# Impact of Queue Management Schemes and TCP Variants on the Performance of 10Gbps High Speed Networks: An Experimental Study

Lin Xue*, Cheng Cui*, Suman Kumar† and Seung-Jong Park*
*Department of Computer Science, Center for Computation & Technology, Louisiana State University, LA, USA
†Department of Computer Science, Troy University, AL, USA
Email: {xuelin, sumank, ccui, sjpark}@cct.lsu.edu

***Abstract*—Queue management schemes at routers and congestion avoidance schemes at end points cooperate to provide good congestion solutions in computer networks. While queue management schemes are still being developed, research on congestion avoidance has come a long way to serve the bandwidth requirement of the networks (e.g. high speed networks, data centers, etc.) at the order of 10Gbps. Because of considerable lack of the evaluation research work, there is no consensus on the choice of the queue management algorithms over these networks. To the best of our knowledge, this is the first work that presents the experimental study of the effect of various queue management schemes on high speed TCP variants in a 10Gbps network environment. Evaluations of queue management schemes such as Drop-tail, RED, CHOKe, and SFB are presented with popular high speed TCP variants such as RENO, HSTCP, CUBIC, and VEGAS over CRON, a 10Gbps high speed network testbed. Performance results are presented for several important metrics of interest such as link utilization, intro-protocol fairness, RTT fairness, delay and computational complexity. We argue the importance of explicit consideration of the basic tradeoffs between TCP variants and router parameters that network designers must face when building high speed networks.**

***Index Terms*—High Speed Networks, Active Queue Management, High Speed TCP Variants, Buffer Sizing**

## I. Introduction

Over the years, the Drop-tail queue mechanism has been under scrutiny and is found to be unsuitable choice to address issues such as transmission control protocols (TCP) global synchronization, underutilization of link bandwidth, high packet drop rate, high transmission delay, and high queuing delay. To address these issues, Random Early Detection (RED) [1] was proposed as an active queue management (AQM) solution in 1993. Thereafter, several AQMs, such as CHOKe(CHOose and Keep for responsive flows, CHOose and Kill for unresponsive flows) [2], SFB(Stochastic Fair Blue) [3] etc., have been proposed. Inspite of so many AQM proposals, there has been a significant lack of comparative performance evaluation studies on real production networks to permit any conclusion on the merits of these QM schemes. There are two major consequences for the lack of comparative studies. Firstly, although these AQMs are theoretically superior to Drop-tail, these AQMs are still scarce in production networks; secondly, there is not much support from the testing to the development of future QM schemes. To address the challenges in design and development of QM schemes in future networks, our focus in this paper is to compare the performance of competing QM proposals in a systematic and repeatable manner in a 10Gbps high speed network environment.

Simulation and experiment are two main methods to perform such studies. Most of the evaluation research works on AQM schemes rely on simulation models, such as Network Simulator 2 (ns-2) or OPNET modeler. However, a linear increase in bandwidth demands for an exponential increase in CPU-time and memory usage for these discrete simulation methods [4] which makes it very difficult to finish the simulation of high speed links in a reasonable time. Besides, to address issues in design and real network deployment of QM schemes demands a real experimental network environment [5].

Since the cost could be expensive for large businesses or ISP networks to deploy the AQM schemes in the Internet, recent network research [6] tries to check if it is beneficial to deploy the AQM schemes in the core routers. The existing research focuses on the nature of the AQM scheme itself, but overlooks the effect of the AQM scheme on transport layer congestion control. Often, the core of the network is a playground for transport protocols and therefore, it is difficult for a network service provider to address issues related to performance of their network. Due to the feedback nature of AQM schemes, TCPs will behave differently according to its own congestion control algorithm. Especially in production networks (or data centers), the pairing of a TCP variant and an appropriate QM scheme to compliment that TCP is highly desirable. In other words, it is highly desirable to know the impact of QM schemes on transport protocols. Therefore, in this paper, the performance metrics for QM schemes are chosen to be very TCP specific. We also consider the metrics such as memory usage and CPU requirement which we find in a direct correlation with ease of deployment and operational costs.

We choose the popular QM schemes for evaluation, including Drop-tail, RED, CHOKe, and SFB. Among high speed TCP variants, we select CUBIC (CUBIC TCP) [7], HSTCP (HighSpeed TCP) [8], RENO (TCP-Reno), and VEGAS (TCP-Vegas) [9]. It was noted that CUBIC, HSTCP and RENO account for $2/3$ of all TCP variants used on the Internet [10].


doi:10.4304/jnw.9.5.1183-1192

VEGAS represents the delay-based TCP, and is the only delay-based TCP supported in the current Linux kernel. We present the results in terms of link utilization, intro-protocol fairness, RTT fairness, delay, and computational complexity.

The organization of this paper is as follows: Section II presents a brief overview of related work. Section III is the background and motivation of this work. Section IV gives the experimental design and setup, the results of the experiment are discussed in Section V, and we conclude our findings in Section VI.

## II. Related Work

To evaluate performance of TCP variants, the authors in [11] and [12] evaluated high speed TCP protocols in a realistic high speed networking environment. The high speed TCP protocols were evaluated against itself in terms of several TCP performance metrics. However, when evaluating all TCP protocols, the authors did not consider the impact of queue management schemes in the router.

In [13], the authors presented a framework to evaluate AQM schemes. Five metrics were chosen to characterize overall network performance of AQM schemes. The authors suggested simulation environments and scenarios, including ns-2 interfaces, traffic models and network topologies. As a continuing work [14], the authors gave simulation based evaluation and comparison of a subset of AQM schemes. Their framework was based on ns-2 simulation which is different than a real-world experiment, especially the 10Gbps high speed networks.

The authors in [15] evaluated new proposed AQM schemes with some specific network scenarios. They proposed a common testbed for the evaluation of AQM schemes which includes a specification of the network topology, link bandwidths and delays, traffic patterns, and metrics for the performance evaluation. Also, the authors realized that AQM schemes need to cooperate closely with TCP. However, they evaluated AQM schemes over regular Internet speed but not 10Gbps speed, and only presented the results of TCP-RENO in their evaluation.

Moreover, the authors in [16] considered router buffer sizing in evaluation of high speed TCP protocol. They conducted an experimental evaluation of CUBIC TCP in small router buffers (e.g. a few tens of packets). Their work highlighted the need for a thorough investigation on the performance of high speed TCP variants with small router buffers for newly emerging high speed networks.

In a recent work [17], the authors found the tradeoff between throughput and fairness in high speed networks. The network performance was evaluated by a model based simulation method, which shows some bottlenecks in evaluating high speed networks. In [18], the authors evaluated the impact of queue management schemes on the performance of TCP over 10Gbps high speed networks. However, the detailed setup of a 10Gbps environment was not unveiled. And some of the research results were not presented such as TCP-VEGAS result, intro-protocol fairness result, memory consumption, etc. In [19], fairness was evaluated thoroughly among heterogeneous high speed TCP variants by using different queue management schemes with varying degrees of buffer sizes, but other metrics have not been fully evaluated yet. The authors in [20] evaluated extensively both fairness and latency of AQM schemes over 10Gbps high speed networks. They proposed a new AQM schemes which works well in terms of fairness and latency over 10Gbps high speed networks.

In this paper, we consider the most important metrics which need to be evaluated for the interrelationship between TCP variants and queue management schemes. Performance metrics have been defined in [21] previously. The authors discussed the metrics to be considered to evaluate congestion control mechanisms for the Internet. They brought 11 metrics in total, which could be used for evaluating new or modified transport layer protocols.

## III. Background and Motivation

### A. Challenges in 10Gbps High Speed Networks

Advances in high speed networking technology coincide with the need for network infrastructure development to support scientific computing, distant learning, e-commerce, health, and many other unforeseen future applications. Consequently, 10Gbps high speed networks, such as Internet2 [22], NRL (National LambdaRail) [23], and LONI (Louisiana Optical Network Initiative) [24], have been the first ones that were developed to connect a wide range of academic institutes. Fig. 1 shows a 10Gbps network infrastructure encompassing the campus network of LSU (Louisiana State University) and MAX (Mid-Atlantic Crossroads). Network operators use Internet2 network stitching [25] to federate all campus networks together, and core routers are responsible for connection across various facilities located in-campus. The 10Gbps high speed network enables resource sharing among different departments at different institutes. As gigabit connectivities are easily available for gigabit-based PCs, servers, data center storage and high performance computing, gigabit networking technology is preferred by many organizations. Therefore, various organizations are increasingly migrating to gigabit links in order to grow their networks to support new applications and traffic types. It is true that the availability of multi-gigabit switches/routers provides the opportunity to build high-performance, high-reliability networks but only if correct design approaches are followed at both hardware and software level.

Our focus in this paper is on network communication protocols running on deployed networks which in combination acts as a life force that brings the best performance to these networks. In particular, we are interested in what really matters when it comes to performance of 10Gbps network, in which we explore the role of router technology in the middle and data communication protocols at the end systems. Our goal in this paper is not to provide a complete design and deployment details which otherwise is an extremely difficult task but a minimal one where combined impact of router parameters and data transport protocols would be critical for network performance. We claim that very simple considerations, which incorporate available technological constraints together with

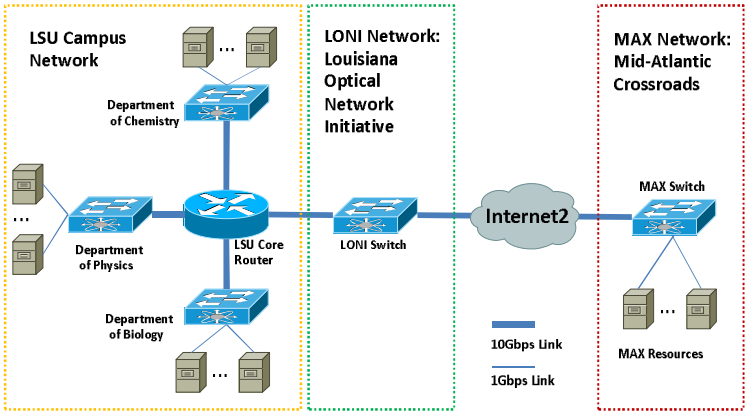



Fig. 1. 10Gbps high speed campus networks connected by Internet2 for large scale scientific computing, distance learning, etc.

relevant network performance metrics, can successfully address this challenge and further reduce the confusion and controversy in design and deployment issues of 10Gbps networks. We argue that one needs to jointly consider the coupling of TCP and router technology in the design and deployment of multi-gigabit networks.

### B. Network Performance Consideration

Previous experimental studies treated evaluation of TCP, evaluation of AQMs (see section II) and effect of router parameters running a particular AQM on TCP separately. Also, these studies do not include 10Gbps bandwidth consideration. In this paper, we propose a complementary approach of combining TCP, router parameters and a high speed network of order of 10Gbps.

The following equation summarizes the dynamics of the TCP congestion window $W(t)$ at time $t$:

$$W(t) = W_{max}\beta + \alpha\frac{t}{RTT} \quad (1)$$

$W_{max}$ is the congestion window size just before the last window reduction, RTT is the round trip delay of this flow, and $\alpha$ and $\beta$ are increase and decrease parameters respectively.

$W_{max}$ depends on router parameters, such as a penalty signal $P$ of queue management schemes , router buffer size $Q$, and the bottleneck capacity $C$:

$$W_{max} \leftarrow \frac{QC}{P} \quad (2)$$

From the equation above, it is clear that the performance of a network depends on the combination of TCP variants, queue management schemes and router buffer sizes. Our argument is consistent with [26], which concluded that the TCP sending rate depends on both the congestion control algorithms and the queue management schemes in the links. A real network measurement on high speed networks shows that burstiness increases with bandwidth because packets degenerate into extremely bursty out flows with data rates going beyond the available bandwidth for short periods of time [27]. It is clear that such surges can easily generate severe penalty signals and further degrade the overall network performance.

The penalty signals for different queue management schemes are also different. When the router buffer is full, Drop-tail drops all the packets. RED drops packets early and randomly according to the queue length. CHOKe extends RED to compare random packets and drops packets for fast flows. SFB uses a bloom-filter to determine fast flows and drops packets from fast flows. Buffer size at the routers also impacts network performance (see section II). Router buffers cause queuing delay and delay-variance. And in the case of underflow, throughput degradation is observed. Appropriate sizing of buffers has been considered a difficult task for router or switch manufacturers.

Given the importance of router parameters, different high speed TCP variants will differ in performance for the same router parameters. We elaborate on this point by considering the impact of penalty signals on congestion window TCP variants in consideration as below:

1) Traditional TCP's AIMD algorithm has the increase parameter $\alpha$ and decrease parameter $\beta$ to be 1 and 0.5 respectively.

2) HSTCP's increase parameter $\alpha$ and decrease parameter $\beta$ are functions of the current window size, namely $\alpha(W)$ and $\beta(W)$. The range of $\alpha(W)$ could be from 1 to 73 packets, and $\beta(W)$ from 0.5 to 0.09.

3) CUBIC updates the congestion window according to a cubic function:

$$W_{CUBIC} \leftarrow C(t - \sqrt[3]{W_{max}\beta/C})^3 + W_{max} \quad (3)$$

where $C$ is a scaling factor, $t$ is the elapsed time since the last window reduction, $W_{max}$ is the window size just before the last window reduction, and $\beta$ is the decrease parameter.

4) VEGAS is a delay-based TCP variant. It has 2 thresholds, $\alpha$ and $\beta$, to control the amount of extra data, i.e $T_{extra}$ = $T_{expected}$ - $T_{actual}$, where $T_{expected}$ is an estimation of expected throughput calculated by $T_{expected}$ = $windowsize$ / $smallestmeasuredRTT$. Window size of VEGAS is updated

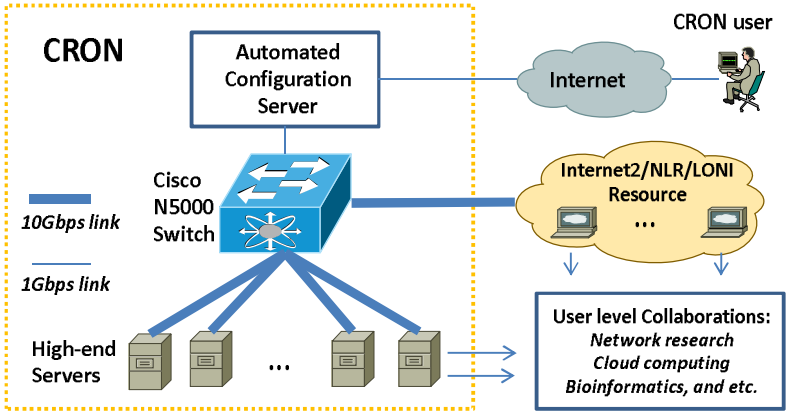



Fig. 2. CRON system architecture consisting of routers, delay links, and high-end workstation operating up to 10Gbps bandwidth

as follows:

If $T_{expected} < \alpha$, window size increased by 1.
If $\alpha < T_{expected} < \beta$, no change in window size.
If $T_{expected} > \beta$, window size decreased by 1.

A detailed understanding of the many facts of network parameters is critical for evaluating the performance of networking protocols, for assessing the effectiveness of proposed protocols, and for developing the next generation high speed networks. In this work, we narrow down our focus to three key components that affect the performance of 10Gbps networks: TCP variants, queue management schemes, and router buffer size.

## IV. Experimental Design

In this section, we present our experimental study design.

### A. CRON Setup

CRON [28] is an emulation-based 10Gbps high speed testbed, which is a cyberinfrastructure of reconfigurable optical networking environment that provides multiple networking testbeds operating up to 10Gbps bandwidth. As shown in Fig. 2, CRON provides users with automatic configuration of arbitrary network topologies with 10Gbps bandwidth. Also, CRON can be federated with other 10Gbps high speed networks, such as Internet2, NLR, and LONI. Details of demonstrations of how to use the CRON testbed could be found here [29].

As dumbbell topology is a widely accepted network topology, we create a dumbbell topology as shown in Fig. 3 in CRON. In the topology, all nodes are Sun Firex4240 servers which have two quad core 2.7-GHz AMD Opteron 2384 processors, 8 GB/s bus, 8GB RAM and 10GE network interface cards. From a software perspective, two pairs of senders and receivers run a modified version of Linux 2.6.34 kernel, which supports TCP variants of CUBIC, HSTCP, RENO, and VEGAS. The routers run a modified version of Linux-2.6.39.3 kernel, which supports queuing disciplines of Drop-tail, RED, CHOKe, and SFB. The delay node runs a modified version of FreeBSD 8.1, which supports a 10Gbps version of Dummynet [30] with 10Gbps bandwidth and enlarged queue size.

We set the RTT on the delay node to 120ms. All the links have a 10Gbps capacity, and the bottleneck link is the one between Router1 and Router2. We set the queue disciplines at the output queue of Router1, where the congestion happens.

By default, we send 10 flows from Sender1 to Receiver1 and 10 flows from Sender2 to Receiver2. We choose the number of flows to be 10 according to recent statistics of network flows of Internet2 [31], which suggested that the number of long-lived TCP flows are always several tens of flows in 10Gbps high speed networks such as Internet2. The duration of each emulation test is 20 minutes, and all tests run for 7 to 10 times. We get the final result based on the average value.

### B. System Tuning for 10Gbps

To get a systematic and repeatable 10Gbps network environment, we perform system tuning and software patching in the CRON testbed.

Firstly, on the senders and receivers with Linux kernel 2.6.34, we enlarge the default TCP buffer size for high speed TCP transmit. According to [32], we implement the zerocopy Iperf to avoid the overhead of data copy from user-space to kernel space, and we enable packets large receive offload (LRO) and TCP segment offload on the NICs. We also set MTU to 9000 Bytes [33].

Secondly, on the routers with Linux kernel 2.6.39.3, the Linux default queuing discipline controller, traffic control (tc), does not support control for CHOKe and SFB in user-space. So we patch tc(8) to support CHOKe and SFB. In kernel-space, we find that for RED, the scaled parameter of maximum queue threshold only supports 24 bit value which means up to only 16MB. So we change it to a 56 bit value to support a higher maximum queue threshold. In addition, we use standard $skbuf$ to forward packets and disable LRO on the router NICs.

Thirdly, on the delay node which runs FreeBSD 8.1, we optimize memory utilization by creating a continuous memory space for received packets to overcome the drawback of the memory fragmentation of $Mbuf$ allocation in FreeBSD. In Dummynet, we change the type of bandwidth from $int$ to $long$ so that its bandwidth capacity gets an improvement from

2Gbps to 10Gbps. We also increase the value of the Dummynet hardware interruption storm threshold.

### C. Queue Parameter Setup

Queue management schemes in consideration along with its parameters are shown in Table 1.

TABLE I
PARAMETER SETUP FOR 4 QUEUE MANAGEMENT SCHEMES

| Queue | Parameter Setup |
|---|---|
| Drop-tail | queue length $limit$: from 1% to 100% BDP |
| RED | queue length $limit$: from 1% to 100% BDP<br>minimum threshold $qth_min$: $0.1 \times limit$<br>maximum threshold $qth_max$: $0.9 \times limit$<br>average packet size $avpkt$: 9000<br>maximum probability $max_P$: 0.02 |
| CHOKe | queue length $limit$: from 1% to 100% BDP<br>minimum threshold $qth_min$: $0.1 \times limit$<br>maximum threshold $qth_max$: $0.9 \times limit$ |
| SFB | queue length $limit$: from 1% to 100% BDP<br>increment of dropping probability $increment$: 0.00050<br>decrement of dropping probability $decrement$: 0.00005<br>Bloom filter uses two 8 x 16 bins<br>target per-flow queue size $target$: 1.5/N of total buffer size<br>N: number of flows<br>maximum packets queued $max$: $1.2 \times target$ |

In [34], the authors suggest that a link needs only a buffer of size O( C / $\sqrt{N}$), where C is the capacity of the link, and N is the number of flows sharing the link. In addition, we vary the router buffer size to examine all the combinations of TCP variants and queue management schemes. In [35], the authors suggest that buffers can be reduced even further to 20-50 packets. Given the significance of their role, we vary the buffer size as 1%, 5%, 10%, 20%, 40%, and 100% of BDP to find out the impact of the router buffer sizing on AQM schemes in 10Gbps high speed networks.

### D. Performance Metrics

(1) Link Utilization. Link utilization is the percentage of total bottleneck capacity utilized during an experiment run.

(2) Intra-protocol Fairness. Intra-protocol fairness represents the fairness among all TCP flows. Long term flow throughput is used for computing fairness according to Jain's fairness index [36].

(3) RTT fairness. RTT fairness is the fairness among TCP flows with different RTTs. Longer RTT TCP flows suffer from lower throughput and shorter RTT flows get more throughput.

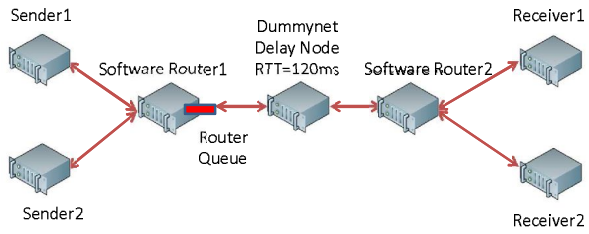



Fig. 3. Experimental topology: dumbbell topology

(4) Delay. Delay is the average end-to-end delay experienced by the flows, also known as Round-trip delay (RTT) of packets across the bottleneck paths. It includes the queuing delay created by the queue at router.

(5) Computational Complexity. Computational complexity is the algorithm space and time complexity of the queue management schemes, which is the memory consumption and CPU usage on the servers in our experiments.

## V. EXPERIMENTAL RESULTS AND DISCUSSION

### A. Link Utilization

We vary queue buffer size to observe the link utilization. Fig.4(a), Fig.4(b), Fig.4(c), and Fig.4(d) show link utilization for queue management schemes as a function of buffer size for CUBIC, HSTCP, RENO with SACK, and VEGAS respectively. With only 1% BDP buffer size on the bottleneck link; almost all of the queue management schemes for all TCP variants show more than 85% link utilization, which is close to the previous sizing router buffer researches [34], [37]. And if the buffer size reaches 10% of BDP, almost all the queue management schemes under all TCPs will get more than 90% link utilization except for TCP-VEGAS.

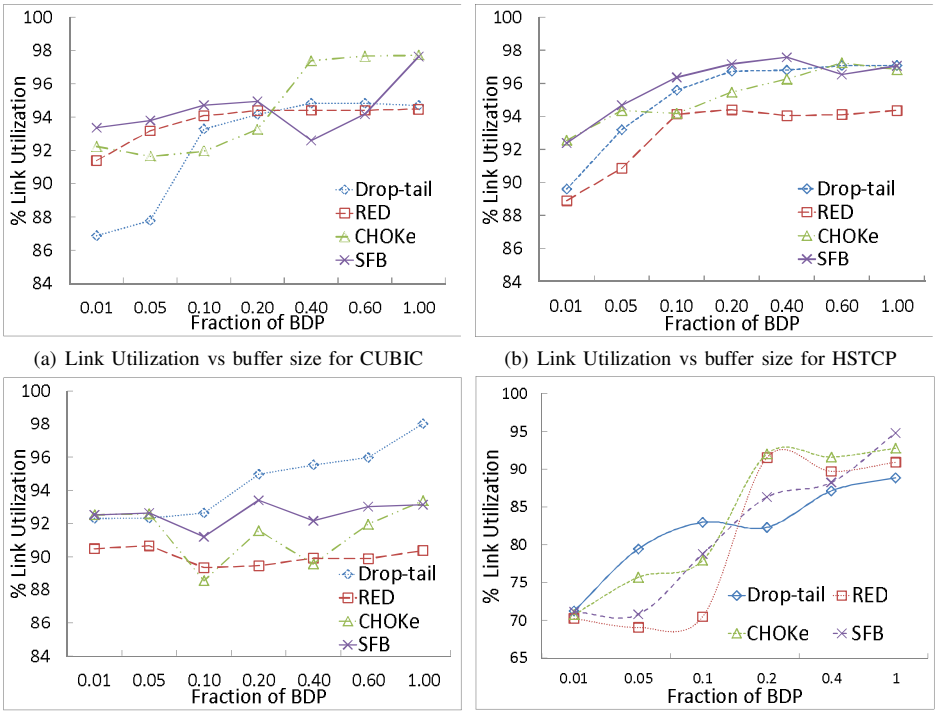
Fig.4(a) is for link utilization of queue management schemes for CUBIC, when the buffer size is very small to 1% BDP, Drop-tail performs worst, while SFB gets highest link utilization among others. If the buffer size increases up to 100% BDP, CHOKe and SFB get higher link utilization. In Fig.4(b), link utilization of queue management schemes is for HSTCP, and SFB almost always outperforms other queuing schemes, while RED almost always gets lowest link utilization than others. Fig.4(c) shows the link utilization of queue management schemes for RENO with SACK, Drop-tail performs best in this case, SFB is still better than the other two, while RED almost always gets the lowest link utilization.

In Fig.4(d), VEGAS shows different link utilization behaviors because of its delay-based nature, which depends on the queue size. In general, when the buffer size becomes larger, all queue management schemes get higher link utilization. In the case of less than 10% BDP, Drop-tail almost always gets the highest link utilization. In the case of more than 10% BDP, AQM schemes almost always get higher link utilization. The reason is when the queue size becomes larger, AQM schemes do not have early drops because VEGAS controls the queue size in a limited range, and therefore, link utilization of the AQM scheme improves.

### B. Intra-protocol Fairness

Intra-protocol fairness is the fairness among TCP flows with the same kind of TCP. In our evaluation, Jains fairness index is calculated for intra-protocol fairness among 20 flows with same TCP variant and same RTT of 120ms.

Fig.5 shows the intra-protocol fairness for these 20 flows. In general, CUBIC shows the highest fairness, which has a fairness index in the range of 0.97 to 1. HSTCP seconds with a fairness index in the range of 0.94 to 0.99. RENO is third,



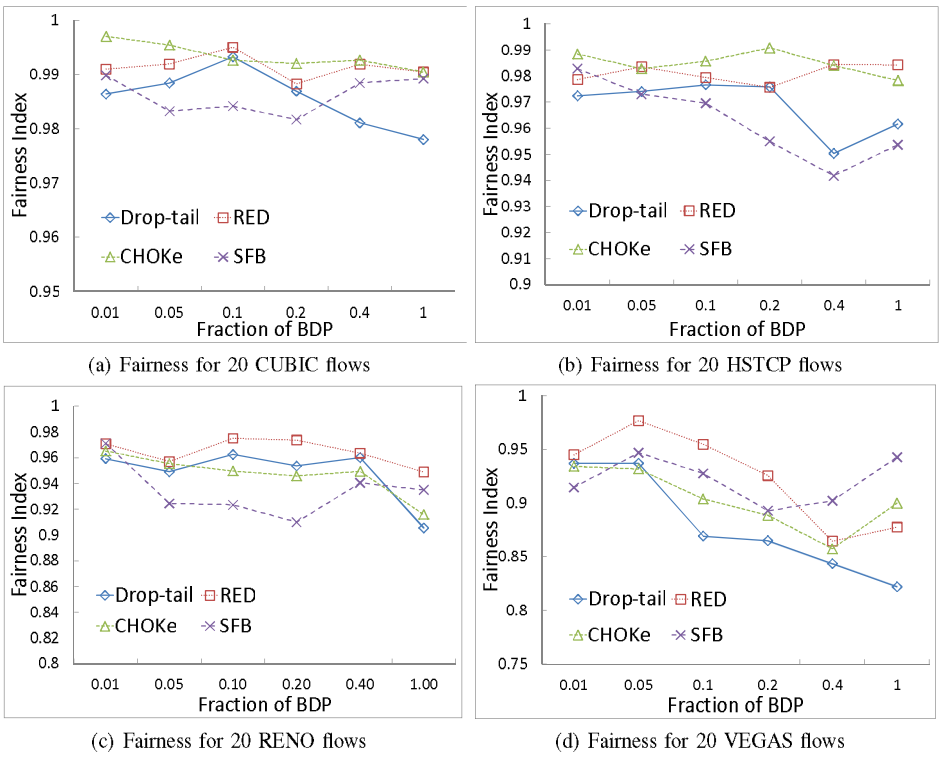

(a) Link Utilization vs buffer size for CUBIC

(b) Link Utilization vs buffer size for HSTCP

(c) Link Utilization vs buffer size for RENO

(d) Link Utilization vs buffer size for VEGAS

Fig. 4. Link Utilization as a function of buffer size in each TCP variant




(a) Fairness for 20 CUBIC flows

(b) Fairness for 20 HSTCP flows

(c) Fairness for 20 RENO flows

(d) Fairness for 20 VEGAS flows

Fig. 5. Fairness among 20 flows as a function of buffer size in each TCP variant (RTT = 120ms)

which has a fairness index higher than 0.89. VEGAS is last with a fairness index higher than 0.8.

Fig.5(a) shows the case for CUBIC, RED and CHOKe have a very high intro-protocol fairness around 0.99 fairness index. SFB generally shows lower intro-protocol fairness than other queue management schemes. According to our observation, although SFB has bucket drops to limit the fast flows, SFB always has more tail drops than other queue management schemes in high speed networks because of its more complex queuing mechanism. That is the reason SFB can not perform as fairly as other AQM schemes.

Fig.5(b) is the result for HSTCP, which is similar to the case of CUBIC. RED and CHOKe still show higher intro-protocol fairness, while Drop-tail and SFB show lower intro-protocol fairness.

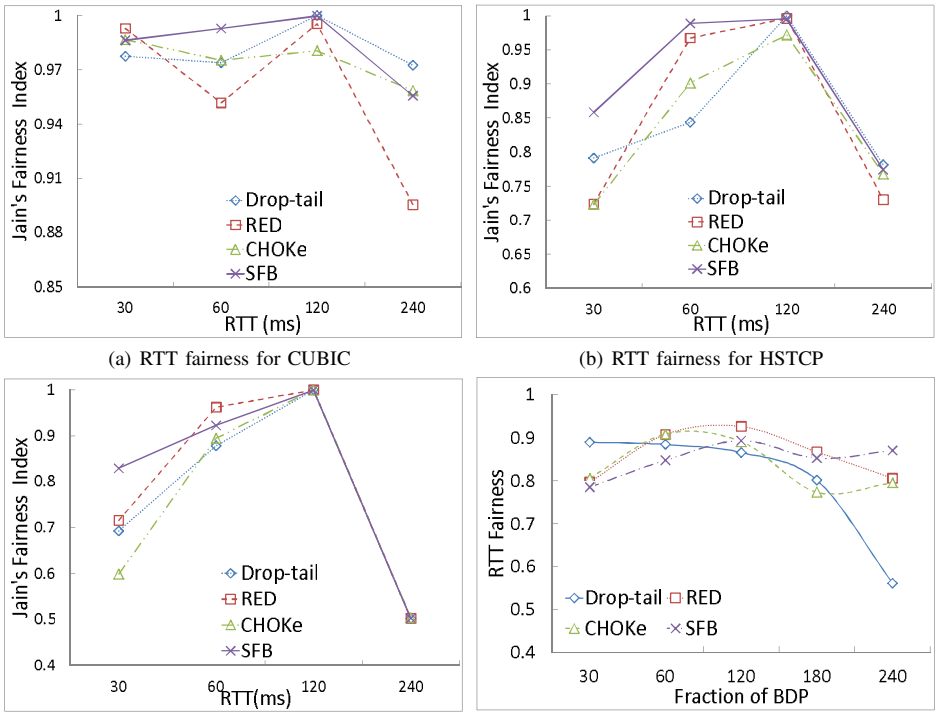
In the case of RENO, Fig.5(c) shows that RED always gets the highest intro-protocol fairness. Drop-tail is second, and CHOKe is the third. The slow instinct of RENO makes AQM schemes have a similar performance to Drop-tail in terms of fairness. Whenever RENO flows have early drops from AQM schemes, it takes some time for the flows to recover, which in consequence degrades the fairness of AQM schemes. SFB still almost always shows the lowest intro-protocol fairness because of having more tail drops than others.

Fig.5(d) shows the case for VEGAS. Since VEGAS is a delay-based TCP variant, it keeps the queue size as small as possible. AQM schemes all get better fairness than Drop-tail. Drop-tail makes relatively large changes on the queue size, and therefore it performs relatively unfair.

### C. RTT Fairness

We measure RTT fairness by Jain's fairness index for 20 competing flows of the same TCP variant but different RTTs. In these 20 flows, 10 of them have a fixed RTT, while the other 10 flows have a different RTT. The RTT of 10 flows from Sender1 is fixed at 120ms, while the RTT of the other 10 from Sender2 is changed as 30ms, 60ms, 120ms, and 240ms respectively. We set the bottleneck buffer to 10% BDP because link utilization of the bottleneck link shows good performance with buffer size 10% BDP in general. Also, in this case queuing delay can be neglected (we present the detailed delay result in Section V-D).

Fig. 6(a) shows CUBIC RTT fairness for four queue management schemes. In our measurement, CUBIC shows very good behavior of RTT fairness. Even under 240ms RTT, every queue management scheme shows more than 90% of RTT fairness. SFB performs better than others; while RED gets least fairness. Fig. 6(b) and 6(c) show HSTCP's and RENO's RTT fairness cases. HSTCP's RTT fairness is better than RENO's for all queue management schemes, and both HSTCP's RTT fairness and RENO's RTT fairness are quite lower than CUBIC's. We can still see SFB shows the best for all the TCP variants under different RTT scenarios, and when one RTT increases to 240ms, we get a low RTT fairness in both cases. Fig. 6(d) shows VEGAS's RTT fairness. Every queuing scheme gets around 0.8 to 0.9 fairness index except that of Drop-tail which gets a low RTT fairness in the case of 240ms RTT.



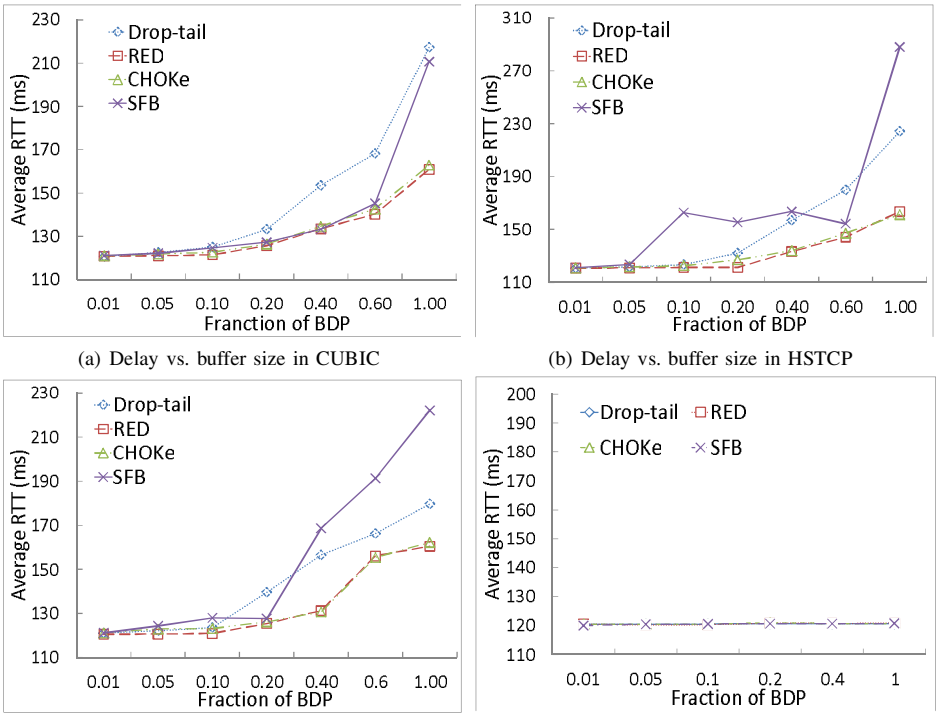

(a) RTT fairness for CUBIC

(b) RTT fairness for HSTCP

(c) RTT fairness for RENO

(d) RTT fairness for VEGAS

Fig. 6. RTT fairness as a function of RTT in each TCP variant




(a) Delay vs. buffer size in CUBIC

(b) Delay vs. buffer size in HSTCP

(c) Delay vs. buffer size in RENO

(d) Delay vs. buffer size in VEGAS

Fig. 7. Delay as a function of buffer size in each TCP variant

### D. Delay

We observe performance in delay by varying buffer size. Since this measurement is based on round trip propagation delay of 120ms, the average RTT will be $120 + [0, max\_queuing\_delay]$. As shown in Fig. 7(a) for CUBIC result, Drop-tail always has more queuing delays than others. SFB is the second one with more queuing delay than RED and CHOKe. Fig. 7(b) shows result for HSTCP, we can still see Drop-tail and SFB show more queuing delay than the others, and SFB shows an oscillation, and exhibits queue delays more than double the amount of propagation delays. In RENO of Fig. 7(c), we observe similar results, Drop-tail and SFB queuing delays grow faster than the others. RED and CHOKe show nearly the same behavior and both grow more smoothly with the increase in buffer size as compared to Drop-tail and SFB.

Fig. 7(d) shows results for VEGAS. For all queue management schemes, VEGAS almost does not create any queuing delay. In all cases, the average RTTs for VEGAS are only 120ms, which is the propagation delay we set. This is because VEGAS itself maintains the queue in a very small size, such as several packets. The results confirm that delay-based TCP variant maintains a stable and small queue size in 10Gbps high speed networks.

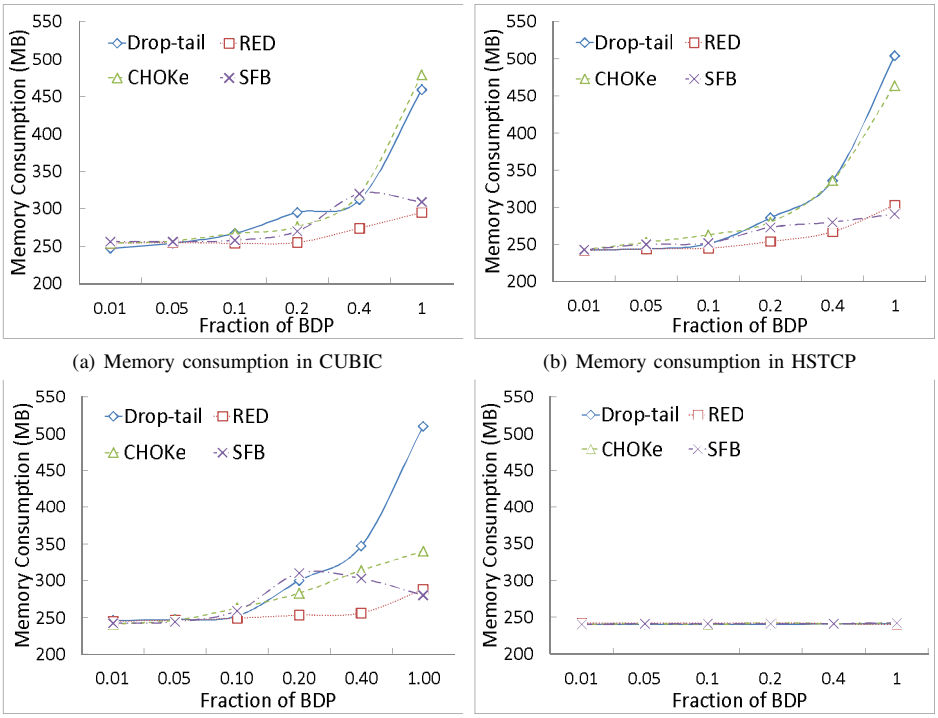
### E. Computational Complexity

Fig. 8 shows the average memory consumption in a function of buffer size on the bottleneck router. We can see that for CUBIC, HSTCP and RENO, the general trend is that when the buffer size increases, the memory consumption increases. When the buffer size is 100% BDP, the memory consumption reaches more than 500MB. Drop-tail generally needs more memory than other AQM schemes. Fig. 8(d) shows that VEGAS almost does not create any additional memory for queuing mechanisms, because it maintains a very small size queue.

The CPU usage results reveal less than 10% of total CPU usage in all of our experiment, so we do not list the detailed CPU usage result here.

## VI. CONCLUSION

In this paper, we present the experimental study of the interplay of queue management schemes and high speed TCP variants over a 10Gbps high speed networking environment. TCP specific performance metrics such as link utilization, fairness, delay, and computational complexity are chosen to compare the impact of queuing schemes on the performance of TCP-RENO, CUBIC, HSTCP and VEGAS. Our test reveals that Drop-tail is most suited for TCP-RENO and observed to be worst for CUBIC and HSTCP. In our experiment scenario, we observe at least 10% BDP of buffer size is required for more than 90% link utilization. RED exhibits higher fairness as compared to other QMs for all the TCP variants. SFB is shown to be effective in RTT fairness improvement. TCP VEGAS shows very low queuing delay and memory consumption. In summary, we observe differences in performance of QMs for different TCP variants.




(a) Memory consumption in CUBIC

(b) Memory consumption in HSTCP

(c) Memory consumption in RENO

(d) Memory consumption in VEGAS

Fig. 8. Memory consumption as a function of buffer size in each TCP variant

We hope that even a preliminary understanding of key factors, when combined with critical performance metrics, can provide a perspective that is easily understood and can serve as guidelines for network designers of 10Gbps high speed networks. Also, the results of the study address the current need for the research on the impact of queue management schemes on the performance of the high speed TCP variants. It is also desirable to observe the same impacts on a more realistic experimental environment by considering background traffic. For our future work, it is interesting to observe the impact of these queue management schemes in a wide range of different network topologies. In this paper, although our focus has been on homogeneous TCP flows, we expect a different behavior in the case of heterogeneous TCP flows. The presented work supports further research work on the design and deployment issues of queue management schemes for high speed networks.

***Acknowledgement:*** This work has been supported in part by the NSF CC-NIE Grant #1341008 and DEPSCoR project N0014-08-1-0856.

## REFERENCES

[1] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *Networking, IEEE/ACM Transactions on*, vol. 1, no. 4, pp. 397–413, 1993.

[2] R. Pan, B. Prabhakar, and K. Psounis, "Choke-a stateless active queue management scheme for approximating fair bandwidth allocation," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2. IEEE, 2000, pp. 942–951.

[3] W. Feng, D. Kandlur, D. Saha, and K. Shin, "Stochastic fair blue: A queue management algorithm for enforcing fairness," in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3. IEEE, 2001, pp. 1520–1529.

[4] S. Kumar, S. 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.

[5] S. Floyd and V. Paxson, "Difficulties in simulating the internet," *IEEE/ACM Transactions on Networking (TON)*, vol. 9, no. 4, pp. 392–403, 2001.

[6] P. Mrozowski and A. Chydzinski, "On the deployment of aqm algorithms in the internet," in *Proceedings of the 11th WSEAS international conference on Mathematical methods and computational techniques in electrical engineering*. World Scientific and Engineering Academy and Society (WSEAS), 2009, pp. 276–281.

[7] S. Ha, I. Rhee, and L. Xu, "Cubic: A new tcp-friendly high-speed tcp variant," *ACM SIGOPS Operating Systems Review*, vol. 42, no. 5, pp. 64–74, 2008.

[8] S. Floyd, "Highspeed tcp for large congestion windows," *RFC 3649*, 2003.

[9] L. Brakmo, S. O'malley, and L. Peterson, *TCP Vegas: New techniques for congestion detection and avoidance*. ACM, 1994, vol. 24, no. 4.

[10] P. Yang, W. Luo, L. Xu, J. Deogun, and Y. Lu, "Tcp congestion avoidance algorithm identification," in *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*. IEEE, 2011, pp. 310–321.

[11] S. Ha, Y. Kim, L. Le, I. Rhee, and L. Xu, "A step toward realistic evaluation of high-speed tcp protocols," in *Proc. International Workshop on Protocols for Fast Long-Distance Networks (PFLD-net2006)*, 2006.

[12] S. Ha, L. Le, I. Rhee, and L. Xu, "Impact of background traffic on performance of high-speed tcp variant protocols," *Computer Networks*, vol. 51, no. 7, pp. 1748–1762, 2007.

[13] A. Bitorika, M. Robin, and M. Huggard, "An evaluation framework for active queue management schemes," in *Modeling, Analysis and Simulation of Computer Telecommunications Systems, 2003. MASCOTS 2003. 11th IEEE/ACM International Symposium on*. IEEE, 2003, pp. 200–206.

[14] A. Bitorika, M. Robin, M. Huggard, and C. Mc Goldrick, "A comparative study of active queue management schemes," in *Proceedings of IEEE ICC 2004*, vol. 201. Citeseer, 2004, p. 6.
[15] L. Chrost and A. Chydzinski, "On the evaluation of the active queue management mechanisms," in *Evolving Internet, 2009. INTERNET'09. First International Conference on*. IEEE, 2009, pp. 113–118.
[16] S. Jain and G. Raina, "An experimental evaluation of cubic tcp in a small buffer regime," in *Communications (NCC), 2011 National Conference on*. IEEE, 2011, pp. 1–5.
[17] S. Kumar, L. Xue, and S.-J. Park, "Impact of loss synchronization on reliable high speed networks: A model based simulation," *Journal of Computer Networks and Communications*, vol. 2014, 2014.
[18] 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 *Computing, Networking and Communications (ICNC), 2012 International Conference on*. IEEE, 2012, pp. 315–319.
[19] L. Xue, S. Kumar, C. Cui, and S.-J. Park, "An evaluation of fairness among heterogeneous TCP variants over 10gbps high-speed networks," in *37th Annual IEEE Conference on Local Computer Networks (LCN 2012)*, 2012, pp. 348–351.
[20] 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 *Computer Communications and Networks (ICCCN), 2013 22nd International Conference on*. IEEE, 2013, pp. 1–7.
[21] S. Floyd, "Metrics for the evaluation of congestion control mechanisms," 2005.
[22] Internet2, "Internet2," 2012, http://www.internet2.edu/.
[23] NLR, "National LambdaRail," 2012, http://nlr.net/.
[24] LONI, "Louisiana Optical Network Initiative," 2012, http://www.loni.org/.
[25] GENI, "Network Stitching," 2012, http://groups.geni.net/geni/wiki/GeniNetworkStitching/.
[26] A. Tang, J. Wang, S. Low, and M. Chiang, "Equilibrium of heterogeneous congestion control: Existence and uniqueness," *Networking, IEEE/ACM Transactions on*, vol. 15, no. 4, pp. 824–837, 2007.
[27] D. Freedman, T. Marian, J. Lee, K. Birman, H. Weatherspoon, and C. Xu, "Exact temporal characterization of 10 gbps optical wide-area network," in *Proceedings of the 10th annual conference on Internet measurement*. ACM, 2010, pp. 342–355.
[28] CRON, "CRON Project: Cyberinfrastructure for Reconfigurable Optical Networking Environment," 2011, http://www.cron.loni.org/.
[29] "CRON demonstration," 2011, https://www.cron.loni.org/crondemo.php/.
[30] M. Carbone and L. Rizzo, "Dummynet revisited," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 2, pp. 12–20, 2010.
[31] Internet2, "Internet2 Netflow Data," 2012, http://netflow.internet2.edu.
[32] T. Yoshino, Y. Sugawara, K. Inagami, J. Tamatsukuri, M. Inaba, and K. Hiraki, "Performance optimization of TCP/IP over 10 gigabit ethernet by precise instrumentation," in *Proceedings of the 2008 ACM/IEEE conference on Supercomputing*. IEEE Press, 2008, p. 11.
[33] Y. Wu, S. Kumar, and S. Park, "Measurement and performance issues of transport protocols over 10 gbps high-speed optical networks," *Computer Networks*, vol. 54, no. 3, pp. 475–488, 2010.
[34] G. Appenzeller, I. Keslassy, and N. McKeown, "Sizing router buffers," in *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 4. ACM, 2004, pp. 281–292.
[35] M. Enachescu, Y. Ganjali, A. Goel, N. McKeown, and T. Roughgarden, "Routers with very small buffers," in *Proc. IEEE Infocom*, vol. 6. Citeseer, 2006.
[36] R. Jain, A. Durresi, and G. Babic, "Throughput fairness index: an explanation," in *ATM Forum Contribution 99*, vol. 45, 1999.
[37] S. Kumar, M. Azad, and S. Park, "A fluid-based simulation study: The effect of loss synchronization on sizing buffers over 10gbps high speed networks," in *PFLDNeT*, 2010.

**Lin Xue** is a PhD student in Department of Computer Science, Louisiana State University. He received M.S. in computer science from Beijing University of Posts and Telecommunications in 2008 and B.S. in computer science from China Agricultral University in 2005. His research interests focus on performance of high-speed networks, high-speed queue management schemes, and high-speed congestion controls.

**Cheng Cui** recieved his Ph.D. in computer science from Louisiana State University. He graduated from Xidian University, Xi'an, China in 2006 with a bachelor's degree in School of Software Engineering. His research interests include Traffic Measurement in High Speed Networks; TCP Congestion Control for 10Gbps Network and Beyond; Cluster Networking and Cloud Computing

**Suman Kumar** is an assistant professor in the Computer Science Department at Troy University. He leads the Trojan Advaned Computer Knowledge and Networking group (TrACKNet) group. He recieved his Ph.D. from Louisiana State University and a degree in Electronics and Communication Engineering from the Indian Institute of Technology (IIT), BHU, India. His current research interests include all aspect of computer networks, high performance computing, and data mining.

**Seung-Jong Park** is an associate professor of Computer Science Department at Louisiana State University. He received Ph.D. from the school of Electrical and Computer Engineering at Georgia Institute of Technology, 2004. He also received a B.S. degree from Computer Science at Korea University, Seoul, Korea and a M.S. degree in Computer Science from KAIST (Korea Advanced Institute of Science and Technology), Teajon, Korea in 1993 and 1995, respectively. His research focuses on issues involving Networking and Data Intensive Computing.