Reordering of IP Packets in Internet

Xiaoming Zhou and Piet Van Mieghem

Delft University of Technology
Faculty of Electrical Engineering, Mathematics, and Computer Science
P.O. Box 5031, 2600 GA Delft, The Netherlands
{X.Zhou,P.VanMieghem}@ewi.tudelft.nl

Abstract. We have analyzed the measurements of the end-to-end packet reordering by tracing UDP packets between 12 testboxes of RIPE NCC. We showed that reordering quite often happens in Internet. For bursts of 50 100-byte UDP packets, there were about 56% of all the streams arrived at the destinations out-of-order. We studied the extent of the reordering in these streams, and observed that most reordered streams have a relative small number of reordered packets: about 14% of all the streams have more than 2 reordered packets in a bursts of 50 UDP packets. In addition, we showed that packet reordering has a significant impact on UDP performance since reordering adds a high cost of recovering from the reordering on the end host. On the other hand, packet reordering does not have a significant impact on the UDP delay. We also compared the reordered stream ratios in the different directions of Internet paths, and observed that reordered stream ratios are asymmetric, but they vary largely from testbox-to-testbox.

1 Introduction

Reordering, the out-of-order arrival of packets at the destination, is a common phenomenon in the Internet [3][1], and it frequently occurs on the latest, high-speed links. The major cause of reordering has been found to be the parallelism in Internet components (switches) and links [1]. For example, due to load balancing in a router, the packets of a same stream may traverse different routers, where each packet experiences a different propagation delay, and thus may arrive at the destination out-of-order. Reordering depends on the network load, although below a certain load very little reordering occurs. Reordering may also be caused by the configuration of the hardware (i.e., multiple switches in a router) and software (i.e., class-based scheduling or priority queueing) in the routers.

The interest in analyzing end-to-end reordering is twofold. First, reordering greatly impacts the performance of applications in the Internet. In a TCP connection, the reordering of three or more packet positions within a flow may cause fast retransmission and fast recovery multiple times resulting in a reduced TCP window and consequently in a drop in link utilization and, hence, in less throughput for the application [5]. For delay-based real-time service in UDP (such as VoIP or video conference), the ability to restore order at the destination will

likely have finite limits. The deployment of a real-time service necessitates certain reordering constraints to be met. For example, in case of VoIP, to maintain the high quality of voice, packets need to be received in order, and also within 150 millisecond (ms). To verify whether these QoS requirements can be satisfied, knowledge about the reordering behavior in the Internet seems desirable. Second, these end-to-end reordering measurements may shed light on the underlying properties of the current topology and traffic patterns of the Internet.

In this paper, packet reordering is measured between 12 Internet testboxes of RIPE TTM (Test Traffic Measurement) [4] project. Our observations are based on packet level traces collected through the network. The main aim lies in the understanding of the nature of reordering. The remainder of the paper is structured as follows. In Section 2, we review the related work relevant to this paper. In Section 3, we describe the methodology used to observe reordering behavior. Our experiments are explained in Section 4. Finally, we conclude in Section 5.

2 Related Work

Quantifying the extent of reordering has been initiated by Paxson [3], and has been further investigated by Bennett *et al.* [1], and Jaiswal *et al.* [6]. During Dec. 1994 and Nov-Dec. 1995, Paxson measured the reordering ratio by tracing 20,000 TCP bulk transfers between 35 computers on the Internet. His results showed that the fraction of sessions with at least one reordering (data or acknowledgements) was 12% and 36% across the two measurement periods. About 2% of all of the data packets and 0.6% of the acknowledgements arrived out of order in the first measurement (0.3% and 0.1% in the second measurement). In 1998, Bennett *et al.* did their experiments by measuring over active TCP probe flows through the MAE-East Internet exchange point. They reported reordering in two different ways: for bursts of five 56-byte ICMP-ping packets, over 90% has at least one reordering event. After isolating a host with significant reordering, they reported that 100 packet bursts of 512-byte each produce similar results. In 2002, Jaiswal *et al.* measured and classified out-of-sequence packets in TCP connections at a single point in the Sprint IP backbone. Their results showed that the magnitude of reordering (1.57%) is much smaller than those in [1][3].

Our work distinguishes from the above in terms of experimental setup since we used and analyzed real network traces based on UDP streams.

3 Problem Description and Definitions

A packet is classified as a reordered or out-of-order packet if it has a sequence number smaller than its predecessors. Specifically, let M streams, denoted as $(S_1, \ldots, S_M)$, be the total number of streams sent from node A to B. In each stream S_i consisting of K packets, we assign to each packet j a sequence number a_j, which is a consecutive integer from 1 to K in the order of the packet emission

and so we establish the source sequence as $(a_1, \ldots, a_K)$. Assume an output sequence $(b_1, \ldots, b_P)$ of stream S_i observed at the receiving node B, where $P \leq K$ be the total number of packets received out of the K packets sent. The amount $K - P$ is due to loss. The sequence is said to be in order if for each index k $(1 \leq k \leq P)$ holds $b_k < b_q$ $(0<q<k)$, else the stream is said to arrive at the destination out-of-order, and the packet k is a reordered packet in the reordered stream. The total number of reordered packets in stream S_i is written as L_i. For example, for the sequence of an arrived reordered stream (1,2,3,5,4,7,6,8), there are 2 reordered packets (packet 4 and packet 6), which leads to $L = 2$. Note that in our paper reordering does not correlate with loss (same as [2][8][9]). For example, a received stream (1,2,3,4,5,6,8) is considered as in order.

We denote the reordered stream ratio by

$$R_{AB} = \frac{M_R}{M_a} \tag{1}$$

where M_a is the total number of received streams out of M streams sent and M_R is the total number of streams having at least one reordered packet. The reordering asymmetry is defined to be the difference of two ratios $|R_{AB}\text{-}R_{BA}|$.

Let $U_n = \Pr[L_n > 0]$ denote the unconditional reordered stream probability for the received stream n. And let C_n denote the probability that a stream $n+1$ is reordered given that the previous stream n was reordered, defined by

$$C_n = \Pr[L_{n+1} > 0 | L_n > 0] \tag{2}$$

In order to predict whether a reordered packet will be useful in a receiver buffer with finite limit, for each reordered packet k $(1 \leq k \leq P)$, this paper studies two more metrics: packet lag P_L and time lag T_L. Packet lag is proposed in [6] and refers to the number of packets k $(1 \leq k \leq P)$, with a sequence number greater than the reordered packet that have been received before the reordered packet itself. Thus,

$$P_L = \sum_{q=1}^{k-1} 1_{b_k < b_q} \tag{3}$$

where the indicator function 1_y defined as 1 if the condition y is true and otherwise it is zero. For example, consider two packet sequences (2,1,3,4,5,6,7,8) and (2,3,4,5,6,7,8,1) which both consist of one reordered packet (packet 1), due to the different arrival positions of packet 1 in the two received sequences, then $P_L = 1$ for the previous sequence, while $P_L = 7$ for the latter sequence. For a receiver with a finite buffer or a time constraint, recovering the latter sequence from reordering may be impossible.

Let t_k $(1 \leq k \leq P)$ be the 1-way delay of packet k. T_L is defined as the difference between the delay t_k of the reordered packet k and its expected delay $t_{k'}$ without reordering,

$$T_L = |t_k - t_{k'}| \tag{4}$$

In practice, $t_{k'}$ is replaced by $\min(t_1, .., t_P)$.

P_L is useful to evaluate the impact of reordering on TCP's performance since $P_L \geq 3$ would trigger the fast retransmit algorithms that halve the TCP sender's congestion window. We believe P_L is also a useful metric to study the impact of reordering events on UDP's performance. In addition to the P_L, T_L is a delay-based metric to more precisely evaluate the impact of reordered packets on the end hosts. For delay sensitive applications based on UDP, reordering can have a drastic effect on the application's performance. For example, in case of VoIP, to maintain the high quality of voice, packets need to be received in order, and also before playback time. If a reordered packet arrives after its playback time has elapsed, that packet may be treated as lost.

4 Experiment Results

In the following we analyze the end-to-end packets reordering measurements performed in 12 "testboxes" of RIPE TTM project. The TTM infrastructure consists of approximately 60 measurement testboxes scattered over Europe (and a few in the US and Asia). Due to the synchronizations with GPS in all testboxes, RIPE TTM achieves a delay accuracy within 10 μs. We have analyzed the data collected between 12 test-boxes; where 3 hosts are located in the Netherlands, 2 in Great Britain, and 1 in Sweden, Slovakia, Belgium, Australia, USA, Denmark and Greece. 12 testboxes participated in two experiments. Firstly, between each sender-destination pair of measurement boxes, IP probe-streams of a back-to-back burst of 50 100-byte UDP packets, called probe-streams, are continuously transmitted with interarrival times of about 30 seconds, resulting in a total of about 360 probe-streams in 3 hours from 5 to 8 PM (Greenwich Mean Time) on October 16, 2003. Secondly, we repeated the same experiment with a burst of 100 UDP packets in 3 hours from 5 to 8 PM on October 17, 2003. In order to get the snapshot of traffic patterns in the Internet, we limited each measurement to a total of 3 hours. We denoted the first experiment by N_{50} and the second by N_{100}. The difference between N_{50} and N_{100} may indicate how Internet packet dynamics change under two difference load situations. In a complete graph, ideally, 12 test-boxes should consist of exactly 132 unidirectional links. In practice, the experiment generally consisted of 104 unidirectional paths due to the erroneous effects during the measurement.

To limit the influence of large packet loss, we only analyzed those streams which received at least 90% of all their total packets (i.e. a valid arrival stream has at least 45 UDP packets in N_{50} and 90 in N_{100}).

4.1 Reordered Probe-Stream Ratio R_{AB}

R_{AB} can give insight how often reordering happened in the probe-streams. For each sender-destination pair we examined each arrival stream by checking its arrival sequence order. We calculated how many reordered probe-streams have been received over 3 hours (there are approximately 360 probe-streams). Table 1

summarizes the total number of observed UDP streams and the packets in these streams on 104 paths over 3 hours. The results of our experiment (Table 1) indicate that reordering quite often occurs in the probe-streams. In N_{50}, about 56% of the probe-streams included at least one packet delivered out-of-order, while 66% did in N_{100}. Overall, 6% of all the received packets in N_{50} arrived reordered while 5.6% in N_{100}. It is interesting to note that this large fraction of reordering in streams is also reported by Bennett et al. [1] (over 90% with at least one reordering event). On the other hand, our results of the number of probe-stream that experience reordering is lower compared to the number in [1]. This discrepancy may be caused by methodological differences between the studies: we used UDP probe-stream in one-way, while the authors in [1] used TCP round-trip measurements.

Received data	N_{50}	N_{100}
UDP streams	36762	32691
Reorder streams	20445	21649
UDP packets	1655120	2828834
Reordered UDP packets	101018	158413
Measurement duration	3 hours	3 hours

Table1. Details of the packets used to measure the reordering on 104 paths

We observed that a large fraction (>26%) of all UDP probe-stream suffered from reordering on all the experiment paths. In general, the probe-streams in N_{100} are more often reordered than those in N_{50}. For example, the average (over the 104 paths) is $E[R_{AB}] = 0.53$, where the standard deviation is $\sigma_{50} = 0.12$ in N_{50}. While $E[R_{AB}] = 0.65$ and $\sigma_{100} = 0.12$ in N_{100}. This is because higher traffic load likely contributes more to reordering.

We also observed that reordering varies greatly from testbox-to-testbox, for instance more than 70% of the streams transmitted from some testboxes in West Europe to two testboxes (a site in Australia, and another in Nottingham, Great Britain) in N_{50} arrived out-of-order; much higher than the 56% overall average, while 80% in N_{100}. This is may be caused by the heavy load at the links to these two testboxes.

4.2 Reordered Packet Lengths *L*

In order to quantify the extent of reordering, for each source-destination pair, we examined each arrival stream by checking its arrival sequence order and by calculating the reordered packet length L (the number of reordered packets).

Figure 1(a) plots a probability density function (pdf) of how many reordered packets are observed in N_{50} and in N_{100}. We found that the pdf of the reordered length has a relative heavy tail. Specifically, $L = 0$ for 44% of total probe-streams in N_{50}, $L = 1$ for 32% and $L = 2$ for another 10% of them. The maximum of L was 49 (about 0.15%). While $L = 0$ for 34% of N_{100}, $L = 1$ for 28% and $L = 2$ for another 12% of them in N_{100}. This suggests that most individual reordered streams have a relatively small number of lengths. Fitting the probability density

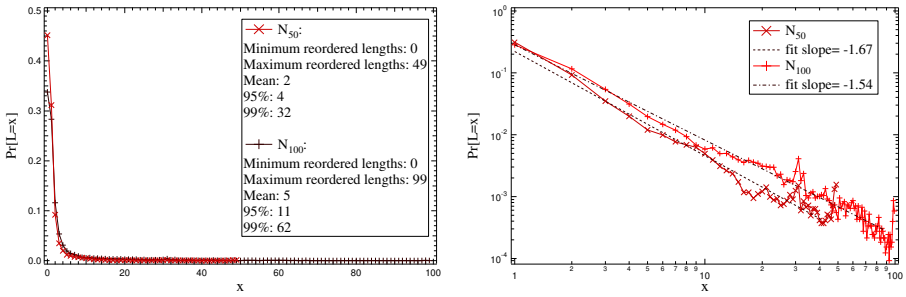

Fig. 1. (a) The pdf of the reordered packet lengths L for the RIPE data sets. (b) The pdf of the reordered packet lengths L and the power law fit

function of L on a log-log scale seems to indicate power law behavior for L. A power law is defined as $\Pr[L = x] \simeq C \cdot x^{-b}$, where the exponent b is the power law exponent and slope in a log-log plot. Figure 1(b) shows the exponent $b_{50} = 1.67$ in N_{50} and $b_{100} = 1.54$ in N_{100}, which are shown in dotted lines.

We found that each IP packet in a sequence had nearly a same probability to be reordered. This suggests that the cause of reordering acts upon a stream of IP packets almost as a Poisson process.

4.3 Packet lag P_L and Time lag T_L

In this section, we analyzed P_L and T_L on 104 unidirectional paths. To measure P_L, for each source-destination pair, we examined each arrival stream by checking its arrival sequence order. For each reordered packet in a reordered stream, we determined P_L by calculating how many packets with greater sequence numbers have been received before it.

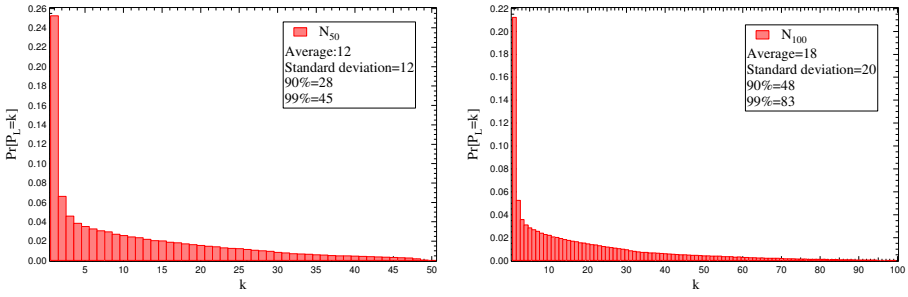

Fig. 2. The pdf of packet lag P_L for 2 data sets

The resulting pdf of P_L (Fig 2) shows that only around 40% of reordered packets occurs within 1, 2 or 3 packets in N_{50}, while around 35% in N_{100}. More-

over, the long tails of the distributions (up to 49 in N_{50}, while 99 in N_{100}) certainly impact the UDP performance because the reordering adds a high cost of recovering on the end host with finite buffer.

In the following, we computed the time lags T_L of different reordered packets. For each source-destination, we examined each arrival stream by checking its arrival sequence order to determine the reordered packets. For each reordered packet, we determined T_L by calculating the difference between the 1-way delay and the minimal delay in its sequence $\min(t_1, ..., t_P)$. Each time lag T_L of a reordered packet was normalized by the minimal one-way delay of the packets in its sequence, thus

$$T = \frac{(t_i - \min(t_1, .., t_P))}{\min(t_1, .., t_P)} \tag{5}$$

A normalized time lag T around 0 means that T_L is very small compared to 1-way delay and a normalized T of 1 means that T_L is very comparable to 1-way delay.

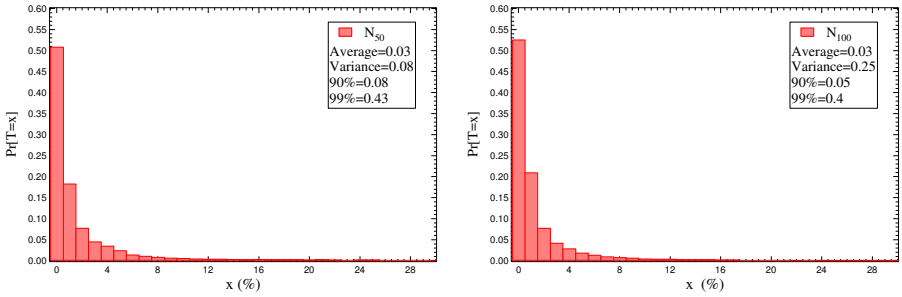

Fig. 3. The pdf of normalized time lag T for 2 data sets

Figure 3 shows the pdf of time lags normalized by the minimal one-way delay of its sequence. In our experiments the 90% percentile of the normalized time lag was 5% of the minimal one-way delay, 99% percentile of the normalized time lag was 40% of the minimal one-way delay. The Figures 3(a) and 3(b) indicate that most of the reordered time lag is very small, which suggests that packet reordering does not have a significant impact on the UDP delay since the reordering does not add large delay on the end hosts. However, the time lag can be also very large (up to 4 times the minimal one-way delay in N_{50} , while 17 times in N_{100}). Due to scale limitation, we did not show this large value in the figures.

4.4 Dependence of Reordered Probe-streams

For each source-destination pair, we calculated the unconditional reordered streams probability U. For each reordered stream, we examined whether the next stream was out-of-order or not to calculate the conditional stream probability C.

Figure 4 presents the measured values of the conditional reordering probability C and the unconditional reordering probability U in N_{50} and N_{100}: U is close to C for most paths. The weak dependence between two consecutive steams tells us that once a stream is reordered, the probability that the next stream is reordered does not seem to depend on whether the first was reordered or not. The effects that cause reordering seem to affect bursts at random, very similar to a Poisson process (which is memoryless).

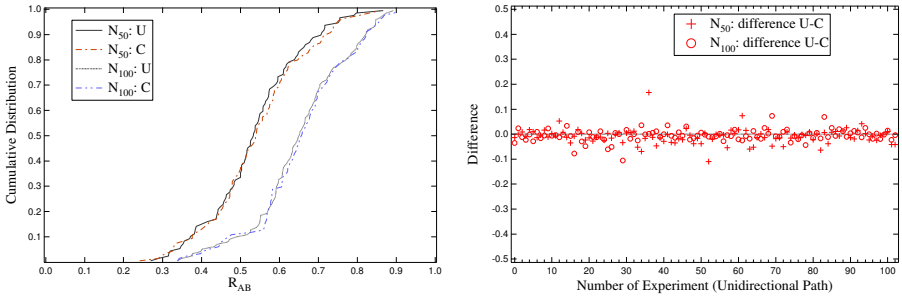

Fig. 4. The difference between U and C for 2 data sets

These observations are expected: the interarrival time between streams is large (30 seconds). The measurement shows that this interarrival time seems to be long enough to treat the streams as independent.

4.5 Asymmetry of Reordered Probe-streams

Since unidirectional packet delay and traffic are highly asymmetric, and it would be no surprise if reordering is asymmetric as well. For a UDP-based application, such as VoIP, asymmetric reordering may result in a transaction in which one party has an acceptable quality of service while the other has not. In this section, we analyzed the asymmetry of reordered stream ratios R_{AB} and R_{BA}. We omitted pairs for which the probe-streams in one of the directions were missing or received less than 50% of all the streams sent (there are approximately 180 probe-streams), leaving the data from in total 39 pairs.

For the purpose of analysis, we defined the DAR to be the degree of asymmetry of reordering as:

$$DAR = \frac{|R_{AB} - R_{BA}|}{\min(R_{AB,} R_{BA})} \tag{6}$$

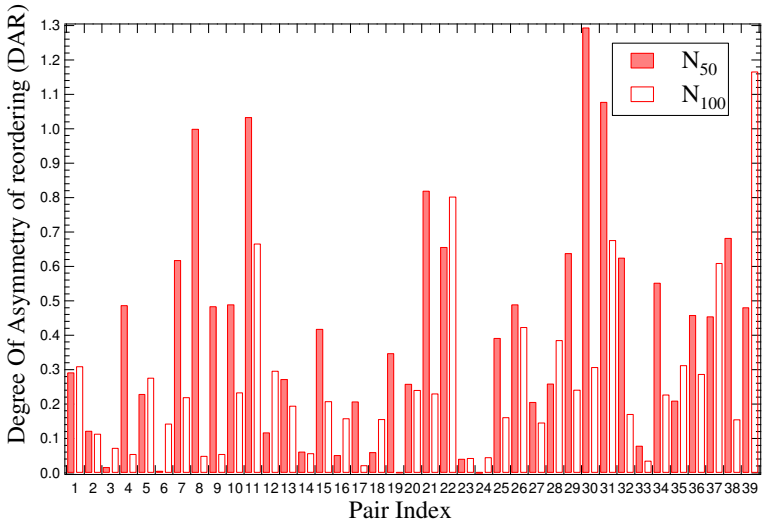

Fig. 5. Degree of Asymmetry of Reordered streams in all 39 symmetric traces. The average (over the 38 traces) is $E[DAR] = 0.41$, $\sigma_{50} = 0.32$ in N_{50}, while $E[DAR] = 0.26$, $\sigma_{100} = 026$ in N_{100}.

The function DAR plays a special role in the sense that it quantifies asymmetry, i.e., a DAR around 0 suggests the difference of R_{AB} and R_{BA} is very small compared to $\min(R_{AB}, R_{BA})$. Figure 5 presents the plots of unidirectional packet reordering ratios of all traces in our measurements.

We observed that the asymmetry exists on all traces, but it varies largely from testbox-to-testbox. For example, depending on the measurement set, DAR ranges from only 0.001 up to 1.29. We note that in N_{50}, 4 of the 38 traces have DAR larger than 1, and 3 of the 4 traces consist of a testbox in Slovakia. We have also studied and observed a similar behavior for N_{100}. Further, 72% of the probe-streams in N_{50} experienced reordering on a path from Greece to Slovakia (where the average probe-packets delay (on this link) is 25.1 ms, whereas the standard deviation is 19.5 ms and the hopcount is 15), while 34% of the probe-streams experienced in the opposite direction (where the average probe-packets delay is 25.9 ms, whereas the standard deviation is 10.6 ms, and hopcount is 17 or 18). We don't argue that the site-specific behavior reflect general Internet behavior, since it is found in [3] that site-specific effects can completely change. However, we suspect that the difference in stream reordering may be caused by the routing policies of the nodes on the path.

5 Conclusions

In this paper, we have analyzed the measurements of the end-to-end packet reordering by tracing UDP packets between 12 testboxes in RIPE NCC. Our results lead to several observations:

– Packet reordering is a frequent phenomenon in Internet. For bursts of 50 100-byte UDP packets, there were about 56% of the probe-streams with at least one reordering event, while about 66% for bursts of 100 100-byte UDP packets.
– Most individual streams have a relatively small number of reordering events. For bursts of 50 100-byte UDP packets, there were about 14% of probe-streams with more than two reordering events, while about 26% for bursts of 100 100-byte UDP packets. Also, the heavy tails on Figure 1(b) suggest that fitting the probability density function of reordered packet length L on a log-log scale seems to indicate power law behavior for L.
– Packet reordering has a significant impact on the UDP performance since the reordering increases a high cost of recovering on the end host. On the other hand, packet reordering does not have a significant impact on the UDP delay.
– The large interarrival time (30 seconds) between streams seems to be long enough to treat the streams as independent.
– The asymmetry of reordered streams ratios exist on all experiment pairs, but it varies greatly from testbox-to-testbox.

Acknowledgement: We thank Dr. Henk Uijterwaal of RIPE NCC for his support with the measurements of the packet reordering.

References

1. C. R. Bennett, C. Patridge and N. Shectman. "Packet Reordering is Not Pathological Network Behavior", Trans. on Networking IEEE/ACM, December 1999.
2. A. Morton, L. Ciavattone, G. Ramachandran and J.Perser. "draft-ietf-ippm-reordering-04.txt", IETF, work in progress.
3. V.Paxson. Automated Packet Trace Analysis of TCP Implementations. In Proceedings of the 1997 SIGCOMM Conference, pages 167-179, Cannes, France, September 1997.
4. RIPE Test Traffic Measurements, http://www.ripe.net/ttm
5. Michael Laor and Lior Gendel, The Effect of Packet Reordering in a Backbone Link on Application Throughput, IEEE Network, September 2002.
6. S.Jaiswal, G. Iannaccone, C. Diot, J.Kurose and D.Towsley, "Measurement and Classification of Out-of-Sequence Packets in a Tier-1 IP Backbone", Proceedings of the ACM SIGCOMM Internet Measurement Workshop 2002, November 6-8, Marseille, France.
7. G.Iannaccone, S.Jaiswal and C.Diot, "Packet Reordering Inside the Sprint Backbone". Technical Report TR01-ATL-062917, Sprint ATL, June 2001
8. M. Allman and E. Blanton. "On Making TCP More Robust To Packet Reordering,", ACM Computer Communication Review, 32(1), January 2002.
9. V. Paxson, G. Almes, J. Mahdavi and M. Mathis. "Framework for IP Performance Metrics", RFC 2330, May 1998.