Robustness to Packet Reordering in High-speed Networks

Sumitha Bhandarkar and A. L. Narasimha Reddy
Dept. of Electrical and Computer Engineering
Texas A & M University
{sumitha,reddy}@ee.tamu.edu

***Abstract*— In this paper we investigate the impact of packet reordering on the performance of high-speed protocols. Our results show that even small fraction of packet reordering can severely impair the performance of these protocols. We then investigate the benefits of using delayed congestion response (TCP-DCR) with the high-speed protocols. Our results indicate that the benefits in terms of avoiding performance degradation is significant, even at very high levels of packet reordering. In the absence of any packet reordering, the protocol behavior in terms of fairness among competing flows or impact on bottleneck link drop rates remains unmodified.**

I. Introduction

In the past few years several high-speed TCP protocols [2]-[6] have been proposed for improving the performance of TCP in high BDP networks. Due to the aggressive nature of bandwidth probing, these protocols allow the congestion window to grow to a very large value quickly, in order to make efficient use of available bandwidth. For instance, for a single TCP flow sending 1500 byte packets on a 1Gbps link with 120ms RTT, the window size can grow as large as 10,000. However, all the new proposals still rely on the *3 dupack* heuristic for determining a packet loss. As a result, even a small amount of packet reordering can cause severe degradation of performance, making these protocols behave no better than the standard TCP variants.

Over the years several measurement studies [9]-[15] have been conducted for determining the amount of packet reordering prevalent in the Internet. These studies have reported seemingly contradictory results. Some claim that packet reordering is caused by pathological behavior of mis-configured network components, while others declare that it is not pathological and is caused mainly by parallelism in network components. Whatever the case may be, since significant amount of research is being carried out for improving the performance of TCP in high-speed networks, it seems prudent to make these protocols more robust to packet reordering.

In addition to modifications to the TCP congestion control algorithms, several other mechanisms such as parallel routers, multi-path routing and multi-homing are being investigated for improving the performance in high-speed networks. It is possible that these may lead to higher amount of packet reordering in the Internet. Designers of these Internet components go to great lengths to avoid packet re-sequencing [16]. As pointed out in [18], improving the performance of TCP to packet reordering imposes less restriction on designers of these Internet components, allowing for possibly more efficient designs.

Several studies [18]-[22] have proposed mechanisms for improving the performance of TCP in the face of packet reordering. In this paper we focus on one of the schemes called *TCP-DCR* [1], [22]. Using ns-2 simulations we show that packet reordering can result in severe degradation of performance of the highspeed protocols. However, when modifications suggested in [22] are applied to the highspeed protocols, even at high levels of packet reordering, the performance benefits are not compromised. In networks which do not contain any packet reordering, the use of modifications in [22] has minimal impact and does not impact the fairness properties of the highspeed protocols and has minimal impact on bottleneck link drop-rates.

The rest of the paper is organized as follows. In Section II we overview the literature to understand the extent of packet reordering in the Internet. In Section III we provide a brief overview of the different highspeed protocols. This is followed by Section IV where we show the impact of packet reordering on highspeed protocols and the benefits that can be obtained by using delayed congestion response with these protocols. Finally, we conclude the paper in Section V.

II. Packet Reordering on the Internet

Over the past years, several different measurement studies were conducted to determine the level of packet reordering in the Internet. The measurements were conducted at different network locations using different methodologies during different time periods. These studies have presented observations that are seemingly contradictory to each other.

In [9], the author pioneered the first large-scale measurement study of Internet packets by conducting 20,000 bulk TCP transfers of 100 Kbytes each between 35 Internet sites. In two sets of measurements conducted during December 1994 and November-December 1995, the author found 2% and 0.3% reordering of data packets (0.6% and 0.1% of ACKs) respectively. At least one packet was delivered out of sequence for 36% of the packets in the first measurement and 12% in the second measurement. Other main observations were that reordering was asymmetrical, some paths were sometimes subject to high levels of reordering and the effects were strongly site specific. The two main causes identified for causing the problems were route fluttering and router updates and hence they claimed the reordering behavior was mainly *pathological* or not very usual.

More recent studies have also claimed that packet reordering is not a commonly occurring phenomenon on the Internet - however, their results indicate the presence of a non-zero amount of packet reordering in the Internet. An extensive study of packet dynamics for low-bitrate MPEG-4 video streams over paths with more than 5000 routers conducted in November 1999 to May 2000 is presented in [11]. The results of this study indicated that packet reordering while rare, does occur. The study presented in [12] looked at 19 million TCP connections on the Sprint backbone and was conducted during February 2002 and October 2002. The results indicated that the packet reordering was observed in 0.03 to 0.72% of all the data packets (0.15 to 4.9% of all the connections). Finally, measurements made in China [13] during May-June 2003 by tracing 208 connections with 3.3 million data packets using a web-crawler on 10,647 web sites indicated that 5.79% of the sites, and 3.2% of the packets exhibited packet reordering at least once. Of the sites that exhibited reordering 20% of the sites had a reordering frequency of more than 80%, indicating strong site dependency.

These results were directly contradicted in [10] where the authors claim that packet reordering is not pathological behavior on the Internet and is prevalent at significantly high levels. Their study consisted of measurements conducted on 140 Internet hosts connected to the MAE-EAST exchange during December 1997 and January 1998. The methodology used was significantly different from that in [9], since the authors chose to send back-to-back bursts of 50 ICMP-ping packets of 56-byte for conducting the first measurement and a 100-packet bursts of 512-byte packets for the second measurement. From the first measurement, they observed that the probability of a session experiencing packet reordering was 90%. From the second measurement they inferred that reordering was a function of network load. Further study indicated that the main cause for reordering was parallelism in Internet components and links due to link-level striping and the multiple paths that a packet can take within the switching devices.

Results from the October 2003 study presented in [14] indicate similar results - that packet reordering was observed in about 56% of all the streams and the leading cause was pointed to be parallelism in the Internet components. Two sets of measurements were conducted, first by sending back-to-back bursts of 50 100-byte UDP packets and second by sending bursts of 100 UDP packets. More reordering was found in the second measurement compared to the first measurement. Another study presented in [15] supported with observations of high levels of packet reordering. The study was conducted using UDP flows in high-speed networks and the authors point to a high correlation between packet rate and observed reordering. They conclude that for high bandwidth applications protocols should be as resilient to packet reordering as they are to packet loss.

Based on these studies, there seem to be two categories of observations. One set claims that packet reordering is pathological and an artifact of some mis-configuration/ misbehavior of network components. Some of these studies observe that packet reordering may be highly local with few sites/links exhibiting high levels of reordering. The other category of results show that packet reordering is not pathological but is widely prevalent and has a possibility of getting worse, since the cause for the reordering is parallelism in Internet components which will only increase as network speed/capacity increases. But both sets of studies point to non-zero amounts of packet reordering in the Internet and we show through experiments that even a small percentage of packet reordering can be harmful to high-speed protocols.

Another observation is that the measurement studies that indicate high levels of reordering ([10], [14] and [15]) used bursts of ICMP or UDP packets for probing while the other measurements were mainly TCP based or used low-bitrate traffic. This indicates that when packets arrive in bursts at a parallel router or switch, it may be characterized by higher packet reordering. This conjecture has been made in [12] as well. Additionally, in [10] authors show that packet reordering is dependent on the network load. This is collaborated in [15] where the authors show that reordering increases as the packet rates increase or conversely, the inter-packet arrival time in the core of the network reduces. This seems to indicate that packet reordering cannot be dismissed easily since the network load on the Internet keeps steadily increasing. The site-dependency of observing higher levels of reordering has also been linked to heavy loads. The aggressive nature of the high-speed protocols changes the behavior of TCP and the packets sent by a flow no longer uses the conservative additive increase policy. As a result, it may result in higher burstiness at the routers, and we conjecture that this could consequently lead to packet reordering. Thus, it is important that protocols aimed at high capacity networks be resilient to packet reordering.

Several different solutions have been proposed in literature to make TCP robust to packet reordering. In [18] and [19] the authors present schemes for improving the reordering robustness of TCP that use DSACKs [20] or timestamps [21] to identify the possible amount of reordering. Once the amount of reordering is estimated, the threshold *delthresh* for responding as if the packet is lost, is modified accordingly. However, the need for identifying the exact amount of reordering in the network requires these schemes to use complex state and algorithms, which may not be desirable in highspeed networks. TCP-DCR [1], [22] on the other hand, aims to improves the reordering robustness of TCP by uniformly delaying the congestion response by one RTT. The simple change in *delthresh* is easy to implement and it has been shown in [22] that it is effective in improving the robustness of TCP-SACK to packet reordering without significant impact on the behavior in case of congestion. In this paper we focus on the solution presented in TCP-DCR, and verify that it avoids performance degradation in the presence of packet reordering, when used in conjunction with highspeed protocols as well.

III. Highspeed Protocols

Several studies [2]-[7] have proposed the modification of congestion control algorithms of TCP for improving performance in highspeed networks. LTCP [2] applies the concept of layering to the congestion control algorithm of TCP to increase its aggressiveness of probing for bandwidth. High-speed TCP [3] uses a congestion window response function that has a higher slope than TCP. Scalable TCP [4], uses multiplicative increase/multiplicative decrease response, to ensure that the congestion window can be doubled in a fixed number of RTTs. BIC-TCP [5] modifies the congestion response function to use binary search with additive increase and multiplicative decrease. HTCP [6] uses response function similar to High-speed TCP but modifies the increase parameter based on time since last drop. FAST TCP [7] relies on the delay-based bandwidth estimation of TCP Vegas [8] and is optimized for Gbps links. While all these solutions provide improved resiliency and robustness to packet losses such that the congestion window can grow to large values, none of them consider improved resiliency to packet reordering. In this paper we focus on the four schemes LTCP, High-speed TCP, BIC-TCP and HTCP.

IV. Experimental Evaluation

The objective of this paper is to show the impact of packet reordering on high speed protocols, and to show that the use of delayed congestion response can help avoid the performance degradation. Experimental evaluation is conducted using simulations on the ns-2 simulator.

We first illustrate the impact of packet reordering on high-speed protocols. The topology consists of a bottleneck link of capacity 1Gbps and delay 23ms between two routers R1 and R2. A source connected to the router R1 sends data to a receiver connected to the router R2. The access link between the routers and the end-nodes has a capacity of 1Gbps and a delay of 1ms. The buffers size at the bottleneck link is set to the delayXbandwidth product.

Packet reordering is simulated by randomly choosing packets based on a uniform distribution and delaying them. The packet delay is chosen from a normal distribution with mean 25ms and 8ms. Since the RTT of the flow is 50ms, the packet reordering in most cases is less than one RTT, but there in a non-zero probability that packets may be delayed by more than one RTT. Fig.1 shows the throughput of the flow for the different protocols, as the fraction of packets delayed is varied from 1E-6 to 1E-1 resulting in potential reordering of 0.0001% to 10% of the packets.

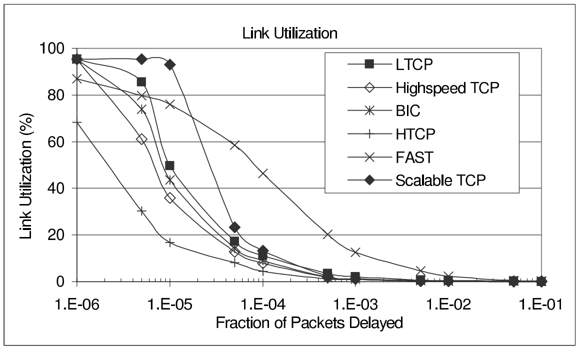


Fig. 1. Impact of Packet Reordering on High-speed Protocols

From the figure we see that most protocols have high link utilization when the fraction of packets delayed is small such as 1E-6. HTCP being the only exception, shows slightly degraded performance even at such low rate of packet reordering. As the fraction of packets delayed is increased, the degradation in throughput is drastic for almost all the protocols. For a packet delay rate of 1E-4 (i.e., 0.01% of the packets are delayed), the link utilization by all the protocols with the exception of FAST is well below 20% of the link capacity. Even though FAST shows less drastic degradation of its throughput, it nevertheless falls below 20% link utilization by the time the fraction of packets delayed is increased to 0.05%.

It has been conjectured in [14] and [12] that packet reordering in the network may be caused by the parallelism in the Internet components. As the bandwidth of links continues to increase, we could expect the parallelism to increase. Designing high-speed switches to avoid packet reordering is an area of research in itself. There has been work proposed in the literature explicitly to ensure that packet ordering is maintained in such switches - e.g., [16]. In [16], the authors point out that while packet reordering is not strictly prohibited in an Internet router [17], it is required to avoid throughput degradation of TCP flows. Hence, the performance of TCP protocol imposes requirements of almost in-order delivery on the design of network components.

Next we study the performance of highspeed protocols when delayed congestion response (DCR) is used. Results are shown for LTCP, Highspeed TCP, BIC-TCP and HTCP. Simulations are conducted for - (a) reordering only (percentage of packets delayed varied from 1E-6 to 0.1) (b) congestion only (number of flows varied from 2 to 1000) and (c) both congestion and packet reordering (similar to (a) but with 50 competing flows). Unless otherwise mentioned, the bottleneck link has capacity 1Gbps and delay of 23ms and the access links have a capacity of 1Gbps and delay 1ms. The buffers at the bottleneck link are set to the delayXbandwidth product. The simulations are run for a period of 400 seconds and only the data collected between 100 and 300 seconds is presented to ensure that we capture the steady state behavior. These experiments on a single bottleneck link topology are followed by a more complex topology with several bottleneck links, flows in both directions and the traffic consisting of a mix of long term flows and short web-like traffic.

A. Packet Reordering Only

Fig.2 shows the throughput of the highspeed protocols with the DCR modifications as the fraction of packets delayed is increased from 1E-6 to 0.1. This experiment is similar to the previous experiment and the results can be compared directly with those in Fig.1. As seen from the figure, the throughput remains high even at high levels of packet reordering is most cases. Note that the delay in congestion response is set to one RTT as suggested in [1] to avoid the triggering of spurious timeouts. However, the model used here for generating delay in packets could result in packet reordering of more than one RTT. Since these events will result in window reduction, depending on the when and how repeatedly these events occur, the throughput may be slightly lower than the optimal as seen in some cases.

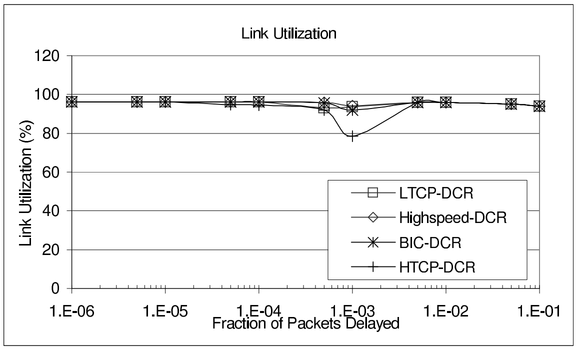


Fig. 2. Link Utilization with Packet Reordering

B. Congestion Only

Next we study the impact of delaying congestion response when there is no packet reordering in the network. The topology is similar to the previous experiments. However, the number of flows is now increased from 2 to 1000, creating different levels of congestion. Fig 3 shows the link utilization as the number of flows is varied. As seen from the results, the link utilization remains similar irrespective of whether delayed congestion response is used or not.

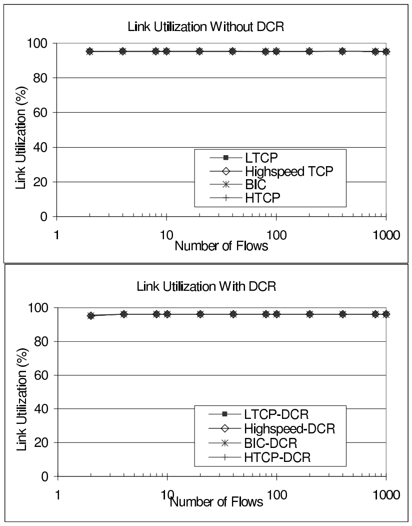


Fig. 3. Link Utilization with Only Congestion

While it is important that the utilization of the bottleneck link is high, it should be done in such a way that the different flows sharing the bottleneck link get a fair share. In order to verify this we examine the Jain Fairness Index[23] of the flows with the different protocols for the above experiment. Fig.4 shows the results. From the figure, we see that the Jain Fairness Index remains high even when delayed congestion response is used, indicating that the bandwidth is shared in a relatively fair manner.

Finally, we examine the bottleneck link drop-rate when delayed congestion response is used. Fig.5 shows the results. Note that, high-speed TCP protocols are in general characterized by high levels of packet losses due to their aggressive probing techniques. Results show that using delayed congestion response does not make the average number of bottleneck link buffer overflows any worse.

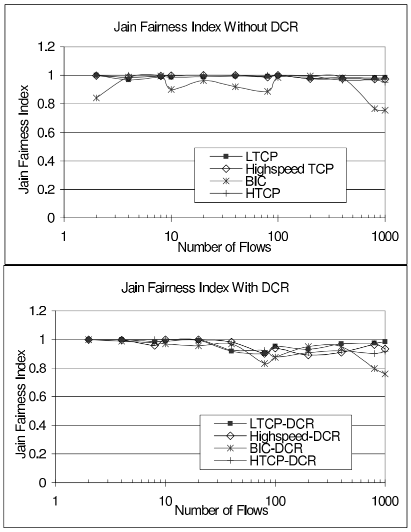


Fig. 4. Jain Fairness Index with Only Congestion

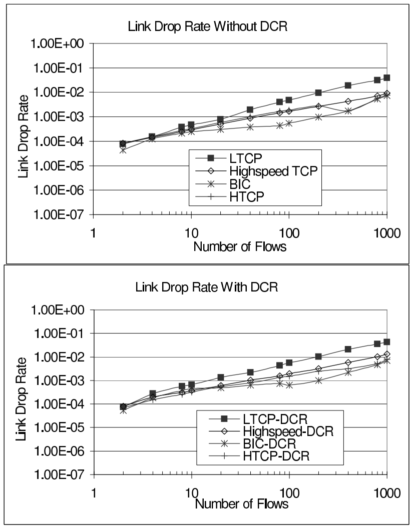


Fig. 5. Bottleneck Link Drop-rate with Only Congestion

C. Both Congestion and Packet Reordering

Next we examine the behavior when the network consists of both packet reordering and packet losses due to congestion. This experiment is similar to the experiment with only packet reordering, except that the bottleneck link is now shared by 50 flows. Fig 6 shows the link utilization of the different highspeed protocols with and without DCR. Since 50 flows share the bottleneck link, the aggregate link utilization of the different high-speed protocols without DCR is improved. However, at higher levels of packet reordering, the degradation, is still very drastic. When DCR is used however, the link utilization is maintained high even for high levels of packet reordering.

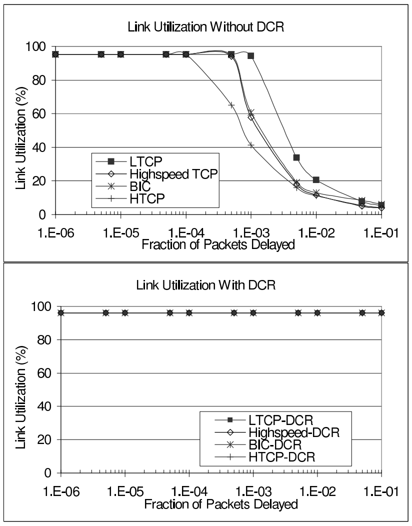


Fig. 6. Link Utilization with Both Congestion and Packet Reordering

We next observe the Jain Fairness Index. Fig. 7 shows the results. Similar to the case with only congestion, the Jain Fairness Index remains high even when DCR is used.

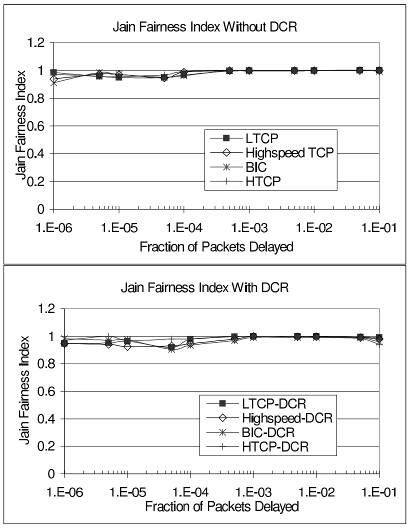


Fig. 7. Jain Fairness Index with Both Congestion and Packet Reordering

Next, we examine the bottleneck link drop-rate. Fig.8 shows the results. When DCR is not used, the link is not fully utilized. In such cases, the link drop rate is negligible. However, when DCR is used, the link is fully utilized and hence the link droprate remains constant across the different simulations.

D. Simulation with More Complex Topologies

In this experiment we verify that the benefits of using DCR are available in complex topologies as well. The network in this simulation consists of four bottleneck links between five routers. Each router is connected to a cloud of nodes. Traffic goes from one cloud to the other in the directions as shown in Fig 9. The traffic from each cloud consists of a mix of 20 long-term flows and 100 web-sessions. The router R3 simulates a mis-configured router and results in reordering 1% of the packets passing through it. The delay used for reordering the packets uses the same model as before.

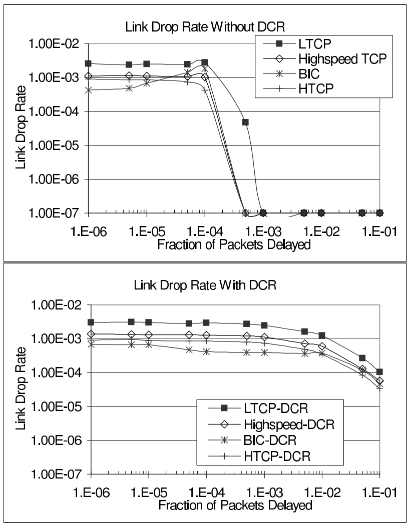


Fig. 8. Bottleneck Link Drop-rate with Both Congestion and Packet Reordering

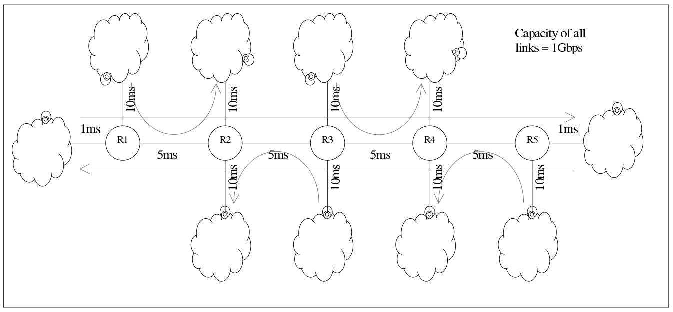
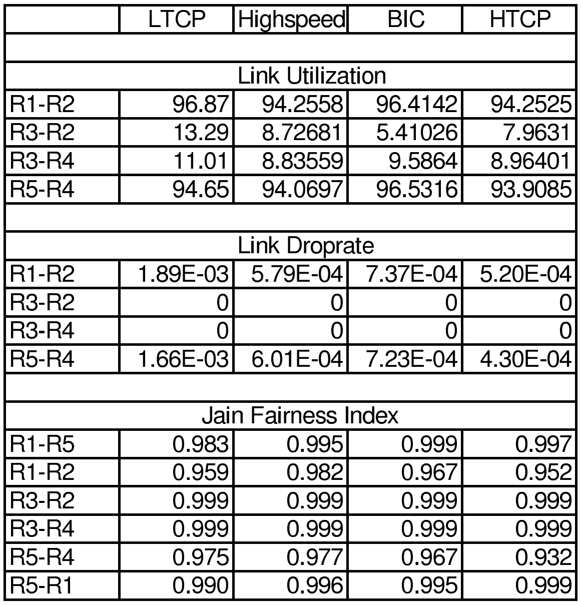


Fig. 9. Topology with Multiple Bottleneck Links, Forward as well as Reverse Traffic, and Long-term as well as Short Web-like Flows

Fig.10 shows the link utilization and drop-rates on each of the bottleneck links as well as the Jain Fairness Index of the long-term flows between each pair of source/destination, when DCR is not used. From the table we see that the utilization sees drastic degradation on the links associated with the 'mis-configured' router R3. The drop-rate on the links R3-R2 and R3-R4 that suffer underutilization, is negligible, while the drop-rate on links R1-R2 and R5-R4, remain what is characteristic for each protocol. The Jain Fairness Index is high for all protocols, indicating that for the topology considered, all protocols maintain similar throughput amongst competing flows irrespective of whether the link is fully utilized or not.

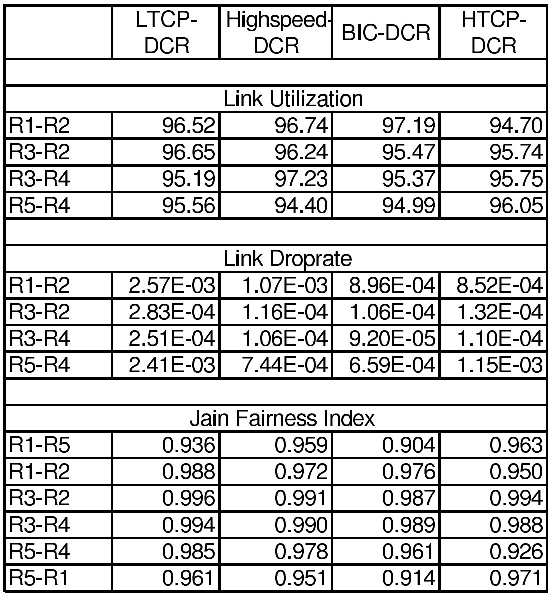
Next we repeat the experiment, with all the highspeed protocols using the DCR modifications. Fig.11 shows the results. From the table, we see that the link utilization for all the protocols remain high even on links associated with the "mis-configured" router. Since the links is no longer under-utilized the bottleneck link drop-rates are non-negligible, but the value remains close to the characteristic values for the protocol as see in the table in 10. Finally, we note that the Jain



	LTCP	Highspeed	BIC	HTCP
Link Utilization				
R1-R2	96.87	94.2558	96.4142	94.2525
R3-R2	13.29	8.72681	5.41026	7.9631
R3-R4	11.01	8.83559	9.5864	8.96401
R5-R4	94.65	94.0697	96.5316	93.9085
Link Droprate				
R1-R2	1.89E-03	5.79E-04	7.37E-04	5.20E-04
R3-R2	0	0	0	0
R3-R4	0	0	0	0
R5-R4	1.66E-03	6.01E-04	7.23E-04	4.30E-04
Jain Fairness Index				
R1-R5	0.983	0.995	0.999	0.997
R1-R2	0.959	0.982	0.967	0.952
R3-R2	0.999	0.999	0.999	0.999
R3-R4	0.999	0.999	0.999	0.999
R5-R4	0.975	0.977	0.967	0.932
R5-R1	0.990	0.996	0.995	0.999

Fig. 10. Link Utilization, Link Drop-rate and Jain Fairness Index for Flows on a Multiple Bottleneck Link Router with a Mis-configured Router that Causes Packet Reordering, when DCR is not used.

Fairness Index remains high, indicating that the use of DCR does not compromise the fairness behavior among competing flows.



	LTCP-DCR	Highspeed-DCR	BIC-DCR	HTCP-DCR
Link Utilization				
R1-R2	96.52	96.74	97.19	94.70
R3-R2	96.65	96.24	95.47	95.74
R3-R4	95.19	97.23	95.37	95.75
R5-R4	95.56	94.40	94.99	96.05
Link Droprate				
R1-R2	2.57E-03	1.07E-03	8.96E-04	8.52E-04
R3-R2	2.83E-04	1.16E-04	1.06E-04	1.32E-04
R3-R4	2.51E-04	1.06E-04	9.20E-05	1.10E-04
R5-R4	2.41E-03	7.44E-04	6.59E-04	1.15E-03
Jain Fairness Index				
R1-R5	0.936	0.959	0.904	0.963
R1-R2	0.988	0.972	0.976	0.950
R3-R2	0.996	0.991	0.987	0.994
R3-R4	0.994	0.990	0.989	0.988
R5-R4	0.985	0.978	0.961	0.926
R5-R1	0.961	0.951	0.914	0.971

Fig. 11. Link Utilization, Link Drop-rate and Jain Fairness Index for Flows on a Multiple Bottleneck Link Router with a Mis-configured Router that Causes Packet Reordering, with the use of DCR.

V. Conclusions

In this paper we show that even a small percentage of packet reordering can degrade the performance of several recently proposed high speed protocols significantly. Through extensive simulations on the ns-2 simulator, we show that using DCR with these high speed protocols can help retain the performance benefits in the presence of packet reordering. Protocol characteristics such as fairness and bottleneck link behavior are not impacted by the addition of DCR even when packet losses are due to congestion. Use of DCR with high-speed protocols helps protect the performance in the presence of packet reordering as well as helps ease the requirements of strict in-order delivery, imposed on the design of future high-speed network components.

References

[1] Sumitha Bhandarkar, A. L. Narasimha Reddy, Mark Allman and Ethan Blanton, "Improving the robustness of TCP to Non-Congestion Events", *RFC 4653*, August 2006.

[2] Sumitha Bhandarkar, Saurabh Jain and A. L. Narasimha Reddy, "LTCP: Improving the Performance of TCP in Highspeed Networks", *ACM SIGCOMM Computer Communication Review*, Volume 36 , Issue 1 , January 2006.

[3] Sally Floyd, "HighSpeed TCP for Large Congestion Windows", *RFC 3649* December 2003.

[4] Tom Kelly, "Scalable TCP: Improving Performance in HighSpeed Wide Area Networks", *ACM Computer Communications Review*, April 2003.

[5] Lisong Xu, Khaled Harfoush, and Injong Rhee, "Binary Increase Congestion Control for Fast Long-Distance Networks", To appear in *Proceedings of IEEE Infocom 2004*, March 2004.

[6] D.J. Leith and R. Shorten, "H-TCP Protocol for High-Speed Long Distance Networks", *PFLDNet 2004*, February 2004

[7] Cheng Jin, David X. Wei and Steven H. Low, "FAST TCP: motivation, architecture, algorithms, performance", *IEEE Infocom*, March 2004.

[8] L. Brakmo, S. O'Malley, and L. Peterson, "TCP Vegas: New techniques for congestion detection and avoidance", *Proceedings of ACM SIGCOMM '94*, August 1994.

[9] V. Paxson, "End-to-end Internet packet dynamics", *IEEE/ACM Transactions on Networking*, 7(3):277–292, 1999.

[10] Jon C. R. Bennett, Craig Partridge, and Nicholas Shectman, "Packet reordering is not pathological network behavior", *IEEE/ACM Transactions on Networking*, 7(6):789–798, 1999.

[11] D. Loguinov and H. Radha, "End-to-End Internet Video Traffic Dynamics: Statistical Study and Analysis", *IEEE INFOCOM*, June 2002.

[12] S. Jaiswal, G. Iannaccone, C. Diot, J. Kurose and D. Towsley, "Measurement and Classification of Out-of-sequence Packets in Tier-1 IP Backbone", *IEEE INFOCOM*, Mar. 2003.

[13] Yi Wang, Guohan Lu, Xing Li, "A Study of Internet Packet Reordering", *ICOIN 2004*, February 2004.

[14] Xiaoming Zhou, Piet Van Mieghem, "Reordering of IP Packets in Internet", *PAM 2004*, April 2004.

[15] Ladan Gharai, Colin Perkins, Tom Lehman, "Packet Reordering, High Speed Networks and Transport Protocol Performance", emphICCCN 2004, October 2004.

[16] I. Keslassy and N. McKeown, "Maintaining packet order in twostage switches," *Proceedings of the IEEE Infocom*, June 2002.

[17] F. Baker, "Requirements for IP version 4 routers," *RFC 1812*, June 1995.

[18] M. Zhang, B. Karp, S. Floyd, and L. Peterson, "RR-TCP: A Reordering-Robust TCP with DSACK," *ICSI Technical Report TR-02-006*, Berkeley, CA, July 2002.

[19] E. Blanton and M. Allman, "On Making TCP More Robust to Packet Reordering," *ACM Computer Communication Review*, January 2002.

[20] Sally Floyd, Jamshid Mahdavi, Matt Mathis and Matt Podolsky, "An Extension to the Selective Acknowledgement (SACK) Option for TCP," *RFC 2883*, July 2000.

[21] R. Ludwig and M. Meyer, "The Eifel Detection Algorithm for TCP," *RFC 3522*, April 2003.

[22] Sumitha Bhandarkar and A. L. Narasimha Reddy, "TCP-DCR: Making TCP Robust to Non-Congestion Events", *Proceedings of Networking 2004 conference*, May 9-14 2004.

[23] Dah-Ming Chiu and Raj Jain, "Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks", *Computer Networks and ISDN Systems*, June 1989.