

FAST TCP over Optical Burst Switched Networks: Modeling and Stability Analysis

Basem Shihada^{*}, Sami El-Ferik[†], and Pin-Han Ho[‡]

^{*}Mathematical and Computer Science & Engineering Division, KAUST, Thuwal, Saudi Arabia

[†]Department of Systems Engineering, KFUPM, Dhahran, Saudi Arabia

[‡]Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada

Abstract— FAST TCP is important for promoting data-intensive applications since it can cleverly react to both packet loss and delay for detecting network congestion. This paper provides a continuous time model and extensive stability analysis of FAST TCP congestion-control mechanism in bufferless Optical Burst Switched Networks (OBS). The paper first shows that random burst contentions are essential to stabilize the network, but cause throughput degradation in FAST TCP flows when a burst with all the packets from a single round is dropped. Second, it shows that FAST TCP is vulnerable to burst delay and fails to detect network congestion due to the little variation of round-trip time, thus unstable. Finally it shows that introducing extra delays by implementing burst retransmission stabilizes FAST TCP over OBS. The paper proves that FAST TCP is not stable over barebone OBS. However, it is locally, exponentially, and asymptotically stable over OBS with burst retransmission.

I. INTRODUCTION

TCP has been subject to a tremendous amount of research, which aims to improve its performance over networks with various transmission characteristics [1]. A large number of previously reported enhancements have modified TCP such that it adapts to the newly evolving network transmission characteristics. These modifications have mostly considered best-effort traffic and buffer-oriented routing mechanisms which are essential in the current IP-based Internet.

Optical Burst Switching (OBS) networks tend to be bufferless in nature [2]. Its fundamental components are control packets and data bursts. A control packet is first launched by the edge node and travels through the OBS switches to set up the virtual path based on predefined traffic parameters. The corresponding data burst is sent at the edge without any acknowledgements indicating the success or failure of the path setup. With such one-way signalling, OBS can achieve all-optical cut-through of each data burst while being subject to burst contention in case multiple data bursts intend to reserve a common switch fabric at the same time [2].

There are two types of OBS networks, the barebone OBS and the OBS with burst contention resolution [12]. Barebone OBS does not employ any burst contention resolution scheme such as burst retransmission, fiber delay lines, or deflection routing [1]. Burst losses are permanent and are not subject to recovery within the OBS domain. On the other hand, OBS with burst contention resolution attempts to recover non-congestion burst losses. Often, burst retransmission and

deflection routing are employed. In our study we consider burst retransmission instead of any other contention resolution schemes because it is easy to implement compared to other contention resolution schemes such as fiber delay lines or deflection routing [1] and generates traceable retransmission values.

In traditional packet-switched networks, IP packets are stored and forwarded at each intermediate router, as such the network congestion arises as a result of buffer overflow. Thus, a packet-drop event indicates network congestion. However, barebone OBS does not rely on the usage of network internal queuing or pre-routing calculations. Furthermore, packet loss occurs due to random burst contention, which does not necessarily reflect the true status of network congestion; as such, TCP could interpret the occurrence of a packet drop event improperly [1].

From the TCP perspective, network congestion is detected either when the TCP observes packet losses and/or packet delays. Recall from [1,3], segment aggregation causes the increase of round-trip time (RTT), and may impose some vicious effects on TCP congestion identification. TCP flows with high access-network bandwidth suffer from a timeout (TO) as a result of a single burst loss because the TCP segments of the entire congestion window (w) that are assembled in the burst are lost. On the other hand, TCP flows with low to medium access-network bandwidth collect triple duplicates (TD) since only a portion of a congestion window's TCP segments are assembled in the contended burst [4].

FAST TCP [4-6] uses the minimum measured RTT ($baseRTT$), the estimated average queuing delay (RTT), as well as the packet loss, to accurately estimate the available network bandwidth and the network congestion. FAST TCP has been selected for our study since it is beneficial for promoting data-intensive applications. It cleverly reacts to both packet loss and packet delay in network congestion identification. It has loss and delay components that incorporate to adjust the number of packets on-the-fly according to the network condition. Therefore, FAST TCP can be used as a reference model for dropping-based and delay-based TCPs. From the application perspective, FAST TCP is considered high-bandwidth type of flow and operates for a relatively long period which is expected to take an important role in some mission-critical applications such as grid and cloud computing applications. Through the study performed in [5], FAST TCP obtained the best throughput performance compared with

HSTCP [7], STCP [8], and Reno under IP-based networks.

When FAST TCP runs over OBS, the packet delay (primarily due to burst assembly and propagation delay) will not vary under a fixed source-routing scheme. Thus, FAST TCP can suffer from false-congestion identification when random burst losses occur in barebone OBS [9,10]. When FAST TCP runs over OBS with retransmission, the OBS link-layer can recover a dropped burst without notifying any upper layer (*i.e.*, TCP and IP). This comes at the expense of introducing additional delay for the TCP segments contained in that optical burst. When FAST TCP detects a sudden increase in RTT for the segments assembled in the burst that was retransmitted [12], it cannot tell whether the sudden increase in the RTT is due to network congestion or due to the retransmission in lightly-loaded OBS domain.

To summarize, FAST TCP may fall in false congestion detection as a response to burst losses and/or suddenly increased RTT due to reasons other than network congestion in both barebone OBS and OBS with retransmission. This makes FAST TCP subject to false congestion identification and may seriously impair the TCP throughput. Accordingly, experimental study in [11] indicated that FAST TCP is not suitable for OBS networks. Motivated by the interesting scenario, this paper investigates the FAST TCP over OBS networks. The paper studies the behavior and stability of FAST TCP over OBS networks. Our results show that in the absence of burst contention, FAST TCP is rather unstable over barebone OBS, where the window dynamics contain multiple eigenvalues at the limit of the stability range. The occurrence of burst contentions, although degrades FAST TCP throughput performance, can enhance the network stability. We also show that employing random burst contentions supported by burst retransmissions stabilizes FAST TCP over OBS and makes it suitable to operate. The study is positioned as the first step towards the problem by gaining deeper understanding of behavior and stability.

The rest of the paper is organized as follows. Section II highlights the related work and relevant stability analysis. In Section III, we model FAST TCP using the discrete-time model over barebone OBS and OBS with burst retransmission. In Section IV, we analyze local stability of FAST TCP over OBS. Section V analyzes FAST TCP global stability. Section VI shows numerical results and finally Section VII concludes the paper.

II. RELATED WORK

Analyzing TCP as a distributed feedback congestion-control mechanism located at the end-system that coordinates with the network core queues operating at the level of fluid-flow is considered a challenging problem [1]. It is even more challenging considering next-generation high-speed bufferless networks. Relevant to this study, the fluid modeling technique has added a new dimension for modeling large number of TCP flows. However, fluid modeling approach imposes strict assumptions, such as (1) having very large number of TCP flows, (2) with Poisson arrival of loss events, and (3) with strong correlation between losses in one RTT while being

independent among the other RTTs. Regarding the first assumption, there is no sufficient evidence that the number of TCP flows is sufficiently high at the OBS edge node. In OBS, since both random burst drops and dropping due to persistent congestion may occur, the second assumption is subject to further investigation. The third assumption can partially be justified as that in the barebone OBS the RTT is more or less fixed. This is because the third assumption can only hold for the TCP flows which can emit the TCP segments of their entire window while being assembled in one burst (e.g., fast flows [3]).

The synchronization model proposed in [13] benefits from the *ack*-clocking to include the burstiness factor in the fluid model. Note that the fluid model assumes that there is no burstiness and the TCP rates of different flows are differentiable. Therefore, it may take infinitely long time to converge. The synchronization approach has been used to model FAST TCP and obtain its stability in [13]. In order to obtain sufficient analysis for TCP performance while considering TCP stability, the synchronization modeling approach needs to capture the bufferless nature of OBS links (*i.e.*, fixed TCP RTTs), the burst aggregation factors, and the burst-loss distribution.

The stability analysis presented in [4-6] derived the conditions for local and global stability of FAST TCP using feedback delays. Generally, the system is stable when delays of the sources are small. In [14], the authors presented a scalable congestion control mechanism that is decentralized, stable at the equilibrium point, achieves high throughput, and satisfies fairness. They have showed that protocols based on Explicit Congestion Notifications (ECN) appear to have a good approximation of the above characteristics in large-bandwidth delay products. Extending this study to OBS domain raises a question; will ECN achieve similar characteristics in OBS networks? Simulation and modeling-based studies presented in [15-17] have tackled this question, but they lack a stability analysis.

In [18], authors studied multi-path routing and dual congestion control stability. They found that controlling traffic that split among outgoing links leads to oscillation instability. The authors focused on studying the stability of a system with single bottleneck network shared with traffic sources with heterogeneous delays. They showed that the system achieves stability after certain amount of time. They also proved global stability for the delay differential equations under certain constrains.

In the next section we present a stability model for FAST TCP over bufferless burst-switched networks.

III. THROUGHPUT MODEL FOR FAST TCP OVER OBS

FAST TCP is widely considered as a high-speed version of TCP Vegas that aims to maintain a constant number of packets in queues throughout the network. Similar to TCP Vegas, FAST TCP uses both average and the minimum measured RTT to estimate the number of packets in the network queues. The number of packets in the network queues is taken by the

TCP sender to determine whether it should increase or decrease the sending rate, and the way of sending rate adjustment distinguishes FAST TCP from TCP Vegas [1]. While TCP Vegas maintains fixed size adjustments to the rate, FAST TCP uses an adaptive mechanism to adjust the sending rate: it can dramatically increase the sending rate when the system is far from equilibrium state, while reducing the increment or decrement of sending rate when the system is close to the equilibrium state. It has been observed that FAST TCP can effectively improve the convergence speed and the stability [4-6]. The network model shown in Fig. 1 is adopted, while the various speeds of TCP flows are considered. Such TCP flow speeds are of our great interest due to the following three reasons: (1) the scenario of high speed TCP service provisioning on OBS networks is attractive and is envisioned to take an important role in future network applications, such as grid and cloud computing; (2) the fast TCP flows with high bandwidth are most vulnerable to false congestion identification in the OBS domain, which could fatally harm TCP throughput. Note that the loss of a burst simply results in a TCP timeout [1], which results in TCP going back to the slow-start (*i.e.*, $w=1$); (3) our stability analysis captures the correlation between FAST TCP flows that have their packets assembled in single bursts. Unlike the study in [5], depending on the flow speed (*e.g.*, fast, medium, or low [3]), if a burst is retransmitted, the induced delay will affect all the active flows which have their packets assembled in that burst [3]. Therefore, congestion windows for all sources will decrease or increase at the same time. Such correlation distinguishes the behavior of FAST TCP flows running over OBS from the ones running on buffer-oriented IP networks.

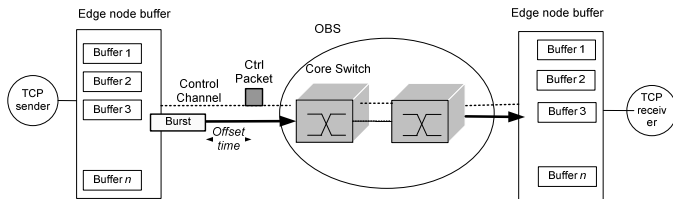


Fig. 1. The adopted model of FAST TCP over OBS networks

The following table lists the notations adopted in the modeling processes.

p	: Burst dropping probability
p_c	: Burst contention probability
p_{nc}	: Probability of no burst contention
p_{sr}	: Probability of a burst contended but successfully retransmitted
c_l	: Capacity of link l
α_i	: Number of packets in the queue for the i^{th} flow.
γ	: Constant between 0 and 1
d_i	: Round trip propagation delay of the i^{th} flow.
T_{b_i}	: Time-based threshold for burst assembly of the i^{th} flow.

RTT_i	: Measured round-trip time without retransmission of the i^{th} flow
$baseRTT_i$: Minimum measured RTT of the i^{th} flow
RTT_i^r	: RTT with retransmission of the i^{th} flow
R_i	: Measured retransmission delay of the i^{th} flow
RTT_i^f	: Forward feedback delay of the i^{th} flow
RTT_i^b	: Backward feedback delay of the i^{th} flow
X_i	: Sending rate of the i^{th} flow
I	: Identity matrix
K	: Routing matrix
w_i	: Congestion window for the i^{th} flow

In Fig. 1, the network has a set of l links with a finite capacity denoted as c_l . There exists a set of i FAST TCP flows. The FAST TCP agent updates its window in every fixed period. Let d_i denote the round-trip propagation delay, and T_{b_i} denote the total burst assembly delay at both of the OBS edge nodes. Let R_i denote the time required to perform burst retransmission for packets belonging to flow i . Let the congestion window of a source i at time t to be $w_i(t)$. Let the routing matrix $K_{li} = 1$ if the source i uses the link l and 0 otherwise. From Eq. (7) in [5], each FAST TCP flow maintains a certain number of packets α in the network at the equilibrium, which is defined as,

$$\alpha_i(w_i, R_i) = \begin{cases} \alpha_i w_i & \text{if } R_i = 0, \text{ barebone} \\ \alpha_i & \text{otherwise, retransmission} \end{cases} \quad (1)$$

According to Eq. (5) in [4], FAST TCP periodically updates the congestion window in every fixed period of time called update time:

$$w = \min \left\{ 2w, (1-\gamma)w + \gamma \left(\frac{baseRTT}{RTT} w + \alpha \right) \right\} \quad (2)$$

where $baseRTT$ is the minimum RTT observed, and RTT is the measured round trip time, $\gamma \in (0,1]$. The overall structure for the congestion control system is presented in Fig. 2. This structure is our congestion control reference used in the following sections.

A. FAST TCP over Barebone OBS

In a barebone OBS network, the network adopts a fixed source-routing scheme, such that the round-trip delay is primarily the sum of propagation and burst assembly delay, $RTT_i = baseRTT_i = d_i + T_{b_i}$, which does not vary in the presence of a steady traffic load. FAST TCP sending rate of a source i at time t is,

$$x_i(t) = \frac{w_i(t)}{RTT_i} \quad (3)$$

The aggregate rate at certain link l is,

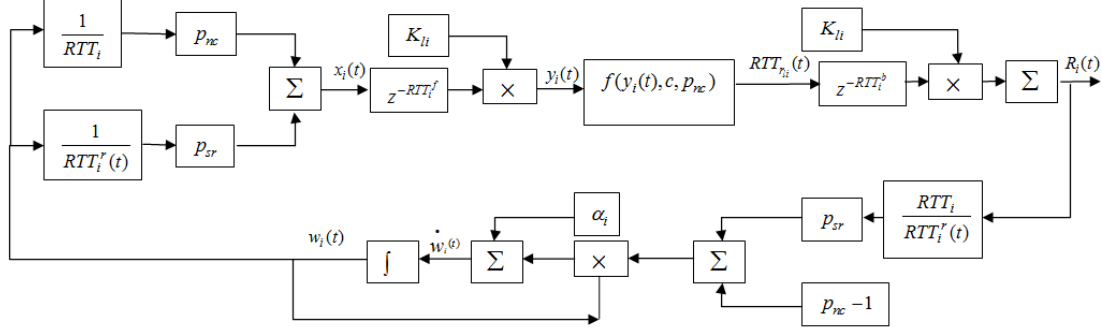


Fig. 2 Congestion Control Structure

$$y_i(t) = \sum_i K_{li} x_i(t) \quad (4)$$

In barebone OBS, $baseRTT$ and RTT are equal. We model Eq. (2) in discrete-time equation as,

$$w_i(t+1) = \gamma (w_i(t) + \alpha_i w_i(t)) + (1-\gamma)w_i(t) \quad (5)$$

We model Eq. (5) using continuous time model; by jointly considering Eq. (1),

$$\frac{dw_i(t)}{dt} = \gamma \alpha_i w_i(t) \quad (6)$$

From Eq. (6), FAST TCP can neither effectively boost the available bandwidth nor detect network congestion in barebone OBS networks. Furthermore, burst losses occur due to contentions even if the OBS network is not congested. In FAST TCP flows where all packets in the window of TCP flow are assembled into a single burst, FAST TCP suffers from false congestion detection if the burst collides with any other and is dropped at various traffic loads [9,10]. The sender then triggers timeout (TO) retransmission and enters the slow start phase, which significantly reduces the flow throughput.

B. FAST TCP over OBS with Burst Retransmission

When the OBS network is heavily loaded, random burst contentions frequently occur. OBS can recover some burst losses in the OBS domain by employing the burst retransmission mechanism at the edge nodes [12]. FAST TCP will often detect the increases in RTTs due to the burst retransmission. Hence, FAST TCP can only detect network congestion state based on the variation pattern of the measured RTTs during burst retransmissions.

Although a burst collision can be resolved in the OBS domain via retransmission and the packets assembled in the retransmitted burst may successfully arrive at their destination, the packets must experience a longer RTT. The delay of the packets in the retransmitted bursts is denoted by RTT_i^r :

$$RTT_i^r(t) = d_i + R_i(t) \quad (7)$$

where $R_i(t)$ is a random variable representing the additional delay due to the retransmission, which is obviously affected by

at which node the previous burst collided and was lost. In specific, the evaluation of $R_i(t)$ needs to consider the number of links that the previous burst has traversed and the burst buffering delay at the edge.

Similar to [10,12,15], we identify two types of successful rounds as follows: (1) the rounds that experience contention but in which bursts are successfully retransmitted, and (2) the rounds that do not experience burst contention. From [12], the probability of a successful round that experiences contention but in which bursts are successfully retransmitted is:

$$p_{sr} = \frac{p_c - p}{1 - p} \quad (8)$$

The probability of a successful round that does not experience burst contention can be calculated as

$$p_{nc} = \frac{1 - p_c}{1 - p} \quad (9)$$

Since burst retransmission introduces a sudden delay increase for the contending bursts that successfully reach the destination, FAST TCP reduces its window size according to Eq. (2), leading to reduced throughput. Thus, the FAST TCP sending rate can be expressed as:

$$x_i(t) = \left(\frac{w_i(t)}{RTT_i} p_{nc} + \frac{w_i(t)}{RTT_i^r(t)} p_{sr} \right) \quad (10)$$

The dynamic properties near the equilibrium of the throughput behavior described in Eq. (10), nonetheless, is nonlinear and intractable in most cases. Thus, in the following section we will first refine the nonlinear model in Eq. (10), followed by linearization of the model to make it easy manipulated. The effect of delays is essential to the proposed stability analysis, and should be included in the throughput model. For this purpose, we take into account the forward and backward delays in the propagation of delays. We define the forward feedback delay from source i to link l as RTT_i^f , and the backward feedback delay from link l to source i as RTT_i^b . The feedback delays RTT_i^f and RTT_i^b are time-varying in OBS with burst retransmission since the notification of burst loss could take place at any core node along the route with a specific physical distance to the edge node. Considering Fig.

1, it requires longer time to retransmit a burst that was dropped at the third OBS switch compared to the one dropped at the first switch. The aggregate rate at certain link l is:

$$y_l(t) = \sum_i K_{li} \left(\frac{w_i(t - RTT_i^f)}{RTT_i} p_{nc} + \frac{w_i(t - RTT_i^f)}{RTT_i^r (t - RTT_i^f)} p_{sr} \right) \quad (11)$$

Let $R_i(t)$ be the retransmission time of TCP source i over link l . Thus the round-trip delay for source i that experience burst retransmission can be expressed as:

$$R_i(t) = \sum_l K_{li} RTT_i^r (t - RTT_i^b) \quad (12)$$

Thus we can rewrite Eq. (5) as:

$$w_i(t+1) = \gamma \left(w_i(t) p_{nc} + w_i(t) \frac{RTT_i}{RTT_i^r} p_{sr} + \alpha_i \right) + (1 - \gamma) w_i(t) \quad (13)$$

in continuous time model and from Eq. (1), we have:

$$\frac{dw_i(t)}{dt} = \begin{cases} \gamma \left(w_i(t) p_{nc} + w_i(t) \frac{RTT_i}{RTT_i^r} p_{sr} + \alpha_i - w_i(t) \right) & p_{sr} > 0 \\ \gamma \alpha_i w_i(t) & p_{sr} = 0 \end{cases} \quad (14)$$

In FAST TCP, packet transmission is clocked at the same rate as the throughput of the received flows [3]. In buffer-oriented networks, the link queuing delay is determined implicitly by the sources' congestion window [5], while in OBS, the retransmission delay takes place independently from the sources' congestion window and the link capacity. However, from [13] the frequency of burst contention increases by approaching the link saturation. Therefore, the retransmission delay vector, $R_i(t) = R_{il}(t)$, for all links l is determined as,

$$\sum_i K_{li} \left(\frac{w_i(t - RTT_i^f)}{RTT_i} p_{nc} + \frac{w_i(t - RTT_i^f)}{RTT_i^r (t - RTT_i^f)} p_{sr} \right) \begin{cases} = c_l & \text{if } R_i > 0 \\ \leq c_l & \text{if } R_i = 0 \end{cases} \quad (15)$$

where R_i is given by Eq. (12).

IV. LOCAL STABILITY IN OBS

In this section, we investigate the local stability of FAST TCP over OBS. The main result of this study indicates sufficient conditions for local stability [14]. In particular, the role played by burst contention in stabilizing the network is highlighted. Indeed, the results indicate that FAST TCP over OBS with burst retransmission has dynamic modes bounded by those of FAST TCP over OBS barebone and those of FAST TCP with the retransmission delay. We show that burst contention improves the quality and stability of FAST TCP over OBS only in the presence of burst retransmission.

Theorem 1: FAST TCP over OBS with retransmission is

1. Locally stable, provided that p_{nc} given by Eq. (9) satisfies $0 < p_{nc} < 1$,
2. Locally exponentially asymptotically stable, If $R_i = \sigma RTT_i$ and $0 < p_{nc} < 1$.

When contention is not likely to occur, FAST TCP over OBS is unstable, for congestion window increases before reaching the limit, recall, Fig. 2.

Corollary 1: If $p_{nc} = 0$, FAST TCP over OBS with retransmission is unstable. The rest of this section is dedicated to prove Theorem 1. The proof takes the following steps:

1. Linearization of the system model described by Eqs. (8)-(13).
2. Expression of the linearized system model in a matrix form within the Z transform domain.
3. Study of the stability conditions for different values of probability of contention.

Define

$$\partial w_i(t - RTT_i^f) = w_i(t - RTT_i^f) - w_i(t) \quad (16)$$

$$\partial R_i(t - RTT_i^f) = R_i(t - RTT_i^f) - R_i(t)$$

By linearizing Eq. (15)

$$\sum_i K_{li} \left(\frac{\partial w_i(t - RTT_i^f)}{RTT_i} p_{nc} + \frac{\partial w_i(t - RTT_i^f)}{RTT_i^r} p_{sr} \right) - \sum_i K_{li} \left(\frac{w_i}{(RTT_i^r)^2} \partial R_i(t - RTT_i^f) p_{sr} \right) = 0 \quad (17)$$

Using

$$(K_f(z))_{li} = \begin{cases} z^{RTT_i^f} & \text{if } K_{li} = 1 \\ 0 & \text{if } K_{li} = 0 \end{cases} \quad \text{and} \quad (K_b(z))_{li} = \begin{cases} z^{RTT_i^b} & \text{if } K_{li} = 1 \\ 0 & \text{if } K_{li} = 0 \end{cases}$$

, and

$$L(z) = (1 - \gamma) I - \gamma \left((p_{nc} I + p_{sr} M) - DBK_b^T(z) \right) \left(K_f(z) BK_b^T(z) \right)^{-1} K_f(z) D^{-1} (p_{nc} I + p_{sr} M) \quad (18)$$

where the matrices D, M, C, and B can be defined as follows:

$$D = \text{diag}(RTT_i), \quad M = \text{diag} \left(\frac{RTT_i}{RTT_i^r} \right), \quad M = DC^{-1},$$

$$C = \text{diag}(RTT_i^r), \quad B = \text{diag} \left(\frac{w_i}{(RTT_i^r)^2} \right)$$

$$K_f(z) D^{-1} (p_{nc} I + Mp_{sr}) W(z) - p_{sr} K_f(z) B R(z) = 0 \quad (19)$$

Using Eq. (12), $R(z) = R_b^T(z) RTT^r(z)$

$$\begin{pmatrix} I & -K_b^T(z) \\ p_{sr} K_f(z) B & 0 \end{pmatrix} \begin{pmatrix} R(z) \\ RTT^r(z) \end{pmatrix} = \begin{pmatrix} 0 \\ K_f(z) D^{-1} (p_{nc} I + Mp_{sr}) \end{pmatrix} W(z) \quad (20)$$

From Eq. (19) an input-output relationship between the transmission delay at the source and the window size can be obtained,

$$\frac{R(z)}{W(z)} = K_b^T(z) \left(K_f(z) B K_b^T(z) \right)^{-1} K_f(z) D^{-1} \left(\frac{p_{nc}}{p_{sr}} I + M \right) \quad (21)$$

On the other hand, Eq. (13) can be written as,

$$w_i(t+1 - RTT_i^f) = \gamma w_i(t - RTT_i^f) \left(p_{nc} + \frac{RTT_i}{RTT_i^r(t - RTT_i^f)} p_{sr} \right) \quad (22)$$

$$+ \gamma \alpha_i w_i(t - RTT_i^f) + (1 - \gamma) w_i(t - RTT_i^f)$$

Linearizing Eq. (21) leads to

$$\delta w_i(t - RTT_i^f + 1) = \gamma \left(p_{nc} + \frac{RTT_i}{RTT_i^r(t - RTT_i^f)} p_{sr} \right)$$

$$\delta w_i(t - RTT_i^f) - \gamma \frac{w_i(RTT_i)}{(RTT_i^r)^2} p_{sr} \delta R_i(t - RTT_i^f)$$

$$+ (1 - \gamma) \delta w_i(t - RTT_i^f)$$

Using Z-transform, the matrix representation of this equation follows:

$$zW(z) = \gamma \left((p_{nc} I + Mp_{sr})W(z) - DBp_{sr} R(z) \right) + (1 - \gamma)W(z) \quad (23)$$

$$zW(z) = \gamma (p_{nc} I + Mp_{sr})W(z) - \gamma DBK_b^T(z) \left(K_f(z) BK_b^T(z) \right)^{-1} K_f(z) D^{-1} (p_{nc} I + p_{sr} M)W(z) + (1 - \gamma)W(z) \quad (24)$$

Stability of $W(z)$ depends on the dynamic term,

$$L(z) = \gamma (p_{nc} I + Mp_{sr}) - \gamma DBK_b^T(z) \left(K_f(z) BK_b^T(z) \right)^{-1} K_f(z) D^{-1} (p_{nc} I + p_{sr} M) + (1 - \gamma)I \quad (25)$$

Eq. (25) can be seen as a convex weighted sum of two terms based on the probability of contention.

$$L(z) = p_{nc} J_1(z) + p_{sr} J_2(z) = J_1(z) (p_{nc} I + p_{sr} M), \quad (26)$$

where,

$$J_1(z) = \gamma \left[I - DBK_b^T(z) \left(K_f(z) BK_b^T(z) \right)^{-1} K_f(z) D^{-1} \right] + (1 - \gamma)I \quad (27)$$

and

$$J_2(z) = \gamma \left[M - DBK_b^T(z) \left(K_f(z) BK_b^T(z) \right)^{-1} K_f(z) D^{-1} M \right] + (1 - \gamma)I \quad (28)$$

Let $\Gamma_{J_\ell}^\gamma$ represents the set of eigenvalues of $J_\ell(z)$ and $\Gamma_{J_\ell}^\gamma$ is the one when $\gamma = 1$, $\ell = 1, 2$.

Lemma 1: When $p_{nc} = 0$ and $\gamma = 1$, the eigenvalues of $L(z)$ have the following properties :

1. There are L zero eigenvalues with the corresponding eigenvectors as the columns of matrix $DBK_b^T(z)$
2. The only nonzero eigenvalues are equal to $\lambda = 1$.

Proof: see appendix.

Lemma 2: When $\gamma = 1$ and

$p_{nc} = 0$, $p_{sr} = 1$, $z = e^{j\omega}$, with $\omega \in [0, 2\pi)$, the eigenvalues of $L(z)$ have the following properties :

1. There are L zero eigenvalues with the corresponding eigenvectors as the columns of matrix $M^{-1}DBK_b^T(z)$
2. The nonzero eigenvalues have moduli less than 1 if $R_{\max} - R_{\min} < 1/4$, where $R_{\max} = \max_i R_i$ and $R_{\min} = \min_i R_i$
3. If R_i is multiple of $baseRTT$, $R_i = \sigma(RTT_i)$, where $\sigma > 0$, then all the nonzero eigenvalues are equal to $\lambda = \frac{1}{\sigma + 1}$.

Proof: see appendix.

Lemma 3: When $\gamma = 1$ and

$0 < p_{nc} < 1$, $p_{sr} + p_{nc} = 1$, $z = e^{j\omega}$, with $\omega \in [0, 2\pi)$, the eigenvalues of $L(z)$ have the following properties :

1. There are L zero eigenvalues with the corresponding eigenvectors as the columns of matrix $(p_{nc} I + p_{sr} M)^{-1} DBK_b^T(z)$
2. The nonzero eigenvalues have moduli less than 1 if $R_{\max} - R_{\min} < 1/4$, where $R_{\max} = \max_i R_i$ and $R_{\min} = \min_i R_i$
3. If R_i is multiple of $baseRTT$, $R_i = \sigma(RTT_i)$, where $\sigma > 0$, then all the nonzero eigenvalues are equal to $\lambda = \frac{\sigma p_{nc} + 1}{\sigma + 1} < 1$.

Proof: see appendix.

V. GLOBAL STABILITY IN OBS

In this section, we analyze the global stability of FAST TCP over OBS. The analysis of global stability will be addressed under the assumption that the network is composed of one link and that the feedback delays are negligible [5]. Under this assumption, Eq. (15) can be reduced,

$$\sum_i \left(\frac{p_{nc}}{RTT_i} + \frac{p_{sr}}{RTT_i + R_i(t)} \right) w_i(t) = \begin{cases} = c_l & \text{if } R_{i_l} > 0 \\ \leq c_l & \text{if } R_{i_l} = 0 \end{cases} \quad (29)$$

Where

$$w_i(t+1) = \gamma \left(w_i(t) p_{nc} + w_i \frac{RTT_i}{RTT_i^r} p_{sr} + \alpha_i \right) + (1 - \gamma) w_i(t) \quad (30)$$

Define,

$$Y(t) = \sum_i \left(\frac{w_i(t)}{RTT_i} \right) \quad (31)$$

Which represents the aggregated load of barebone FAST TCP over OBS (i.e. $p_{nc} = 1$ and $p_{sr} = 0$).

Lemma 4: There exist $k_l > 0$ such that the following are true for all $t > k_l$

1. With probability 1 , p_{nc} will be different
2. $v(t+1) = (1 - \gamma)v(t)$, (32a)

where

$$v(t) = Y(t) - \left(\sum_i \frac{\alpha_i}{RTT_i} + c \right) \quad (32b)$$

Proof:

$$Y(t+1) = \sum_i \left(\frac{w_i(t+1)}{RTT_i} \right) \quad (33)$$

Using Eq. (30),

$$Y(t+1) = \sum_i \frac{w_i(t+1)}{RTT_i} = \gamma \sum_i \left(\frac{w_i(t)}{RTT_i} p_{nc} + w_i(t) \frac{1}{RTT_i^r} p_{sr} + \frac{\alpha_i}{RTT_i} \right) + (1-\gamma) \sum_i \frac{w_i(t)}{RTT_i}$$

$$Y(t+1) = \gamma \sum_i \left(\frac{w_i(t)}{RTT_i} p_{nc} + w_i(t) \frac{1}{RTT_i^r} p_{sr} \right) + \gamma \sum_i \frac{\alpha_i}{RTT_i} + (1-\gamma) \sum_i \frac{w_i(t)}{RTT_i}$$

$$Y(t+1) \leq (1-\gamma)Y(t) + \gamma c + \gamma \sum_i \frac{\alpha_i}{RTT_i}$$

In particular, at initial startup the value of $p_{nc}=1$ and therefore initially we have,

$$Y(t+1) = Y(t) + \gamma \sum_i \frac{\alpha_i}{RTT_i}$$

$$Y(t+1) = Y(t_0) + \gamma(t+1) \sum_i \frac{\alpha_i}{RTT_i}$$

This is linearly increasing as a function of t and will reach the link capacity c at,

$$Y(t) = Y(t_0) + (k_1)\gamma \sum_i \frac{\alpha_i}{RTT_i} = c$$

$$k_1 = \frac{c - Y(t_0)}{\gamma \sum_i \frac{\alpha_i}{RTT_i}}$$

In such case, $p_c=1$ and therefore p_{nc} is close to 0.

$$v(t) = Y(t) - \sum_i \frac{\alpha_i}{RTT_i} - c \quad (34)$$

If Eq. (29) is verified with equality for a given value of $p_{nc} < 1$ then,

$$Y(t+1) = (1-\gamma)Y(t) + \gamma c + \gamma \sum_i \frac{\alpha_i}{RTT_i}$$

$$v(t+1) = Y(t+1) - \sum_i \frac{\alpha_i}{RTT_i} - c$$

$$= (1-\gamma)Y(t) + \gamma c + \gamma \sum_i \frac{\alpha_i}{RTT_i} - \sum_i \frac{\alpha_i}{RTT_i} - c$$

$$v(t+1) = (1-\gamma)v(t)$$

The retransmission time $R_i(t)$ varies from zero, as a low limit, and up to theoretically infinity. However, according to the FAST TCP, the retransmission time cannot exceed two *baseRTT* after which the packet will be dropped, therefore,

$$R_{\min} = \min_i R_i, \quad R_{\max} = \max_i R_i \quad \text{and} \quad R_i(t) > 0$$

The values of R_{\min} and R_{\max} are both bounded. For instance, in case of FAST TCP over OBS, $R_{\min} = \min RTT_i$ and $R_{\max} = \max RTT_i$. Furthermore, we define,

$$\mu_i(t) = \frac{R_i(t)}{RTT_i + R_i(t)} = \frac{R_i(t)}{RTT_i^r(t)}, \quad 0 \leq \mu_i(t) \leq 1 \quad \text{for all } t.$$

$$\eta_i(t) = \frac{p_{sr} w_i(t) - \alpha_i}{\alpha_i RTT_i} - \frac{1}{R_i(t)}$$

$$\eta_{\min}(t) = \min_i \eta_i(t), \quad \eta_{\max}(t) = \max_i \eta_i(t)$$

Lemma 5: In case $p_{sr} \neq 0, (p_{nc} < 1)$, two positive numbers, δ_1 and δ_2 , exist such that,

$$\eta_{\min}(t) < \delta_1 (1-\gamma)^t, \quad \eta_{\max}(t) > -\delta_2 (1-\gamma)^t$$

Proof: Using Eq. (34),

$$v(t) = Y(t) - \left(\sum_i \frac{\alpha_i}{RTT_i} + c \right)$$

$$v(t+1) = Y(t+1) - \left(\sum_i \frac{\alpha_i}{RTT_i} + c \right)$$

$$v(t+1) - v(t) = Y(t+1) - Y(t) \quad (\text{using Eq. (32a)})$$

$$= \gamma v(t) = \gamma(1-\gamma)v(t-1) = \gamma(1-\gamma)^{t-k_2} v(k_2)$$

$$Y(t+1) - Y(t) < \gamma(1-\gamma)^{t-k_2} |v(k_2)|$$

$$Y(t+1) - Y(t) < \kappa(1-\gamma)^t \quad (35)$$

Where $\kappa = \gamma(1-\gamma)^{-k_2} |v(k_2)|$, k_2 is a time index, and $t > k_2$. In FAST TCP over OBS, the update of the congestion window measured at two successive instants of time,

$$w_i(t+1) - w_i(t) = \gamma \left[-w_i(t) + p_{nc} w_i(t) + p_{sr} w_i(t) \frac{RTT_i}{RTT_i^r(t)} + \alpha_i \right]$$

$$= \gamma \left[-p_{sr} w_i(t) + p_{sr} w_i(t) \frac{RTT_i}{RTT_i^r(t)} + \alpha_i \right]$$

$$w_i(t+1) - w_i(t) = \gamma \frac{R_i(t)}{RTT_i^r(t)} \left[\frac{RTT_i^r(t)}{R_i(t)} (\alpha_i - p_{sr} w_i(t)) + p_{sr} w_i(t) \frac{RTT_i}{R_i(t)} \right]$$

$$= -\gamma \frac{R_i(t) \alpha_i RTT_i}{RTT_i^r(t)} \left[\frac{(p_{sr} w_i(t) - \alpha_i)}{\alpha_i RTT_i} - \frac{1}{R_i(t)} \right]$$

$$= -\gamma \mu_i \alpha_i \eta_i R_i(t)$$

(36)

where,

$$\mu_i = \frac{RTT_i}{RTT_i^r(t)}; \quad \eta_i = \frac{(p_{sr} w_i(t) - \alpha_i)}{\alpha_i RTT_i} - \frac{1}{R_i(t)}$$

In particular, for $p_{sr} = 0, (p_{nc} = 1)$, $\eta_i = -\frac{RTT_i^r(t)}{R_i(t)RTT_i}$

and Eq. (36) will be reduced to the additive increase model

$$w_i(t+1) - w_i(t) = \gamma \alpha_i$$

which is strictly increasing as previously indicated. On the other hand, if $p_{sr} \neq 0, (p_{nc} < 1)$, then the equilibrium is reached when,

$$w_i^*(t) = \frac{RTT_i^r \alpha_i}{R_i p_{sr}} \quad (37)$$

Eq. (37) could be verified by equating Eq. (14) to 0. Select δ_1 such that,

$$\delta_1 N \gamma \mu_{\min} \alpha_{\min} \frac{R_{\min}}{RTT_{\max}} > \kappa \quad (38)$$

Assuming that a time $t \geq k_2$ such that $\eta_{\max}(t) \leq -\delta_1(1-\gamma)^t$ exists, the one step ahead increment change of $Y(t)$ is given by

$$\begin{aligned} Y(t+1) - Y(t) &= \sum_i \frac{w_i(t+1) - w_i(t)}{RTT_i} \\ &= \sum_i \frac{(-\eta_i) \gamma \mu_i \alpha_i R_i}{RTT_i} \\ &\geq \frac{\gamma \mu_{\min} \alpha_{\min} R_{\min}}{RTT_{\max}} \sum_i (-\eta_i) \\ &\geq \frac{\gamma \mu_{\min} \alpha_{\min} R_{\min}}{RTT_{\max}} N \eta_{\max} \geq \delta_1 N (1-\gamma)^t \frac{\gamma \mu_{\min} \alpha_{\min} R_{\min}}{RTT_{\max}} \\ &\geq \kappa (1-\gamma)^t, \end{aligned}$$

which contradicts Eq. (35). Therefore, $\eta_{\max}(t) > -\delta_1(1-\gamma)^t$.

Similar steps can be followed to prove that $\eta_{\min}(t) < \delta_2(1-\gamma)^t$. $\eta_i(t)$ represents the most important term in the update rate of the congestion window for it decides about the existence of the equilibrium point as well as the stability. The following Lemma shows that the difference between $\eta_{\max}(t) - \eta_{\min}(t)$ is exponentially bounded.

Lemma 6: Let $L(t) = \eta_{\max}(t) - \eta_{\min}(t)$, and δ_3 and δ_4 , two positive numbers, such that for $t \geq k$ and for $p_{sr} \neq 0, (p_{nc} < 1)$. the following inequalities are verified.

- 1) $L(t) \geq 0$
- 2) $L(t+1) \leq (1 - p_{sr}\gamma + p_{sr}\gamma\mu_{\max})L(t)$
- 3) $L(t) \leq \delta_4 (1 - p_{sr}\gamma + p_{sr}\gamma\mu_{\max})^t$

Proof: See appendix.

Lemma 7: For $p_{sr} \neq 0, (p_{nc} < 1)$, and $L(t) = \eta_{\max}(t) - \eta_{\min}(t)$, both $\eta_{\max}(t)$ and $\eta_{\min}(t)$ converge to zero exponentially.

Proof:

$$\begin{aligned} \eta_{\max}(t) &= L(t) + \eta_{\min}(t) \\ &< \delta_4 (1 - p_{sr}\gamma + p_{sr}\gamma\mu_{\max})^t + \delta_1 (1-\gamma)^t \\ -\delta_2 (1-\gamma)^t &< \eta_{\max}(t) < \delta_4 (1 - p_{sr}\gamma + p_{sr}\gamma\mu_{\max})^t + \delta_1 (1-\gamma)^t \end{aligned}$$

Theorem 2: On single link and neglecting feedback delays, FAST TCP over OBS is

- 1) Unstable and no finite equilibrium point exists if $p_{sr} = 0, (p_{nc} = 1)$,
 - 2) Globally asymptotically stable, if $p_{sr} \neq 0, (p_{nc} < 1)$,
- with $w_i^*(t) = \frac{RTT_r \alpha_i}{R_i p_{sr}}$

VI. FAST TCP SIMULATION OVER OBS

Simulation is conducted using NS-2, where the NSF network topology shown in Fig. 3 is adopted as the OBS core network. The distance along each link is in *km*. Each link has 8 wavelengths operating at 10 *Gbps*. Each link consists of one bi-directional control channel used for the control signalling. The control header processing time is set to be 1 μsec . The core nodes implement the LAUC-VF channel scheduling algorithm [2]. In order to obtain precise protocol performance results, we obtain our simulation results from FAST TCP flows originating at edge node E_1 , and leaves at edge node E_{14} . The total number of multiplexed flows at the edge node is 10^3 . Packets are routed from E_1 to E_{14} passing by core switches 8 and 9. The edge nodes, E_1 and E_{14} , are connected to the network core nodes with a 10ms propagation delay.

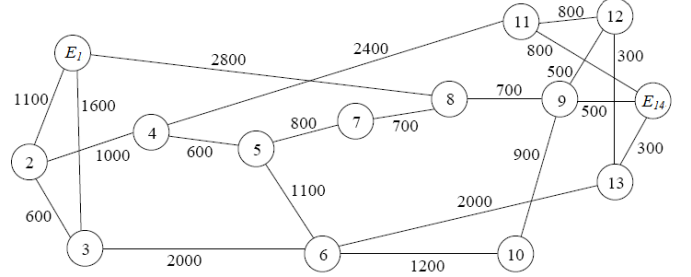


Fig. 3. The network topology adopted in the simulation.

In order to simulate fast flows [3], we adapted the time-based burst assembly algorithm, where the burst timeout threshold is set to 6ms. Also, a high access bandwidth, 100Mbps, is allocated to each flow. It has been shown that fast flows are severely affected by random burst losses and burst delays. In the presence of burst loss, fast flows lose the entire congestion window which results into timeout. In the presence of sudden increase in RTT, the entire congestion window will be affected. We anticipate that medium to slow flows will be much less affected compared to fast flows. The packet delay in the access network is set to be constant such that the effect due to the burst retransmission delay is the only reason for longer round-trip time of each packet. In the scenario of burst retransmissions, a burst subject to any contention is allowed to be retransmitted only once in order to have the best chance of meeting the timeout threshold of FAST TCP. Retransmission is triggered when the piggyback control packet notifies the edge node of a failure in reserving the fabric. Random burst contention phenomena occurs at core nodes only. The average

RTT between E_1 and E_{14} in the simulation is approximately 100ms.

In the simulation, FAST TCP senders and receivers are attached to the OBS edge nodes E_1 and E_{14} respectively. Other TCP flows were generated from every network edge to cause random burst contention. The random burst contention probability p_c ranges in $[10^{-5}, 10^{-2}]$. An FTP application is initiated to generate TCP segments with an average size of 512B. In all experiments, the maximum window size of TCP is 10^3 segments. In order to ensure that FAST TCP traffic fills the link bandwidth, we extended the simulation time to 10^3 s.

In Fig. 4 we show the average burst RTT in the presence of different burst contention probabilities. We observed that in the presence of light network congestion (*i.e.*, low burst contention probability), the average burst RTT increased as some bursts were successfully delivered after the first attempt of burst retransmission. However, in the presence of high network congestion, the average burst RTT dropped as the bursts are often lost in the network and failed to be successfully received after the first retransmission attempt.

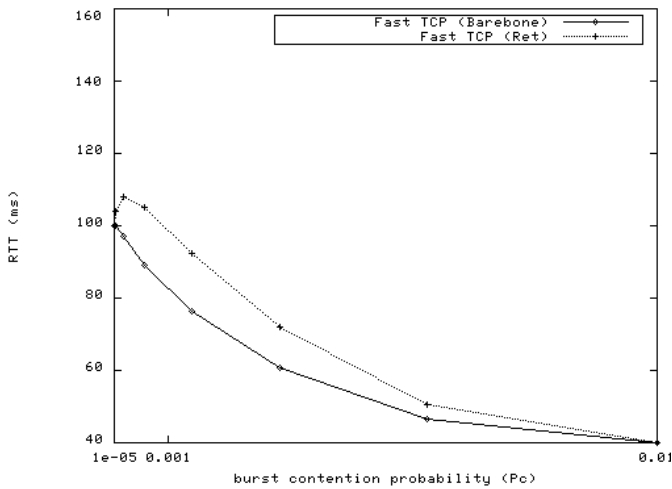


Fig. 4. FAST TCP RTT vs. burst contention probability in both barebone and OBS with burst retransmission

In Fig. 5, we provide a closer insight on FAST TCP congestion window evolution in the presence of different burst contention probabilities. We observed that in the presence of light network congestion, the congestion window stabilizes since the underlying OBS network can successfully retransmit the bursts. On the other hand, in barebone OBS, FAST TCP congestion window suffers from a significant and frequent losses when the burst contention probability increases. This is due to the fact that in barebone OBS, the burst contention probability resulted in an immediate burst loss.

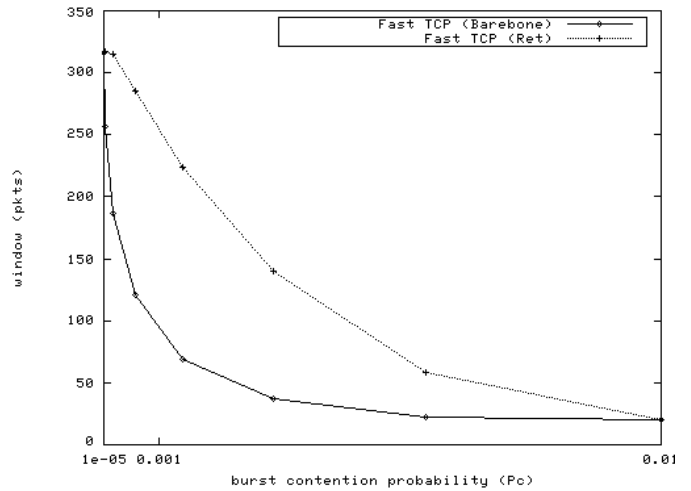


Fig. 5. FAST TCP congestion window vs. burst contention probability in both barebone and OBS with burst retransmission

In the following simulation studies, we observe the throughput of FAST TCP fast flows over a barebone OBS network and OBS with burst retransmission. In Fig. 6, the results clearly demonstrate that FAST TCP failed to utilize the bandwidth efficiently over barebone OBS network. While with one attempt of burst retransmission, burst loss was solidly reduced. Such a decrease is contributed by a better precision on the detection of congestion state in the network, which avoids triggering the congestion avoidance mechanism. Furthermore, we observe from Fig. 7 that reducing burst losses while varying RTT over OBS with burst retransmission can significantly improve the stability of FAST TCP and achieve higher throughputs. It is clear that the FAST TCP congestion window has a much smoother transition which supports the analysis concluded by *Theorem 1* and *Corollary I* in section IV.

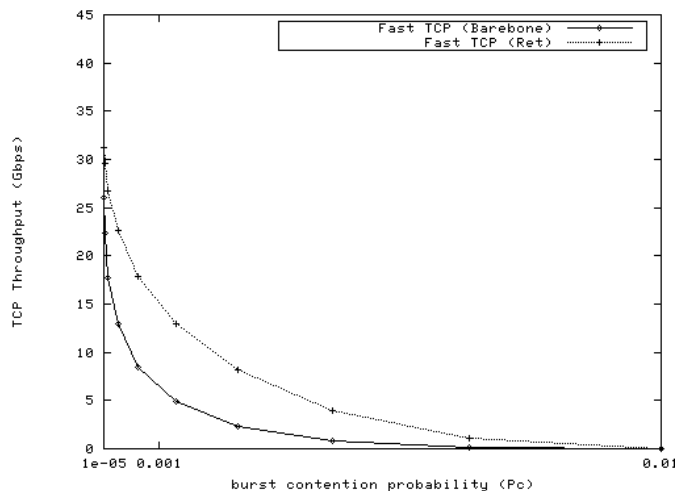


Fig. 6. FAST TCP throughput over OBS network

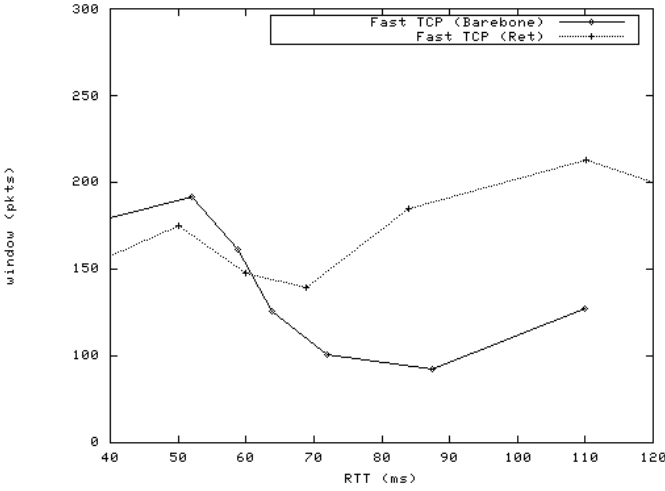


Fig. 7. FAST TCP congestion window vs. RTT in both barebone and OBS with burst retransmission

Fig. 8 shows the evolution of FAST TCP congestion window over barebone OBS and OBS with burst retransmission at different burst loss probabilities. We observe that FAST TCP has smoother window transition over OBS with burst retransmission. This verifies the better capability by using the FAST TCP in dealing with false congestion identification due to random burst contention in OBS network, where a higher throughput can be achieved. Also, the result confirms that FAST TCP stabilizes well in the presence of random contentions and random variation of RTT.

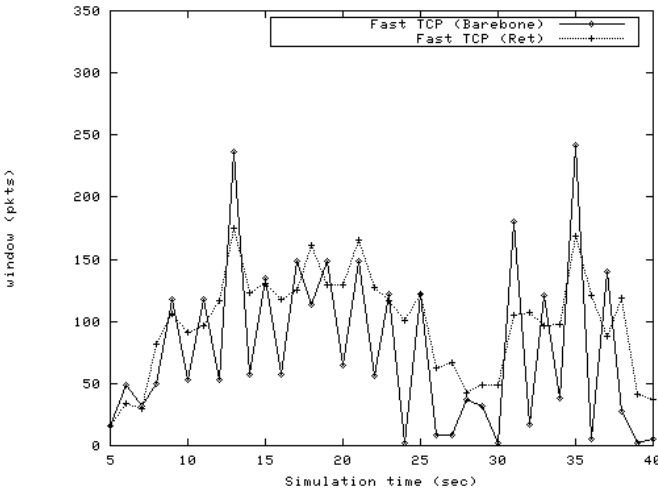


Fig. 8. FAST TCP congestion window over barebone OBS and OBS with burst retransmission

Fig. 8 illustrates the dynamic stability of the congestion window. One can observe the amplitude of oscillations of the FAST TCP over barebone OBS. This instability affect the regularity of the network performance and, in many occasions, forces the congestion window to be zero hence dropping all the packets. In the FAST TCP over OBS with retransmission case, the congestion window does not drop to zero and manage to reach a minimum required size to still be able to

handle the traffic in the network. The stability property of the FAST TCP over OBS with retransmission makes the oscillations less frequent with smaller amplitude giving a smoother and regular performance of the network.

VII. CONCLUSIONS

In this work, we introduced a model and stability analysis of FAST TCP over OBS networks. The network stability was analyzed, which led to the conclusion that the occurrence of random burst contention, although degrades the throughput performance, can enhance the network stability. In the absence of contention, our results indicated that FAST TCP is rather unstable over barebone OBS, where the window dynamics contains multiple eigenvalues at the limit of the stability range. We have also found that FAST TCP over OBS with retransmission is locally, exponentially, and asymptotically stable. Numerical results showed that the burst transmission has yielded remarkable enhancements on the congestion window size and network throughput in presence of burst contention. It also significantly contributes to the FAST TCP stability.

APPENDIX

Proof Lemma 1:

$$\text{At } \gamma = 1, p_{nc} = 1 (p_{sr} = 0),$$

$$L(z) = J_1(z) = (I - DBK_b^T(z) (K_f(z)BK_b^T(z))^{-1} K_f(z) D^{-1})$$

Multiplying both sides by $DBR_b^T(z)$

$$J_1(z)DBR_b^T(z) = (DBR_b^T(z) - DBR_b^T(z)) = 0 \quad (A1)$$

Let

$$A = L(z) - \lambda I$$

$$A = (1 - \lambda) I - DBK_b^T(z) (R_f(z)BK_b^T(z))^{-1} K_f(z) D^{-1}$$

$$|A| = |(1 - \lambda) I - DBK_b^T(z) (K_f(z)BK_b^T(z))^{-1} K_f(z) D^{-1}| = 0$$

This is also true for

$$|(K_f(z)BK_b^T(z))|.$$

$$|(1 - \lambda) I - DBK_b^T(z) (K_f(z)BK_b^T(z))^{-1} K_f(z) D^{-1}| = 0 \quad (A2)$$

Using the Schur complement, the above equation is equivalent

$$\begin{vmatrix} K_f(z)BK_b^T(z) & K_f(z) D^{-1} \\ DBK_b^T(z) & (1 - \lambda) I \end{vmatrix} = 0 \quad (A3)$$

If $(1 - \lambda) I$ is singular, then $\lambda = 1$. Otherwise, the determinant in Eq. (A3) is equal to

$$\begin{aligned} & |(1 - \lambda) I| \cdot |K_f(z) [B - D^{-1}((1 - \lambda) I)^{-1} DB] K_b^T(z)| = \\ & -(1 - \lambda)^n \left(\frac{\lambda}{1 - \lambda} \right)^n |K_f(z)BK_b^T(z)| \neq 0 \end{aligned} \quad (A4)$$

Therefore, $\lambda = 1$ is the unique nonzero eigenvalue.

Proof of Lemma 2:

$$\text{At } \gamma = 1, p_{nc} = 0; p_{sr} = 1$$

$$L(z) = J_2(z) = \left[I - DBK_b^T(z) (K_f(z)BK_b^T(z))^{-1} K_f(z) D^{-1} \right] M$$

Multiplying both sides by $M^{-1}DBK_b^T(z)$

$$J_2(z)M^{-1}DBK_b^T(z) = J_1(z)DBK_b^T(z) = 0$$

Let $A = J_2(z) - \lambda I$

$$|A| = \left| (M - \lambda I) - DBK_b^T(z) (K_f(z)BK_b^T(z))^{-1} K_f(z) D^{-1} M \right| = 0 \quad (\text{A5})$$

This is also true for

$$|(K_f(z)BK_b^T(z))|. \quad (\text{A6})$$

$$\left| (M - \lambda I) - DBK_b^T(z) (K_f(z)BK_b^T(z))^{-1} R_f(z) D^{-1} M \right| = 0$$

Using the Schur complement, the above equation is equivalent to

$$\begin{vmatrix} K_f(z)BK_b^T(z) & K_f(z) D^{-1} M \\ DBK_b^T(z) & (M - \lambda I) \end{vmatrix} = 0. \quad (\text{A7})$$

If $(M - \lambda I)$ is singular, then $\lambda = \frac{RTT_i}{RTT_i^r}$ for $M = \text{diag} \left[\frac{RTT_i}{RTT_i^r} \right]$.

Therefore $\lambda = \frac{RTT_i}{RTT_i^r} < 1$, for $R_i > 0$. If $R_i \xrightarrow{w.p.1} 0$, then $\lambda \xrightarrow{w.p.1} 1$.

If $R_i \approx \sigma(RTT_i)$, then $\lambda = \frac{1}{\sigma + 1} < 1$. Otherwise, if $(M - \lambda I)$ is

nonsingular, the determinant in (34) can be computed as

$$|M - \lambda I| \cdot |K_f(z) [B - D^{-1}M(M - \lambda I)^{-1}DB] K_b^T(z)| = 0 \quad (\text{A8})$$

Which leads to

$$|K_f(z) [B - M(M - \lambda I)^{-1}B] K_b^T(z)| = 0 \quad (\text{A9})$$

Using the identity $P(P - I)^{-1} = I + (P - I)^{-1}$, with $P = \frac{M}{\lambda}$,

Eq. (A9) becomes

$$|K_f(z) \left[\left(I - \frac{M}{\lambda} \right)^{-1} B \right] K_b^T(z)| = 0 \quad (\text{A10})$$

$$\lambda \left| K_f(z) \text{diag} \left[\left(\frac{1}{\lambda - \mu_i} \right) \beta_i \right] K_b^T(z) \right| = 0$$

As proven in [6], when $\gamma = 1$, $p_{nc} = 0$, $p_{sr} = 1$, and $z = e^{j\omega}$, with $\omega \in [0, 2\pi)$, the nonzero eigenvalues have moduli less than 1 if $R_{i\max} - R_{i\min} < \frac{1}{4}$, where

$R_{i\max} = \max_i R_i$, and $R_{i\min} = \min_i R_i$. Therefore the network is stable.

In particular, if $R_i = \sigma(RTT_i)$, then

$$\left| K_f(z) \text{diag} \left[\left(\frac{1}{\lambda - \mu_i} \right) \beta_i \right] K_b^T(z) \right| = \left(\frac{1}{\lambda - \frac{1}{\sigma + 1}} \right)^n |K_f(z)BK_b^T(z)| \neq 0 \quad (\text{A11})$$

Therefore, all nonzero eigenvalues are equal to $\lambda = \frac{1}{\sigma + 1} < 1$.

Proof of Lemma 3:

at $\gamma = 1$, $p_{nc} < 1$, $p_{sr} < 1$, and $p_{nc} + p_{sr} = 1$

Let $A = L(z) - \lambda I$. $L(z)$ also can be written as,

$$L(z) = p_{nc} J_1(z) + p_{sr} J_2(z) = J_1(z) (p_{nc} I + p_{sr} M)$$

Multiplying both sides of $L(z)$ by $(p_{nc} I + p_{sr} M)^{-1} DBK_b^T(z)$,

$$L(z) \left((p_{nc} I + p_{sr} M)^{-1} DBK_b^T(z) \right) = J_1(z) DBK_b^T(z) = 0$$

Columns of $\left((p_{nc} I + p_{sr} M)^{-1} DBK_b^T(z) \right)$

$$|A| = \begin{vmatrix} (p_{nc} - \lambda) I + p_{sr} M - \\ DBK_b^T(z) (K_f(z)BK_b^T(z))^{-1} K_f(z) D^{-1} (p_{nc} I + p_{sr} M) \end{vmatrix} = 0 \quad (\text{A12})$$

which is also true for

$$|(K_f(z)BK_b^T(z))|.$$

$$\begin{vmatrix} (p_{nc} - \lambda) I + p_{sr} M - \\ DBK_b^T(z) (K_f(z)BK_b^T(z))^{-1} K_f(z) D^{-1} (p_{nc} I + p_{sr} M) \end{vmatrix} = 0 \quad (\text{A13})$$

Using the Schur complement, the above equation is equivalent to

$$\begin{vmatrix} K_f(z)BK_b^T(z) & K_f(z) D^{-1} (p_{nc} I + p_{sr} M) \\ DBK_b^T(z) & (p_{nc} - \lambda) I + p_{sr} M \end{vmatrix} = 0, \quad (\text{A14})$$

which can be rearranged as

$$\begin{vmatrix} p_{nc} \begin{bmatrix} K_f(z)BK_b^T(z) & K_f(z) D^{-1} \\ DBK_b^T(z) & (1 - \lambda) I \end{bmatrix} + \\ (1 - p_{nc}) \begin{bmatrix} K_f(z)BK_b^T(z) & K_f(z) D^{-1} M \\ DBK_b^T(z) & (M - \lambda I) \end{bmatrix} \end{vmatrix} = 0 \quad (\text{A15})$$

As p_{nc} varies from 0 to 1, the eigenvalues loci, Eq. (A15), moves from within the circle of radius 1 to limit of stability ($\lambda = 1$). On the other hand,

if $p_{nc} (1 - \lambda) I + p_{sr} (M - \lambda I)$ is singular then

$$\lambda = p_{nc} + p_{sr} \frac{RTT_i}{RTT_i^r} = p_{nc} + (1 - p_{nc}) \frac{RTT_i}{RTT_i^r} \quad (\text{A16})$$

λ is a continuous strictly increasing real function of p_{nc} taking values in the interval $\left[\frac{RTT_i}{RTT_i^r}, 1 \right]$ and when

$$R_i = \sigma RTT_i, \lambda = \frac{p_{nc}\sigma + 1}{\sigma + 1}.$$

Otherwise, when $p_{nc}(1-\lambda)I + p_{sr}(M - \lambda I)$ is not singular, Eq. (43) is equivalent to

$$\left| K_f(z) \left[B - D^{-1} (p_{nc} I + p_{sr} M) (p_{nc} I + p_{sr} M - \lambda I)^{-1} DB \right] K_b^T(z) \right| = 0,$$

and using the identity $P(P-I)^{-1} = I + (P-I)^{-1}$, the determinant can be rearranged

$$\left| K_f(z) \left[\left(I - \frac{p_{nc} I + p_{sr} M}{\lambda} \right)^{-1} B \right] K_b^T(z) \right| = 0 \quad (A17)$$

$$\left| K_f(z) \text{diag} \left[\left(\frac{\lambda}{(\lambda-1)p_{nc} + p_{sr}(\lambda-\mu_i)} \right) \right] BK_b^T(z) \right| = 0$$

As proven in [5], when $\gamma = 1$ and

$p_{nc} = 0, p_{sr} = 1, z = e^{j\omega}$, with $\omega \in [0, 2\pi)$, the nonzero roots

have moduli less than 1 if $R_{\max} - R_{\min} < \frac{1}{4}$, where

$$R_{\max} = \max_i R_i, \text{ and } R_{\min} = \min_i R_i.$$

Particularly, when $R_i = \sigma RTT_i$

$$\left| K_f(z) \text{diag} \left[\left(\frac{\lambda}{\lambda - p_{nc} - p_{sr} \frac{1}{\sigma + 1}} \right) \right] BK_b^T(z) \right| =$$

$$\left(\frac{\lambda}{\lambda - p_{nc} - p_{sr} \frac{1}{\sigma + 1}} \right)^n \left| K_f(z) BK_b^T(z) \right| \neq 0 \quad (A18)$$

Therefore, the nonzero eigenvalues are all equal to

$$\lambda = \frac{p_{nc}\sigma + 1}{\sigma + 1} < 1.$$

Proof of Lemma 6:

$$\eta_i(t+1) = \frac{(p_{sr}w_i(t+1) - \alpha_i)}{\alpha_i RTT_i} - \frac{1}{R_i(t+1)}$$

$$\eta_i(t) = \frac{(p_{sr}w_i(t) - \alpha_i)}{\alpha_i RTT_i} - \frac{1}{R_i(t)}$$

$$\eta_i(t+1) - \eta_i(t) = \frac{(p_{sr}w_i(t+1) - \alpha_i)}{\alpha_i RTT_i} - \frac{1}{R_i(t+1)} - \frac{(p_{sr}w_i(t) - \alpha_i)}{\alpha_i RTT_i} + \frac{1}{R_i(t)}$$

$$\begin{aligned} \eta_i(t+1) - \eta_i(t) &= \frac{p_{sr}}{\alpha_i RTT_i} (w_i(t+1) - w_i(t)) - \frac{1}{R_i(t+1)} + \frac{1}{R_i(t)} \\ &= -p_{sr}\gamma \frac{R_i(t)}{RTT_i^r(t)} \eta_i - \frac{1}{R_i(t+1)} + \frac{1}{R_i(t)} \\ &= -p_{sr}\gamma \left(1 - \frac{RTT_i}{RTT_i^r(t)} \right) \eta_i(t) - \frac{1}{R_i(t+1)} + \frac{1}{R_i(t)} \end{aligned} \quad (A19)$$

Define

$$a_i = 1 - p_{sr}\gamma + p_{sr}\gamma \frac{RTT_i}{RTT_i^r(t)} = 1 - p_{sr}\gamma(1 - \mu_i(t)) \quad (A20)$$

$$0 < a_i \leq 1$$

then

$$\eta_i(t+1) = a_i \eta_i(t) - \frac{1}{R_i(t+1)} + \frac{1}{R_i(t)} \quad (A21)$$

Depending on the flow speed, if a delay occurred in the link due to retransmission, this delay will affect all the sources active on the link [3]. Congestion windows for all sources will decrease or increase at the same time. Therefore, for two sources i and j , $\eta_i(t)$ and $\eta_j(t)$ will have always the same sign

and

$$\eta_i(t+1) - \eta_j(t+1) = a_i \eta_i(t) - a_j \eta_j(t)$$

$$L(t) = \eta_i(t) - \eta_j(t) \leq a_{\max} (\eta_{\max}(t-1) - \eta_{\min}(t-1))$$

$$L(t) \leq a_{\max} L(t-1) \leq a_{\max}^2 L(t-2) \leq a_{\max}^{t-k} L(k) \leq \delta_4 a_{\max}^t$$

$$\delta_4 = a_{\max}^{-k} L(k)$$

$$(A22)$$

REFERENCES

- [1] B. Shihada and P-H. Ho, "Transport Control Protocol (TCP) in Optical Burst Switched Networks: Issues, Solutions, and Challenges", IEEE Communications Surveys and Tutorials, **10**, 70-86 (2008).
- [2] Y. Xiong, M. Vandenhoude, and H. Cankaya, "Control architecture in optical burst-switched WDM networks", IEEE Journal on Selected Areas in Communications, **18**, 1838-51 (2000).
- [3] A. Detti and M. Listanti, "Impact of segment aggregation on TCP Reno flows in optical burst switching networks", *Proceedings of IEEE International Conference on Computer Communication* (IEEE 2002), pp. 1803 - 1812.
- [4] C. Jin, D. Wei, and S. Low, "FAST TCP: Motivation, Architecture, Algorithms, Performance," *Proceedings of IEEE International Conference on Computer Communication* (IEEE 2004), pp. 2490 - 2501.
- [5] J. Wang, D. X. Wei, S. Low, "Modeling and Stability of FAST TCP", *Proceedings of IEEE International Conference on Computer Communication* (IEEE 2005), pp. 938-948.
- [6] C. Jin, D. Wei, et al, "FAST TCP: From Theory to Experiments", IEEE Network, **19**, 4-11, (2005).
- [7] Floyd, "High-Speed TCP for large congestion windows", RFC 3649, Experimental (2003).
- [8] T. Kelly, "Scalable TCP: improving performance in highspeed wide area networks", ACM SIGCOMM Computer Communication Review, **33**, 83-91 (2003).
- [9] X. Yu, C. Qiao, and Y. Liu, "TCP implementation and false time out detection in OBS networks," *Proceedings of IEEE International Conference on Computer Communication*, (IEEE 2002), pp. 774-784.

- [10] B. Shihada, Q. Zhang, and P-H. Ho, "Threshold-based TCP Vegas over Optical Burst Switched Networks", *Proceedings of IEEE International Conference on Computer Communications and Networks* (IEEE 2006), pp. 119-124.
- [11] L. Li, H. Xiaobin, W. Jian, Y. Yawei, C. Shuri, and L. Jintong, "Experimental comparison of high-speed transmission control protocols on a traffic-driven labeled optical burst switching network test bed for grid applications", *J. Opt. Netw.*, **8**, 491-503 (2009).
- [12] Q. Zhang, V. Vokkarane, Y. Wang, and J. Jue, "Analysis of TCP over optical burst switched networks with burst retransmission," *Proceedings of IEEE Global Communication Conference* (IEEE 2005), pp.1978-1983.
- [13] A. Tang, D. Wei, S. H. Low, and M. Chiang, "Heterogeneous congestion control: Efficiency, fairness and design," *Proceedings of IEEE International Conference on Network Protocols* (IEEE 2006), pp. 127-136.
- [14] F. Paganini, Z. Wang, J. Doyle, and S. Low, "Congestion Control for High Performance, Stability, and Fairness in General Networks", *IEEE/ACM Transactions on Networking*, **13**, 43-65 (2005).
- [15] B. Shihada and P-H. Ho, "A Novel TCP with Dynamic Burst-Contention Loss Notification over OBS Networks", *Elsevier Journal of Computer Networks*, **52**, 461-471 (2008).
- [16] B. Shihada, P-H. Ho, F. Hou, X. Jiang, S. Horiguchi, M. Guo, and H. Mouftah, "BAIMD: A Responsive Rate Control for TCP over Optical Burst Switched (OBS) Networks", *Proceedings of IEEE International Conference on Communications* (IEEE 2006), pp 2550 - 2555.
- [17] B. Shihada, P-H. Ho, and Q. Zhang, "TCP-ENG: Dynamic Explicit Congestion Notification for TCP over OBS Networks", *Proceedings of IEEE International Conference on Computer Communications and Networks* (IEEE 2007), pp. 516-521.
- [18] E. Mallada and F. Paganini, "Stability of node-based multipath routing and dual congestion control", *Proceedings of IEEE Conference on Decision and Control* (IEEE 2008), pp. 1398 - 1403.