

Research Article

Impact of Loss Synchronization on Reliable High Speed Networks: A Model Based Simulation

Suman Kumar,¹ Lin Xue,² and Seung-Jong Park²

¹ Department of Computer Science, Troy University, Troy, AL 36082, USA

² School of Electrical Engineering and Computer Science, Center for Computation and Technology, Louisiana State University, Baton Rouge, LA 70802, USA

Correspondence should be addressed to Seung-Jong Park; sjpark@cct.lsu.edu

Received 29 October 2013; Accepted 11 December 2013; Published 19 January 2014

Academic Editor: Liansheng Tan

Copyright © 2014 Suman Kumar et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Contemporary nature of network evolution demands for simulation models which are flexible, scalable, and easily implementable. In this paper, we propose a fluid based model for performance analysis of reliable high speed networks. In particular, this paper aims to study the dynamic relationship between congestion control algorithms and queue management schemes, in order to develop a better understanding of the causal linkages between the two. We propose a loss synchronization module which is user configurable. We validate our model through simulations under controlled settings. Also, we present a performance analysis to provide insights into two important issues concerning 10 Gbps high speed networks: (i) impact of bottleneck buffer size on the performance of 10 Gbps high speed network and (ii) impact of level of loss synchronization on link utilization-fairness tradeoffs. The practical impact of the proposed work is to provide design guidelines along with a powerful simulation tool to protocol designers and network developers.

1. Introduction

As one of the basic characteristics of computer networks, a dynamical system, TCP flow synchronization/desynchronization, is very important and interesting. In fact, level of loss synchronization is proven to be the major impact factor for the performance of computer networks. Modeling the loss synchronization has been a challenging task for network researchers especially for high speed networks. A few studies have concentrated on loss synchronization studies on high speed networks such as [1–3]. The work in [1] presents an analytical model using M/M/1/K queuing model approximations that is only valid for HighSpeed TCP (HSTCP) [4]. The work in [2, 3] presents synchronization statistics in a high speed network environment via NS2 simulation. However, both [2, 3] do not answer the question: how does loss synchronization level affect the performance of high speed TCP variants? Or how does loss synchronization affect the design of high speed networks? Also, these works do not address 10 Gbps high speed networks.

Hardware technologies and network applications have been bringing rapid changes in protocols at transport layer as well as at network layer. At the same time, network community must understand the behavior of these protocols in order to support research and development of next generation networks. This understanding is especially important to improve the robustness of protocol implementations and network applications. In general, networking protocol developers have to repeat a cycle consisting of two steps: they design and evaluate until they satisfy performance, and then they deploy protocols in real environment. While network simulation is a well accepted and widely used method for performance analysis and evaluation, it is also well known that packet-based simulators like NS2 and Opnet cannot be used in the case of high speed or large scale networks because of their inherent bottlenecks in terms of message overhead and CPU execution time [5–7]. In that case, model based approach with the help of a set of coupled differential equations is preferred for the simulation. However, current fluid model is

not suitable to perform performance studies on high speed networks of the order of 10 Gbps because of its treatment of loss synchronization by fairly dropping the packets at the bottleneck link on congestion events, which of course is not a realistic assumption. With these motivations, in this paper, we present a loss synchronization/desynchronization model that is scalable to many orders of 10 Gbps links. We demonstrate the usage of our model by performing two important studies on 10 Gbps high speed networks: (1) the impact of loss synchronization on sizing buffers and the countereffect of these two on the performance of 10 Gbps high speed networks and (2) throughput-fairness tradeoffs. To the best of our knowledge, no study is there to explore the role of different levels of loss synchronization on TCP performance over 10 Gbps high speed networks. Our work promotes the understanding and possible development of ongoing protocol development work for high speed networks.

2. Contribution and Paper Organization

2.1. A User Configurable Loss Synchronization Module for Fluid Simulation. In this paper, we show that normal fluid models do not capture the loss synchronization phenomena in the simulation. Through this paper, we propose a user configurable loss synchronization module that can be attached to the bottleneck queue to break the all flow loss synchronization in fluid simulation. Presented loss synchronization module can be easily coupled with fluid models for testing and furthering the development process of future protocol for high speed networks.

2.2. Performance of High Speed TCP Variants for Different Buffer Sizes on 10 Gbps High Speed Networks. We perform an extensive study on the impact of buffer sizes on the performance of high speed TCP variants on 10 Gbps high speed networks for different levels of loss synchronization. This work further motivates the exploration of the relationship among synchronization behavior of high speed flows, buffer sizes, and congestion control on high speed networks of the order of 10 Gbps and beyond.

2.3. Performance Tradeoffs in High Speed Networks of the Order of 10 Gbps. High speed networks are characterized by high capacity and low delay providing that high-capacity telecommunication networks are based on fiber optics technologies. It is observed that high speed TCP flows are aggressive. They greedily steal bandwidth from other flows and exhibit unfairness on these networks. Studies show that there is a strong correlation between loss synchronization and utilization-fairness of protocols. Through this paper, we present a preliminary study of the role of loss synchronization on utilization-fairness tradeoff. The presented study provides a reference point for the effect of loss synchronization on the performance of high speed protocols.

This paper is organized as follows. Section 3 presents background and motivation behind this work. In Section 4, we propose a loss-synchronization module for the fluid simulation and present the simulation setup for high speed

networks. Section 5 presents some basic simulation results using this model. Section 6 gives a brief summary of research work on this topic. Section 7 summarizes our work and presents conclusion and possible future research direction in this area.

3. Background and Motivation

3.1. Loss Synchronization on High Speed Networks. In [8], 2000 flows sharing a bottleneck link are simulated using NS2. It is observed that at the time of congestion event not all the flows have packet losses. In reality, it is observed that high speed networks have a few high speed flows sharing the bottleneck link [9] and exhibit an increased level of synchronization among high speed flows [10, 11]. It is to be noted that several high speed versions of TCPs such as CUBIC [12], FastTCP [13], HSTCP [4], S-TCP [14], BIC-TCP [15], and HAMILTON-TCP [16] were developed and some of them got adopted on the end systems and these high speed TCP variants are designed to be aggressive on high speed links.

To explain this, we refer to fluid model equations for congestion control algorithm for high speed TCP variants presented as below for some TCP flow i :

$$\frac{dW_i(t)}{dt} = \frac{a(W_i(t) < M_i)}{R_i(t)} - W_i(t) b \lambda_i(t), \quad (1)$$

where W_i is congestion window, R_i is round trip time, a and b are defined as increment and decrement factors, respectively, M_i is the maximum window size, and $\lambda_i(t)$ is the loss arrival rate. New congestion control algorithms are often characterized by higher values of a and by lower values of b , that is, by a conservative reduction of cwnd upon a congestion event. It is observed that burstiness increases with bandwidth because packets degenerate into extremely bursty outflows with data rates going beyond the available bandwidth for short periods of time [10] which causes unfair packet drops. Therefore, when a large number of flows see packet drops around the same time, they will synchronously react by reducing their congestion window sizes. This may lead to a significant reduction in instantaneous throughput of the system. In an unfair AQM scheme, however, only a few flows will record packet losses and rest of the flows will keep the link busy and, therefore, the aggregate congestion window will change less significantly. Considering Figure 1(a), there are three flows numbered 1, 2, and 3. Flows 1 and 2 record packet loss at time t_1 and flows 1, 2, and 3 record packet loss at t_3 .

In Figure 1(b), we show how different flows mix up while reaching the bottleneck link. In the figure, flow 1 would lose more packets than flow 2 and flow 3 might get accommodated in the queue because queue is not only dropping packets but also keeping the link busy by accepting new packets.

3.2. Desynchronized Losses and High Speed TCPs. Recent years have seen many studies focusing on sizing buffers because of statistical multiplexing of different flows in the internet [17–24] and high speed networks [1, 25]. Several

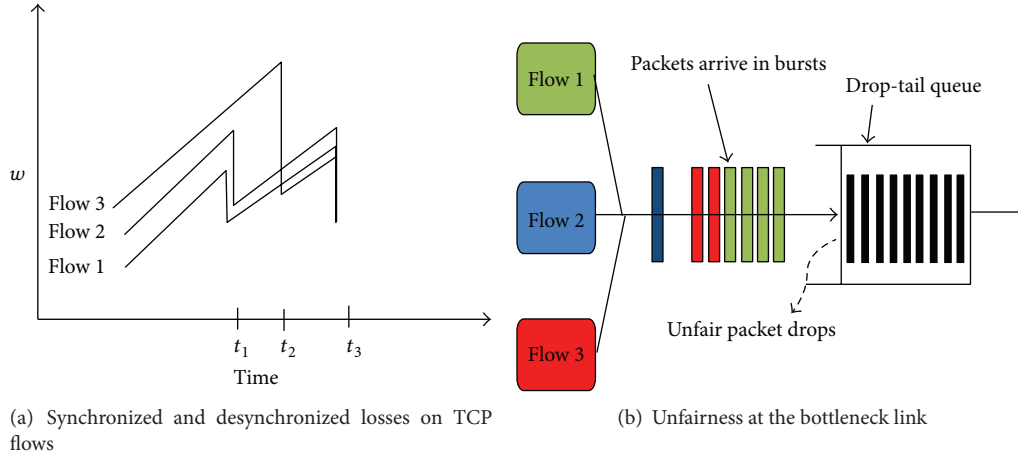


FIGURE 1: Loss synchronization in action.

research projects argued that desynchronized flows can give high link utilization and also affect fairness among coexisting flows on bottleneck link. In this section, our intention is to see if change in loss synchronization can impact the fairness. For this, first we focus on an aggressive protocol pairing with a less aggressive protocol. Second, we focus on two similar aggressive flows with different starting times sharing a bottleneck link. Clearly, STCP and standard TCP satisfy our requirements for aggressive and conservative flows, respectively [26, 27]. Flows are fully synchronized when all the flows record packet losses and partially synchronized when only a fraction of all the flows record packet losses in a congestion event. We consider a standard dumbbell topology with two flows sharing a 1 Gbps link and propagation delay is 120 ms. Synchronization level is enforced when queue size increases to a certain limit. For partial synchronization, we force the flows with higher congestion window to drop first. We measure fairness by using Jain's fairness index [28].

Figure 2(a) shows when both flows are synchronized. Simulation time is 200 sec and we observe that STCP occupies the major bandwidth (see [26] where STCP occupies 99.2% of bandwidth). In the case of dropping only one flow with highest sending rate at the time of congestion as shown in Figure 2(b), the fairness improves.

Figure 3 shows two STCP flows with low and high RTTs competing for bandwidth. In Figure 3(a), we see that the flow with low RTT consumes the major bandwidth and the other flow starves because it can not update its congestion window values as frequently as the one with smaller RTT. In Figure 3(b), when we enforce only one flow to lose the packet during the time of congestion, there is a good improvement in fairness as well as link utilization.

STCP flows are very sensitive to their start-up times and exhibit slow convergence. Therefore, more often homogeneous STCP flows exhibit unfairness (e.g., see [27]) especially for short flows. In Figure 4, we start the second flow after 10 seconds. When losses are synchronized (in Figure 4(a)), STCP starting after 10 seconds gets lesser bandwidth while the flow that started early enjoys major chunk of available bandwidth. On the other hand, we observe improvement in

fairness when we enforce the partial synchronization of the flows as shown in Figure 4(b).

Therefore, through our simple example, we observe a strong correlation between level of flow synchronization and throughput and fairness. Moreover, we aim to explore the impact of loss synchronization on utilization-fairness tradeoffs.

4. Fluid Model Implementation for Loss Synchronization

4.1. Background. From network perspective, drop policies need to address mainly three issues: when to signal congestion, which flows should be dropped, and how many flows should be dropped.

Which flows should be dropped? To answer this question we review some of the existing policies, like RED and drop-tail. Drop-tail drops the whole tail and costs all the flows which are queued in the stream. This results in multiple losses. RED addresses this question more intelligently by dropping the packets probabilistically.

How many flows are to be dropped? Drop-tail drops all the flows which are lagging behind in the queue and RED takes probabilistic approach to drop packets. However, we do believe that it needs some information to process to make certain decisions like *when* to drop, *which ones* to drop, and *how many* of *which ones* to be dropped.

Based on our discussion above, following are the factors for the design of loss synchronization module.

Drop Threshold: defined as a maximum queue size. If the queue size exceeds the maximum queue size, congestion will be signaled by dropping packets.

Flow Selection Policy: a selection policy to select flows to drop packets.

Synchronization Level: defined as number of flows to be dropped.

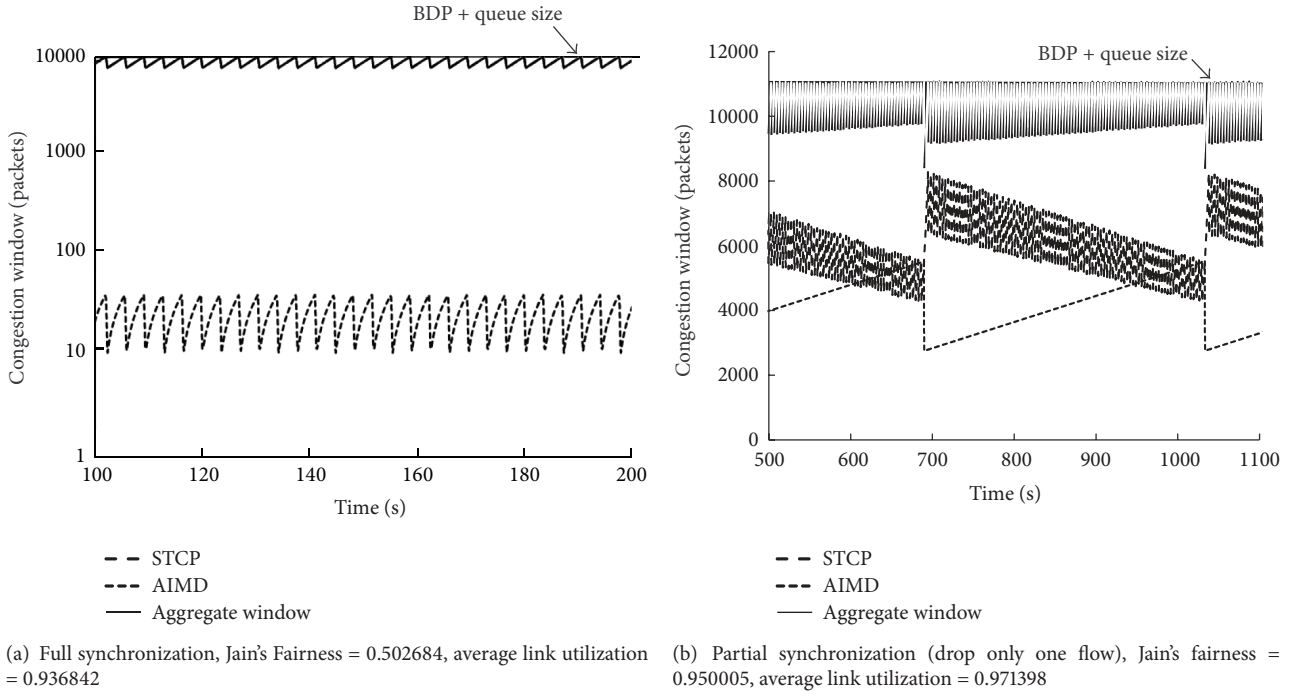


FIGURE 2: Congestion window versus time for STCP and AIMD pairing on a bottleneck link = 1 Gbps, RTT = 120 ms, buffer size = 0.1 BDP (1000 1.5 KB packets).

4.2. *A General Fluid Model and Its Limitations.* For reference, we rewrite the original fluid model equations as below:

- F_i : A set of ordered queues traversed by i th flow
- W_i : Congestion window for i th flow
- R_i : Round trip time for i th flow
- λ_i : Loss indication rate for i th flow
- q_l : Queue size associated with l th link
- C_l : Service capacity/bandwidth for l th link
- p_l : Packet drop probability at l th queue
- q_l^{\max} : Maximum queue size associated with l th link
- n_l : Denotes number of flows traversing l th link
- A_l^i : Arrival rate of i th flow at l th link.

In fluid model [29, 30], packet chunks are modeled as a fluid at a sender according to the following ordinary differential equations:

$$\frac{dW_i(t)}{dt} = \frac{a(t)}{R_i(t)} - W_i(t) b(t) \lambda_i(t). \quad (2)$$

When the packet fluid reaches the queue, the queue checks for the incoming rate and adjusts the queue size according to equation below:

Queue size:

$$\frac{dq_l(t)}{dt} = -1(q_l(t) > 0) C_l + \sum_{i=1}^{n_l} A_l^i(t), \quad (3)$$

where $q_l(t) > 0$ and can have only positive value. $ARsum_l$ is sum of the arrival rates of all flows at queue l , and l is bottleneck queue in our case.

Consider

$$ARsum_l = \sum_{i=1}^{n_l} A_l^i(t). \quad (4)$$

Under the overload or overfilling, the drop-tail queue generates the loss probability according to the equation below:

$$p_l(t) = \begin{cases} 0, & q_l(t) < q_l^{\max} \\ \max\left(\frac{ARsum_l - C_l}{ARsum_l}, 0\right), & q_l(t) = q_l^{\max}. \end{cases} \quad (5)$$

This loss probability is proportionally divided among all flows passing the queue as shown in Figure 5.

$$\lambda_i(t) = \sum_{l \in F_i} A_l^i(t) p_l(t). \quad (6)$$

In reality, TCP flows' burstiness induces dynamics of certain degree of synchronization among TCP flows sharing the bottleneck link. In addition, during the time of congestion, losses are not evenly distributed among TCP flows and TCP flows with larger congestion windows are more likely to get affected. The model does not account for this behavior and, therefore, it is not very useful in performance studies in the presence of desynchronized flows.

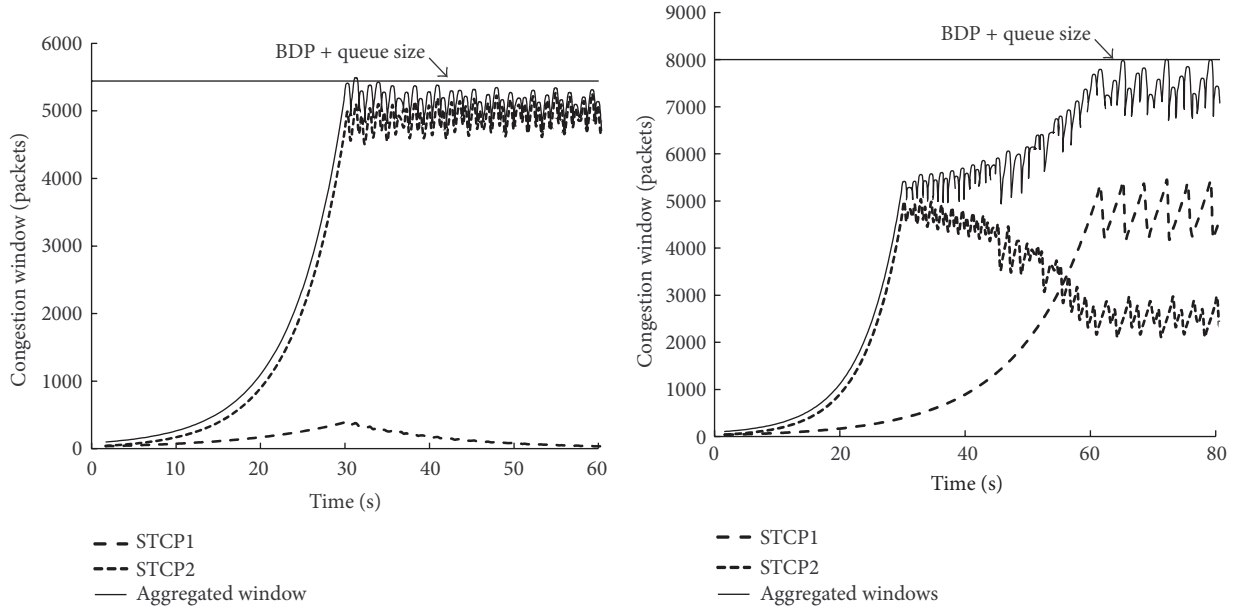


FIGURE 3: Congestion window versus time for 2 STCP flows (Flow1 RTT = 40 ms, Flow2 RTT = 120 ms) a bottleneck link = 1 Gbps, buffer size = 0.1 BDP (1000 1.5 KB packets).

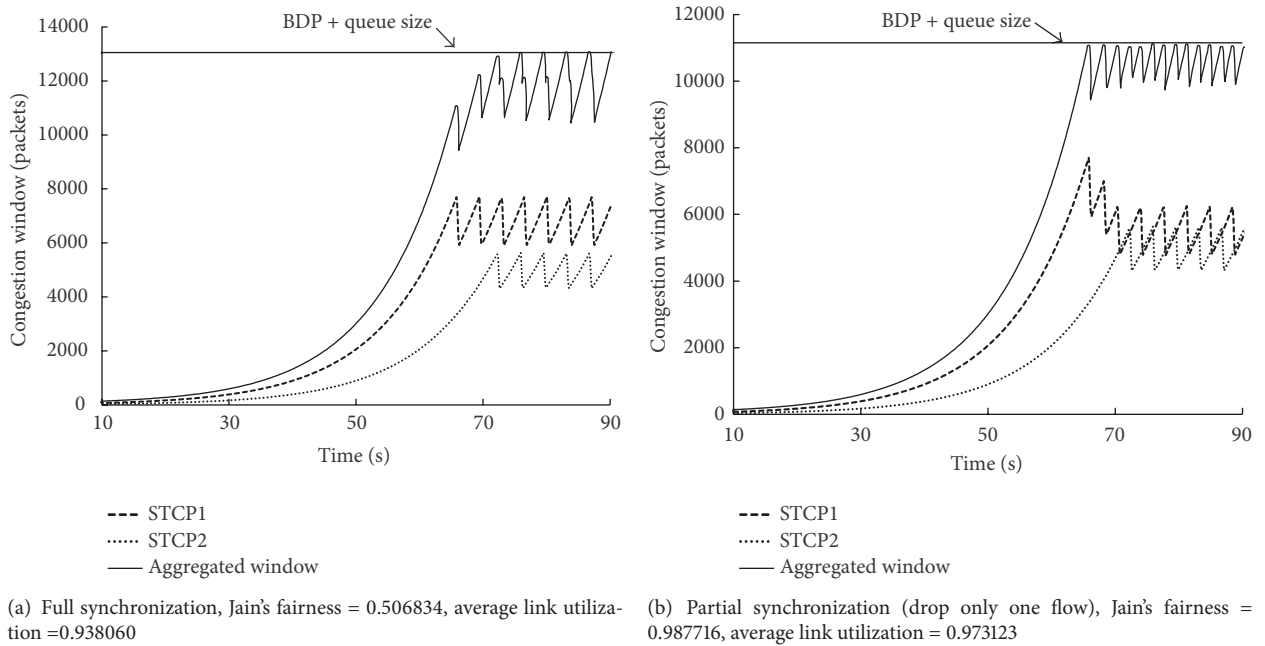


FIGURE 4: Congestion window versus time for 2 STCP flows (flow 2 start 10 sec. after flow 1) on a bottleneck link = 1 Gbps, RTT = 120 ms, buffer size = 0.1 BDP (1000 1.5 KB packets).

4.3. A Loss-Synchronization Fluid-Simulation Module. Our loss-synchronization simulation module consists of two parts as illustrated in Figure 6.

- (1) Loss-synchronization controller: the controller controls the loss synchronization factor at the time of congestion. The loss synchronization factor can

be user given or derived from a distribution or experimental data at any congestion event. The loss synchronization factor is an integer value and defines how many flows are to record packet losses. For the description below, loss synchronization factor is denoted as m_k at the k th congestion. Suppose there

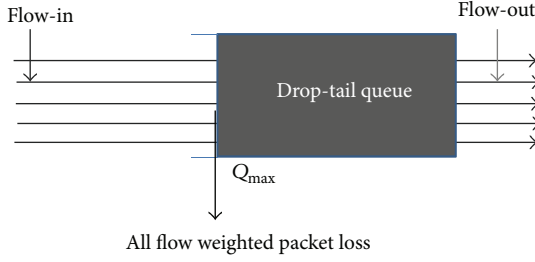


FIGURE 5: Operation of a drop-tail queue under the fluid model simulation.

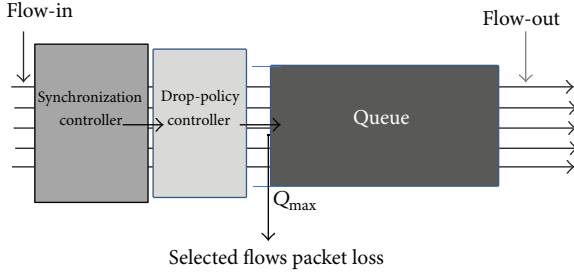


FIGURE 6: Operation of the loss-synchronization module on a queue under fluid model simulation.

are N number of flows in the network and out of those N , m_k flows are experiencing packet losses. Therefore, following boundary value holds for any selected m_k :

$$m_k \in (1, N). \quad (7)$$

- (2) Packet-drop policy Controller: loss-synchronization controller passes the information to the packet-drop policy controller. The packet-drop policy controller selects m_k TCP flows to drop at the time of congestion by using a priority matrix that defines the order of flows to record packet losses and pass this information to the queue. Specifically, at the time of k th congestion, the packet-drop policy controller determines the priority matrix $P^k = [D_1^k, D_2^k, \dots, D_N^k]$, where D_i^k defines some important time varying decision parameter for flow i at k th congestion events. $D_i^k > D_j^k$ indicates that packets in flow i have higher drop probability than flow j . We define Pl_k as the set of m flows selected based on the priority matrix. We define Pl as a policy through which these m flows get selected. Therefore, all the flows $i \in Pl$ satisfy following relationship:

$$\sum_{i \in Pl_k} \lambda_i(t) = \text{ARsum}_l - C. \quad (8)$$

The above equation means that every loss is accounted and distributed among the flows (since burstiness is a random and stochastic phenomenon of TCP flows). In both the models above, we assume that congestion occurs when the buffer can not accept any more packets and total arrival rate exceeds the

link capacity. Therefore, there is a duality between both the models in terms of loss rates at the queue. Synchronization at the packet level in reliable networks indicates the phase relationship between the wavelike variations of congestion window values of different TCP flows. However, it is to be noted that differential equation model of TCPs is used which is inherently oscillatory in nature and, therefore, whether these oscillations are phase synchronized or not depends on the loss indication signal controlled by the parameter m_k approximating the notion of congestion event duration to the notion of point of congestion.

4.4. High Speed Networks Simulation Setup. For high speed networks we make following assumptions.

- (i) Congestion events occur when bottleneck buffer is full.
- (ii) Highest rate flows are more prone to record packet losses.
- (iii) High speed TCP flows' burstiness induces higher level of synchronization.

The first assumption is obvious and states that buffer overflow causes congestion in the network. Second assumption relates to the fact that high speed TCP flows are aggressive and higher burstiness is attributed to high congestion window values; the assumption that highest sending rate flows have higher probability to record losses is heuristically justified (in [11], it is observed that larger burstiness increases the probability of packet losses) if not very accurate. To understand the third assumption, we refer to [2] that shows high speed TCP flows intent to show some level of loss synchronization. Fairness in packet drops (i.e., dropped packets are uniformly distributed among all the flows) can create synchronization among flows whereas unfair packet drops (i.e., only some of the flows record packet drops) can lead to reduced synchronization. Below, we outline the values for the loss synchronization simulation module for queue.

To select random m_k at any congestion event k , we define a parameter X which is defined to give the minimum level of synchronization (i.e., the ratio of synchronized flows and total number of flows is no less than X):

$$m_k \in (XN, N). \quad (9)$$

It is observed that when congestion happens multiple number of flows record packet losses. Selection of X not only satisfies a least certain level of loss synchronization but also does not avoid any degree of synchronization higher than X . Hence, the definition of X is reasonable for high speed networks case. It is to be noted that the definition of X may not be suitable to all levels of statistical multiplexing for high speed TCPs but it presents a very simple reference point covering a wide range of synchronization behavior of high speed TCP flows.

In our study, Pl_k is the set of m_k flows with highest sending rates that is denoted by priority matrix whose elements are arrival rate of flow i ($= D_i^k$) at the bottleneck queue.

Incremental parameter a and decremental parameter b for variants of TCPs are outlined in [5].

5. Simulation Results

To calculate link utilization, we sample the departure rate at the bottleneck link as defined below:

$$U = \frac{\sum_s \sum_{i=1}^{n_l} \text{Dep}_l^i(t_s)}{C_l \times \sum_s} \quad (10)$$

In the above equation, Dep_l^i is the departure rate of flow i at bottleneck link l , s denotes the sampling instances, and C_l is the capacity of the bottleneck link. Therefore, to present our result we calculate the link utilization by taking the average of sampled total departure rate at bottleneck link.

Our second performance matrix is fairness. Long term flow throughput is used for computing fairness according to Jain's fairness index [28].

We consider 10 flows of high speed TCP variants on 10 Gbps link competing for bandwidth. A network in general consists of several queues. The behavior of TCP congestion control algorithm mainly depends on the congestion on the most congested link [31]. Therefore, in this work we consider the case of 10 persistent high speed TCP flows sharing the same bottleneck link on a dumbbell topology.

The queuing discipline policy is drop-tail with loss-synchronization module. Mean field theory of buffer sizes [32] suggests that at least 60% of flows record packet losses during a congestion event. Since high speed networks are more different than the internet, we consider two cases of X , $X = 0.3$ and $X = 0.6$, to introduce lower and higher level of synchronization among high speed connections ($X = 1$ refers to the case when all flows record packet losses). Values for configurable parameter m_k are drawn from normal distribution. We consider the effect of only the basic congestion control algorithms on bottleneck link and, therefore, the mechanisms like time-out, slow start, and fast retransmit are not simulated. We perform fluid simulation by solving differential equations for high speed TCP variants. Since our approach involves random number generation, we ran the simulation 10 times each for 3000 s and averaged them for each result presented in this section. It is to be noted that throughput results presented in this section are taken from the time of first congestion event (i.e., when the flows are stabilized, we record the first congestion since the start of simulation).

Before presenting our results, we verify our model with previously published results in [1]. Topology presented in Figure 7 is used for all the results presented in this section.

5.1. Model Verification. To verify our proposed model compared with the Boston model presented in [1], we used the NS2 simulator. For this validation, high speed TCP (HSTCP) is used; bottleneck link C is set to 1 Gbps; packet size is 1 KB; and RTT is ranging from 115.5 ms to 124.5 ms with average RTT of 120 ms. The buffer size of the bottleneck buffer is varied as a fraction of 12 500 packets corresponding to the BDP of largest RTT ($= 124.5$ ms). We use the same parameter sets as suggested in [4]. Figure 8 shows that the Boston model has a large amount of difference compared with NS2 simulation results in case of low and moderate

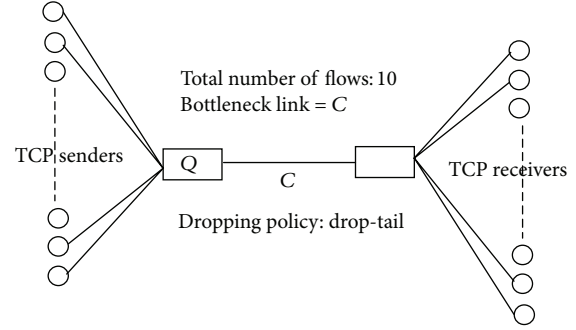


FIGURE 7: Simulation topology with 10 flows sharing a bottleneck.

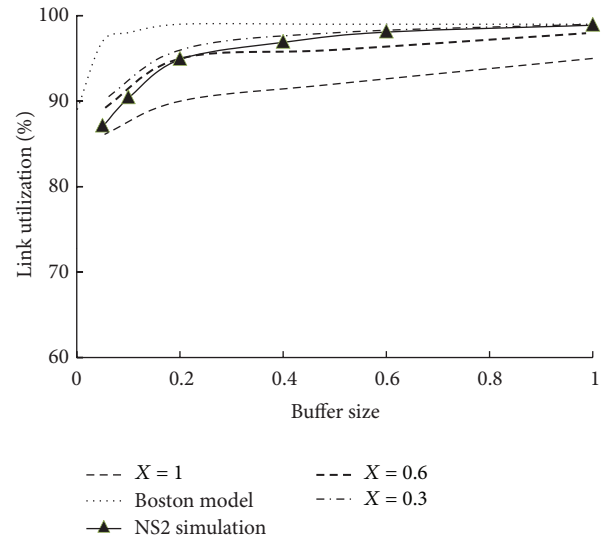


FIGURE 8: Validation with previously published results (average link utilization as function of buffer size fraction (max 12 500)).

buffer sizes. However, the proposed model's results shown in Figure 8 give a closer match with NS2 simulation results. We also observe that when all the flows are synchronized ($X = 1$), fluid model does not match with the NS2 simulation result. As we decrease the synchronization ($X = 0.6$ and $X = 0.3$ case), the utilization improves and NS2 simulation result matches with fluid simulation results. We conclude that fluid simulation with synchronization module presented in this work can provide more accurate result than the Boston model.

5.2. Link Utilization as a Function of Buffer Size on 10 Gbps High Speed Networks. In this section, we show simulation results for AIMD, HSTCP, CUBIC, and HTCP [33] for 10 Gbps link (results for rest of the high speed TCP variants will be included in an extended paper). For the results presented in this subsection, we set $C = 10$ Gbps, link delay $= 10$ ms, RTTs of 10 flows are ranging from 80 ms to 260 ms with average RTT $= 170$ ms. We use average RTT to calculate BDP which gives 141 667 1500 B packets maximum. Number of flows is 10.

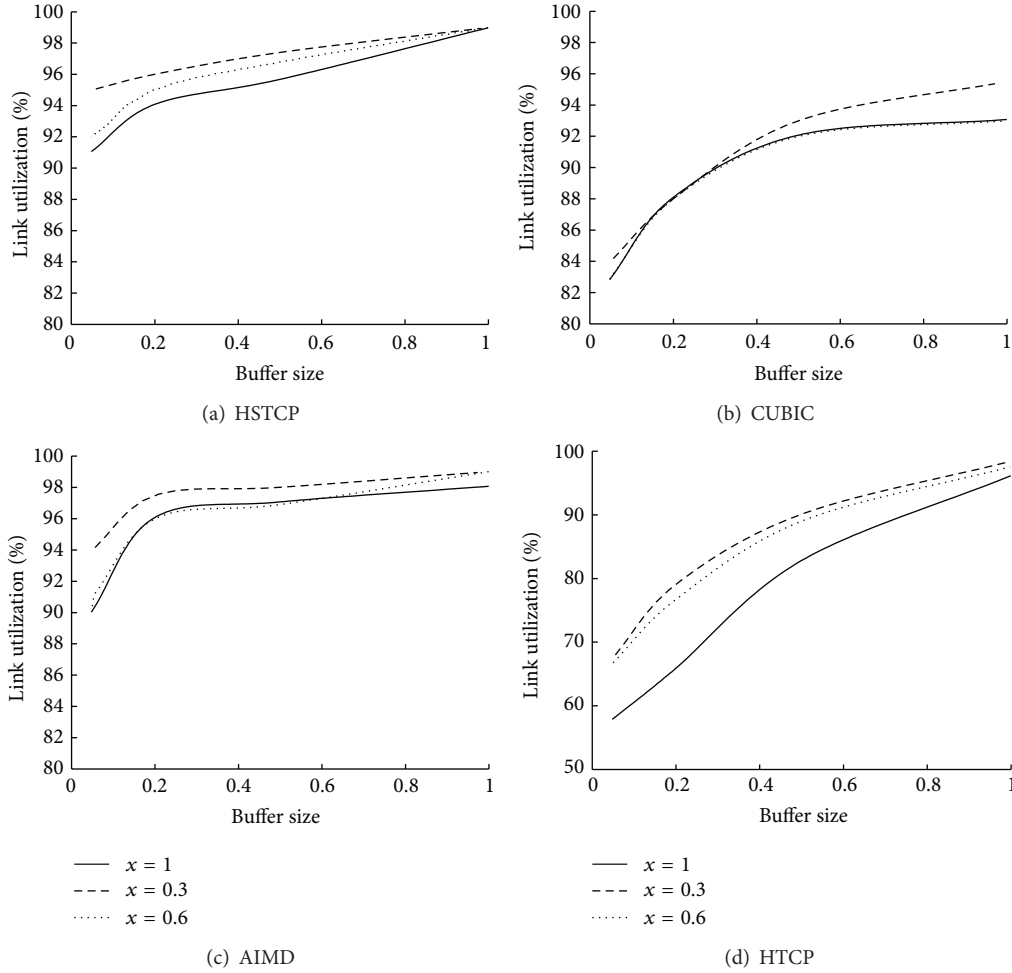


FIGURE 9: Average link utilization as function of fraction of buffer size on 10 Gbps link (max buffer = 141 667 packets).

In Figure 9(a), we show link utilization as a function of fraction of maximum buffer size for three different values of X . We observe as buffer size increases link utilization improves. As shown in the previous results, desynchronization improves the link utilization (with synchronization parameter $X = 0.3$, higher link utilization is achieved as compared to $X = 1$ and $X = 0.6$).

Simulation result for CUBIC is shown in Figure 9(b), the performance of CUBIC TCP is drastically affected by smaller buffer size. We observe less than 90% link utilization for all the three values of X . We also observe for buffer sizes greater than 20% of BDP reduction in level of synchronization improves the performance. However, we observe that higher partial synchronization ($X = 0.6$) does not show any improvement than all synchronization ($X = 1$). We also observe HSTCP performs better than CUBIC TCP with various buffer sizes. CUBIC TCP is designed to be more fair, whereas HSTCP has RTT fairness problem. Although CUBIC and HSTCP have some performance differences, we believe there has to be some tradeoff between fairness and link utilization. Fairness of the TCP flows is out of scope of this paper and is left for further exploration.

AIMD result is presented in Figure 9(c). We observe for the case of $X = 1$ that 10 AIMD flows behave like 10 parallel TCP flows. We observe that link utilization improves with the lower degree of synchronization. It is also observed that AIMD outperforms CUBIC and is comparable to HSTCP for all buffer size cases. HSTCP and AIMD results are close to each other because congestion control algorithm of HSTCP emulates the parallel TCP flows. HTCP (Figure 9(d)) performs poorly for small buffer sizes. We observed that frequent losses render the ability of synchronized HTCP flows to utilize the available bandwidth when 0.05 BDP buffer size is used. However, desynchronized flows are able to show better performance.

The main reason for the poor performance of CUBIC and HTCP as compared to AIMD and HSTCP is attributed to its improvement of fairness. In desynchronized environment, both CUBIC and HTCP mark the last congestion event by setting the values of W_{\max} and *time since last congestion even* parameters, respectively. However, some flows miscalculate this congestion point if they do not record any packet loss. The inherent idea of these two TCP mechanisms is to be fair with other flows traversing the same bottleneck link. There is

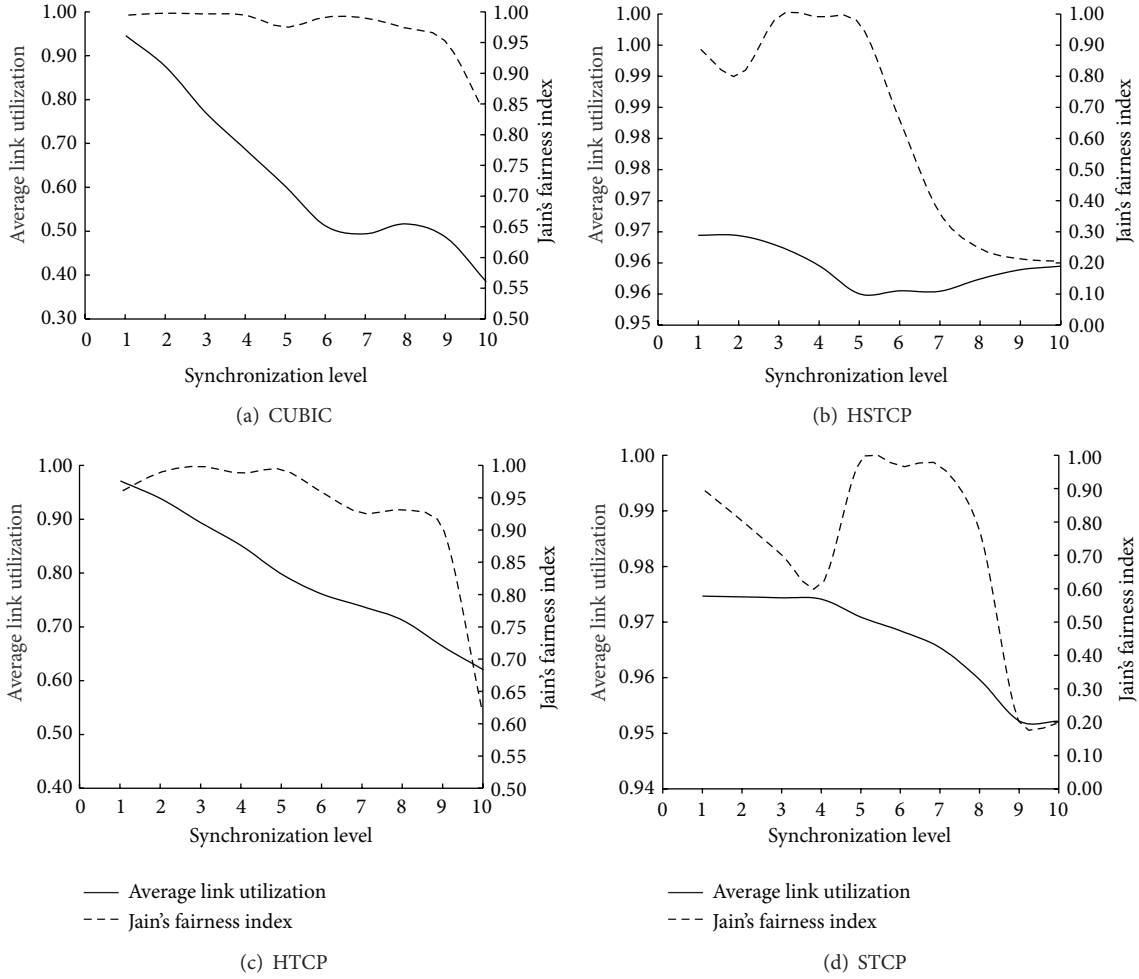


FIGURE 10: Fairness and link utilization versus Synchronization level of 10 CUBIC flows on 10 Gbps link with average RTT = 170 ms and Buffer Size = 0.1 BDP (10 000 1.5 KB packets).

an intrinsic tradeoff between fairness and efficiency of these protocols.

5.3. Synchronization Level versus Utilization-Fairness. In this section, we present our study on the impact of level of loss synchronization on link utilization and fairness tradeoffs. We relax our assumption on loss synchronization. Instead of choosing it from a distribution, the value of m is fixed for each measurement. We plot the value of m_k as synchronization level on x -axis.

5.3.1. Homogeneous TCP Flows. We discuss the results for homogeneous TCP flows. Figure 10(a) shows the variation of utilization and fairness with variety of synchronization levels “ m ”, which stands for how many flows to be dropped at congestion event. We observe that when all the flows are synchronized CUBIC shows good fairness but poor link utilization. As synchronization level decreases, utilization improves and the fairness index is almost 1. It is interesting to see the behavior of HSTCP in Figure 10(b). We observe as

m increases fairness first decreases and then there is a slight increase in fairness. We observe high link utilization for m values 3 to 5. However, we believe there has to be an intrinsic tradeoff between the two as far as m is concerned.

There is a linear decrease in utilization as m increases for HTCP (see Figure 10(c)). We observe highly synchronized flows decrease the link utilization. In this case RTT unfairness impacts the performance. However, HTCP performs well in a wide range of values of m .

We observe in Figure 10(d) that fairness of STCP decreases first and then improves when synchronization level increases. For link utilization, we see a slight decrease. STCP with its multiplicative increase approach has been a very unfair protocol in majority of the evaluation studies and its fairness can be drastically impacted by different variables. It has been observed in Section 9 of [34] that STCP has unstable intraprotocol fairness behavior. We observe that intraprotocol fairness of STCP is also drastically impacted by the loss synchronization level. However, by observing the fairness and utilization trend with m , we can say there have to be some values of m for a given configuration of flows where it

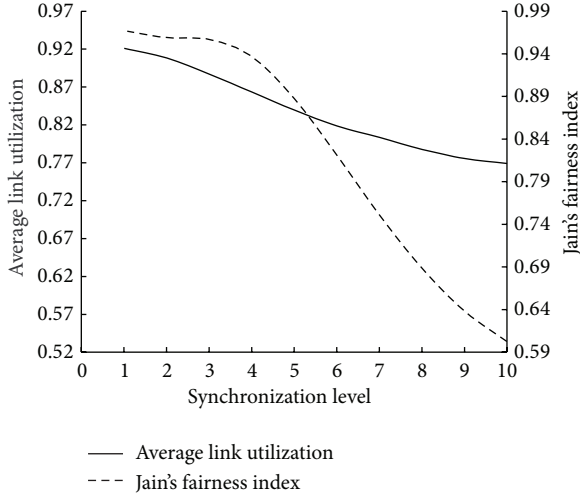


FIGURE 11: Fairness and link utilization versus synchronization level of 10 AIMD flows on 10 Gbps link with average RTT = 170 ms and buffer size = 0.1 BDP (10 000 1.5 KB packets).

can still achieve good results. Furthermore, we see the fairness and link utilization for 10 AIMD flows in Figure 11. It seems that low level of synchronization can help address the issue of RTT unfairness. We observe that decreasing m improves link utilization as well as fairness. With the observation in this section, we conclude that controlled synchronization would help to address some challenging issues for homogeneous TCP flows.

5.3.2. Heterogeneous TCP Flows. Fairness and utilization results for heterogeneous TCP flows are presented in Figure 12. In this scenario, we have two flows of each TCP variant considered in this simulation study. It is interesting to observe that fairness and link utilization both improve as synchronization parameter m decreases below 4. However, since we ran the simulation for a long time, the resulting fairness and utilization correspond to long time average. The slower flows get dropped as often as faster flows do when synchronization parameter is lower than 4. Highly synchronized flows include the slower flows frequently at the time of congestion and, therefore, slower flows take a long time to reach peak window size after being punished. Hence, we see a poor level of fairness and link utilization for m values greater than 4. Also, we observe a strong impact of loss synchronization on these performance factors.

6. Related Work

There are a few works available that attempt to conduct a performance analysis in high speed networks. In [2, 3], the authors presented a study of loss synchronization phenomena in a high speed network setting using drop-tail and RED respectively, for different buffer sizes. However, their studies only validated loss synchronization effect of these two specific queue management schemes. In [35], the authors generalized

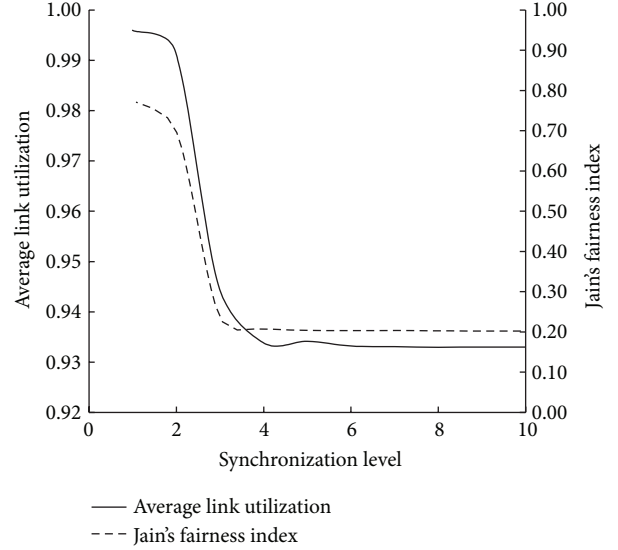


FIGURE 12: Fairness and link utilization versus synchronization level of 10 mixed TCP flows on 10 Gbps link with average RTT = 170 ms and buffer size = 0.1 BDP (10 000 1.5 KB packets).

FAST TCP to a so-called Generalized FAST TCP to characterize the fairness issue while fairness issue is directly relevant to synchronization. The authors in [36] considered impact of loss synchronization on fairness associated with FAST TCP operations due to inaccurate estimation of the round-trip propagation delay. In a recent work [25], experimental study of the impact of queue management schemes on high speed TCP variants over high speed networks was presented. Different combinations of queue management schemes and TCP variants were evaluated in terms of link utilization, fairness, delay, and packet loss rate. In follow-up works, authors in [37, 38] evaluated interprotocol fairness for heterogeneous TCP flows over high speed networks. CUBIC, HSTCP, and TCP-SACK were mixed at a 10 Gbps bottleneck link, and the fairness properties among heterogeneous TCP flows were explored. The authors elaborated a queue management scheme to solve unfairness issues in high speed networks. However in above works, impact of loss synchronization on the performance of high speed networks was not explored.

Next, we like to mention a few notable works on sizing buffers in Internet environment. In particular, the rule-of-thumb, first stated in [39] and further studied in [40], was challenged in [31]. The author in [39] assumed that there are N long lived flows which are stochastically multiplexed at a router buffer requiring $BDP/N^{0.5}$. On the similar hypothesis, [17] proposed the buffer size is $0.63 BDP/N^{0.5}$. In [18], the sufficient buffer size of $BDP^2/32N^3$ can provide near 100% link utilization. The assumption that the number of flows in the network remains constant was further investigated in [19]. The authors in [19] concluded that depending on the core-to-access speed ratio, the buffer size and the number of flows are not independent of each other and, therefore, these two quantities should not be treated independently. And $O(1)$ buffer sizes are good enough for near 100% link

utilization given the core-to-access ratio is large. The authors in [20] considered packet loss rate as an important metric and attempted to bound the loss rate to small value to achieve good link utilization with small buffers. Packet loss rate is proportional to N^2 , where N is the number of flows and hence, shown to be an important parameter while designing router buffers [22]. Some researchers also tried to attempt the problem from a different perspective. For example, in [21], input/output capacity ratio was considered to be important metrics as far as end user is concerned.

In high speed networks, [1] presented an analytical model focusing on the effect of buffer size on HSTCP performance in a 1 Gbps network. The results shown in this work argue that a smaller buffer can be sufficient to give near 100% link utilization. However, the authors in [1] assumed that there are many long lived flows in the network. Their study is solely focused on HSTCP and does not apply to other high speed TCP variants.

7. Conclusion

To develop next generation high speed networks, we must understand the dynamics, which govern network performance, such as the relationship among congestion control algorithms, bottleneck buffers, and queuing policies. We focus on a simple yet critical setup to get a clear understanding of the underlying mechanisms behind impact of loss synchronization on congestion control and queue management mechanism. A modified fluid model is presented to accommodate the phenomena of different degree of loss synchronization and unfair packet drops on high speed networks.

Simulation results for HSTCP, CUBIC, AIMD, and HTCP are presented to show the effect of different buffer sizes on link utilization. We present the fluid simulation results showing the effect of different buffer sizes on bottleneck link utilization for high speed TCP flows. We observe the following.

- (i) Buffer size of less than 10% of BDP is sufficient to achieve more than 90% link utilization for both HSTCP and AIMD.
- (ii) Both CUBIC and HTCP require larger buffer size for better performance.
- (iii) Increase or decrease of loss synchronization levels does not show much improvement in the performance of desynchronized HTCP and CUBIC flows, whereas lower synchronization further improves the link utilization for HSTCP and AIMD.

Our study indicates that the synchronization level can be used to improve the performance of the network in terms of fairness and bottleneck link utilization. We present our findings for 10 Gbps high speed networks for both homogeneous and heterogeneous TCP flows. We observe that desynchronization of packet losses among high speed TCP flows plays an important role in the link utilization, fairness, and design of buffer size of bottleneck queue. Presented work can act as a powerful tool that can be used in the design, analysis, and comparison of transport protocols and queue management

schemes over high speed networks. Although we present a simple analysis, explorations of more complicated scenario can be expanded from this work.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] D. Barman, G. Smaragdakis, and I. Matta, "The effect of router buffer size on highspeed TCP performance," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '04)*, pp. 1617–1621, December 2004.
- [2] S. Hassayoun and D. Ros, "Loss synchronization and router buffer sizing with high-speed versions of TCP," in *Proceedings of the IEEE INFOCOM Workshops*, Phoenix, Ariz, USA, April 2008.
- [3] S. Hassayoun and D. Ros, "Loss synchronization, router buffer sizing and high-speed TCP versions: adding RED to the mix," in *Proceedings of the IEEE 34th Conference on Local Computer Networks (LCN '09)*, pp. 569–576, October 2009.
- [4] S. Floyd, "High Speed TCP for large congestion windows," Tech. Rep. RFC 3649, 2003.
- [5] S. Kumar, S.-J. Park, and S. Sitharama Iyengar, "A loss-event driven scalable fluid simulation method for high-speed networks," *Computer Networks*, vol. 54, no. 1, pp. 112–132, 2010.
- [6] S. Kumar, S.-J. Park, S. S. Iyengar, and J.-H. Kimn, "Time-adaptive numerical simulation for high speed networks," in *HPCNCS, ISRST*, pp. 198–205, 2007.
- [7] Y. Sakumoto, H. Ohsaki, and M. Imase, "A method for accelerating flow level network simulation with low-pass filtering of fluid models," *Information and Media Technologies*, vol. 8, no. 3, pp. 797–805, 2013.
- [8] D. Wischik, "Buffer requirements for high-speed routers," in *Proceedings of the 31st European Conference on Optical Communication (ECOC '05)*, vol. 5, pp. 23–26, 2005.
- [9] <http://www.internet2.edu/presentations/fall-03/20031013-Net-Flow-Shalunov.pdf>.
- [10] D. A. Freedmanxz, T. Marianx, J. H. Leey, K. Birmanx, H. Weatherspoonx, and C. Xuy, "Exact temporal characterization of 10 Gbps optical wide-area network," in *Proceedings of the 10th Internet Measurement Conference (IMC '10)*, pp. 342–355, November 2010.
- [11] R. Takano, Y. Kodama, T. Kudoh, M. Matsuda, and F. Okazaki, "Realtime burstiness measuremen," in *Proceedings of the 4th International Workshop on Protocols for Fast Long-Distance Networks (PFLDnet '06)*, 2006.
- [12] S. Ha, I. Rhee, and L. Xu, "CUBIC: a new TCP-friendly high-speed TCP variant," *ACM SIGOPS Operating System Review*, vol. 42, no. 5, pp. 64–74, 2008.
- [13] C. Jin, D. X. Wei, and S. H. Low, "FAST TCP: motivation, architecture, algorithms, performance," in *Proceedings of the Conference on Computer Communications (IEEE INFOCOM'04)*, pp. 2490–2501, Hong Kong, China, March 2004.
- [14] T. Kelly, "Scalable TCP: improving performance in highspeed wide area networks," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 2, pp. 83–91, 2003.
- [15] L. Xu, K. Harfoush, and I. Rhee, "Binary increase congestion control (BIC) for fast long-distance networks," in *Proceedings of*

- the Conference on Computer Communications (IEEE INFOCOM '04)*, pp. 2514–2524, March 2004.
- [16] R. N. Shorten and D. J. Leith, “H-TCP: TCP for high-speed and long-distance networks,” in *Proceedings of the 4th International Workshop on Protocols for Fast Long-Distance Networks (PFLD-net '04)*, 2004.
 - [17] D. Wischik and N. McKeown, “Part I: buffer sizes for core routers,” *SIGCOMM Computer Communication Review*, vol. 35, no. 3, pp. 750146–784833, 2005.
 - [18] K. Avrachenkov, U. Ayesta, and A. Piunovskiy, “Optimal choice of the buffer size in the Internet routers,” in *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference (CDC-ECC '05)*, pp. 1143–1148, Seville, Spain, December 2005.
 - [19] A. Lakshmikantha, R. Srikant, and C. Beck, “Impact of file arrivals and departures on buffer sizing in core routers,” in *Proceedings of the 27th IEEE Communications Society Conference on Computer Communications (INFOCOM '08)*, pp. 529–537, April 2008.
 - [20] A. Dhamdhere, H. Jiang, and C. Dovrolis, “Buffer sizing for congested internet links,” in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM '05)*, pp. 1072–1083, Miami, Fla, USA, March 2005.
 - [21] R. S. Prasad, C. Dovrolis, and M. Thottan, “Router buffer sizing revisited: the role of the output/input capacity ratio,” in *Proceedings of the ACM CoNEXT Conference—3rd International Conference on Emerging Networking EXperiments and Technologies (CoNEXT '07)*, pp. 1–12, ACM, December 2007.
 - [22] R. Morris, “TCP behavior with many flows,” in *Proceedings of the 1997 International Conference on Network Protocols, IEEE Computer Society (ICNP '97)*, Washington, DC, USA, 1997.
 - [23] D. Wischik, “Fairness, QoS, and buffer sizing,” *SIGCOMM Computer Communication Review*, vol. 36, no. 1, p. 93, 2006.
 - [24] M. Wang and Y. Ganjali, *The Effects of Fairness in Buffer Sizing*, Atlanta, Ga, USA, 2007.
 - [25] L. Xue, C. Cui, S. Kumar, and S.-J. Park, “Experimental evaluation of the effect of queue management schemes on the performance of high speed TCPs in 10Gbps network environment,” in *Proceedings of the International Conference on Computing, Networking and Communications (ICNC '12)*, pp. 315–319, February 2012.
 - [26] M. Tekala and R. Szabo, “Modeling Scalable TCP friendliness to NewReno TCP,” in *International Journal of Computer Science and Network Security*, vol. 7, no. 3, pp. 89–96, March 2007.
 - [27] Y.-T. Li, D. Leith, and R. N. Shorten, “Experimental evaluation of TCP protocols for high-speed networks,” *IEEE/ACM Transactions on Networking*, vol. 15, no. 5, pp. 1109–1122, 2007.
 - [28] R. Jain, A. Duresi, and G. Babic, “Throughput fairness index: an explanation,” in *ATM Forum Contribution*, vol. 45, 1999.
 - [29] V. Misra, W.-B. Gong, and D. Towsley, “Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED,” in *Proceedings of ACM SIGCOMM Conference*, pp. 151–160, September 2000.
 - [30] Y. Liu, F. L. Presti, V. Misra, D. Towsley, and Y. Gu, “Fluid models and solutions for large-scale IP networks,” in *Proceedings of the International Conference on Measurement and Modeling of Computer Systems (ACM SIGMETRICS '03)*, pp. 91–101, June 2003.
 - [31] G. Appenzeller, I. Keslassy, and N. McKeown, “Sizing router buffers,” in *Proceedings of the Conference on Computer Communications (ACM SIGCOMM '04)*, pp. 281–292, Portland, Ore, USA, September 2004.
 - [32] M. Wang, “Mean-field analysis of buffer sizing,” in *Proceedings of the 50th Annual IEEE Global Telecommunications Conference (GLOBECOM '07)*, pp. 2645–2649, November 2007.
 - [33] D. Leith and R. Shorten, “H-TCP: TCP for High-speed and Long-distance Networks,” in *Proceedings of the Protocols for Fast Long-Distance Networks (PFLDnet '04)*, Argonne, 2004.
 - [34] S. Ha, Y. Kim, L. Le, I. Rhee, and L. Xu, “A step toward realistic performance evaluation of high-speed TCP variants,” in *Proceedings of the 4th International Workshop on Protocols for Fast Long-Distance Networks (PFLDNet '06)*, 2006.
 - [35] C. Yuan, L. Tan, L. L. H. Andrew, W. Zhang, and M. Zukerman, “A Generalized FAST TCP scheme,” *Computer Communications*, vol. 31, no. 14, pp. 3242–3249, 2008.
 - [36] L. Tan, C. Yuan, and M. Zukerman, “FAST TCP: fairness and queuing issues,” *IEEE Communications Letters*, vol. 9, no. 8, pp. 762–764, 2005.
 - [37] L. Xue, S. Kumar, C. Cui, and S.-J. Park, “An evaluation of fairness among heterogeneous TCP variants over 10Gbps high-speed networks,” in *Proceedings of the 37th Annual IEEE Conference on Local Computer Networks (LCN '12)*, pp. 348–351, Clearwater, Fla, USA, 2012.
 - [38] L. Xue, S. Kumar, C. Cui, P. Kondikoppa, C.-H. Chiu, and S.-J. Park, “AFCD: an approximated-fair and controlled-delay queuing for high speed networks,” in *Proceedings of the 22nd IEEE International Conference on Computer Communications and Networks (ICCCN '13)*, pp. 1–7, 2013.
 - [39] V. Jacobson, “Congestion avoidance and control,” in *Proceedings of the ACM SIGCOMM Computer Communication Review*, pp. 314–329, 1988.
 - [40] C. Villamizar and C. Song, “High performance TCP in ANSNET,” *SIGCOMM Computer Communication Review*, no. 5, pp. 45–60, 1994.

