Spatial-Temporal Analysis of passive TCP Measurements

Eli Brosh Galit Lubetzky-Sharon Yuval Shavitt

***Abstract*— In this paper we look at TCP data which was passively collected from an edge ISP, and analyze it to obtain some new results and deeper understanding of TCP loss process. The focus of our study is to identify the 'root cause' links, i.e., the links that are responsible for the majority of the losses or reorders found on the end-to-end TCP connection. We suggest a new root cause criterion and a cost-effective algorithm to identify the root cause links. The algorithm incorporates a new out-of-sequence packet classification technique. We test our algorithm on the collected and simulated data and analytically justify its correctness. The simulation results show that the algorithm has a 95% detection rate with 10% false detection rate.**

We also analyze TCP temporal loss process, and found that the burst loss size is geometrically distributed. We analyze the TCP time-out loss indication under the Bernoulli loss model, which is the simplest model that can cause a geometric distribution, and show that the behavior of the TCP loss process is not different than when tail drop is assumed.

I. Introduction

TCP is the transport protocol which is carrying most of the current Internet data. Thus understanding its characteristics can help ISPs, equipment manufacturer, and protocol designers. In this paper we look at TCP data which was passively collected from an edge ISP, and analyze it to obtain some new understanding of TCP. The data is comprised of two samples that include about 49 Million TCP sessions[1] and over 260 Million packet headers. The collection from an edge ISP is one of the few done from this vantage point. We concentrate on the spatial identification of low performing links and on the temporal spreading of the loss process. We also revisit some measurements that where done at backbone ISPs. We believe our data is typical to many stub ASes that serve private and business clients connected by either dial-up or ADSL modems.

Our main focus is to identify the Internet links that are responsible for poor end-to-end TCP performance, i.e., the network links with significant losses and reorders. In contrast to previous studies [1], [2], [3], [4] that attempt to infer the link loss characteristics using active measurements techniques (e.g., by sending back-to-back unicast packets to different destinations or by using multicast packet probes) we make inferences based on TCP traffic passively collected at a single point within the network. Passive measurements enable us to leverage large amounts of traffic (millions of TCP sessions) without the overhead of active probes which may bias the results.

While there were a few passive measurements attempts in the past, they have different goals and requirements, and thus cannot be directly compared with our study. For instance, [5] and [6] detect shared congestion links using end-to-end passive measurements, where the former requires the senders to cooperate by timestamping the packets and the latter requires the measurement point to observe a reasonable amount of the traffic processed by the bottleneck links.

The closest study to ours is the one done by Padmanabhan *et al.* [7] which attempts to estimate the loss rate on network links. For this end, high complexity analysis such as Bayesian inference and linear programming were used. In our case we only want to identify the *root cause* links, which are the links that have the highest loss or reorder rates compared to their neighborhood. This enables us to deploy a much simpler algorithm which we developed. This algorithm uses topology information which is acquired by performing traceroute to the IP addresses in the sampled data. We evaluate the performance of the proposed algorithms using simulations and real-world Internet traffic. We found that our algorithm has a very high detection rate and a law false detection rate.

One of the challenges in our work was to identify the loss and reorder events. We developed an efficient packet classification technique which is used to infer the loss and reorder rates of individual TCP flows. Our algorithm differs from the recent algorithm by Jaiswal *et al.* [8] since it does not require the use of both directions of

[1]In this study we use the term TCP session to refer to a one-way direction of a TCP connection

a TCP connection. Using a one way TCP analysis is useful in the current Internet that shows high percentage of TCP asymmetricity, e.g., [8] indicates that 10% of the flows in a backbone ISP are asymmetric while our traces indicate that 5% of the flows in an edge ISP are asymmetric. In addition, our algorithm is much simpler since it does not attempt to infer the TCP state, instead, we leverage the IP identifer field (similar to [9], which uses this field to actively measure packet reordering).

Our temporal analysis looked, at the first time in a passive TCP study, at the distribution of the loss burst size. For loss bursts studied based on active measurements see [10], [11]. We have found that it roughly matches the geometric distribution; this incites us to use the Bernoulli loss model which is the simplest model to explain such a distribution. To measure the impact of the loss model on TCP throughput analysis, we analytically derive a formula for the probability that the loss indication is due to a time-out event. Interestingly, we have found that Bernoulli and tail drop loss models have the same structure, which implies that the well-known TCP throughput formulaes in [12], [13], [14], [15] originally developed under the tail-drop assumption characterize TCP's behavior under both loss models.

The remainder of the paper is constructed as follows. We begin in Section II by describing the experimental setup used to gather the TCP traffic. In Section III we discuss the methodology, the challenges, and the algorithms used to identify the low performance links. In Section III-A we present the root cause identification algorithm and in Section III-B we evaluate its performance via simulations. In Section III-C we present the rules used to infer and classify the out-of-sequence behavior in the observed sessions. We apply our methods to real-world Internet samples and present the results in Section III-D. The second part of the study that deals with the process of consecutive packet losses is described in Section IV. In Section IV-A we analyze the effect of the Bernoulli loss model assumption on the modelling and analysis of TCP throughput. Section V concludes this paper.

II. Experimental Setup

The network that hosted our experiment belongs to a large Internet Service Providers in Israel, thus we could monitor large amounts of traffic with diverse characteristics: both private and business customers connected via dial-up and ADSL modems. We were connected to a mirror port of a Cisco switch that combines multiple trunks of client switches to an outgoing router connected to the internet. The results we report here are from two samples taken at the same ISP but from different locations. For this study, we sampled only the IP and TCP headers of each packet due to privacy concerns, and to save space. The first sample was taken on July 2002, from business ADSL clients, over 2.5 days. Most of the traffic, around 97%, was TCP, spread over 61 million sessions and 6 million data packets. The second sample was taken on December 2002 (see Table I). It contains 17 hours of sampling and it includes both business and private clients. This sample has 225 million data packets. Due to policy routing and load balancing at the ISP we were not always able to capture both directions a TCP session. For about 5% of the TCP sessions, we only saw one direction of the connection.

	July Sample	Dec Sample
Duration	2.5 days	17 hours
# TCP packets	61,316,032	224,982,834
# TCP sessions	5,877,269	37,531,953

TABLE I
SUMMARY OF THE TRACES

III. Spatial Loss Analysis

In this section we discuss the methodology and algorithms used to identify the root cause links from passively gathered TCP traces. We employ a comprehensive approach that includes both an algorithmic solution which correlates topological information with the loss and reorder rates of TCP flows and a packet classification algorithm that infers the required flow characteristics.

Our first step is to derive the network topology formed by the routing paths of the end-to-end TCP sessions. To determine this topology we collected all the IP addresses of the end-hosts found in the TCP traces and performed `traceroute` from the sampling point to a subset of the top $10,000$ end-hosts generated the largest number of packets. This gave us the network region where most of our traffic flows, and ensures statistical robustness. The set of routing paths from the sampling node to the end-hosts forms a directed acyclic graph (DAG) which was fairly close to a tree.

To determine the loss and reorder rates of the paths comprising the DAG we develop a packet classification technique. The classification technique, detailed in Section III-C, is based on analyzing TCP sequence number and IP identifier patterns and identifies the various causes of TCP sequencing problems: packet retransmissions by TCP senders, and network-generated packet reordering

or duplication. The reorder rate of a path is calculated by measuring the ratio of the number of reordered packets to the total amount of packets on this path. However, estimating loss rates is more challenging since there is not necessarily a one-to-one correspondence between packet retransmissions and packet losses. The discrepancy between these measures is attributed primarily to spurious time-outs [16], [17] which occur when the round-trip time (RTT) suddenly increases and may cause unnecessary retransmissions.

Since we can accurately measure the size of a loss burst that occurs before the measurement point using sequence number gaps (see Section IV), we only need to consider spurious time-out inaccuracies for retransmissions arising from losses that occur after the measurement point. To reduce potential inaccuracies we follow the assumption of Benko and Veres [18] that a large set of consecutively retransmitted packets is most likely caused by a spurious time-out event, and exclude from the loss ratio computation the retransmission bursts due to losses after the measurement point that their size exceeds some threshold, e.g., three. The low occurrence of large loss bursts (e.g., in Figure 9 such bursts account for 1.5% of all bursts), implies that this process would most likely eliminate the majority of spurious time-out inaccuracies.

We assume that the routing paths and their loss and reorder rates remain stable during the analysis. These assumptions are influenced by the findings of previous passive and active measurement studies [19], [7]. For example, the findings of Padmanabhan *et al.* [7] which are based on passive measurements of traffic flows between a wide-range of clients and a popular Internet server indicate that loss rates are likely to remain stable for periods of minutes.

While estimating a path's loss rate is a straightforward task, deriving the loss rate of an internal network link is more challenging due to the lack of a unique mapping from path loss rates to the loss rate of an individual link [20]. Therefore, we seek to find a solution to a simplified problem: detecting network links that are likely to have high loss or reorder rates compared to their neighborhood. We term such links as *root cause links*. In the following section we define this notion formally and present a cost-effective heuristic for solving this problem. We then evaluate its performance via simulations. For simplicity of presentation we describe this algorithm in the context of the loss performance measure. Nonetheless, the proposed heuristic is applicable to other multiplicative performance measures, such as packet reordering[2] or packet duplication.

Using our algorithm we were able to analyze the Internet traffic traces and derive root cause links for losses and reorders. The obtained results are described in Section III-D. We avoided packet duplication analysis due to the rareness of such events which may bias the results significantly. For example, in our traces only 1.3% of all TCP's sequencing problems are due to in-network packet duplication, and similar proportion was also obtained in [8] using samples taken from a backbone link of a backbone ISP.

A. Root Cause Identification Heuristic

Given a set of paths and an associated set of loss rates, r_i, our goal is to detect the lossy links, also termed root cause links. For the loss process we assume a Bernoulli model where each link drops a packet independently of others with some fixed probability. Ideally, we would like to find the links that their loss probability exceed a desired threshold. However, to reduce the complexity of the problem we propose an alternative root cause criterion. A link (u, v) is considered to be a root cause if the difference between its loss probability and the maximum loss probability of the links entering or leaving either v or u is larger than a pre-defined threshold δ.

For the identification process we use the notion of average loss rates. The average loss rate of link l, denoted by $\hat{p_l}$, is defined as a weighted loss rate mean taken over the paths that include l such that $\hat{p_l} = \sum_{j:l \in t_j} w_j r_j$ where t_j is the set of links on path j and w_j [3] is the weight (number of packets) of path j. Let us now analyze the properties of the calculated link loss rates. The underlying assumption is that the input loss rates capture the loss probabilities of their paths, and thus we have $r_i = 1 - \prod_{l \in t_i}(1 - p_l)$, where p_l is the loss probability of link l. Expanding the weighted mean of $\hat{p_l}$ we get $\hat{p_l} = 1 - \sum_{j:l \in t_j} w_j \prod_{k \in t_j}(1 - p_k)$. The latter equation can be alternatively expressed as

$$\hat{p_l} = p_l + (1 - p_l)e_l \tag{1}$$

where $e_l = 1 - \sum_{j:l \in t_j} w_j \prod_{k \in t_j, k \neq l}(1 - p_k)$ is the contribution of the links on the paths that share l, excluding l itself, to the average loss rate.

This implies that the average loss rate of a link can be viewed as a biased estimator of its loss probability. To

[2]Packet reordering can be viewed as a multiplicative measure since there is a high probability that a packet is reordered only once on the sender to receiver path.

[3]For clarity we omit the normalization factor $\sum_{j:l \in t_j} w_j$ from the equation.

determine the mean value and the variance of the bias we assume that the loss probabilities of the contributing links $\{k : k, l \in t_j, k \neq l\}$ are i.i.d random variables with mean μ_p and variance σ_p^2. For the simplicity of the analysis we also assume that all the paths have the same number of edges h. Under these assumptions it can be shown that

$$\begin{aligned} E(\hat{p_l} - p_l) &= (1 - p_l)(1 - (1 - \mu_p)^{h-1}) \\ V(\hat{p_l} - p_l) &= (1 - p_l)^2 \sum w_j^2 \cdot \\ &\quad ((\sigma_p^2 + (1 - \mu_p)^2)^{h-1} - (1 - \mu_p)^{2(h-1)}) \end{aligned} \quad (2)$$

Using the above equations we can deduce the following observations:

- The mean value and the variance of the bias tend to constant values when p_l is upper bounded by a small value, as is often the case in modern IP networks.
- The size of the variance is largely determined by the term $\sigma_p^2 + (1 - \mu_p)^2$, which is the second moment of the success probability random variable. Specifically, if $\sigma_p^2 + (1 - \mu_p)^2 < 1$ the variance of the bias decreases exponentially as h increases.
- The accuracy of the estimator is proportional to the weights of the paths that share a link. Therefore, for a particular setting, e.g., a link that is shared by many flows of significant weights, the bias may be small enough to produce an estimator that tends to the real loss probability value.

For the detection process we use a simple rule that implements the root cause criterion using the average loss rates of the links. That is, if a link has either incoming links or outgoing links with an average loss rate that is lower by at least δ than its own average loss rate, this link is classified as a root cause link. As noted earlier, δ is the detection threshold. It is important to note, that the difference between the loss probability estimators for two adjacent links has a bias which is far lower than the estimator bias for a single link. This is since the terms in the single link bias that are due to flows that run through both links cancel each other. This improves our root cause accuracy and justify the use of our algorithm.

The formal description of the proposed algorithm is given in Figure 1. The input is a directed acyclic graph $G = (V, E)$ and a rate function r that specifies the average loss rate of each link in E, where V and E are the node set and link set, respectively. The output is the root cause link set denoted by S.

For the Internet data analysis and the simulations δ was set by default to zero and the weights were assigned

Algorithm Root-Cause-Identify(G, r)

1. $S \leftarrow \emptyset$
2. for each $(u, v) \in E$ do
3. if $\forall z : (z, u) \in E \quad r(u, v) - r(z, u) > \delta$
4. or $\forall w : (v, w) \in E \quad r(u, v) - r(v, w) > \delta$
5. $S \leftarrow S \bigcup (u, v)$
6. return S

Fig. 1. Heuristic for identifying the root cause links

in proportion to the amount of traffic on the paths such that the average loss rate of a link represents the ratio of losses and total packets on this link.

B. Performance Evaluation

In this section we evaluate the average performance of the root cause identification heuristic using simulations. Our main objective is to investigate the effectiveness of the detection process including its false alarm and miss detection characteristics.

For the simulations we use a DAG topology generated in several steps. First, a random tree is constructed where the degree of each node is randomly and uniformly chosen from the discrete interval $[1, d]$, where d denotes the maximum node degree. Then, each path without branching is collapsed to a single link. Finally, a small number of extra edges (set to 10% of the node count in our simulations) is added to the graph by repeatedly selecting at random a pair of nodes not connected by a link. We assume that all flows originate from the root node and terminate at the tree leaf nodes. Each leaf node represents an end-host and thus is assigned the unique root to leaf tree path. An additional path is assigned per each extra link by randomly and uniformly selecting a root to leaf DAG path that includes this extra link.

For each simulation configuration we randomly generate a DAG topology, a loss probability vector with a size that matches the number of links in the DAG, and flow sizes for the paths. We use two alternative distributions to generate the loss probability vector: Zipf distribution with $\alpha = 1$, and uniform distribution. The range of the loss probabilities is selected to be $0-0.04$ to correspond to typical Internet loss settings [7]. The flow sizes are randomly selected from the a Zipf distribution with $\alpha = 1$.

In each simulation configuration our experiment is repeated 5000 times, where the links are randomly and

uniformly assigned a permutation of the loss probability vector. This permutation determines the path loss rates as described in Section III-A. In each repetition the root cause identification algorithm is used to gather statistics per each true and estimated loss probability element, such as whether the corresponding link is classified as a root cause, whether it is classified as a root cause under both true and estimated probabilities, and miss detection (false positive) and false alarm (false negatives) rates.

Figures 2–4 depicts the simulation results for a setting with 200 nodes, maximum degree of 10, and Zipf loss probabilities. The upper graph in Figure 2 shows the portion of experiments where the links that correspond to an estimated loss probability value are classified as root causes by our algorithm, marked as total detections; and the portion of experiments where such links are classified as root causes under our algorithm and the root cause criterion (i.e., those that obey the root cause criterion for both the estimated and the true loss probabilities), marked as true detections. For reference, the root cause criterion curve (see Section III-A) is also presented. The lower graph in Figure 2 presents the same data as a function of the true loss probabilities.

From these graphs we see that the likelihood (i.e., the portion of experiments) of a root cause classification increases as the loss value increases, which is a desired property of the proposed criterion. For large loss probabilities above 0.02 the detection likelihood is more than 80%. For smaller values the likelihood curve decreases, such that links that correspond to loss probabilities below 0.001 are classified as a root cause in no more than 25% of the runs.

Figure 3 illustrates the ratios of detections, miss detections, and false alarms, in respect to the root cause criterion. For large estimated loss values, e.g., above 0.02 (which covers the upper third of the probability range), the heuristic has a high detection ratio around 95% and a small ratio of miss detections and false alarm below 10%. For smaller probabilities there is a drop in the detection ratio and an increase in the false ratios, such that the miss detection ratio reaches 100% for very small loss values. This behavior (the increase in the false ratios for small loss probabilities) is expected since small probability values are more vulnerable to estimation errors. However, since we are interested in the upper range of loss probabilities we can safely ignore these high error ratios.

Figure 4 shows the relationship between the true and the average estimated loss probabilities values. The graph reveals a constant bias which is consistent with Equation

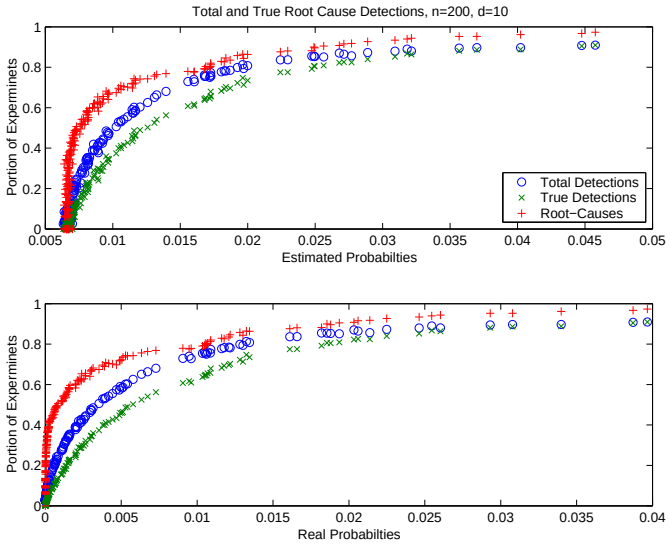

Fig. 2. Portion of root cause detections in a 200 node topology, d=10, and Zipf loss distribution

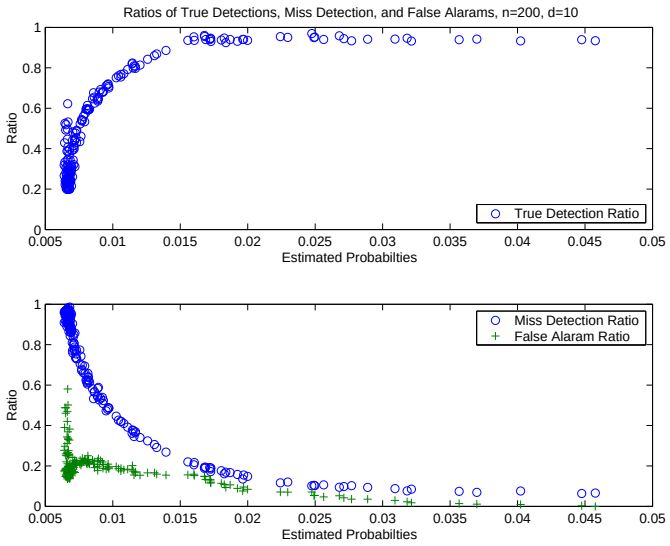

Fig. 3. Ratios of valid and false detections for a 200 node topology, $d = 10$, and Zipf loss distribution

2. Computing the bias analytically using Equation 2 we get the value of 0.0075 (in this configuration the average path length h is around 3 and the average probability μ_p is 0.0038), which is sufficiently close to the bias shown in the graph of 0.0067.

To check the consistency of the results we considered several DAG configurations where the number of nodes, denoted by n, ranged from 100 to 5000 and the maximum node degree d varied from 5 to 10. For these experiments we obtained similar results and therefore the corresponding graphs are omitted. To test the sensitivity

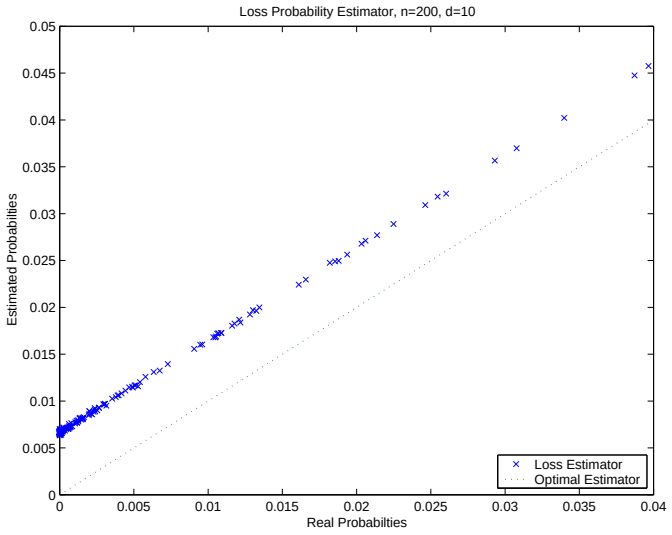

Fig. 4. Loss probability estimators for a 200 node topology, $d = 10$, and Zipf loss distribution

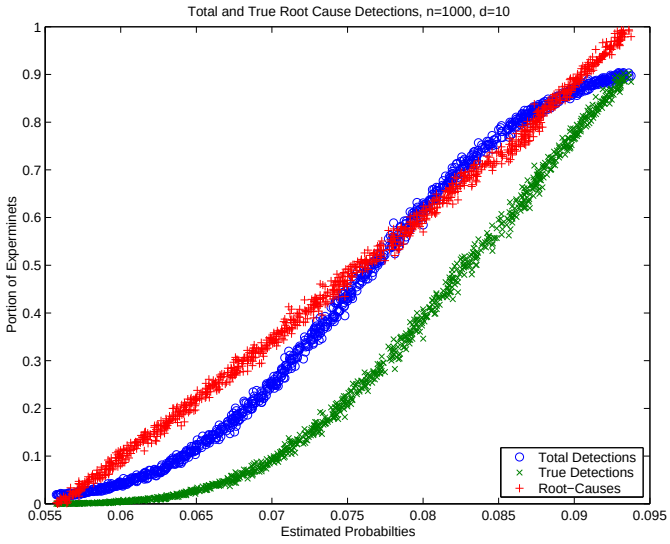

Fig. 5. Portion of root cause detections for a 1000 nodes topology, $d = 10$, and uniform loss distribution

of the algorithm to the type of distribution used to derive the loss rates, we repeated the previous settings using a uniformly generated loss probabilities. The results indicate that the root cause detection curves (total and true) exhibit slower decrease compared to the Zipf case, and that error ratio bounds are similar for the two distributions. Figure 5 shows the root cause detection results for a 1000 node topology, maximum degree $d = 10$, and loss probabilities derived using a uniform distribution. In this configuration the average loss probability is 0.02 generating a bias of 0.05. Note that this setting may not represent typical Internet scenarios due to the relatively high loss rate considered.

The main conclusion that can be drawn from these simulations is that the algorithm is very efficient in detecting the lossy links that obey the root cause criterion. As noted earlier, for root cause links in the third upper range of loss probabilities it achieves a high detection ratio, typically above 95%, while maintaining a low error ratio, typically below 10%. The detection rate can meet almost any arbitrary threshold requirement, due to the monotone increase (decrease) of the detection (false) ratio as a function of the loss probability size, by the adequate selection of the loss range of the root cause links.

Another conclusion that can be deduced is that there is a positive correlation between the links detected as root causes and their true loss probability, i.e., the likelihood of classifying a link as a root cause increases as its loss value increases, where the exact form of the correlation depends upon the link loss distribution and the routing paths. In general, the estimated loss probabilities have a constant but unknown bias compared to the real probabilities. However for specific settings, e.g., when we have only a small number of sporadic links with high losses that are shared by many flows, the estimator may tend to the real value and thus the heuristic can successfully identify links that their absolute value is above a desired threshold. This is a valuable property since such scenarios are typical in the Internet.

C. Packet Classification Technique

In order to measure the loss and reorder rates of selected connections we develop a packet classification technique based on examining the out-of-sequence TCP packets and their IP identifier pattern.

Following Jaiswal *et al.* [8] we define an out-of-sequence (OOS) packet to be a packet that its TCP sequence number is smaller than previously observed sequence numbers in that connection. Such a packet is generated by one of the following events: *(1) Retransmission.* The loss of a data packet triggers the sender to retransmit a packet with a previously used sequence number. *(2) Reordering.* The network changes the original ordering and causes a packet to arrive before its proceeding packet *(3) Duplication.* The network duplicates the original packet and generates at least two packets with the same sequence number. Note that the causes and the impact of these anomalies have been extensively studied [21], [8], [9], [22], [11], [23].

We begin the process by extracting the observed TCP connections. Given an identified connection we

analyze the data direction headers, i.e., the sender to receiver data headers, and classify the out-of-sequence packets. Observe that our technique is dependant on the data header fields only (i.e., it doesn't rely on the acknowledgement packets in the reverse direction) and thus can be applied to each direction of a TCP connection separately.

Our classification technique leverages two header fields: the sequence number field in the TCP header and the identification (ID) field in the IP header. The TCP sequence number field identifies the sequence number of the first byte of data carried in the segment and is used to guarantee TCP's in-order reliable delivery. The identification field uniquely identifies each transmitted IP datagram. In practice, most Berkeley-derived TCP/IP implementations guarantee the delivery of unique IDs by having the IP layer increase a global variable each time an IP datagram is sent [24], [9]. This implies that the IDs of a packet flow emitted from a sender forms a monotonic increasing sequence, i.e., given two packets x and y where x is emitted before y we have that the ID of x is smaller than the ID of y. Since the assumption about the ID field is implementation dependent we verified its consistency in several common operating systems including Windows 2000/XP and several Linux variants. Thus, we can expect this assumption to be valid for the larger majority of the sampled traffic.

Given a data packet x we denote its IP identifier and sequence number by $id(x)$ and $seq(x)$, respectively. To classify the observed packets we use simple rules to identify the scenarios resulting from the different types of events.

- **Retransmission - due to a loss after the measurement point.** Assume that we observe packet x and an earlier instance of x, denoted by x', such that both packets have different IDs and a common sequence number. In this scenario, illustrated in Figure 7, we assume that the original instance x' was lost after passing the measurement point, and thus classify x as a retransmission.
- **Retransmission - due to a loss before the measurement point.** Assume that current packet x has not been previously observed (i.e., $seq(x)$ is detected for the first time) and that its ID is in order. Where the in-order property is determined by comparing x's ID with the ID of the earliest packet with a sequence number larger than $seq(x)$, denoted by x''. That is, x'' represents the packet succeeding x's original instance. In this scenario, illustrated in Figure 8, we assume that the original instance was lost before reaching the measurement point, and thus classify x as a retransmission.
- **Reordering.** Assume that the current packet x has not been previously observed and its ID is out of order, i.e., $id(x) < id(x'')$. In this scenario, we assume that the order of packets was inverted, and thus classify x as a reorder.
- **Duplication.** Assume that we observe packet x and a previous instance such that both packets have equal sequence numbers and equal IDs. In this scenario, we classify x as a duplicate.

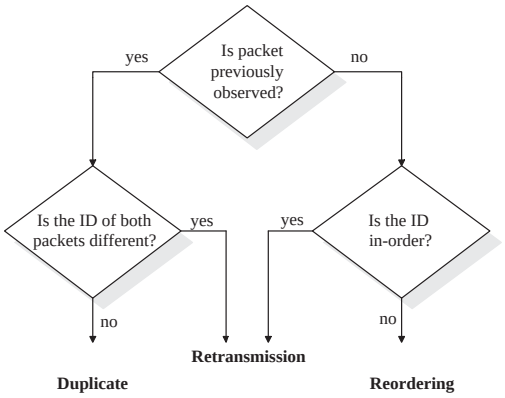

Fig. 6. Classification process of out-of-sequence packets

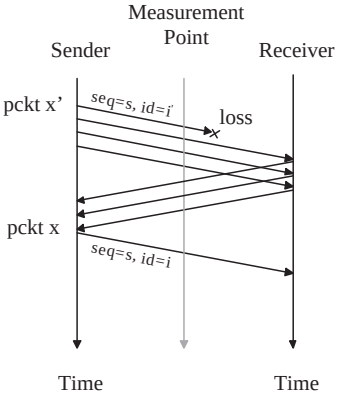

Fig. 7. Retransmission due to a loss after the measurement point

The complete classification process is illustrated in Figure 6.

Our simple classification technique is based on the IP identifier field assumption and thus is prone to errors due to non-standard implementations of TCP/IP stacks. Although we may weaken the effect of this type of error by observing the reverse direction of the connection (i.e., the acknowledgment path) and inferring TCP's state (as done, for example, in [8]), we decided not to do so. One reason is that the alternative approach may prove to be

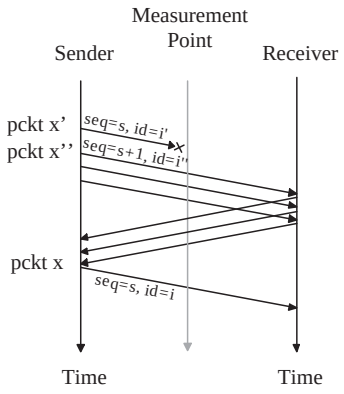

Fig. 8. Retransmission due to a loss before the measurement point

less accurate than expected, due to timing related errors involved with estimating the sender's retransmission time-out (RTO) or RTT parameters, or reverse direction flow processing errors due to dropped acknowledgements (ACKs). The one-way classification capability of our approach is valuable due to the support for asymmetric connections which is useful in the case of traffic load balancing by an ISP.

The accuracy of the classification is also dependant upon the comprehensiveness of the viewed data [8]. For example, the classification process cannot detect the loss of an entire congestion window of packets that happens before the measurement point as well as the loss of the first packet in a TCP connection.

D. Internet Measurement Results

In this section, we apply the root cause identification heuristic to the Internet traffic sample from Dec. 2002 (see Section II for further details) and evaluate its effectiveness. In addition, we apply the OOS classification technique to all the TCP traffic in the Internet samples and report the obtained results.

For the link identification process our aim was to capture the most lossy links and thus we considered a reduced topology that includes only the 150 links and 100 nodes with the worst average loss rates, where the average loss rate of a node or a link is computed by matching all the traffic flowing through it and calculating the average losses it experiences. We are aware that using this particular setting we might miss most of the last mile losses [7], although, we did capture two last mile links that connect portal servers.

Given the resulting forest, i.e., the collection of connected components, we applied the root cause heuristic and identified the lossy links. The same identification process was repeated for the packet reordering measure as well. Given that we don't have previous knowledge on the real error rates of the Internet links we verified our results by visualization of the graph of the most problematic nodes and their connected links. This graph is too dense to be presented in the paper format (for a viewable version we used an A2 size paper). Interestingly, out of the 150 worst performing links, only 15% were identified as root cause links.

Link type
Internal link in a US service provider
Israeli ISP – Korean ISP
Israeli ISP – UK Software company
Israeli ISP – Israeli portal
Israeli ISP – Israeli portal
Internal link in an Israeli portal
Link between 2 Israeli ISPs
Israeli ISP – US telecom
Israeli ISP and a US ISP
2 US ISPs

TABLE II
WORST LOSSY LINKS, HIGHEST TO LOWEST

Link type
Link between 2 Major US ISPs
Israeli cable ISP – small Israeli ISP
Internal link in an Israeli ISP
US ISPs, Denver - NJ
Israeli ISP – Israeli Portal

TABLE III
WORST REORDERED LINKS, HIGHEST TO LOWEST

To our surprise the most lossy link was an internal link in a US service provider. The second most congest link was between an Israeli ISP and a Korean ISP, which we suspect to be a satellite connection. The top loss root cause links are given in Table II, sorted by the loss rate: highest to lowest. The topological link locations were derived manually using databases such as the `whois` database. Similarly, we give the top reorder root cause links in Table III. Observe that there is no direct correlation between the highest lossy links and the highest reordered links. The results also indicate that most of the top-rated links, i.e., those with significant losses or reorders, are inter-ISP links rather than intra-ISP links. These results are consistent with the findings of [7] which indicate that links with significant losses tend to be located across AS boundaries.

We now proceed to describe packet classification

	July Sample	Dec. Sample
TCP sessions	5877269	37531953
Out-of-sequence	916961 (15%)	2016325 (5.3%)
Retransmissions	512877 (10%)	1008464 (2.8%)
Reorders	460539 (7.8%)	1067425 (2.8%)
Duplicates	14970 (1.6%)	12122 (0.06%)

TABLE IV
SUMMARY OF OUT-OF-SEQUENCE SESSIONS

	July Sample	Dec. Sample
TCP Packets	61316032	224982834
Out-of-sequence	2337501 (3.8%)	3503015 (1.5%)
Retransmissions	1646638 (70.44%)	2010863 (57.4%)
Reorders	648596 (27.64%)	1453315 (41.48%)
Duplicates	42267 (1.8%)	38837 (1.1%)

TABLE V
SUMMARY OF OUT-OF-SEQUENCE PACKETS

results. Out of the 40 million TCP sessions 6.7% had experienced an out-of-sequence event (i.e., an OOS packet), 3.8% experienced a retransmission event, and 3.5% experienced a reorder event. A small percentage of these flows, less than 0.5%, experienced a combination of different types of events. The classification results for the collected sessions are given in Table IV. For each event the table indicates the number of TCP sessions with this event both in absolute numbers and in relative percentage (in respect to the total session count). Overall, 2% of the 286 million packets we observed were classified as out-of-sequence packets. As expected, the majority of these packets, around 62%, is caused by retransmissions, reordering comes second with 36%, and packet duplication appears to be a rare event that account for only 1.3% of the OOS packets. The classification of the sampled TCP packets is given in Table V. This table is structured as Table IV and it shows the absolute number and relative proportion of the packets (in respect to the OOS packets) in each category.

We didn't compare our results with the result of other active measurement studies such as [11], [22] due to major methodological differences between passive and active inferences [25]. Instead, we compare our findings based on traffic samples from an Israeli ISP with the results of a passive measurement study in a Tier-1 IP backbone [8]. Although the results exhibit large variance, it is interesting to see that both studies provide similar insights, namely that packet reordering and duplication effect only a small portion of the problematic packets in the Internet, using different classification methods. Unlike our study, Jaiswal *et al.* [8] infers the causes of sequencing problems by observing both directions of a connection and replicating the sender's TCP state.

IV. LOSS BURST ANALYSIS

In this section we study the process of consecutive packet losses, i.e., *loss bursts*. We develop a simple methodology to infer the degree to which packet loss occurs in bursts from passive measurements of TCP traffic, and investigate how efficiently TCP deals with such bursty losses. Finally, we note that the observed loss patterns may better match the Bernoulli loss model, and investigate the effect of this assumption on the modelling and analysis of TCP throughput.

We begin with inferring loss bursts. As noted earlier in Section III, the inference of losses from retransmissions is challenging due to spurious time-outs. To handle this challenge we consider the case were the loss burst can be accurately determined, i.e., when the loss bursts occur before the measurement point. The estimation is unbiased if the considered bursts are representative samples of the entire 'population' of bursts, e.g., when the measured loss bursts are independent of one another and uncorrelated with the location of the measurement point. Also, to get meaningful results, the amount of data in the reduced sample set should be large enough. The first requirement is achieved by the location of our measurement point very close to one of the connection end-points, and the second requirement is satisfied by our Internet traces which contain tens of thousands of burst samples.

The basic idea behind the inference method is to detect a sequence number gap (due to losses that occur before the measurement point) and count the number of retransmitted packets used to fill this gap. Using this method we can detect loss bursts that contain variable length packets, which are common in many application level protocols that operate above TCP, e.g., HTTP.

Given a trace of TCP packets we classify the packets using the technique described in Section III-C and filter the results to consider only the retransmissions that occur before the measurement point. After the filtering we can identify a loss burst using the corresponding retransmission burst. For this purpose we seek a retransmitted packet, denoted by x, that its sequence number is lower than the previously observed packet, denoted by y. The size of the lost burst is computed by counting the number of unique packets following y (and not seen

before) that cover completely the sequence number gap $[seq(x), seq(y)]$.

Using passive measurements to infer loss bursts enables us to consider a large amount of data at the expense of introducing potential inaccuracies. One potential source of errors is the lack of timing analysis in our technique, e.g., we cannot determine whether all the consecutive losses belong to a single congestion window, and thus may incorrectly interpret multiple bursts as a single merged burst. Inaccuracies may also result from TCP/IP implementations that combine the data of several lost packets into a single retransmission. We expect this phenomena to have a minor impact on the results due to the measured rareness of the event in the alternative scenario of retransmission bursts due to losses after the measurement point.

Figure 9 presents the normalized loss burst histograms for our two Internet samples (see Section II). The largest burst that we captured was of 31 packets. However, only a small number of sporadic bursts had more than 20 consecutive losses, and therefore we limit the graphs accordingly. For the burst computation we considered 0.25% of all the observed retransmission bursts: the July 2002 curve is based on 40458 loss bursts, and the December 2002 curve is based on 108002 bursts. As expected, single packet losses account for the large majority of the bursts, around 83%, and double losses occupy 12% of the bursts. It is interesting to see that both histograms are very similar although derived from traces with different traffic characteristics (there is a nearly perfect matching for bursts of 7 packets or less, and minor discrepancy for larger bursts due to the low number of large size burst samples). We compared the loss burst histograms for different times of the day, morning, early afternoon, evening, and night, and found them similar. The only significant difference among the four curves is that the evening losses have more than 10% higher probability for a single loss than the others. The consistency of the results strengthens the validity of our methodology.

Another aspect of bursty losses we investigate is how efficiently TCP deals with them. Packet loss can be detected by TCP in one of two ways, either by the reception of triple-duplicate (TD) ACKs at the sender, or by time-outs. To measure the performance of TCP loss recovery mechanisms we classify the first packet in each retransmission burst according to its trigger, TD ACK or time-out, using the technique of TCP state replication [26]. That is, the retransmission is classified according to the inferred state, slow-start or fast

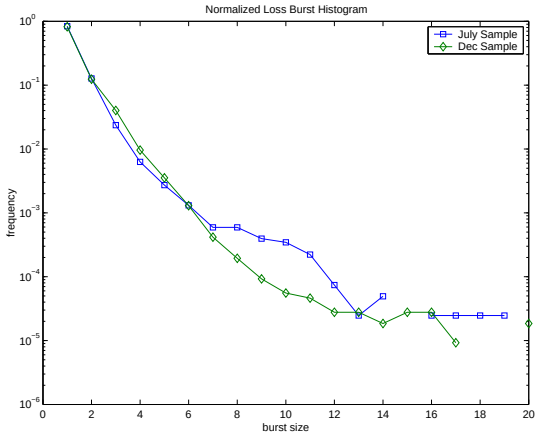

Fig. 9. Loss burst histograms for Internet samples

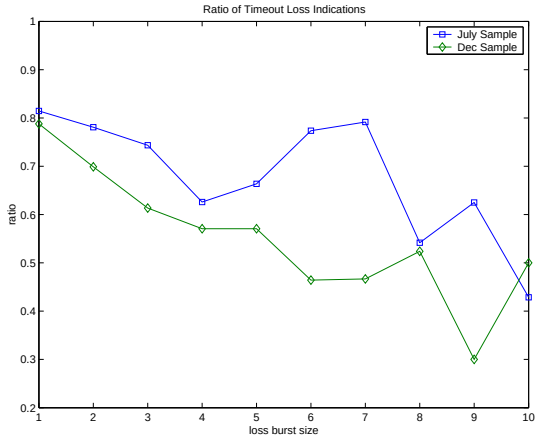

Fig. 10. Proportion of time-out loss indications in the Internet samples

recovery/retransmit. Figure 10 presents the proportion of time-out indications as the function of the loss burst size. The results indicate a low correlation between the loss burst size and the amount of time-outs, which contradicted our initial expectations that the amount of time-outs in TCP connections is positively correlated with the degree of loss burstiness that the connections experience. This may suggest that TCP efficiently uses the recent improvements such as SACK, fast recovery, and improved RTT estimation to recover from large loss bursts.

The plots of the measured loss burst histograms in Figure 9 roughly match the geometric distribution. This fits recent studies [10], [11] that argue that Internet packet losses can be modeled by loss episodes whose length can be approximated using the geometric distribution. The above results encourage us to revisit the common drop-tail modeling assumption, used in many TCP throughput

studies [12], [13], [14], [15] that incorporate the effect of the TD and TO loss indications.

In the next section we analytically derive the probability that the loss indication is a time-out considering the Bernoulli loss model which may better capture the current loss patterns in the Internet. The Bernoulli loss model, which is the most fundamental model to produce the geometric distribution, may represent the deployment of Internet buffer management algorithms such as Random Early Detection (RED) [27], which drop packets uniformly at random during congestion periods.

A. Loss Indication Analysis

Our analysis is conducted within the well-known framework of [12]. We preserve the notations and the relevant assumptions of this model: we assume that packets are send in rounds, and a packet is lost in a round independently of any packet loss in other rounds. However, we replace the original drop-tail loss assumption with the Bernoulli loss assumption: each packet in a round is dropped with probability p independently of others.

Our goal is to analytically derive, Q, the probability that a loss indication ending a TD period is a time-out (TO), where the TD period is the period between two loss indications. For this purpose we examine the round at which a loss indication occurs, which is refereed to as the 'penultimate' round. Let w be the congestion window size at this round.

As shown in [12] a TO would occur if the number of packets in the penultimate round is less than or equal to three, or that the number of packets successfully delivered in the last round is less than three. Therefore, $\hat{Q}(w)$, the probability that a loss is a TO as a function of w is given by:

$$\hat{Q}(w) = \begin{cases} 1 & \text{if } w \leq 3 \\ \frac{\sum_{k=0}^{2} B(w,k) + \sum_{k=3}^{w} B(w,k) \sum_{m=0}^{2} B(k,m)}{1-(1-p)^w} & \text{o.w.} \end{cases} \quad (3)$$

Where $B(w,k)$ is the probability that k packets are ACKed in a round of w packets, and the quantity in the denumerator is due to the condition that there is at least one loss in the round. In the Bernoulli loss model $B(w,k) = \binom{w}{k} p^{w-k}(1-p)^k$.

After algebraic manipulation we get the following bound for $w > 3$

$$\hat{Q}(w) \leq \frac{\sum_{k=0}^{2} B(w,k)(1 + (1-p)^k(-1 + (2-p)^w))}{1-(1-p)^w} \quad (4)$$

To derive an approximation of $\hat{Q}$, we may use L'Hopital's rule when $p \to 0$, as done in [12], and get that $Q \approx \min\{1, \frac{3}{E[W]}\}$.

Alternatively, we use a more accurate approximation that considers TCP's congestion window size, W, which is assumed to be uniformly distributed on the discrete interval $[0, w_{max}]$. The probability that a loss event is TO is calculated using the taylor series expansion about the point $p \to 0$

$$\begin{aligned} Q &= \textstyle\sum_{w=1}^{w_{max}} \hat{Q}(w) P[W=w] \\ &\approx \min\{1, \tfrac{1}{w_{max}}(6 + 96p - 32p^2 + o(p^3))\} \end{aligned} \quad (5)$$

Given the sawtooth behavior of TCP the average congestion window size can be approximated in steady state to $\frac{3}{4}$ of the maximal value of the congestion window [28], and thus we assign $E[W] = \frac{3}{4} w_{max}$. Using this assumption and considering small values of p, Q can be approximated

$$Q \approx \min\{1, \frac{4.5}{E[W]}\} \quad (6)$$

Observe that both approximation methods yield a similar result (different only by a multiplicative factor), which closely matches the time-out probability in the drop-tail model [12], $Q \approx \min\{1, \frac{3}{E[W]}\}$. This implies that the behavior of the TCP loss process is similar under both models, and that the TCP throughput formulas in [12], [13], [14], [15] can also be used to characterize TCP's behavior for the Bernoulli loss model.

V. Conclusions

In this study we address the issue of identifying the low performance network links from passively collected TCP traffic. We propose a root cause criterion that reduces the complexity of identifying low-performance links and consequently develop a comprehensive solution for the problem. Our solution includes both a cost-effective identification algorithm and a simple packet classification algorithm that infer the various causes of TCP sequencing problems such as packet loss, reordering and duplication.

We find that the identification algorithm is very efficient in detecting the lossy and reordered links that obey the root cause criterion. For lossy links, it typically achieves a high detection rate above 95% and a false detection rate below 10%. Furthermore, we show that our method is able to estimate the true loss and reorder rates of the network internal links up to a constant bias, and present scenarios for which the loss estimator tends to its real value. Applying our algorithms to Internet samples

gathered at an edge ISP we find that the majority of the lossy links are inter-ISP links.

To derive the loss and reorder rates of the observed sessions we develop a simple methodology that infers and classifies the observed out-of-sequence packets. A novelty of our packet classification technique is that it requires only one direction of the TCP connection, and thus can be applied to asymmetric TCP flows. Using our Internet samples we find that packet loss is significantly more frequent than packet reordering and duplication.

Another aspect of our study includes TCP's temporal loss process. We found that the burst loss size is geometrically distributed. We then analyze the TCP time-out loss indication under the Bernoulli loss model, which is the simplest model that can cause a geometric distribution, and show that the behavior of the TCP loss process is similar under both model. This implies that the various TCP throughput formula are applicable to both models, and thus can be used to characterize TCP's behavior in the presence of the two major queuing disciplines used by modern routers, tail-drop and RED.

REFERENCES

[1] R. Caceres, N. G. Duffield, J. Horowitz, D. F. Towsley, and T. Bu, "Multicast-based inference of network-internal characteristics: Accuracy of packet loss estimation," in *IEEE INFOCOM*, New-York, NY, USA, 1999.

[2] N. G. Duffield, F. L. Presti, V. Paxson, and D. F. Towsley, "Inferring link loss using striped unicast probes," in *IEEE INFOCOM*, Anchorage, Alaska, USA, Apr. 2001.

[3] S. Ratnasamy and S. McCanne, "Inference of multicast routing trees and bottleneck bandwidths using end-to-end measurements," in *IEEE INFOCOM*, New-York, NY, USA, 1999.

[4] A. B. Downey, "Using pathchar to estimate internet link characteristics," in *ACM SIGCOMM*, 1999.

[5] D. Rubenstein, J. F. Kurose, and D. F. Towsley, "Detecting shared congestion of flows via end-to-end measurement," *IEEE/ACM Transactions on Networking*, vol. 10, no. 3, pp. 381–395, 2002.

[6] D. Katabi, I. Bazzi, and X. Yang, "A passive approach for detecting shared bottlenecks," in *IEEE ICCCN*, October 2001.

[7] V. Padmanabhan, L. Qiu, and H. Wang, "Server-based inference of internet performance," in *IEEE INFOCOM*, San Francisco, CA, USA, April 2003.

[8] 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," in *IEEE INFOCOM*, San Francisco, CA, USA, April 2003.

[9] J. Bellardo and S. Savage, "Measuring packet reordering," in *ACM/USENIX Internet Measurement Workshop*, Marseille, France, Nov. 2002.

[10] Y. Zhang, V. Paxson, and S. Shenker, "The stationarity of internet path properties," in *ACIRI Technical Report*, May 2000.

[11] V. Paxson, "End-to-end Internet packet dynamics," *IEEE/ACM Transactions on Networking*, vol. 7, no. 3, pp. 277–292, 1999.

[12] J. Padhye, V. Firoiu, D. Towsley, and J. Krusoe, "Modeling TCP throughput: A simple model and its empirical validation," in *ACM SIGCOMM*, Sept. 1998.

[13] E. Altman, K. Avrachenkov, and C. Barakat, "A stochastic model of TCP/IP with stationary random losses," in *ACM SIGCOMM*, Sept. 2000.

[14] N. Cardwell, S. Savage, and T. Anderson, "Modeling TCP latency," in *IEEE INFOCOM*, Tel-Aviv, Israel, Mar. 2000.

[15] B. Sikdar, S. Kalyanaraman, and K. S. Vastola, "An integrated model for the latency and steady-state throughput of TCP connections," *Performance Evaluation*, vol. 46, no. 2-3, pp. 139–154, 2001.

[16] A. Gurtov and R. Ludwig, "Responding to spurious timeouts in TCP," in *IEEE INFOCOM*, April 2003.

[17] M. Allman, W. M. Eddy, and S. Ostermann, "Estimating loss rates with TCP," *ACM Performance Evaluation Review*, vol. 31, no. 3, pp. 12–24, Dec. 2003.

[18] P. Benko and A. Veres, "A passive method for estimating end-to-end TCP packet loss," in *IEEE GLOBECOM*, Taipei, Taiwan, 2002.

[19] Y. Zhang, N. Duffield, V. Paxson, and S. Shenker, "On the constancy of internet path properties," in *ACM SIGCOMM Internet Measurement Workshop*, Nov. 2001.

[20] Y. Tsang, M. Coates, and R. Nowak, "Passive unicast network tomography based on TCP monitoring," Rice University, ECE Dept., Tech. Rep. TR-0005, Nov. 2000.

[21] M. Laor and L. Gendel, "The effect of packet reordering in a backbone link on application throughput," *IEEE Network*, vol. 16, no. 5, pp. 28–36, Sept./Oct. 2002.

[22] J. C. Bennett, C. Partridge, and N. Schectman, "Packet reordering is not pathological network behaviour," *IEEE/ACM Transactions on Networking*, vol. 6, no. 7, pp. 789–798, 2000.

[23] S. B. Moon, "Measurement and analysis of end-to-end delay and loss in the internet," Ph.D. dissertation, Dept. of CS, University of Massachusetts at Amherst, Feb. 2000.

[24] W. R. Stevens, *TCP/IP Illustrated*. MA: Addison-Wesley, Nov. 1994, vol. 1.

[25] P. Barford and J. Sommers, "A comparison of active and passive methods for measuring packet loss," *University of Wisconsin-Madison Technical Report*, October 2002.

[26] S. Jaiswal, G. Iannaccone, C. Diot, J. Kurose, and D. Towsley, "Inferring TCP connection characteristics through passive measurements," in *IEEE INFOCOM*, Hong-Kong, 2004.

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

[28] J. F. Kurose and W. R. Ross, *Computer Networking: A Top-Down Approach Featuring the Internet*. Addison-Wesley, 2000.