol. 24, no. 4, pp. 24–35, 1994.

[11] K. Tan, J. Song, Q. Zhang, and M. Sridharan, "A compound TCP approach for high-speed and long distance networks," Microsoft Research, Tech. Rep. MSR-TR-2005-86, Jul. 2005.

[12] A.Baiocchi, A. Castellani, and F. Vacirca, "YeAH- TCP: Yet Another Highspeed TCP," in Fifth International Workshop on Protocols for FAST Long-Distance Networks (PFLDnet-07), February 2007, pp. 37–42. [Online]. Available:http://wil.cs.caltech.edu/pfldnet2007/paper/YeAHTCP.pdf

[13] T. A.Edwan, "Improved algorithms for TCP congestion control," Ph.D. dissertation, Computer Science, 2010. [Online]. Available: https://dspace.lboro.ac.uk/2134/7141

[14] J. Padhye, V. Firoiu, D. Towsley, and J. Krusoe, "Modeling TCP throughput:A simple model and its empirical validation," Proceedings of the ACM SIGCOMM '98 conference on Applications, technologies, architectures, and protocols for computer communication, pp. 303–314, 1998. [Online]. Available: citeseer.ist.psu.edu/padhye98modeling.html

[15] S. Belhareth, L. Sassatelli, D. Collange, D. Lopez-Pacheco, and G. Urvoy-Keller, "Understanding tcp cubic performance in the cloud: A mean-field approach," in 2013 IEEE 2nd International Conference on Cloud Networking (CloudNet), Nov 2013, pp. 190–194.

[16] W. Bao, V. W. S. Wong, and V. C. M. Leung, "A model for steady state throughput of TCP CUBIC," in 2010 IEEE Global Telecommunications Conference GLOBECOM 2010, Dec 2010, pp. 1–6.

[17] S. Fortin-Parisi and B. Sericola, "A markov model of tcp throughput, goodput and slow start," Performance Evaluation, vol. 58, no. 2, pp. 89 – 108, 2004, distributed Systems Performance. [Online].Available:http://www.sciencedirect.com/science/article/pii/S0166531604000860

[18] C. Casetti and M. Meo, "A new approach to model the stationary behavior of TCP connections," in Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064), vol. 1, 2000, pp. 367–375 vol.1.

[19] E. Altman, K. Avrachenkov, and C. Barakat, "A stochastic model of TCP/IP with stationary random losses," IEEE/ACM Trans. Netw., vol. 13, no. 2, pp. 356–369, 2005.

[20] S. Poojary and V. Sharma, "Analytical model for congestion control and throughput with TCP CUBIC connections," in 2011 IEEE Global Telecommunications Conference - GLOBECOM 2011, Dec 2011, pp. 1–6.

[21] G. Vardoyan, N. S. V. Rao, and D. Towsley, "Models of TCP in high-BDP environments and their experimental validation," in 2016 IEEE 24th International Conference on Network Protocols (ICNP), Nov 2016, pp. 1–10.

[22] J. Olsén, "Stochastic modeling and simulation of the tcp protocol," Ph.D. dissertation, Avdelningen för matematisk statistik, 2003.

[23] M. Allman, V. Paxson, and W. Stevens, "TCP congestion control," Internet Requests for Comments, RFC Editor, RFC 2581, April 1999.

[24] R Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, 2017. [Online]. Available: https://www.R-project.org/
[25] O. Source, "The network simulator - ns-2," Jan 2008, http://www.isi.edu/nsnam/ns/.
[26] R. L. C. Hadrien Bullot and R. Hughes-Jones, "Evaluation of advanced tcp stacks on fast long-distance production networks," Journal of Grid Computing, vol. Volume 1, no. 4, pp. 345–359, Dec. 2003.

Talal A.Edwan main work has been in analysis of network performance evaluation, protocols and algorithms. He received a BSc in Electronics Engineering with a First Class Honours from Princess Sumaya University for Technology (Jordan) in 2002, MSc in Networks Engineering with Distinction from University of Plymouth (UK) in 2005, and a PhD in Computer Networks from Loughborough University (UK) in 2010. Dr Talal is an Assistant Professor in the Department of Computer Engineering at PSUT since 2012, his research interests are: Computer Networks, Networked Embedded Systems, Network Congestion Control, Queueing Theory, and Performance Evaluation of Computer Systems/Networks.

Email: t.edwan@psut.edut.jo

Paper Handling Data:

Submitted: 31.07.2018
Received in revised form: 15.10.2018
Accepted: 17.10.2018
Corresponding author: Dr. Talal A.Edwan
Department of Computer Engineering, Princess Sumaya University for Technology, Amman, Jordan