A Markov Chain Model for TCP NewReno over Optical Burst Switching Networks†

V. Bimal, T. Venkatesh, and C. Siva Ram Murthy
Department of Computer Science and Engineering
Indian Institute of Technology Madras, Chennai - 600036, India
{vbimal,tvenku}@cse.iitm.ernet.in, murthy@iitm.ac.in

Abstract—Study of the performance of Transmission Control Protocol (TCP) over Optical Burst Switching (OBS) networks has been an important problem of research lately. In this work, we propose an analytical model for a TCP NewReno source to derive the steady-state throughput in presence of burst assembly process and burst losses. The source model uses a Markov chain based evolution of congestion window and the network model characterizes the distribution of burst size for a general assembly process which is used to estimate the impact of a burst loss on the number of packets lost. A fixed-point iteration method is then used to jointly solve the source model and the network model to obtain the TCP send rate. We validate the proposed analytical model through simulations. Results highlight the importance of accounting for slow start and fast retransmit phases in the model.

I. INTRODUCTION

Optical Burst Switching (OBS) [1] that uses the best of both circuit switching and packet switching technologies is envisioned to be the technology to support ever-increasing traffic in the Internet. At the same time, the emergence of today’s Internet based on the Internet Protocol (IP) has revolutionized the telecommunication industry. Since all forms of existing end-user applications such as, HTTP, FTP, Telnet use the ubiquitous Transmission Control Protocol (TCP), TCP traffic accounts for approximately 90% of the total Internet traffic. Hence, analyzing the performance of TCP traffic over OBS networks is an important area of research.

At an ingress node of the OBS network, packets from different sources to the same destination are aggregated into a burst and transmitted along the path entirely in optical domain. A control packet with information on the burst length and arrival time is sent ahead of the data burst to reserve resources on the path only for the burst duration. At the egress node, the data bursts are disassembled into packets and sent to the destination. The core network is assumed to be bufferless and hence, one or more of the contending bursts are dropped at a core node. These contention losses do not indicate serious congestion unlike losses due to buffer overflow in the Internet. When a TCP sender perceives burst losses (multiple simultaneous packet losses) in OBS network, it responds with a drastic window reduction and lowers the send rate. A good analytical model that accurately models the behaviour of a TCP sender is required to study the impact of bursty losses on TCP throughput.

There is widely available literature on TCP modeling for the Internet from which we refer only those that are closely related to our work. We use the fixed-point method introduced in [2] to model TCP flows with a Markovian model for the TCP source. Extensions to [2] are proposed in [3], [4] that consider other variants of TCP as well as correlated/bursty packet losses. A closed queueing network model for TCP was proposed in [5]. Work in [6], [7] considers only the extreme cases of fast and slow TCP sources to study the effect of send rate on burst losses in OBS networks. It is based on extensions of the renewal theory based model [8] that considers the cyclic evolution of TCP window in terms of rounds. A simple formulation of TCP Reno send rate which uses a generic speed source is presented in [9], but it does not model the burst assembly process in detail.

We extend the Markovian model for the TCP source used in [3], [4] and use fixed-point method to solve the interaction with the network model iteratively that include the effect of burst aggregation and burst losses in OBS networks. Measurements [10] show that TCP NewReno is the most widely used TCP implementation in the Internet. We model the Slow-but-Steady variant of TCP NewReno [11] in which the retransmission timer is reset after each partial acknowledgement (ACK) during the fast recovery phase. We also use the partial window deflation algorithm in which if a sender receives a partial ACK, NewReno retransmits the first unacknowledged packet, then decreases the window by the amount of new data acknowledged and adds one packet back. Also, the sender recovers one lost packet per round trip time (RTT). The main contributions of this paper are as follows:

- A detailed model for TCP NewReno source with a generic send rate considering slow start, congestion avoidance, fast retransmit/fast recovery, timeout and exponential back-off states for the TCP window.
- Impact of the burst assembly process on TCP send rate is characterized by finding the exact distribution of burst size based on a general arrival rate at the ingress. Dynamic behaviour of TCP window is modeled by considering the loss of bursts of variable size which results in a variable number of packets getting lost.

We describe the analytical model in Section II, validate it

†This work was supported by Microsoft Corporation and Microsoft Research India under Microsoft Research India PhD Fellowship Award.
*Author for correspondence.
through simulations in Section III and conclude in Section IV.

II. DESCRIPTION OF THE MODEL

We propose an analytical model for the send rate of TCP NewReno which considers the bursty losses in OBS network. We assume that the reader is familiar with the working of TCP NewReno [11]. The model consists of a TCP source and a network component. The source model for TCP computes the send rate of the source, which becomes the offered load for the network model. The network model considers the burst assembly process and calculates the statistical distribution of the burst size for an arrival rate. The burst size distribution computed by the network model is in turn used by the source model to compute the send rate. Considering this interaction between the TCP source and network model, we use fixed point iteration method to compute the average TCP send rate.

A. Network Model

Due to the bufferless switching in OBS network, burst losses occur randomly. We assume a Bernoulli loss model for the losses and do not compute burst loss probability explicitly (assumed to be given). In this model, the TCP source is connected to the ingress with a lossless access network. We consider only a single TCP flow feeding the burst assembler. We assume that the packet arrival at the ingress node follows a stationary Poisson process with its parameter as the average throughput of the TCP source. This is reasonable because packet arrivals appear Poisson distributed in the sub-second burst assembly period [12]. We assume that each packet contains one TCP segment. The average RTT of the TCP connection includes the delay in the access, as well as burstification/deburstification delay in the OBS core network. We neglect the burstification/deburstification process in the reverse path and assume no losses in the reverse path. We also neglect the impact of delayed acknowledgements.

We assume a Min-Burst-Length-Max-Assembly-Period (MBMAP) assembly algorithm [13] which sends out a burst when the burst size exceeds the minimum burst length ($b_{th}$) or when the assembly period ($T_b$) times out, whichever happens first. At first we evaluate the distribution for number of packets in a burst to calculate the number of packets lost by a burst loss. For the MBMAP assembly process, we characterize the burst size distribution in terms of packets. Consider an exponential packet size distribution with mean $\mu_p$ and let $\lambda_p$ be the average send rate of the TCP source. Two events can occur at the assembly queue: Let $A_1$ be the event that the $T_b$ times out, and $A_2$ be the event of a burst size exceeding $b_{th}$ given that $n$ packets arrive by the time a burst is ready to be sent. Probability that event $A_1$ occurs is given by the probability that the sum of the size of $n$ packets is less than $b_{th}$ given as,

$$Prob\{A_1\} = \prob \left\{ \sum_{i=1}^{n} P_i < b_{th} \right\} = F_{S_n}(b_{th})$$

$$= 1 - \sum_{k=0}^{n-1} e^{-b_{th}/\mu_p} (b_{th}/\mu_p)^k/k!$$

where, $F_{S_n}(\cdot)$ is the $cdf$ of an Erlang distribution, which is the sum of $n$ exponentially distributed random variables (packet sizes) each with a mean $\mu_p$, and $P_i$ is an exponential random variable representing the size of a packet in Bytes. $Prob\{A_2\} = 1 - Prob\{A_1\}$ as $A_1$ and $A_2$ are mutually exclusive.

For a Poisson arrival process, when $A_1$ occurs, the number of packets in a burst is a Poisson random variable with mean $\lambda_p T_b$. Similarly, when $A_2$ occurs, the number of packets in a burst is a Poisson random variable with mean $b_{th}/\mu_p$ for exponential packet sizes (elaborated in [14]). From Eq. 1, we get the pmf of the number of packets in a burst, i.e., the probability that there are $n$ ($n = 1, 2, \ldots$) packets in a burst as,

$$f_{N_p}(n) = \frac{((\lambda_p T_b)^n e^{-\lambda_p T_b})}{n!} Prob\{A_1\} + \frac{((b_{th}/\mu_p)^n e^{-b_{th}/\mu_p})}{n!} Prob\{A_2\}$$

B. Source Model

The dynamics of the TCP congestion window ($W$) are modeled using a homogeneous discrete time Markov chain, viz., slow start (SS), congestion avoidance (CA), fast retransmit/fast recovery (FR), time out (TO), and exponential back-off states (B). Let $W_m$ be the maximum receiver window size. The state space is defined as,

$$\{SS, W, W_{th}\} \in \{(SS, 1, 3), (SS, 2, 3), (SS, 1, 4), \ldots, (SS, 2^\lceil\log_2 W_{m/2}\rceil - 1, W_{m/2})\}$$

$$\{CA, W\} \quad \quad W \in \{1, 2, 3, \ldots, W_m\}$$

$$\{FR, W\} \quad \quad W \in \{4, 5, \ldots, W_m\}$$

$$\{TO, W\} \quad \quad W \in \{2, 3, \ldots, W_m\}$$

$$\{B, bck\} \quad \quad \{B, 4\}, \{B, 8\}, \{B, 16\}, \{B, 32\}, \{B, 64\}$$

The trivial case of slow start threshold $W_{th} = 1, 2$ is built into the congestion avoidance phase. The parameter $bck$ in $B$ states denotes the exponential back-off factor. When a loss is detected from triple duplicate ACKs, the window size at that time should be atleast 4 (atleast three packets have to be sent successfully after the lost packet(s)). So FR states start from a window size of 4. We neglect the case of a packet loss immediately at the start of a TCP connection (when $W = 1$), therefore TO states start from $W = 2$.

To compute the stationary probabilities of the Markov chain we need to compute the state transition probabilities. A single burst loss can trigger multiple packet losses in a window as determined by the burst size. We view a TCP window as being composed of variable number of bursts (group of packets), which makes it easier to analyze the impact of bursty losses. We use the burst size distribution (in terms of TCP packets) from the network model to evaluate the distribution of number bursts in a window. Let $f_{N_b}(W)$ be the conditional pmf of the number of bursts $N_b$ given a window size $W$, and $B_i$ be a sequence of iid random variables each representing the size of
the $i^{th}$ burst in number of packets. The probability that there are $n$ bursts in a window $w$ is given by,

$$f_{N_{w}}(N_{w} = n|W = w) = \text{Prob}(\sum_{i=1}^{n} B_{i} = w) \equiv g(n, w)$$

(4)

For clarity, henceforth we use $g(n, w)$ to indicate the probability that $w$ consists of $n$ bursts of size $B_{i}$ as defined in Eq. 4. $g(n, w)$ can be determined by taking the $n$-fold convolution of the pmf of the number of packets in a burst (i.e., $f_{N_{b}}$).

To obtain the state transition probabilities for the Markov chain, we define three types of transition probabilities: $P_{NL}(W)$ is the probability that all the packets in a window $W$ are successfully transmitted, $P_{TO}(W)$ is probability of a TCP sender going to a $TO$ state due to multiple packet losses in a window $W$, and $P_{FR}$ is probability of a TCP sender going to an $FR$ state on receiving triple duplicate ACKs.

$P_{NL}(W)$ can be obtained by conditioning on the number of bursts in a window using $g(n, w)$ as

$$P_{NL}(W|\text{ n bursts in } W) = (1 - p)^{n}$$

$$\Rightarrow P_{NL}(W) = \sum_{n=1}^{W} (1 - p)^{n}g(n, W)$$

(5)

where, $p$ denotes the burst loss probability. The number of bursts in a window can vary from 1 to $W$ (when each burst has only one packet). The probability of losing $i$ bursts in a window $W$, $P_{nl}(i|W)$ is also obtained by conditioning on the number of bursts in a window as,

$$P_{nl}(i|W) = \sum_{n=1}^{W} nC_{i} p^{i}(1 - p)^{n-i}g(n, W)$$

(6)

NewReno goes into a $TO$ state if less than 3 duplicate ACKs are received and retransmission timer expires. Therefore $P_{TO}(W)$

is the probability of a loss resulting in less than 3 packets being successfully transmitted (i.e., $P_{3}$) given by,

$$P_{TO}(W) = (1 - P_{NL}(W)) \times P_{3}$$

(7)

Let $x$ be the number of bursts lost resulting in a $TO$ state. To find $P_{3}$, the sum of $x$ burst sizes ($B_{s}$) should be equal to either $W-2$, $W-1$, or $W$. Conditioning it with the probability that $x$ bursts are lost in a window $W$, $(x$ from 1 to $W$), we get

$$P_{3} = \sum_{x=1}^{W} \sum_{k=W-2}^{W} \text{Prob}\left(\sum_{i=1}^{x} B_{i} = k\right) P_{nl}(x|W)$$

(8)

where $B_{i}$ is pmf of the burst size.

From the law of total probability we can write $P_{FR}(W)$ as,

$$P_{FR}(W) = (1 - P_{NL}(W) - P_{TO}(W)) , W \geq 4$$

(9)

Next, we obtain the transition probability from state $V_{k}$ to a state $V_{k+1}$ for each type of state. When window has only one packet which means only one burst, the probability of

\footnote{Note that $P_{TO}(W) = 1 - P_{NL}(W)$ when $2 \leq W < 4$ because transition to $FR$ state is not possible.} sending it successfully is taken as $1 - p$. For the $SS$ states, i.e., $V_{k} = (SS, W, W_{th})$, window size increases exponentially in the absence of packet loss. When a loss occurs, transition can occur to a $TO$ state or an $FR$ state, depending on the current window size at the loss. The various transitions possible with the respective probabilities are given by,

$$V_{k+1} = \begin{cases} (SS, 2W, W_{th}), & w.p P_{NL}(W), \quad W < W_{th} \\ (SS, 2W, W_{th}), & w.p (1 - p), \quad W = 1 \\ (CA, W_{th}), & w.p P_{NL}(W), \quad 2W \geq W_{th} \\ (FR, W), & w.p P_{FR}(W), \quad 4 \leq W < W_{th} \\ (TO, W), & w.p (1 - P_{NL}(W)), \quad 2 \leq W < 4 \\ (B, 4), & w.p p_{b}, \quad W = 1 \end{cases}$$

(10)

where, “w.p” denotes “with probability”.

When $V_{k} = (CA, W)$ (i.e., a $CA$ state), window increases linearly without loss, and can go to a $TO$, $FR$ or $B$ state depending on the nature of the loss. The transition probabilities to the next state $V_{k+1}$ are given by,

$$V_{k+1} = \begin{cases} (CA, W + 1), & w.p (1 - p), \quad W = 1 \\ (CA, W), & w.p P_{NL}(W), \quad 2 \leq W \leq W_{m} \\ (FR, W), & w.p P_{FR}(W), \quad W = W_{m} \\ (TO, W), & w.p (1 - P_{NL}(W)), \quad 2 \leq W < 4 \\ (B, 4), & w.p p_{b}, \quad W = 1 \end{cases}$$

(11)

From an $FR$ state, i.e., $V_{k} = (FR, W)$, after the loss recovery mechanism of NewReno with partial window deflation, $W$ is reduced by half and $W_{th} = \left[W/2\right]$ [15].

$$V_{k+1} = (CA, \left[W/2\right]), \quad w.p 1, \quad 4 \leq W \leq W_{m}$$

(12)

After a time out loss, the window is reset to 1 and $W_{th}$ is set to $\left[W/2\right]$. So when $V_{k}$ is of the type $(TO, W)$, the transition probabilities are defined as,

$$V_{k+1} = (SS, 1, \left[W/2\right]), \quad w.p 1, \quad 2 \leq W \leq W_{m}$$

(13)

Lastly for the BO states, i.e., $V_{k}$ is of the type $(B, bck)$, the transition probabilities are given by,

$$V_{k+1} = \begin{cases} (CA, 2), & w.p (1 - p) \\ (B, 2 * bck), & w.p p_{b}, \quad bck \neq 64 \\ (B, bck), & w.p p_{b}, \quad bck = 64 \end{cases}$$

(14)

The state transition probabilities for all other transitions other than those defined above are 0.

To determine the send rate of the TCP source, we need to calculate the number of packets sent in each state as well as the sojourn time for each state in the Markov chain. The probability of losing $x_{p}$ packets in a window $W$ can be written in terms of burst loss probability $p$ as,

$$P_{np}(x_{p}|W) = \sum_{k=1}^{x_{p}} g(k, x_{p}) P_{nl}(k|W)$$

(15)

For $CA$, $SS$ and $B$ states, the number of packets sent is simply the current size of the window. For the $TO$ states, less than 3 packets might have been sent successfully in the loss window leading to time out. There is a residual round where 1 or 2 packets are sent. So expected number of packets sent
in a TO state with a window size $W$, denoted by $N_T(W)$ can be written as,

$$N_T(W) = 1 \times P_{\text{retrans}}(W-1|W) + 2 \times P_{\text{retrans}}(W-2|W) \quad (16)$$

For the Slow-but-Steady variant of TCP NewReno with partial window deflation, if $\alpha$ packets are lost in $W$ which triggers a fast retransmit/fast recovery procedure, the total number of packets sent in the remaining FR phase is given by [15], [16].

$$Total^{FR} = (W-\alpha) + \alpha \sum_{k=1}^{W} \max(0, W_{th} - \alpha - k - 1) \quad (17)$$

Therefore the expected number of packets sent in an FR state with a window $W$ is given by,

$$N_F(W) = W + \sum_{i=1}^{W-3} \sum_{k=1}^{i} \max(0, W_{th} - i + k + 1) P_{\text{retrans}}(i|W) \quad (18)$$

Waiting time in SS and CA states when $W \neq 1$ is assumed to be RTT. When $W = 1$ for CA and SS states, the expected time spent is $RTT(1-p) + 2RTO(p)$ where, $RTO$ is the value of retransmission time out. Waiting time for TO and B states is $RTO$ and $RTT(1-p) + bck \times RTO(p)$, respectively. We know that $RTO = RTT + 4RTTVAR$ where, $RTTVAR$ is the standard deviation in RTT. As there is no queuing delay in our network model, the only variation in delay is due to burst assembly process so that $RTO = RTT$. NewReno retransmits each lost packet in one RTT in the FR state and requires approximately one RTT to detect triple duplicate ACKs. Therefore the expected waiting time in an FR state is dependent on the expected number of packets lost in a window $W$. For a transition to an FR state to occur, a maximum of $(W-3)$ packets can be lost in a window $W$. Using Eq. 15, the expected waiting time in FR state for a window $W$ can be written as,

$$RTT \left( \sum_{i=1}^{W-3} i P_{\text{retrans}}(i|W) \right) + RTT \quad (19)$$

Finally, the send rate of the TCP source is defined as,

$$Rate = \frac{\sum_{i=1}^{T_s} \pi(i) N(i)}{\sum_{j=1}^{V} \pi(j) S(j)} \quad (20)$$

where, $T_s$ is the total number of states in the Markov chain, $\pi(i)$ is the stationary probability of the $i^{th}$ state which can be computed by any one of the standard techniques, and $N(i)$ and $S(i)$ are the number of packets sent and waiting time in the $i^{th}$ state in the Markov chain, respectively.

### III. Validation of the Model

In this section we validate the proposed model with simulation results obtained using ns-2 simulator with necessary OBS modules. We use the 14-node, 21-link NSFNET topology with 32 wavelengths in each link (as used in [13]). The maximum receiver window size is set to 128. A burst size threshold of 40kB and an exponential distribution for packet size with mean size of 512 Bytes are used. As we have not modeled loss and delay in the access network, we simulated TCP behaviour only for the OBS network. Though we compute the throughput of a single flow, we considered background TCP flows to generate burst losses in the network. Since we modeled steady-state throughput of TCP, we assumed long-lived TCP flows (source always has data to send). For the analytical results presented, we solved Eq. 20 for the throughput iteratively till the values converge to within $10^{-4}$. The throughput obtained in terms of pkts/s is converted to bps by multiplying with mean packet size. All the results are obtained with 95% confidence level.

Fig. 1 shows a semi-log plot of variation of TCP throughput with burst loss probability (BLP). We set $RTT = 600\, ms$ and $T_b = 3\, ms$. BLP is varied between $10^{-4}$ and 0.1 (shown on log-scale). It can be seen that simulation and analytical results closely match validating the proposed model for TCP send rate. It can be seen that the throughput degrades as the BLP increases.

Fig. 2 and Fig. 3 show the analytical and simulation results depicting the variation of TCP throughput with burst assembly time (BAT) and BLP. The average $RTT$ is set to $500\, ms$. We see that the TCP throughput degrades slowly with increase in burst assembly time.

For higher values of BLP (greater than 0.001) the throughput decreases further for any value of burst assembly time. Increasing assembly time results in correlation gain for TCP throughput negating the loss penalty caused by bursty losses. But as the losses increase, this gain is reduced by the retransmissions triggered at the sender. Since we consider the throughput of a single TCP flow with moderate window sizes, these effects are not pronounced.

Our model has an additional advantage that it can characterize the fraction of time spent in different states for varying end-to-end delay and BLP. Fig. 4 shows the proportion of time spent in FR, TO (including $B$ states) and SS states. The remaining time, not shown in the results is spent in CA states. The fraction of time spent in a state is computed as,

$$T_{Fr} = \frac{\pi(i) S(i)}{\sum_{j=1}^{V} \pi(j) S(j)} \quad (21)$$

where, $T_s$, $S(i)$ and $\pi(i)$ are as explained in Eq. 20. We use an average BLP of 0.15 and $T_s$ of $3\, ms$. We see that among all the states, majority of time is spent in SS states. Fig. 5 shows the fraction of time spent in different states for increasing BLP (in
semi-log scale) for an average $RTT = 500ms$ and $T_b = 3ms$. Observing these results, we draw some conclusions. The TCP sender spends most of the time in SS state in OBS network because of burst losses and smaller window size used. A significant amount (almost 15 – 20%) of the time is spent in FR states, thereby showing that a loss event also triggers receipt of triple duplicate ACKs and not always a time out event. Therefore, a detailed modeling of FR state is essential. Further, as the end-to-end delay increases fraction of time spent in TO states increases and the time spent in FR states slowly decreases, indicating the delay penalty observed for TCP in OBS networks [6]. This can be attributed to the fact that in TO states $W$ is low so that the throughput is less. Similarly, fraction of time spent in TO states increases with BLP.

Fig. 5: Fraction of time spent in different states for increasing BLP

IV. CONCLUSION

In this paper, we developed a complete model for TCP NewReno over OBS network based on a Markov chain representation for the evolution of congestion window. We modeled slow start, congestion avoidance, fast retransmit/fast recovery, timeout and exponential back-off states for the congestion window. The source model computed the send rate considering the burst losses in the network which can result in variable packet losses. Our network model considered the impact of a time and threshold based burst assembly process by computing the exact distribution of burst size. We used a fixed-point iteration method to jointly solve the source and the network model to compute the steady-state throughput of a NewReno sender in OBS network. Our model gives the additional advantage to compute the fraction of time spent in different states in presence of bursty losses. We verified the model with simulations which showed the accuracy of the proposed model. Results also showed that a significant fraction of time is spent in the fast recovery phase and slow start phase for OBS networks which highlighted the need for a complete model. We plan to extend the model for multiple flows of TCP and with background traffic.

REFERENCES