This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE Globecom 2010 proceedings.

A Model for Steady State Throughput of TCP CUBIC

Wei Bao, Vincent W.S. Wong, and Victor C.M. Leung
Department of Electrical and Computer Engineering
The University of British Columbia, Vancouver, Canada
e-mail: {weib, vincentw, vleung}@ece.ubc.ca

***Abstract*— For transmission control protocol (TCP), CUBIC is a TCP-friendly high-speed variant, in which the window size is a cubic function of time since the last loss event. TCP CUBIC is implemented in Linux operating systems and performs well in wired networks with large bandwidth-delay product. Most of the evaluations of TCP CUBIC are conducted via simulations or experiments. Analytical models for TCP CUBIC are few. In this paper, we propose a Markovian model to determine the steady state throughput of TCP CUBIC in wireless environment. The proposed model considers both congestion loss and random packet loss due to fading. We derive the stationary distribution of the Markov chain and obtain the average throughput based on the stationary distribution. Simulations are carried out to validate the analytical model. Results show that the simulated stationary distribution and the average throughput are both very close to our analytical results. Furthermore, we analyze the throughput performance of TCP CUBIC. Results show that random packet loss reduces the normalized average throughput more for end-to-end flow with large bandwidth-delay product. We propose an improvement to increase the throughput performance of TCP CUBIC by moderately increasing the window growth factor and the multiplicative decrease factor.**

I. Introduction

The transmission control protocol (TCP) is one of the core protocols of the Internet protocol (IP) suite. It provides reliable end-to-end connections in the Internet. The TCP congestion control mechanism enables the sender to adjust the *transmission rate* (or equivalently the *congestion window size*) according to the network conditions dynamically.

There are many variations of TCP congestion control mechanisms proposed in the literature. Some of the TCP congestion control protocols which have been deployed in the current Internet include TCP Reno, New Reno [1], and SACK. In these protocols, the window size is reduced if there is a loss event (e.g., three duplicate acknowledgement (ACK), timeout). In some other TCP protocols (e.g., TCP Vegas [2]), the transmission rate is adjusted based on the measured round trip time (RTT) (or queueing link in intermediate links).

There are TCP protocols which are designed specifically for long-distance, high latency links. Examples include Fast TCP [3], BIC (Binary Increase Congestion control) TCP [4], and TCP CUBIC [5]. TCP CUBIC is implemented and used by default in Linux kernels 2.6.19 version. It combines additive increase and binary search increase to achieve good scalability as well as RTT fairness. TCP CUBIC achieves some improvements on TCP BIC. It performs well in wired networks with large bandwidth-delay product. It simplifies the BIC window control and improves TCP-friendliness. In addition, the window growth function of TCP CUBIC is defined in real-time instead of RTT, so that the window growth rate is independent of RTT, which guarantees the RTT fairness.

There have been various analytical models proposed to analyze the performance of TCP in the literature. In [6], Padhye *et al.* developed a simple analytical model to determine the steady state throughput, which is a function of the loss rate and RTT. In [7], Mathis *et al.* proposed a periodic loss model to evaluate the TCP throughput. In [8], Baccelli *et al.* analyzed a scenario where a large number of TCP flows go through a bottleneck router which uses the tail drop policy. In [9], Misra *et al.* analyzed the stationary behavior of the TCP congestion window by using a continuous time Markovian model. In [10], Sikdar *et al.* proposed analytical models to determine the delay and throughput of TCP Reno and SACK. The work in [11] and [12] studied the performance of variants of TCP in wireless scenarios.

For TCP CUBIC, the performance evaluation is mainly conducted via simulations and experiments. Leith *et al.* performed experimental evaluation for TCP CUBIC in [13] and Bateman *et al.* presented a simulation based study of BIC, CUBIC, scalable TCP (STCP), and high-speed TCP (HSTCP) in [14]. Moreover, most of the previous analysis of TCP CUBIC [5] [13] [14] focus on wired networks. There are very few related work on the performance of TCP CUBIC for wireless networks.

Despite the fact that TCP CUBIC has been implemented in Linux operating systems (OS), analytical models for the performance of TCP CUBIC are few. Although analytical models for other variants of TCP have been documented well, they are not valid for TCP CUBIC. The complex, nonlinear window growth function of TCP CUBIC seems to be the bottleneck of establishing an appropriate model for TCP CUBIC. The fluid model in [15] took both random packet loss and congestion loss into consideration, but it is valid only for additive increase multiplicative decrease (AMID) cases. In [16], an increase rate accelerator (AIRA) model was proposed. Although the AIRA model is the extension of AMID model, it cannot be used to model the window growth behavior of TCP CUBIC. In [17], a stochastic model is proposed for STCP, but the model is also not applicable for TCP CUBIC.

In this paper, our goal is to propose an analytical model to analyze the performance of TCP CUBIC in wireless networks. The contributions of our work can be summarized as follows:

- We propose a Markovian model to determine the steady state throughput of TCP CUBIC. The model takes into

978-1-4244-5638-3/10/$26.00 ©2010 IEEE

account both buffer router and fading in the wireless environment by considering both congestion loss and random packet loss.

- We derive the stationary distribution of the Markov chain and obtain the average TCP throughput. The analytical model is validated via simulations.
- We evaluate the throughput performance of TCP CUBIC. Results show that random packet loss reduces the normalized average throughput more for end-to-end flow with large bandwidth-delay product. We propose an improvement to increase the throughput performance of TCP CUBIC by moderately increasing the window growth factor and the multiplicative decrease factor.

The rest of this paper is organized as follows. Section II presents the Markovian throughput model for TCP CUBIC. In Section III, we present the model validation via simulation. We also present the throughput performance of TCP CUBIC under different parameters. Conclusions and future work are given in Section IV.

II. System Model for TCP CUBIC

In this section, we first present the network model and state the assumptions of the system model. We then describe the window behavior of TCP CUBIC as a Markov chain. After that, we derive the stationary distribution of the Markov chain and obtain the steady state throughput based on the stationary distribution.

A. Congestion Loss and Random Packet Loss

Consider the network where the *bottleneck link* is the last hop wireless link. This wireless bottleneck link has a capacity of C bits/sec and is smaller than the capacities of other intermediate links between the source and destination pair. This scenario is applicable to the scenario as in 3GPP (Third Generation Partnership Project) LTE (Long Term Evolution) or WiMAX (Worldwide Interoperability for Microwave Access). The source has a large file to send to the destination. We assume that packet losses are caused by two factors: *congestion loss* and *random packet loss*.

Congestion loss happens when the transmission rate attains the maximum capacity C of the bottleneck link. We assume that the average RTT is a constant, which is a common assumption in loss-based TCP analytical modeling (e.g., [6]). Thus, the maximize congestion window size W is

$$W = C \cdot RTT. \tag{1}$$

Equivalently, congestion loss happens when window size attains the maximum window size W.

Random packet loss is caused by fading or interference in the wireless link. We assume that random packet loss experiences a random Poisson process with rate λ. This assumption has also been made in [15]. Given a time instant t_0, the time duration τ_{loss} from time t_0 to the next loss event is a random variable with an exponential distribution. The probability density function (pdf) of τ_{loss} is

$$f(\tau_{loss}) = \lambda \exp(-\lambda \tau_{loss}), \qquad \tau_{loss} > 0. \tag{2}$$

Given the time instant t_0, the probability that the next loss event happens within the time interval $(t_0 + T_1, t_0 + T_2]$ is

$$P(T_1 < \tau_{loss} \le T_2) = \exp(-\lambda T_1) - \exp(-\lambda T_2). \tag{3}$$

B. Congestion Control for TCP CUBIC

We now introduce some notations to model TCP CUBIC congestion control. Let τ denote the elapsed time from the last window reduction. The window size just before the last window reduction is denoted by x. The constant α denotes the window growth factor. A large value of α implies faster window growth rate. The constant β represents the multiplicative decrease factor. The window reduces to βx at the time of the last reduction.

In TCP CUBIC, the window size is a cubic function of time since the last loss event. Let $w(x, \tau)$ denote the window size as a function of x and τ. The congestion window of CUBIC is determined by [5][1]

$$w(x, \tau) = \alpha \left(\tau - \sqrt[3]{(1-\beta)x/\alpha} \right)^3 + x. \tag{4}$$

The congestion window reduction occurs due to either congestion loss or random packet loss event. When window reduction happens, the window size reduces to β times the window size just before the loss event. After that, it grows according to (4). Let $D(x, y)$ denote the time duration in which the window size grows from βx to y without encountering another loss event, after the last window reduction happened at the value of x. We have

$$D(x, y) = \sqrt[3]{\frac{y - x}{\alpha}} + \sqrt[3]{\frac{(1-\beta)x}{\alpha}}. \tag{5}$$

C. Markov Chain Formulation

We now present our proposed Markovian model. The range of congestion window size $(0, W]$ is partitioned into N equal-sized intervals. The ith interval is $((i-1)W/N, iW/N]$. If the congestion window size is in the ith interval, we regard the window size to be the midpoint of the interval, with value $(i - 0.5)W/N$, denoted by a_i. This is similar to *quantization*: we *map* a continuous range of values $(0, W]$ to a finite set of values $\{a_1, a_2, ..., a_N\}$. The number of intervals N can also be regarded as quantization precision: when N increases, the mapping becomes more precise. The quantization is an essential step of the Markov chain formulation: the finite set of values derived through quantization corresponds to the Markov chain with N states.

The Markov chain is observed at each time instant when there is a window reduction (i.e., loss event). The kth time instant, where $k = 1, 2, \ldots$, corresponds to the kth window reduction. For a TCP CUBIC session, when the kth window reduction is just about to happen, the congestion window size is denoted by x_k. x_k is in one of the N intervals, and is mapped to $\widetilde{x}_k$, $\widetilde{x}_k \in \{a_1, a_2, ..., a_N\}$.

When $\widetilde{x}_k = a_i$, the *state* of the TCP CUBIC session is in the ith state at the time instant k. Let X_k denote the state at

[1]The β in (4) corresponds to $(1-\beta)$ of (1) in [5]. As a result $w(x, 0) = \beta x$, meaning that the window reduces to βx at the time of last reduction.

978-1-4244-5638-3/10/$26.00 ©2010 IEEE

TABLE I
DEFINITIONS OF x_k, $\widetilde{x}_k$ AND X_k

Symbol	Definition
x_k	When the kth window reduction is just about to happen, the congestion window size is x_k, $x_k \in (0, W]$.
$\widetilde{x}_k$	The mapped value of x_k at the time instant k, if $(i-1)W/N < x_k \le iW/N$, then $\widetilde{x}_k = a_i = (i-0.5)W/N$.
X_k	The state at the time instant k, if $(i-1)W/N < x_k \le iW/N$, then $X_k = i$, $X_k \in \{1, 2, \ldots, N\}$.

the time instant k. We have $X_k = i$ and $\widetilde{x}_k = a_i$ if $x_k \in ((i-1)W/N, iW/N]$. Table I shows the definition of x_k, $\widetilde{x}_k$ and X_k.

Let random variable τ_k denote the time duration between the time instant k (i.e., the kth window reduction) and the time instant $(k+1)$ (i.e., the $(k+1)$th window reduction). Given x_k, the maximum value of congestion window size is W. The maximum value of τ_k is $D(x_k, W)$.

Consider the *state sequence* $X_1, X_2, \ldots, X_k$. From (4), the congestion window size at the next loss event after time instant k is only related to x_k and τ_k. That is

$$\begin{aligned} x_{k+1} &= w(x_k, \tau_k) \\ &= \alpha \left(\tau_k - \sqrt[3]{\frac{(1-\beta)x_k}{\alpha}} \right)^3 + x_k, \\ &\qquad \tau_k \le D(x_k, W). \end{aligned} \quad (6)$$

Thus, x_{k+1} is independent of $x_1, x_2, \ldots, x_{k-1}$. We have

$$P(x_{k+1} \mid x_k, x_{k-1}, \ldots, x_1) = P(x_{k+1} \mid x_k). \quad (7)$$

Furthermore, the conditional probabilities of $\widetilde{x}_{k+1}$ and X_{k+1} are as follows:

$$P(\widetilde{x}_{k+1} \mid \widetilde{x}_k, \widetilde{x}_{k-1}, \ldots, \widetilde{x}_1) = P(\widetilde{x}_{k+1} \mid \widetilde{x}_k), \quad (8)$$

and

$$P(X_{k+1} \mid X_k, X_{k-1}, \ldots, X_1) = P(X_{k+1} \mid X_k). \quad (9)$$

Therefore, the state sequence, $X_1, X_2, \ldots$, is a Markov chain.

D. State Transition Probability

Fig. 1 shows the process when the state transits from the ith state at the time instant k to jth ($j \ne N$) state at the time instant $(k+1)$ (i.e., $X_k = i$ and $X_{k+1} = j$). At time instant k, the loss event occurs, the congestion window size is reduced from a_i to βa_i. After that, the congestion window size grows within the interval of $((j-1)W/N, jW/N]$. At the time instant $(k+1)$, the congestion cannot happen below the value βa_i.Thus,

$$P_{ij} = 0, \qquad \text{if } jW/N < \beta a_i. \quad (10)$$

Since $a_i = (i-0.5)W/N$, when the inequality $jW/N < \beta a_i$ holds, it is equivalent to state $j < \beta(i-0.5)$.

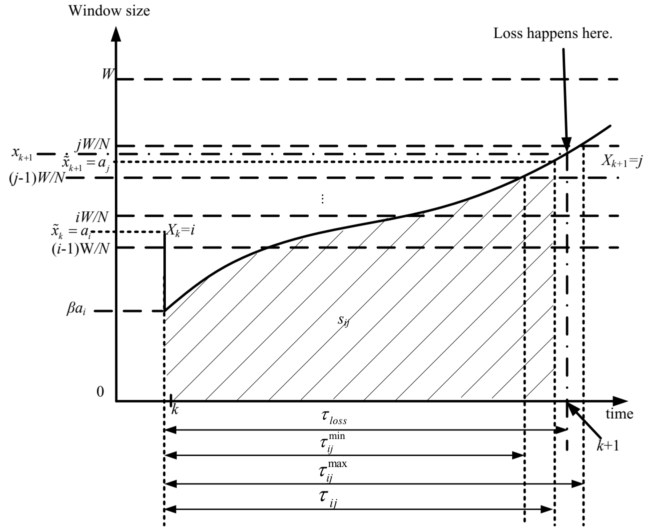

Fig. 1. Congestion window size versus time.

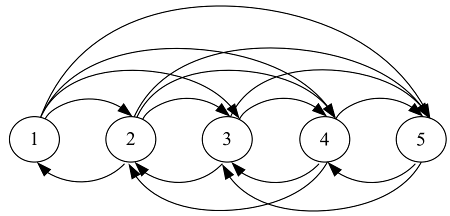

Fig. 2. Markov chain example of $N = 5$ and $\beta = 0.5$. According to (10), (14) and (15), the ith state can transit to the jth state if $j \ge \beta(i-0.5)$.

When the state transits from the ith state to the jth state, it is in the interval $((j-1)W/N, jW/N]$ that a loss event happens. We have

$$\tau_{ij}^{\min} < \tau_{loss} \le \tau_{ij}^{\max}, \quad (11)$$

where $\tau_{ij}^{\min}$ is the minimum time duration that the state transits from the ith state to the jth state; it is the time that the congestion window size grows from βa_i to $(j-1)W/N$ after the window reduction happened at the value of a_i.

$$\begin{aligned} \tau_{ij}^{\min} &= (D(a_i, (j-1)W/N))^+ \\ &= \left(\sqrt[3]{\frac{(j-1)W/N - a_i}{\alpha}} + \sqrt[3]{\frac{(1-\beta)a_i}{\alpha}} \right)^+, \end{aligned} \quad (12)$$

where $(a)^+ = \max(a, 0)$.

$\tau_{ij}^{\max}$ is the maximum time duration that the state transits from the ith state to the jth state. It is given by

$$\begin{aligned} \tau_{ij}^{\max} &= D(a_i, jW/N) \\ &= \sqrt[3]{\frac{jW/N - a_i}{\alpha}} + \sqrt[3]{\frac{(1-\beta)a_i}{\alpha}}. \end{aligned} \quad (13)$$

Thus, according to (3), if $jW/N \ge \beta a_i$ (i.e., $j \ge \beta(i-0.5)$)

and $j \neq N$, then the state transition probability is

$$\begin{aligned} P_{ij} &= P\left((D(a_i,(j-1)W/N))^+ < \tau_{loss} \leq D(a_i, jW/N)\right) \\ &= \exp\left(-\lambda\left(D\left(a_i, \frac{(j-1)W}{N}\right)\right)^+\right) \\ &\qquad - \exp\left(-\lambda D\left(a_i, \frac{jW}{N}\right)\right) \\ &= \exp\left(-\lambda\left(\sqrt[3]{\frac{(j-1)W/N - a_i}{\alpha}} + \sqrt[3]{\frac{(1-\beta)a_i}{\alpha}}\right)^+\right) \\ &\qquad - \exp\left(-\lambda\left(\sqrt[3]{\frac{jW/N - a_i}{\alpha}} + \sqrt[3]{\frac{(1-\beta)a_i}{\alpha}}\right)\right). \end{aligned} \tag{14}$$

When state j is equal to N (i.e., $j = N$), we have

$$P_{iN} = 1 - \sum_{j=1}^{N-1} P_{ij}. \tag{15}$$

Fig. 2 shows an example of the Markov chain when $N = 5$ and $\beta = 0.5$.

E. Stationary Distribution and Throughput

Let $(\pi_1, \pi_2, \dots, \pi_N)$ denote the stationary distribution of the Markov chain, in which π_i is the stationary probability of the ith state. Given the state transition probabilities from (10), (14) and (15), we can derive the stationary distribution $(\pi_1, \pi_2, \dots, \pi_N)$ by solving

$$\sum_{i=1}^{N} \pi_i P_{ij} = \pi_j, \qquad j \in \{1, 2, \dots, N\}, \tag{16}$$

and

$$\sum_{i=1}^{N} \pi_i = 1. \tag{17}$$

The duration that the state transits from the ith state to the jth state is a random variable in the interval of $\left((D(a_i,(j-1)W/N))^+, D(a_i, jW/N)\right]$. When N is large enough, we can regard

$$\tau_{ij} = (D(a_i,(j-0.5)W/N))^+, \tag{18}$$

as the average time duration that the state transits from the ith state to the jth state.

Let $s_{ij} = \int_0^{\tau_{ij}} w(a_i, t)dt$, which is the shaded area in Fig. 1.

$$\begin{aligned} s_{ij} &= \int_0^{\tau_{ij}} w(a_i, t) d\tau \\ &= \int_0^{\tau_{ij}} \left(\alpha\left(t - \sqrt[3]{\frac{(1-\beta)a_i}{\alpha}}\right)^3 + a_i\right) d\tau \\ &= a_i \tau_{ij} + \frac{\alpha}{4}\left((\tau_{ij} - L)^4 - L^4\right), \end{aligned} \tag{19}$$

where $L = \sqrt[3]{\frac{(1-\beta)a_i}{\alpha}}$.

Let r_{ij} denote the total transmission amount while the state transits from the ith state to the jth state. From (19), we have

$$\begin{aligned} r_{ij} &= \int_0^{\tau_{ij}} \frac{w(a_i,t)}{RTT} d\tau \\ &= \frac{s_{ij}}{RTT}. \end{aligned} \tag{20}$$

Let t denote the total transmission time and $r(t)$ denote the total transmission amount during t. $M_{ij}(t)$ denotes the occurrence times of the ith state transition to the jth state during t. When t is large, by the law of large numbers, the limit $\lim_{t\to\infty} \frac{M_{ij}(t)}{\sum_{i=1}^N \sum_{j=1}^N M_{ij}(t)}$ is equal to the occurrence probability of the ith state transition to the jth state. Thus, the average normalized TCP CUBIC throughput is

$$\begin{aligned} \overline{x} &= \frac{\lim_{t\to\infty} \frac{r(t)}{t}}{C} = \frac{RTT}{W} \lim_{t\to\infty} \frac{\sum_{i=1}^N \sum_{j=1}^N M_{ij}(t) r_{ij}}{\sum_{i=1}^N \sum_{j=1}^N M_{ij}(t)\tau_{ij}} \\ &= \frac{1}{W} \lim_{t\to\infty} \frac{\sum_{i=1}^N \sum_{j=1}^N M_{ij}(t)(r_{ij} \cdot RTT)}{\sum_{i=1}^N \sum_{j=1}^N M_{ij}(t)\tau_{ij}} \\ &= \frac{1}{W} \lim_{t\to\infty} \frac{\sum_{i=1}^N \sum_{j=1}^N M_{ij}(t) s_{ij}}{\sum_{i=1}^N \sum_{j=1}^N M_{ij}(t)\tau_{ij}} \\ &= \frac{1}{W} \lim_{t\to\infty} \frac{\sum_{i=1}^N \sum_{j=1}^N \left(\left(M_{ij}(t) / \sum_{p=1}^N \sum_{q=1}^N M_{pq}(t)\right) s_{ij}\right)}{\sum_{i=1}^N \sum_{j=1}^N \left(\left(M_{ij}(t) / \sum_{p=1}^N \sum_{q=1}^N M_{pq}(t)\right) \tau_{ij}\right)} \\ &= \frac{1}{W} \lim_{k\to\infty} \frac{\sum_{i=1}^N \sum_{j=1}^N P(X_{k+1} = j, X_k = i) s_{ij}}{\sum_{i=1}^N \sum_{j=1}^N P(X_{k+1} = j, X_k = i)\tau_{ij}} \\ &= \frac{1}{W} \frac{\sum_{i=1}^N \sum_{j=1}^N \pi_i P_{ij} s_{ij}}{\sum_{i=1}^N \sum_{j=1}^N \pi_i P_{ij} \tau_{ij}}. \end{aligned} \tag{21}$$

III. Performance Evaluation

In this section, we first validate our proposed analytical model via simulation. We then present the TCP throughput results under different parameters.

A. Analytical Model Validation via Simulation

We develop a discrete-event simulator to validate the accuracy of our proposed analytical model. Let M_i, where $i \in \{1, 2, \dots, N\}$, denote the counter of the ith state, K denote the total simulation time. We first initialize W, N, β, α

978-1-4244-5638-3/10/$26.00 ©2010 IEEE

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE Globecom 2010 proceedings.

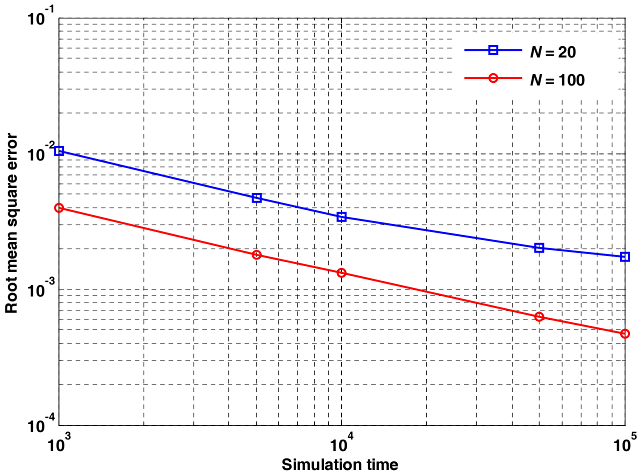

Fig. 3. Root mean square (RMS) error versus total simulation time. When N increases, the RMS error is reduced. ($C = 100$ Mb/s, $RTT = 100$ ms, $\alpha = 1$ Mb/s, $\beta = 0.5$, $\lambda = 1$ s^{-1})

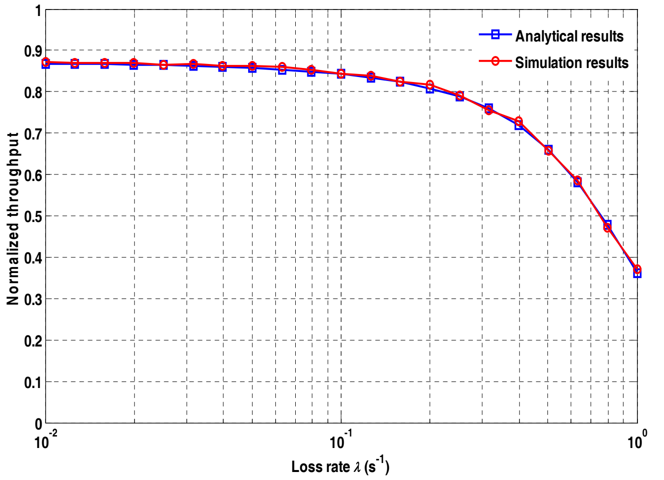

Fig. 4. Analytical and simulated average normalized throughput under different loss rate λ. ($C = 100$ Mb/s, $RTT = 100$ ms, $\alpha = 1$ Mb/s, $\beta = 0.5$, $N = 100$, $K = 1000$)

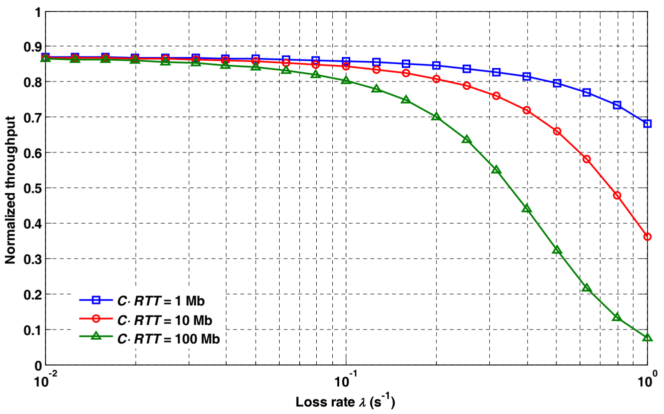

Fig. 5. The normalized average TCP CUBIC throughput under different bandwidth-delay product $C \cdot RTT$ and loss rate λ. ($\alpha = 1$ Mb/s, $\beta = 0.5$, $N = 100$)

and set M_i to be equal to 0 at the beginning; x_1 is randomly generated in $(0, W]$. We simulate the behavior of TCP CUBIC starting from the time instant $k = 1$ to $k = K$. The simulation at each time instant is performed as follows: At each time instant k, we first calculate $\widetilde{x}_k$ and X_k according to x_k and update the state counter (i.e., increase M_{X_k} by 1). Then, we simulate τ_k, which is the time duration from the time instant k to the next loss event. The value of τ_k is set to be equal to $\min(\tau_{loss}, D(x_k, W))$, in which τ_{loss} is randomly generated according to its pdf in (2). Based on x_k and τ_k, the window size at the next window reduction $x_{k+1} = w(x_k, \tau_k)$ can be calculated. The stationary distribution as well as the average TCP CUBIC throughput can then be determined.

For performance metric, we consider the root mean square (RMS) error, which is defined as

$$RMS = \sqrt{\frac{\sum_{i=1}^{N} (\pi_i - \widetilde{\pi}_i)^2}{N}}, \qquad (22)$$

where π_i and $\widetilde{\pi}_i$ are obtained via analytical model and simulation, respectively. The RMS error reflects the gap between the analytical results and the simulation results. Fig. 3 shows the RMS error under different simulation time. The two curves show that when the simulation time is increased, the RMS error will decrease. This illustrates that an increase in simulation time leads to the simulation results being closer to the analytical results. In addition, we notice that when the number of intervals N increases, the RMS error becomes smaller. This illustrates that larger N will lead to more accurate analytical results.

Fig. 4 shows the analytical and simulated average throughput under different loss rate λ. The parameters chosen are as follows: $C = 100$ Mb/s, $RTT = 100$ ms, $\alpha = 1$ Mb/s, and $\beta = 0.5$. This figure shows that the analytical results and simulation results agree with each other.

B. Throughput Performance of TCP CUBIC

We now present the throughput results of TCP CUBIC based on our proposed Markovian model.

Fig. 5 shows the throughput performance of TCP CUBIC under different bandwidth-delay product $C \cdot RTT$ and loss rate λ. The number of intervals N is set to 100. Results show that even when λ is very small, the normalized average throughput will not attain 1. This is because even when there is no random packet loss, there are still congestion losses, causing multiplicative decrease when the window size is equal to W. Fig. 5 also shows that if $C \cdot RTT$ increases, the normalized average throughput decreases. Random packet loss reduces the normalized average throughput of large bandwidth-delay product links more.

Our analytical results show that, for $i, j \in \{1, 2, \ldots, N\}$, $P_{ij}, \pi_i, \frac{s_{ij}}{W}$ and τ_{ij} depend on the value of $\frac{C \cdot RTT}{\alpha}$ (i.e., bandwidth-delay product over window growth factor).

Therefore, the normalized throughput depends on the value of $\frac{C \cdot RTT}{\alpha}$. The effect of an increase of bandwidth-delay product is equivalent to a decrease of the window growth factor, leading to a decrease of the normalized throughput.

978-1-4244-5638-3/10/$26.00 ©2010 IEEE

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE Globecom 2010 proceedings.

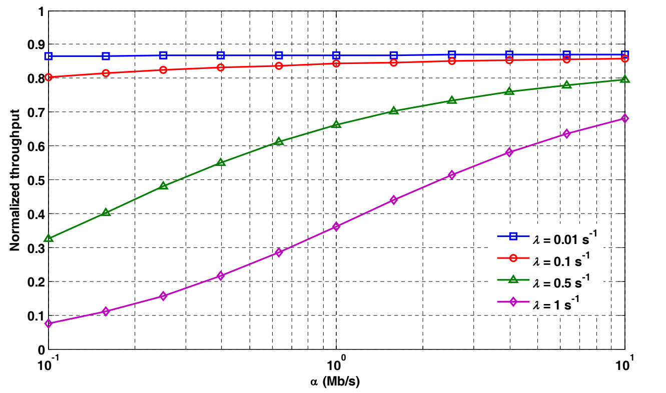

Fig. 6. The normalized average throughput of TCP CUBIC under different window growth factor α. (C = 100 Mb/s, RTT = 100 ms, β = 0.5, N = 100)

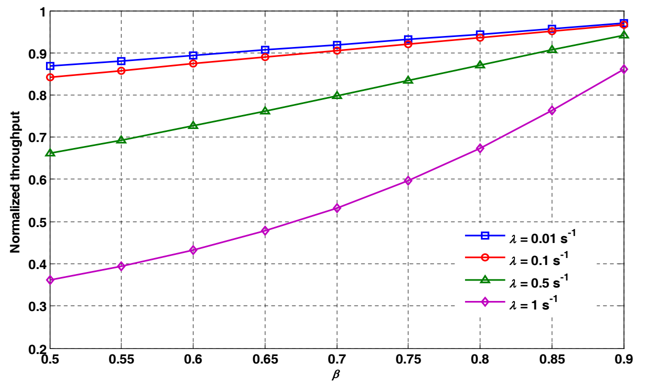

Fig. 7. The normalized average throughput of TCP CUBIC under different β. (C = 100 Mb/s, RTT = 100 ms, α = 1 Mb/s, N = 100)

Therefore, in contrast to the wired cases, large bandwidth-delay product will decrease the normalized throughput of TCP CUBIC in wireless scenarios due to the random packet loss.

Fig. 6 shows the normalized average throughput under different window growth factor α with C = 100 Mb/s, RTT = 100 ms, N = 100, and β = 0.5. Results show that by moderately increasing α, the throughput performance will improve. When $\lambda = 1$ s^{-1}, the average normalized throughput increases by about 806% when α grows from 0.1 Mb/s to 10 Mb/s. When λ is small, increasing α will bring less throughput gain. In the figure, when $\lambda = 0.01$ s^{-1}, the normalized average throughput is not improved much.

Fig. 7 shows the normalized average throughput under different β with C = 100 Mb/s, RTT = 100 ms, N = 100, and α = 1 Mb/s. Results show that by moderately increasing β, the throughput performance will improve. When $\lambda = 1$ s^{-1}, the average normalized throughput increases by about 137% when β grows from 0.5 to 0.9. When λ is small (e.g., $\lambda = 0.01$ s^{-1}), increasing β will bring less throughput gain.

IV. Conclusions

In this paper, we proposed an analytical model to determine the steady state throughput of TCP CUBIC in wireless environment. We considered both congestion loss and random packet loss and established a Markov model. We derived the stationary distribution of the Markov chain and obtained the average throughput based on the stationary distribution. The accuracy of the model was validated via simulation. Based on our proposed model, we evaluated the throughput performance of TCP CUBIC. Results showed that random packet loss reduces the normalized average throughput more for end-to-end flow with large bandwidth-delay product. In wireless environment, we showed that the throughput of TCP CUBIC can be improved by moderately increasing the window growth factor α and the multiplicative decrease factor β. For future work, we plan to extend the model by incorporating other types of loss models for wireless channels.

Acknowledgement

This work is supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada and the British Columbia Innovation Council (BCIC).

References

[1] S. Floyd, T. Henderson, and A. Gurtov, "The NewReno modification to TCP's fast recovery algorithm," *IETF RFC 3782*, Apr. 2004.
[2] L. Brakmo and L. Peterson, "TCP Vegas: End-to-end congestion avoidance on a global Internet," *IEEE Journal on Selected Areas in Communication*, vol. 13, no. 8, pp. 1465–1480, Oct. 1995.
[3] D. X. Wei, C. Jin, S. H. Low, and S. Hegde, "FAST TCP: Motivation, architecture, algorithms, performance," *IEEE/ACM Trans. on Networking*, vol. 14, no. 6, pp. 1246–1259, Dec. 2006.
[4] L. Xu, K. Harfoush, and I. Rhee, "Binary increase congestion control (BIC) for fast long-distance networks," in *Proc. of IEEE Infocom*, Hong Kong, China, Mar. 2004.
[5] I. Rhee and L. Xu, "A new TCP-friendly high-speed TCP variant," in *Proc. PFLDNet'05*, Lyon, France, Feb. 2005.
[6] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," in *Proc. of ACM SIGCOMM*, Vancouver, Canada, Sept. 1998.
[7] M. Mathis, J. Semke, J. Mahdavi, and T. Ott, "The macroscopic behavior of the TCP congestion avoidance algorithm," *ACM SIGCOMM Computer Communication Review*, vol. 27, no. 3, pp. 67–82, July 1997.
[8] F. Baccelli and K. B. Kim, "TCP throughput analysis under transmission error and congestion losses," in *Proc. of IEEE Infocom*, Hong Kong, China, Mar. 2004.
[9] A. Misra and T. J. Ott, "The window distribution of idealized TCP congestion avoidance with variable packet loss," in *Proc. of IEEE Infocom*, New York, NY, Mar. 1999.
[10] B. Sikdar, S. Kalyanaraman, and K. S. Vastola, "Analytic models for the latency and steady-state throughput of TCP Tahoe, Reno, and SACK," *IEEE/ACM Trans. on Networking*, vol. 11, no. 6, pp. 959–971, Dec. 2003.
[11] F. Anjum and L. Tassiulas, "Comparative study of various TCP versions over a wireless link with correlated losses," *IEEE/ACM Trans. on Networking*, vol. 11, no. 3, pp. 370–383, June 2003.
[12] S. Chen, B. Bensaou, and K. L. Hung, "Performance of different TCP variants in IEEE 802.11 WLAN and the TCP-WOW algorithm," in *Proc. of IEEE Globecom*, Honolulu, HI, Dec. 2009.
[13] D. J. Leith, R. N. Shorten, and G. McCullagh, "Experimental evaluation of Cubic-TCP," in *Proc. PFLDNet'07*, Los Angeles, CA, Feb. 2007.
[14] M. Bateman, S. Bhatti, G. Bigwood, D. Rehunathan, C. Allison, and T. Henderson, "A comparison of TCP behaviour at high speeds using ns-2 and Linux," in *Proc. of Communications and Networking Simulation*, Ottawa, Canada, Apr. 2008.
[15] S. Hassayoun, P. Maille, and D. Ros, "On the impact of random losses on TCP performance in coded wireless mesh networks," in *Proc. of IEEE Infocom*, San Diego, CA, Mar. 2010.
[16] I. Psaras and V. Tsaoussidis, "AIRA: Additive increase rate accelerator," in *Proc. of IFIP Networking*, Singapore, May 2008.
[17] F. Baccelli, G. Carofiglio, and M. Piancino, "Stochastic analysis of scalable TCP," in *Proc. of IEEE Infocom*, Rio de Janeiro, Brazil, Apr. 2009.

978-1-4244-5638-3/10/$26.00 ©2010 IEEE