On the Constancy of Internet Path Properties

Yin Zhang, Nick Duffield, Vern Paxson, Scott Shenker

Abstract— **Many Internet protocols and operational procedures use measurements to guide future actions. This is an effective strategy if the quantities being measured exhibit a degree of *constancy*: that is, in some fundamental sense, they are not changing. In this paper we explore three different notions of constancy: mathematical, operational, and predictive. Using a large measurement dataset gathered from the NIMI infrastructure, we then apply these notions to three Internet path properties: loss, delay, and throughput. Our aim is to provide guidance as to when assumptions of various forms of constancy are sound, versus when they might prove misleading.**

I. INTRODUCTION

There has been a recent surge of interest in network measurements. These measurements have deepened our understanding of network behavior and led to more accurate and qualitatively different mathematical models of network traffic. Network measurements are also used in an operational sense by various protocols to monitor their current level of performance and take action when major changes are detected. For instance, RLM [MJV96] monitors the packet loss rate and, if it crosses some threshold, decreases its transmission rate. In addition, several network protocols and algorithms use network measurements to predict future behavior; TCP uses delay measurements to estimate when it should time-out missing packets, and measurement-based admission control algorithms use measures of past load to predict future loads.

Measurements are inherently bound to the present—they can merely report the state of the network at the time of the measurement. However, measurements are most valuable when they are a useful guide to the future; this occurs when the relevant network properties exhibit what we will term *constancy*. We use a new term for this notion, rather than an existing term like "stationarity," in an attempt to convey our goal of examining a broad, general view of the property "holds steady and does not change," rather than a specific mathematical or modeling view. We will also use the term *steady* for the same notion, when use of "constancy" would prove grammatically awkward.

Y. Zhang and N. Duffield are with AT&T Labs—Research, Florham Park, NJ. Email: {yzhang,duffield}@research.att.com.

V. Paxson and S. Shenker are with the AT&T Center for Internet Research at ICSI (ACIRI), International Computer Science Institute, Berkeley, CA. Email: {vern,shenker}@aciri.org.

In this paper we investigate three notions of constancy: mathematical, operational, and predictive. We do so in the context of measurements of three quantities describing Internet paths: packet loss, packet delays, and throughput.

We say that a dataset of network measurements is *mathematically steady* if it can be described with a single time-invariant mathematical model. The simplest such example is describing the dataset using a single independent and identically distributed (IID) random variable. More complicated forms of constancy would involve correlations between the data points. More generally, if one posits that the dataset is well-described by some model with a certain set of parameters, then mathematical constancy is the statement that the dataset is consistent with that set of parameters throughout the dataset.

One example of mathematical constancy is the finding by Floyd and Paxson [PF95] that session arrivals are well described by a fixed-rate Poisson process over time scales of tens of minutes to an hour. However, they also found that session arrivals on longer time scales can only be well-modeled using Poisson processes if the rate parameter is adjusted to reflect diurnal load patterns, an example of mathematical *non-constancy*.

When analyzing mathematical constancy, the key is to find the appropriate model. Inappropriate models can lead to misleading claims of non-constancy because the model doesn't truly capture the process at hand. For instance, if one tried to fit a highly correlated but stationary arrival process to a Poisson model, it would appear that the Poisson arrival rate varied over time.

Testing for constancy of the underlying mathematical model is relevant for modeling purposes, but is often too severe a test for operational purposes because many mathematical non-constancies are in reality irrelevant to protocols. For instance, if the loss rate on a path was completely constant at 10% for thirty minutes, but then changed abruptly to 10.1% for the next thirty minutes, one would have to conclude that the loss dataset was not mathematically steady, since its fundamental parameter has changed; yet one would be hard-pressed to find an application that would care about such a change. Thus, one must adopt a different notion of constancy when addressing operational issues. The key criterion in operational, rather than mathematical, constancy is whether an application (or other operational entity) would care about the changes in the dataset. We will call a dataset *operationally steady* if the quantities of interest remain within bounds considered operationally equivalent. Note that while it is obvious that operational constancy does not imply mathematical constancy, it is also true that mathematical constancy does not imply operational constancy. For instance, if the loss process is a highly bimodal process with a high degree of correlation, but the loss rate in each mode does not change, nor does the transition probability from one mode to the other, then the process would be mathematically steady; but an application will see sharp transitions from low-loss to high-loss regimes and back which, from the application's perspective, is highly non-steady behavior.

Operational constancy involves changes (or the lack thereof) in perceived application performance. However, protocols and other network algorithms often make use of measurements on a finer level of granularity to predict future behavior. We will call a dataset *predictively steady* if past measurements allow one to reasonably predict future characteristics. As mentioned above, one can consider TCP's time-out calculation as using past delays to predict future delays, and measurement-based admission control algorithms do the same with loss and utilization. So unlike operational constancy, which concerns the degree to which the network remains in a particular operating regime, predictive constancy reflects the degree to which *changes* in path properties can be tracked.

Just as we can have operational constancy but not mathematical, or vice versa, we also can have predictive constancy and none or only one of the others, and vice versa. Indeed, as we will illustrate, processes exhibiting the simplest form of mathematical constancy, namely IID processes, are generally impossible to predict well, since there are no correlations in the process to leverage.

Another important point to consider is that for network behavior, we anticipate that constancy is a more useful concept for coarser time scales than for fine time scales. This is because the effects of numerous deterministic network mechanisms (media access, FIFO buffer drops, timer granularities, propagation delays) manifest themselves on fine time scales, often leading to abrupt shifts in behavior, rather than stochastic variations.

An important issue to then consider concerns different ways of how to look at our fine-grained measurements on scales more coarse than individual packets. One approach is to aggregate individual measurements into larger quantities, such as packets lost per second. This approach is quite useful, and we use it repeatedly in our study, but it is not ideal, since by aggregating we can lose insight into the underlying phenomena. An alternative approach is to attempt to *model* the fine-grained processes using a model that provides a form of aggregation. With this approach, if the model is sound, we can preserve the insight into the underlying phenomena because it is captured by the model.

For example, instead of analyzing packet loss per second, we show that individual loss events come in *episodes* of back-to-back losses (§ III-B). We can then separately analyze the characteristics of individual loss episodes versus the constancy of the process of loss episode arrivals, retaining the insight that loss events often come back-to-back, which would be diminished or lost if we instead went directly to analyzing packets lost per second.

Our basic model for various time series is of piecewise steady regions delineated by *change-points*. With a parameterized family of models (e.g. Poisson processes with some rate), the time series in each change-free region (CFR) is modeled through a particular value of the parameter (e.g., the Poisson arrival rate). In fitting the time series to this model, we first identify the change-points. Within each CFR we determine whether the process can be modeled by IID processes. When occurring, independence can be viewed as a vindication of the approach to refocus to coarser time scales, showing the simplicity in modeling that can be achieved after removing small time scale correlations. Furthermore, we can test conformance of inter-event times with a Poisson model within each CFR. Given independence, this entails testing whether inter-event times follow an exponential distribution.

To focus on the network issues, we defer discussion of the statistical methodology for these tests—the presence of change-points, IID processes, and exponential inter-event times—to Appendix A. However, one important point to note is that the two tests we found in the literature for detecting change-points are not perfect. The first test—$CP/RankOrder$—is *biased* towards sometimes finding extraneous change-points. The effect of the bias is to underestimate the duration of steady regions in our datasets. The second test—$CP/Bootstrap$—does not have the bias. However, it is *less sensitive* and therefore misses actual change-points more often. The effect of the insensitivity is to overestimate the duration of steady regions and to underestimate the number of CFRs within which the underlying process can be modeled by IID processes. (See [Zh01] for a detailed assessment of the accuracy of both tests.) To accommodate the imperfection, we apply both tests whenever appropriate and then compare the results. Our hope is to give some bound on the duration of steady regions.

This paper is organized as follows. We first describe the sources of data in Section II. We discuss the loss data and its constancy analysis in Section III, and the delay and throughput data in Sections IV and V. Of these three sections, the first one is much more detailed, as we develop a number of our analysis and presentation techniques therein. We then conclude in Section VI with a brief summary of our results.

II. Measurement methodology

We gathered two basic types of measurements: Poisson packet streams, used to assess loss and delay characteristics, and TCP transfers to assess throughput.[1] Our measurements were all made using the NIMI measurement infrastructure [PMAM98]. NIMI is a follow-on to Paxson's NPD measurement framework, in which a number of measurement platforms are deployed across the Internet and used to perform end-to-end measurements, and it attempts to address the limitations and resulting measurement biases present in NPD [Pa99].

We took two main sets of data, one during Winter 1999–2000 ($\mathcal{W}_1$), and one during Winter 2000–2001 ($\mathcal{W}_2$). For the first, the infrastructure consisted of 31 hosts, 80% of which were located in the United States, and for the second, 49 hosts, 73% in the USA. About half are university sites, and most of the remainder research institutes of different kinds. Thus, the connectivity between the sites is strongly biased towards conditions in the USA, and is likely not representative of the commercial Internet in the large. That said, the paths between the sites do traverse the commercial Internet fairly often, and we might plausibly argue that our observations could apply fairly well to the better connected commercial Internet of the not-too-distant future, if not today.

For Poisson packet streams we used the "`zing`" utility, provided with the NIMI infrastructure, to source UDP packets at a mean rate of 10 Hz ($\mathcal{W}_1$) or 20 Hz ($\mathcal{W}_2$). For the first of these, we used 256 byte payloads, and for the second, 64 byte payloads. `zing` sends packets in selectable patterns (payload size, number of packets in back-to-back "flights," distribution of flight interarrivals), recording time of transmission and reception. While `zing` is capable of using a packet filter to gather kernel-level timestamps, for a variety of logistical problems this option does not work well on the current NIMI infrastructure, so we used user-level timestamps.

[1] See [ZPS00] for related analysis of end-to-end routing based on `traceroute` measurements.

Dataset	# pkt traces	# pairs	# pkts	# thruput	# xfers
$\mathcal{W}_1$	2,375	244	160M	58	16,900
$\mathcal{W}_2$	1,602	670	113M	111	31,700

TABLE I
Summary of datasets used in the study.

By using Poisson intervals for sending the packets, time averages computed from the measurements are unbiased [Wo82]. Packets were sent for an hour between random pairs of NIMI hosts, and were recorded at both sender and receiver, with some streams being unidirectional and some bidirectional. We used the former to assess patterns of one-way packet loss based on the unique sequence number present in each `zing` packet, and the latter to assess both one-way loss and round-trip delay. We did not undertake any one-way delay analysis since the NIMI infrastructure does not provide synchronized clocks.

For throughput measurements we used TCP transfers between random pairs of NIMI hosts, making a 1 MB transfer between the same pair of hosts every minute for a 5-hour period. We took as the total elapsed time of the transfer the interval observed at the receiver between accepting the TCP connection and completing the close of the connection. Transfers were made specifying 200 KB TCP windows, though some of the systems clamped the buffers at 64 KB because the systems were configured to not activate the TCP window scaling option [JBB92]. The NIMI hosts all ran versions of either FreeBSD or NetBSD.

Table I summarizes the datasets. The second column gives the number of hour-long `zing` packet traces, the third the number of distinct pairs of NIMI hosts we measured (lower in $\mathcal{W}_1$ because we paired some of the hosts in $\mathcal{W}_1$ for an entire day, while all of the $\mathcal{W}_2$ measurements were made between hosts paired for one hour), and the total number of measured packets. The fifth column gives the number of throughput pairs we measured, each for 5 hours, and the corresponding number of 1 MB transfers we recorded.

In our preliminary analysis of $\mathcal{W}_1$, we discovered a deficiency of `zing` that biases our results somewhat: if the `zing` utility received a "No route to host" error condition, then it terminated. This means that if there was a significant connectivity outage that resulted in the `zing` host receiving an ICMP unreachable message, then `zing` stopped running at that point, and we missed a chance to further measure the problematic conditions. 47 of the $\mathcal{W}_1$ measurement hours (4%) suffered from this problem. We were able to salvage 6 as containing enough data to still warrant analysis; the others we rejected, though some would have been rejected anyway due to NIMI coordination problems. This omission means that the $\mathcal{W}_1$ data is, regrettably, biased towards underestimating significant network problems, and how they correlate with non-constancies. This problem was fixed prior to the $\mathcal{W}_2$ data collection.

One other anomaly in the measurements is that in $\mathcal{W}_2$ some of the senders and receivers were missynchronized, such that they were not running together for the entire hour. This mismatch could lead to series of packets at the beginning or ending of traces being reported as lost when in fact the problem was that the receiver was not running. We removed the anomaly by trimming the traces to begin with the first successfully received packet and end with the last such. This trimming potentially could bias our data towards underestimating loss outages; however, inspection of the traces and the loss statistics with and without the trimming convinced us that the bias is quite minor.

Finally, our focus in this paper is on constancy, but to soundly assess constancy first requires substantial work to detect pathologies and modal behavior in the data and, depending on their impact, factor these out. We then can identify quantities that are most appropriate to test for constancy. Due to space restrictions and in the interest of brevity, we refer the reader to [ZPS00] for many of the particulars of this assessment of the data.

III. Loss constancy

We begin our analysis of types of constancy with a look at packet loss. We devote significantly more discussion to this section than to the subsequent sections analyzing delay and throughput because herein we develop a number of our analysis and presentation techniques.

Correlation in packet loss was previously studied in [Bo93], [Pa99], [YMKT99]. The first two of these focus on conditional loss probabilities of UDP packets and TCP data/ACK packets. [Bo93] found that for packets sent with a spacing of $\leq$ 200ms, a packet was much more likely to be lost if the previous packet was lost, too. [Pa99] found that for consecutive TCP packets, the second packet was likewise much more likely to be lost if the first one was. The studies did not investigate correlations on larger time scales than consecutive packets, however. [YMKT99] looked at the autocorrelation of a binary time series representation of the loss process observed in 128 hours of unicast and multicast packet traces. They found correlation time scales of 1000 ms or less. However, they also note that their approach tends to underestimate the correlation time scale.

While the focus of these studies was different from ours—in particular, [YMKT99] explicitly discarded non-steady samples—some of our results bear directly upon this previous work. In particular, in this section we verify the finding of correlations in the loss process, but also find that much of the correlation comes only from back-to-back loss episodes, and not from "nearby" losses. This in turn suggests that congestion epochs (times when router buffers are running nearly completely full) are quite short-lived, at least for paths that are not heavily congested.

As discussed in the previous section, we measured a large volume (270M) of Poisson packets sent between several hundred pairs of NIMI hosts, yielding binary-valued time series indexed by sending time and indicating whether each packet arrived at the receiver or failed to do so. For this analysis, we considered packets that arrived but with bad checksums as lost.

There were two artifacts in the data that we had to explicitly adjust for. First, as detailed in [ZPS00], one of the sites exhibited strong 60-second periodicities in its losses. As we did not find such periodicities for any of the other sites, we removed these traces from our analysis as anomalous. Second, if a packet was *replicated* by the network such that multiple copies arrived at the receiver, we treated this as a single arrival, discarding the late arrivals. In general, we found packet replication very rare, but in one trace 16% of the packets arrived twice.

Packet loss in the datasets was in general low. Over all of $\mathcal{W}_1$, 0.87% of the packets were lost, and for $\mathcal{W}_2$, 0.60%. However, as is common with Internet behavior, we find a wide range: 11–15% of the traces experienced no loss; 47–52% had some loss, but at a rate of 0.1% or less; 21–24% had loss rates of 0.1–1.0%; 12–15% had loss rates of 1.0–10%; and 0.5–1% had loss rates exceeding 10%.

Because we sourced traffic in both directions during our measurement runs, the data affords us with an opportunity to assess symmetries in loss rates. We find for $\mathcal{W}_1$ that, similar to as reported in [Pa99], loss rates in a path's two directions are only weakly correlated, with a coefficient of correlation of 0.10 for the 70% of traces that suffered some loss in both directions. However, the logarithms of the loss rates are strongly correlated (0.53), indicating that the order of magnitude of the loss rate is indeed fairly symmetric. While time-of-day and geographic (trans-continental versus intra-USA) effects contribute to the correlation, it remains present to a degree even with those effects removed. For $\mathcal{W}_2$, the effect is weaker: the coefficient of correlation is -0.01, and for the logarithm of the loss rate, 0.23.

A. Individual loss vs. loss episodes

Previously we discussed how an investigation of mathematical constancy should incorporate looking for a good

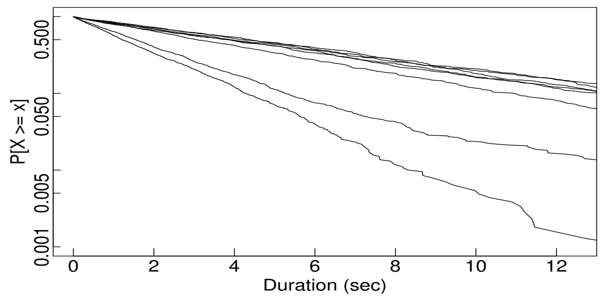

Fig. 1. Example log-complementary distribution function plot of duration of loss-free runs.

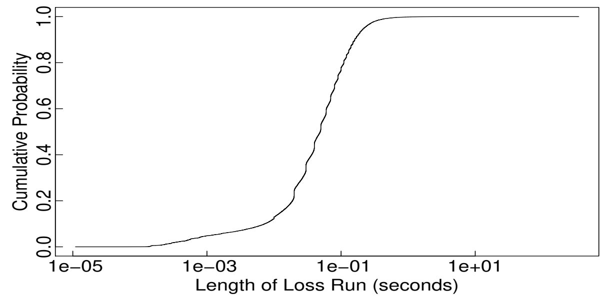

Fig. 2. Distribution of loss run durations.

model. In this section, we apply this principle to understanding the constancy of packet loss processes.

The traditional approach for studying packet loss is to examine the behavior of individual losses [Bo93], [Mu94], [Pa99], [YMKT99]. These studies found correlation at time scales below 200–1000 ms, and left open the question of independence at larger time scales. We introduce a simple refinement to such characterizations that allows us to identify these correlations as due to back-to-back loss rather than "nearby" loss, and we relate the result to the extended Gilbert loss model family [Gi60], [SCK00], [JS00]. We do so by considering not the loss process itself, but the loss *episode* process, i.e., the time series indicating when a series of consecutive packets (possibly only of length one) were lost.

For loss processes, we expect congestion-induced events to be clustered in time, so to assess independence among events, we use the autocorrelation-based Box-Ljung test developed in § A-B, as it is sensitive to near-term correlations. We chose the maximum lag k to be 10, sufficient for us to study the correlation at fine time scales. Moreover, to simplify the analysis, we use lag in packets instead of time when computing autocorrelations.

We first revisit the question of loss correlations as already addressed in the literature. In $\mathcal{W}_1$, for example, we examined a total of 2,168 traces, 265 of which has no loss at all. In the remaining 1,903 traces, only 27% are considered IID at 5% significance using the Box-Ljung Q statistic. The remaining traces show significant correlations at lags under 10, corresponding to time scales of 500–1000 ms, consistent with the findings in the literature.

These correlations imply that the loss process is not IID. We now consider an alternative possibility, that the loss *episode* process is IID, and, furthermore, is well modeled as a Poisson process. We again use Box-Ljung to test the hypothesis. Among the 1,903 traces with at least one loss episode, 64% are considered IID, significantly larger than the 27% for the loss process. Moreover, of the 1,380 traces classified as non-IID for the loss process, half have IID loss episode processes. In contrast, only 1% of the traces classified as IID for the loss process are classified as non-IID for the loss episode process.

Figure 1 illustrates the Poisson nature of the loss episode process for eight different datasets measured for the same host pair. The X-axis gives the length of the loss-free periods in each trace, which is essentially the loss episode interarrival time, since nearly all loss episodes consist of only one lost packet. The Y-axis gives the probability of observing a loss-free period of a given length or more, i.e., the complementary distribution function. Since the Y-axis is log-scaled, a straight line on this plot corresponds to an exponential distribution. Clearly, the loss episode interarrivals for each trace are consistent with exponential distributions, even though the mean loss episode rate in the traces varies from 0.8%–2.7%, and this in turn argues strongly for Poisson loss episode arrivals.

If we increase the maximum lag in the Box-Ljung test to 100, the proportion of traces with IID loss processes drops slightly to 25%, while those with IID loss episodes falls to 55%. The decline illustrates that there is some non-negligible correlation over times scales of a few seconds, but even in its presence, the data becomes significantly better modeled as independent if we consider loss episodes rather than losses themselves.

If we continue out to still larger time scales, above roughly 10 sec, then we find exponential distributions become a considerably poorer fit for loss episode interarrivals; this effect is widespread across the traces. It does not, however, indicate correlations on time scales of 10's of seconds (which in fact we generally find are absent), but rather mixtures of exponentials arising from differing loss rates present at different parts of a trace, as discussed below. Note that, were we not open to considering a loss of constancy on these time scales, we might instead wind up misattributing the failure to fit to an exponential distribution as evidence of the need for a more complex, but steady, process.

All in all, these findings argue that in many cases the fine time scale correlation reported in the previous studies is caused by trains of consecutive losses, rather than intervals over which loss rates become elevated and "nearby" but not consecutive packets are lost. Therefore, loss processes are better thought of as spikes during which there's a short-term outage, rather than epochs over which a congested router's buffer remains perilously full. A spike-like loss process accords with the Gilbert model [Gi60], which postulates that loss occurs according to a two-state process, where the states are either "packets not lost" or "packets lost," though see below for necessary refinements to this model.

A related finding concerns the size of loss runs. Figure 2 shows the distribution of the duration of loss runs as measured in seconds. We see that virtually all of the runs are very short-lived (95% are 220 ms or shorter), and in fact near the limit of what our 20 Hz measurements can resolve. Similarly, we find that loss run sizes are uncorrelated according to Box-Ljung. We also confirm the finding in [YMKT99] that loss run lengths in packets often are well approximated by geometric distributions, in accordance with the Gilbert model, though the larger loss runs do not fit this description, nor do traces with higher loss rates ($> 1\%$); see below.

B. Mathematical constancy of the loss episode process

While in the previous section we homed in on understanding loss from the perspective of looking at loss episodes rather than individual loss, we also had the finding that on longer time scales, the loss episode rates appear to changing, i.e., *non-constancy*.

To assess the constancy of the loss episode process, we apply change-point analysis to the binary time series $\langle T_i, E_i \rangle$, where T_i is the time of the ith observation and E_i is an indicator variable taking the value 1 if a loss episode began at that time, 0 otherwise. In constructing this time series, note that we collapse loss episodes *and* the non-lost packet that follows them into a single point in the time series. (For example, if the original binary loss series is: $0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0$, then the corresponding loss episode series is: $0, 0, 1, 1, 0, 1, 0, 0$.) I.e., $\langle T_{i+1}, E_{i+1} \rangle$ reflects the observation of the second packet after the ith loss episode ended. We do this collapsing because if the series included the observation of the *first* packet after the loss episode, then E_{i+1} would always be 0, since episodes are always ended by a non-lost packet, and we would thus introduce a negative correlational bias into the time series.

Using the methodology developed in § A-A, we then divide each trace up into 1 or more change-free regions (CFRs), during which the loss episode rate appears well-modeled as steady. Figure 3 shows the cumulative distribution function (CDF) for the size of the largest CFR found for each trace in $\mathcal{W}_1$ (solid) and $\mathcal{W}_2$ (dashed). We also plot CDFs restricted to just those traces for which the overall loss rate exceeded 1% ("Lossy traces"). We see that more than half the traces are steady over the full hour. Of the remainder, the largest period of constancy runs the whole gamut from just a few minutes long to nearly the full hour. However, the situation changes significantly for lossy traces, with half of the traces having no CFR longer than 20 minutes for $\mathcal{CP}/RankOrder$ (or 30 minutes for $\mathcal{CP}/Bootstrap$). The behavior is clearly the same for both datasets. Meanwhile, the difference between the results for $\mathcal{CP}/RankOrder$ and those for $\mathcal{CP}/Bootstrap$ is also relatively small—about 10-20% more traces are change-free over the entire hour with $\mathcal{CP}/Bootstrap$ than with $\mathcal{CP}/RankOrder$. This suggests the effect of the bias/insensitivity is not major.

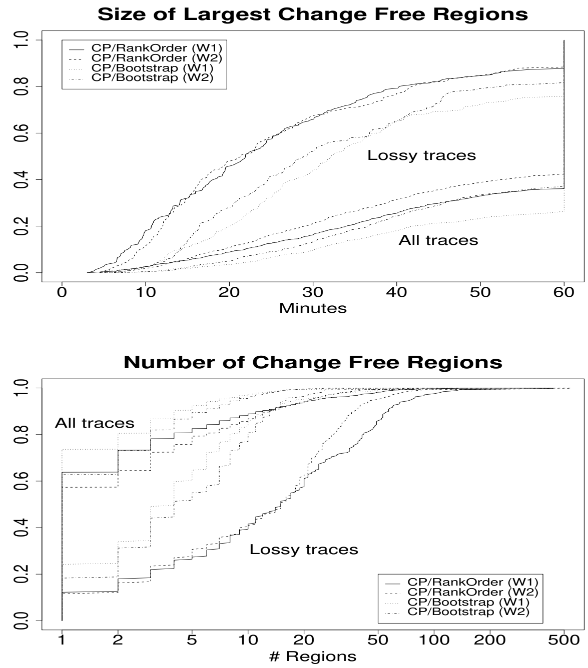

Fig. 3. CDF of largest change-free region (CFR) for loss episodes in $\mathcal{W}_1$ and $\mathcal{W}_2$ datasets, and number of CFRs present. "Lossy traces" is the same analysis restricted to traces for which the overall loss rate exceeded 1%.

We also analyzed the CDFs of the CFR sizes weighted to reflect the proportion of the trace they occupied. For example, a trace with one 10-minute CFR and one 50-minute CFR would be weighted as $\frac{1}{6}10 + \frac{5}{6}50 = 43.3$ minutes, meaning that if we pick a random point in a trace, we will on average land in a CFR of 43.3 minutes total duration. The CDFs for the weighted CFRs have shapes quite similar to those shown above, but shifted to the left about 7 minutes, except for the 60-minute spike on the righthand side, which of course does not change because its weight is 1.

The bottom half of the figure shows the distribution of the number of CFRs per trace. Again, the two datasets

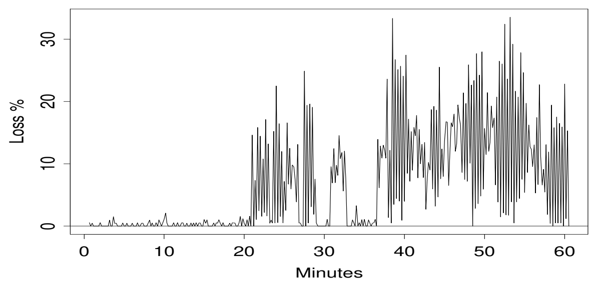

Fig. 4. Example of a trace with hundreds of change-free regions.

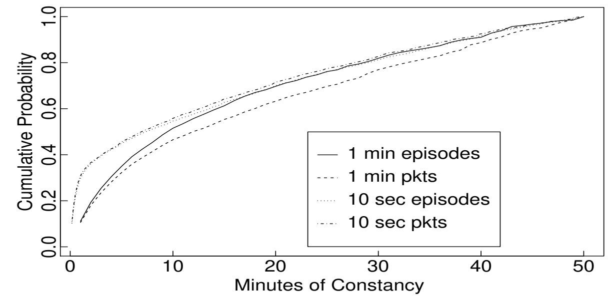

Fig. 5. Operational constancy for packet loss and loss episodes, conditioned on the constancy lasting 50 minutes or less.

agree closely. Over all the traces there are usually just a handful of CFRs, but for lossy traces the figure is much larger, with the average rising from around 5 over all traces to around 20 over the lossy traces. Clearly, once we are in a high-loss regime, we also are in a regime of changing conditions. In addition, sometimes we observe a huge number of CFRs. Figure 4 shows an example of the latter, a trace whose loss episode process divides into more than 400 CFRs.

Once we have divided traces into one or more CFRs, we can then analyze each region separate from the others, having confidence that within the region the overall loss episode rate does not change. Upon applying the Box-Ljung test, we find that 88-92% of the regions are consistent with an absence of lag 1 correlation, and 77-86% are consistent with no correlation up to lag 100. Clearly, within a CFR the loss episode process is well modeled as IID better than over the entire trace (previous section). In addition, applying the Anderson-Darling test (§ A-C) to the interarrivals between loss episodes in a region, we find that 77-85% of the regions are consistent with exponential interarrivals.

Together, these findings solidly support modeling loss episodes as homogeneous Poisson processes within change-free regions. In particular, correlations in loss processes are due to the presence of consecutive losses, rather than nearby losses.

It remains to describe the structure of loss episodes. We do so in the context of the aforementioned Gilbert and extended Gilbert models. For the two-state Gilbert model to hold, we should find that within a loss episode the probability of observing each additional loss remains the same. In particular, the probability that we observe a 2nd loss in an episode, given that we've seen the initial loss of an episode, should be the same as the probability of observing a 3rd loss given that we've seen the 2nd loss. Similarly, the extended Gilbert model allows for k different loss rates for the first k losses after the initial loss, each corresponding to a different state in the model.

Accordingly, we can assess whether k states suffice to describe a given loss process by seeing whether the $k+1$ loss after the initial loss occurs (conditioned on the kth loss) with the same probability as the kth loss does (conditioned on the $k-1$ loss). We made these tests using Fisher's Exact Test [Ri95], and found that, for both $\mathcal{W}_1$ and $\mathcal{W}_2$, 40% of the traces are consistent with Bernoulli loss; 89% with the Gilbert two-state model; 98% with 3 states (extended Gilbert); and 99% with 4 states. However, the models work less well for lossy traces: only 6% are well-modeled as Bernoulli, 68% with 2 states, 85% with 3 states, and 96% with 4 states.

C. *Operational constancy of loss rate*

We now turn to analyzing a different notion of loss rate constancy, namely from an *operational* viewpoint. To do so, we partition loss rates into the following categories: 0–0.5%, 0.5–2%, 2–5%, 5–10%, 10–20%, and 20+%. The role of these categories is to capture qualitative notions such as "no loss," "minor loss," "tolerable loss," "serious loss," "very serious loss," and "unacceptable loss."

For each trace we then analyze how long the loss rate remained in the same category. Figure 5 plots the weighted CDF for four different loss series associated with each trace in $\mathcal{W}_1$: the loss episode rate computed over 1-minute intervals, the raw packet loss rate over 1-minute intervals, and the same but computed over 10-second intervals. (Results for $\mathcal{W}_2$ are virtually identical.) The CDF is weighted by the size of the constancy interval, as mentioned above; thus, we interpret the plot as showing the unconditional probability that at any given moment we would find ourselves in a constancy interval of duration T or less. For example, about 50% of the time we will find ourselves in a constancy interval of 10 min or less, if what we care about is the constancy of loss episodes computed over minute-long intervals (solid line).

An important point is that we truncated the plot to only show the distribution of intervals 50 minutes or less. We characterize longer intervals separately, as these reflect entire datasets that were operationally steady. Since our datasets spanned at most one hour, constancy over the whole dataset provides a lower bound on the duration of constancy, rather than an exact value, and hence differs from the distributions in Figure 5.

For the four loss series, the corresponding probabilities of observing a constancy interval of 50 or more minutes are 71%, 57%, 25%, and 22%. Thus, if we only care about constancy of loss viewed over 1-minute periods, then about two-thirds (57–71%) of the time, we will find we are in a constancy period of at least an hour in duration—it could be quite a bit longer, as our measurements limited us to observing at most an hour of constancy.

We also see that the key difference between the 10 sec and 1 min results is the likelihood of being in a period of long constancy: it takes only a single 10-second change in loss rate to interrupt the hour-long interval, much more likely than a single 1-minute change. If we condition on being in a shorter period of constancy, then we find very similar curves. In particular, if we are not in a period of long-lived constancy, then, per the plot, we find that about half the time we are in a 10-minute interval or shorter, and there is not a great deal of difference in the duration of constancy, regardless of whether we consider one-minute or 10-second constancy, or loss runs or loss episodes.

Finally, we repeated this assessment using a set of cutpoints for the loss categories that fell in the middle of the above cutpoints (e.g., 3.5–7.5%), to test for possible binning effects in which some traces straddle a particular loss boundary. The results are highly similar.

D. Comparing mathematical and operational constancy

We now briefly assess the degree to which we find that the notion of mathematical constancy of loss coincides with the notion of operational constancy of loss. While there are many dimensions in which we could undertake such an assessment, we aim here to only explore the coarse-grained relationship.

We begin by categorizing each trace as either "steady" or "not steady," where the distinction between the two concerns whether the trace has a 20-minute region of constancy; i.e., for mathematical constancy, a 20-minute CFR for the rate of the loss episode process, and for operational constancy, a 20-minute period during which the loss rate did not stray outside one of the particular regions. We then assess what proportion of the traces were neither mathematically nor operationally steady ($\overline{MO}$), mathematically but not operationally ($M\overline{O}$), vice versa ($\overline{M}O$), and both (MO).

For operational constancy evaluated using loss computed over 1 min, we find $\overline{MO}$ = 6–9%, $M\overline{O}$ = 6–15%, $\overline{M}O$ = 2–5%, and MO = 74–83%. (The minor variation in the figures depends on whether for operational constancy we look at loss rate or loss episode rate, and whether we use the first or the second set of loss categories as discussed at the end of § III-C.) Clearly, the notions of mathematical and operational constancy mostly coincide.

However, if we instead evaluate operational constancy using loss rates computed over 10 sec intervals, the figures are significantly different: $\overline{MO}$ = 11%, $M\overline{O}$ = 37–45%, $\overline{M}O$ = 0.1%, and MO = 44–52%. We can summarize the difference as: *Operational constancy of packet loss coincides with mathematical constancy on large time scales such as viewing how loss changes from one minute to the next; but not nearly so well on medium time scales such as looking at 10-second intervals.*

E. Predictive constancy of loss rate

The last notion of packet loss constancy we explore is that of *predictive* constancy, i.e., to what degree can an estimator predict future loss events?

There are a number of different loss-related events we could be interested in predicting. Here, we confine ourselves to predicting the length of the next loss-free run. We chose this event for two reasons: first, we do not have to bin the time series (which predicting loss rate over the next T seconds would require); and second, there are known applications for such prediction, such as TFRC [FHPW00].

The next question is what type of estimator to use. We assess three different types popular in the literature: moving average (MA), exponentially-weighted moving average (EWMA) such as used by TCP [Ja88], and the S-shaped moving average estimator (SMA) used by TFRC. This last is a class of weighted moving average estimators that give higher weights to more recent samples; we use a subclass that gives equal weight to the most recent half of the window, and linearly decayed weights for the earlier half; see [FHPW00] for discussion.

For each of these estimators there is a parameter that governs the amount of memory of past events used by the estimator. For MA and SMA, we used window sizes of $2, 4, 8, 16, 32$; and for EWMA, $\alpha = 0.5, 0.25, 0.125$, and 0.01, where $\alpha = 0.5$ corresponds to weighting each new sample equally to the cumulative memory of previous samples, and $\alpha = 0.01$ weights the previous samples 99 times as much as each new sample.

Once we've defined what estimator to use, we next have to decide how to assess how well it performed. To do so, we compute:

$$\text{prediction error} = E\left[\ \left|\log\left(\frac{\text{predicted value}}{\text{actual value}}\right)\right|\ \right]$$

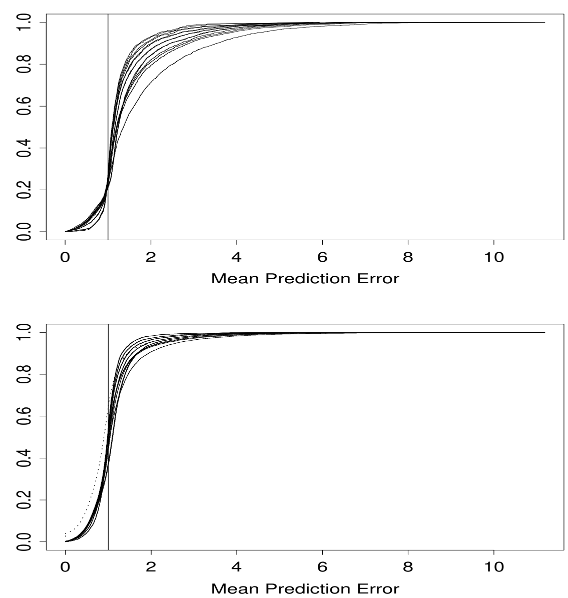

Fig. 6. CDFs of the mean error for a large number of loss predictors, computed over entire traces (top) or change-free regions (bottom).

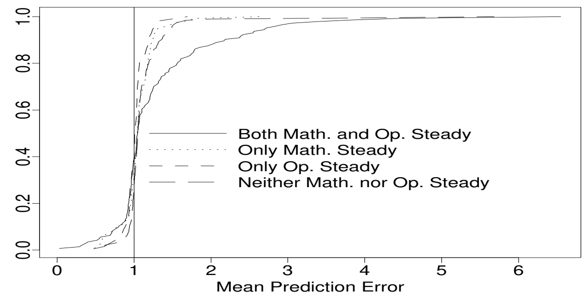

Fig. 7. CDFs of the mean error for EWMA ($\alpha = 0.25$) estimator computed over sets of lossy traces with different types of constancy.

where the expectation, which is computed over each of the events (loss-free runs) in a trace, reflects the ratio by which the estimator typically misses the target. We then compute CDFs that show the range of how well a given estimator performs over all of the traces.

Figure 6 shows the resulting CDFs, computed for all traces (top plot) and for all CFRs within the traces (bottom plot). The vertical line in each plot reflects a prediction error of 1, corresponding to overestimating or underestimating by a factor of e. (It turns out that the best one can achieve, on average, for predicting IID exponential random variables is a prediction error of 1.02.) We have plotted CDFs for all of the different estimators and sets of parameters, and the plot does not distinguish between them because the main point to consider is that virtually all of the estimators perform about the same—*the parameters don't matter, nor does the averaging scheme*.

We interpret this as reflecting that the process does not have significant structure to its short-range correlations that can be exploited better by particular types of predictors or window sizes; all that the estimators are doing is tracking the mean of the process, which varies more slowly than do the windows. There are two exceptions, however. First, in the top plot, the CDF markedly below all the others corresponds to EWMA with $\alpha = 0.01$. That estimator has a lengthy memory (on the order of 100 packets), and accordingly cannot adapt to rapid fluctuations in the loss process. In addition, that estimator will do particularly poorly during a transition between two CFRs, because it will remember the behavior in the older CFR for much longer than the other estimators. We see that in the lower plot, it fares better, because that plot does not include transitions between CFRs.

Also, in the second plot we have added an "oracular" estimator (dotted). This estimator knows the mean loss-free length during the CFR, and always predicts that value. We can see that it does noticeably better than the other estimators about half the time, and comparable the other half. A significant element of its improved performance is that the lower plot is heavily skewed to favoring estimators that do well over traces that are highly non-steady (many CFRs), because each of the CFRs will contribute a point to the CDF. The success of the oracle also suggests that it might be a good general strategy to construct estimators that include an explicit decision whether to restart the estimator, so they can adapt to level shifts in a nimble fashion.

Finally, we repeated the analysis after applying a random shuffle to the traces to remove their correlational structure. Doing so makes only a slight difference in the estimators' performance, reducing the discrepancy between the $\alpha = 0.01$ estimator and the others, and we find that the various estimators do only slightly worse than an oracular estimator applied to the now-IID time series.

We finish with a look at the relationship between how well we can predict loss versus the presence or absence of mathematical and/or operational constancy. As in § III-D, we aim only to understand the coarse-grained relationship, and again we consider a trace mathematically steady if it has a maximum CFR of at least 20 minutes, and operationally steady if it stays within a particular loss region for at least 20 minutes.

Partitioning the lossy ($\geq 1\%$ loss) traces on that criteria, using EWMA with $\alpha = 0.25$ we attain the predictor error CDFs shown in Figure 7. We see that the quality of the predictor is virtually unchanged if we have neither mathematical nor operational constancy, or just one of them. But if we have both, then the predictor's performance is

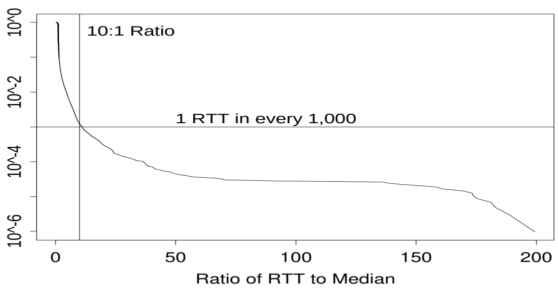

Fig. 8. Complementary distribution of the ratio of RTT samples to the median of their traces, computed for $\mathcal{W}_2$.

worse. This is because in this regime the loss episode process resembles an IID process without significant short-term variations, and the recent samples seen by the estimator provide no help in predicting the next event. In addition, note that if we look at all traces rather than just the lossy traces, the estimators again do worse, because for the type of event we are predicting (interval until the next loss episode), traces with low loss levels provide very few samples to the estimator. However, low loss is also a condition under which we generally won't care about the preciseness of the estimator, since loss events will be quite rare. In summary, *predictors do equally well whether or not we have other forms of constancy, unless we have constancy resembling an IID process with little short-term variation*.

IV. DELAY CONSTANCY

We next turn to exploring the types of constancy associated with packet delays. Mukherjee found that packet delay along several Internet paths was well-modeled using a shifted gamma distribution, but the parameters of the distribution varied from path to path and on time scales of hours [Mu94]. Similarly, Claffy and colleagues found that one-way delays measured along four Internet paths exhibited clear level shifts and non-constancies over the course of a day [CPB93].

For our analysis, we again use the `zing` Poisson packet streams measured on the NIMI hosts. Because the NIMI hosts lack synchronized clocks, we confine our analysis to those datasets with bidirectional packet streams. These are generated by `zing` on host A sending "request" packets host B, and the `zing` on host B immediately responding to each of these by sending back matching "reply" packets, facilitating round-trip measurement at host A. The delay in `zing`'s response is short, usually taking 100–200 μsec, occasionally rising to a few ms.

A. Delay "spikes"

The data totaled 130M RTT measurements made between 613 distinct pairs of hosts. In analyzing it, the first phenomenon we had to deal with is the presence of delay *spikes*. These are intervals (often quite short) of highly elevated RTTs. They are rare, but if unaddressed can seriously skew our analysis due to their magnitude. Figure 8 conveys the size and prevalence of spikes. For each trace, we computed the median of all of the RTT measurements, and then normalized each RTT measurement by dividing it by the median. This allows us to then plot all of the measurements together to assess, in high level terms, the magnitude of RTT variation present in the data. The plot shows the complementary distribution of the RTT-to-median ratio; this style of plot emphasizes the upper tail. For reference we have drawn lines reflecting a ratio of 10:1 (vertical) and a probability of 10^{-3} (horizontal). Clearly, there are a significant number of very large RTTs, though not so many that we would consider them anything other than an extreme upper-tail phenomenon.

To proceed with separating spikes from regular RTT behavior, we need to devise a definition for categorizing an RTT measurement as one or the other. We were unable to find a crisp modality to exploit—the only one present in the plot is for ratios above or below 100:1, but that cutoff point omits many spikes that we found visually—so we settled on the following imperfect procedure: for each new RTT measurement R', we compared it to the previous non-spike measurement, R. If $R' \geq \max(k \cdot R, 250\text{ms})$, then we consider the new measurement a spike; otherwise, we set $R \leftarrow R'$ and continue to the next measurement.[2] We then applied this classification for $k = 2$ and $k = 4$. Doing so revealed two anomalies: a high latency path plagued by rapid RTT fluctuations ranging from 200 ms to 1 sec, and a pair of hosts that periodically jumped their clocks. With the anomalies removed, we find that $k = 2$ categorized $1.1 \cdot 10^{-3}$ of the $\mathcal{W}_1$ RTTs as spikes, and $k = 4$ categorized $3.4 \cdot 10^{-4}$.

Once we had the definition in place, we could check it in terms of "yes, these are really outliers," as follows: for each trace we computed $\overline{x}$ and σ, the mean and standard deviation of the RTT measurements *with the spikes removed*. We then for each spike assessed how many σ it was above $\overline{x}$. For $\mathcal{W}_1$, the $k = 2$ definition leads to spikes that are typically (median) 16.9σ above the mean, with 80% being more than 5.6σ. For $k = 4$, these figures rise to 28σ and 6.6σ.

B. Constancy of body of RTT distribution

The degree to which RTT spikes are indeed outliers points up a need to assess the constancy of the body of

[2]We found the 250 ms lower bound necessary for applying the classifier to traces with quite low RTTs.

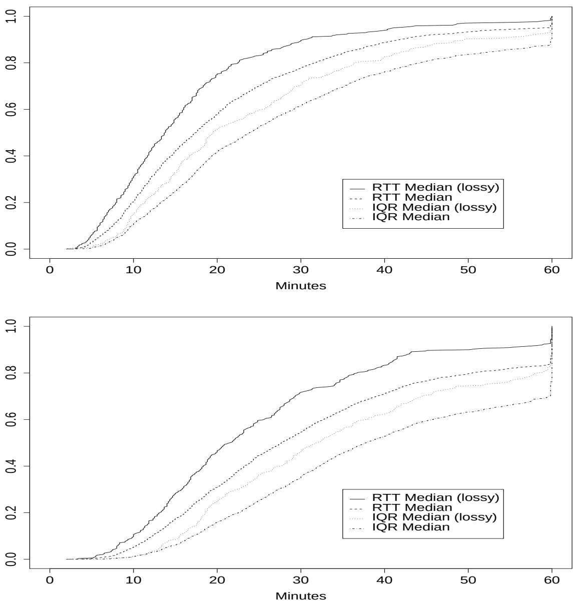

Fig. 9. CDF of largest CFR for median and IQR of packet RTT delays. "Lossy" is the same analysis restricted to traces for which the overall loss rate exceeded 1%.

the RTT distribution separate from that of the RTT spikes. We do so by applying change-point analysis to the median and inter-quartile range (IQR) of the distribution.[3]

Figure 9 shows CDFs of the size of the largest corresponding CFRs. We see that, overall, the median is less steady than the IQR (indeed, we find that IQR change-points appear to often be a subset of median change-points), and both distributions shift about 5 minutes to the left for lossy traces. The striking difference with Figure 3, though, is the absence of entire hours with no change-points. Thus we find that *overall, delay is less steady than loss*, and that, while there's a wide range in the length of steady delay regions, in general delay appears well described as steady on time scales of 10–30 minutes. We can also test the median and IQR (computed over 10-second intervals) for independence within each CFR. Using the Box-Ljung test for up to 6 lags, we find very good agreement (90–92%) with independence.

C. Constancy of RTT spikes

Having characterized the constancy of the packet delay distribution's body, we now turn to the constancy of the RTT spike process. Analogous to our approach for packet loss, we group consecutive spikes into spike episodes, noting that in general the episodes are quite short lived: for example, the median duration of a spike episode (using $k = 2$) in $\mathcal{W}_1$ was 150 ms, and the mean 275 ms.

[3]The IQR of a distribution is the distance between the 25th and 75th percentiles. It serves as a robust counterpart to standard deviation. For IQR change-points, we compute the IQR over ten-second intervals and look for a change in the median of that time series.

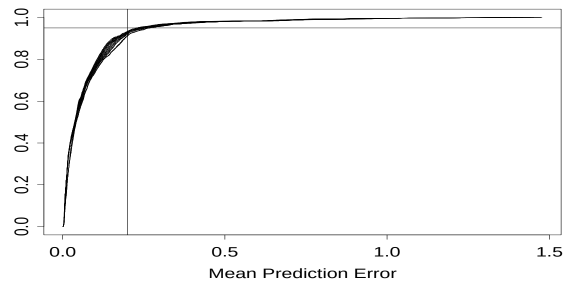

Fig. 10. CDFs of the mean error for a large number of delay predictors.

Upon applying change-point detection to the spike episode process, we find spike episodes even more steady than loss episodes: the process is steady across the entire hour 75% of the time for $k = 2$ spikes, and 90% of the time for $k = 4$ spikes. In addition, we find the interarrivals between spikes are well-modeled as IID exponential, i.e., Poisson.

D. Operational constancy of RTT

Similar to our analysis for loss (§ III-C), we assess the operational constancy of RTTs by partitioning the delays into a set of categories and then assessing the duration of regions over which the measured RTT stays within a single category.

Different applications can have quite different views as to what constitutes good, fair, poor, etc., delay. To have concrete categories, we used ITU Recommendation G.114 [ITU96], which defines three regions: 0–150 ms ("Acceptable for most user applications"), 150–400 ms ("Acceptable provided that Administrations are aware of the transmission time impact on the transmission quality of user applications"), 400+ ms ("Unacceptable for general network planning purposes"). Because these recommendations are for one-way delays and we are analyzing RTTs, we doubled them to form RTT categories, and then sub-divided 0–300 ms into 0–100 ms, 100–200 ms, and 200–300 ms, to allow a somewhat finer-grained assessment.

We find that more than half of the traces have maximum CFRs under 10 min, and 80% are under 20 min. We found virtually no difference whether or not we left RTT spikes in the traces (since they are rare), or when we tested a "shifted" version of the categories similar to the shifted version of loss rates discussed in § III-C. Thus, *not only are packet delays not mathematically steady, they also are not operationally steady.*

E. Predictive constancy of delay

We finish our assessment of different types of delay constancy with a brief look at the efficacy of predicting future RTT values. We again use the families of estimators discussed in § III-E. The events they process are RTT measurements, and our assessment concerns how well they predict the next measurement. Figure 10 shows that the estimators again all perform virtually identically, and that their performance is very good: the vertical line on the plot marks a mean prediction error of 0.2, which corresponds to estimating the next value within a factor of $e^{0.2} \approx 22\%$, and the horizontal line marks 95% of the distribution. We attain virtually identical results whether or not we include RTT spikes in the measurements. Thus, we find that, in contrast with loss (Figure 6), *in general, delay is highly predictable*. Of course, for some applications, the consequence of mispredicting delay can be significant (e.g., a bad TCP retransmission timeout); we are not blithely asserting that applications will find highly predictable those facets of delay that they particularly care about, only that delay in general is highly predictable.

V. Throughput constancy

The last facet of Internet path constancy we study is end-to-end throughput. Compared to loss and delay, throughput is a higher-level path property, a product of the first two plus the dynamics of the transport protocol used. In addition, applications have a wide range of throughput requirements. To keep our analysis tractable, we confine ourselves to a simple notion of throughput constancy, namely the minute-to-minute variations observed in 1 MB TCP transfers. The data we analyzed consisted of 169 runs of 5 hours each, comprising a total of 49,000 connections measured along 145 distinct Internet paths.

Based on a very large packet-level trace collected at a single busy Web server, [BPSSK98] found that the throughput of Web transfers exhibited significant temporal (several minutes) and spatial stability despite wide variations in terms of end-host location and time of day. Their study differs from ours in that the server was a single site, there were many more clients, and the analysis focused on the throughput of Web transfers, which are usually much shorter than our transfers. In other previous work, Paxson found that for a measure of available bandwidth derived from timing patterns in TCP connections, the predictive power of the estimator was fairly good for time periods up to several hours [Pa99].

A. Mathematical constancy of throughput

We applied change-point analysis to the mean of the series of per-minute throughput measurements in each trace. Figure 11 shows the cumulative distribution of the maximum CFR and the weighted average of the duration of the CFRs (per the discussion of Figure 3 previously). We see that few traces are steady over the entire 5-hour period,

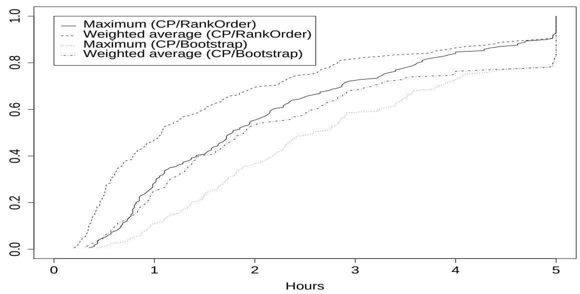

Fig. 11. CDF of maximum and weighted average CFRs for throughput achieved transferring 1 MB using TCP.

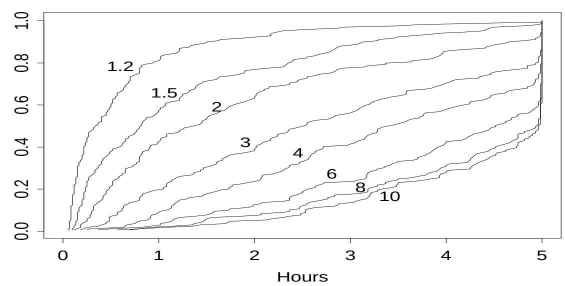

Fig. 12. Distribution of maximum operational constancy regions for $\rho = 1.2$ (leftmost), ..., $\rho = 10$ (rightmost).

and for 60-70%, the largest CFR is 2.5 hours long or less. The weighted averages are shifted over about 45 minutes; half of the time we find ourselves in a change-free region of under 1.5 hours duration.

On the other hand, throughput does not wildly fluctuate minute-by-minute: only 10% of the time do we find ourselves in a CFR of under 20 minutes duration. Similarly, the median number of change-points in a trace is 8. Finally, within CFRs, we find that the individual throughput measurements are well modeled as IID, 92% passing the Box-Ljung test for autocorrelation up to 6 lags; over entire traces, however, this figure falls to 24%.

B. Operational constancy of throughput

We adopt a simple notion of operational throughput constancy, namely whether the observed bandwidth stays in a region for which the ratio between the maximum and minimum observed values is less than a factor of ρ. Figure 12 shows the distribution of the size of the maximum steady regions, for $\rho = 1.2$ through $\rho = 10$. We see that if our operational requirement is for bandwidth not to vary by more than 20% peak-to-peak, then we will only have a few minutes of constancy, but as ρ increases, so too does the maximal constancy, fairly steadily; for peak-to-peak variation of a factor of 3, it is often several hours.

We also find that, due to the wide range in operational constancy as we vary ρ, there is no simple relationship between the mathematical and operational constancy of

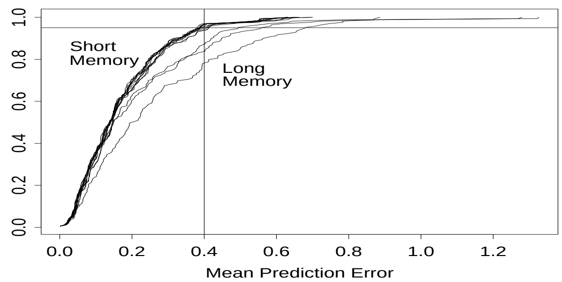

Fig. 13. CDFs of the mean error for a large number of throughput predictors.

throughput. For example, if we classify a trace as operationally steady if it has a maximum CFR of at least 2 hours, then for $\rho = 1.2$, we find $\overline{MO} = 53\%$, $M\overline{O} = 39\%$, $\overline{M}O = 2.4\%$, and $MO = 5.9\%$. But for $\rho = 10$, we have $\overline{MO} = 3.6\%$, $M\overline{O} = 1.2\%$, $\overline{M}O = 51.5\%$, and $MO = 43.8\%$, completely different.

C. Predictive constancy of throughput

We finish our look at different types of throughput constancy with a look at how well an estimator can predict the next observed throughput measurement. Figure 13 shows how the families of estimators discussed in § III-E performed in estimating the next throughput value over each 5-hour trace in its entirety. Almost all of the estimators perform equally well, with 95% of their estimates (horizontal line) yielding an error of 0.4 (vertical line) or lower, corresponding to estimating the next value within a factor of $e^{0.4} \approx 50\%$. However, three estimators do poorly: EWMA with $\alpha = 0.01$, and MA and SMA with windows of 128. These reflect estimators with long memory, as indicated on the plot (the other estimators had windows of 16 or less, or $\alpha \geq 0.125$), indicating that when predicting throughput, remembering observations from a number of minutes in the past is fine, but remembering for more than an hour can mislead the estimator. Finally, we note that for traces that are mathematically steady (maximum CFR $\geq$ 1 hour), the short-memory estimators do nearly twice as well (half the mean error) as they do on all the traces. (We do not attempt a comparison between prediction and operational constancy, since for throughput there is such a wide range of operational constancy depending on the parameter ρ.)

VI. Conclusions

Applications and protocols are becoming more *adaptive* and *network-conscious*. Network operators and algorithms are increasingly relying on measurements to assess current conditions. Mathematical models are playing a larger role in the discussions of Internet traffic characteristics. For each of these developments, one of the key issues is the degree to which the relevant Internet properties hold steady; yet each also involves a quite different notion of constancy. We have discussed how mathematical, operational, and predictive constancy sometimes overlap, and sometimes differ substantially. That they can differ significantly highlights how it remains essential to be clear which notion of constancy is relevant to the task at hand.

This paper can be read on two levels. On one level, we have attempted to shed light on the current degree of constancy found in three key Internet path properties: loss, delay, and throughput. One surprise in our findings is that many of the processes are well-modeled as IID, once we identify change-points in the process's median (loss, throughput) and aggregate fine-grained phenomena into episodes (loss runs, delay spikes). However, IID models are a mixed blessing; they are very tractable, but IID processes are very hard to predict.

The need to refine the analysis by looking for change-points and identifying episodes illustrates how important it is to find the right model. For example, while the loss process itself is both correlated and non-steady, when reduced to the loss episode process, the IID nature of the data becomes evident. This illustrates the importance of considering the constancy of a path property not as a fundamental property in its own right, but only as having meaning in the context of a model, or an operational or protocol need.

Another general finding is that almost all of the different classes of predictors frequently used in networking (moving average, EWMA, S-shaped moving average) produce very similar error levels. Sometimes the predictors perform well, such as when predicting RTTs, and sometimes poorly, because of the IID nature of the data (loss, throughput).

Finally, the answer to the question "how steady is the Internet?" depends greatly on the particular aspect of constancy and the dataset under consideration. However, it appears that for all three aspects of constancy, and all three quantities we investigated, one can generally count on constancy on at least the time scale of minutes.

On another level, our paper tries to carefully distinguish between the three different notions of constancy: mathematical, operational, and predictive. One of the goals of our study was to gather the appropriate set of concepts and tools needed to understand each of these different aspects of constancy. While the detailed results from our measurements may soon prove ephemeral (due to changing traffic conditions), or rendered obsolete (by subsequent and better measurement efforts), we hope that the fundamental concepts and tools developed here might prove longer-lived.

VII. Acknowledgments

Many thanks to Andrew Adams, who did a tremendous amount of NIMI development work to support our extensive measurements, and to his NIMI colleagues, Matt Mathis and Jamshid Mahdavi. We would also like to thank Lee Breslau for valuable discussions; the many NIMI volunteers who host NIMI measurement servers; and Mark Allman for the bulk-transfer measurement software, and comments on this work.

Appendix

I. Statistical Methodology

In this appendix we discuss the three main statistical techniques we use in our analysis, tests for: change-points, independence, and exponential interarrivals.

A. Testing for change-points

We apply two different tests, $\mathcal{CP}/RankOrder$ and $\mathcal{CP}/Bootstrap$, to detect changes in the median. Both tests detect change-points in a two step approach: first identifying a candidate change-point, then applying a statistical test to determine whether it is significant. The combined approach [La96], [Ta00] uses an analysis of ranks in order to detect changes in the median [SC88]. Being based on ranks, the method is resistant, i.e., tolerant to the presence of outliers. Furthermore, the hypotheses underlying these test are quite weak; equality of variances is not required.

Consider first a set of n values $(x_i)_{i=1,\ldots,n}$ comprising a segment of a given time series. Construct the rank r_i of each x_i within the set, i.e., 1 for the smallest and n for the largest. Compute the cumulative rank sums $s_i = \sum_{j=1}^{i} r_i$. The basis of the test is that if no change point is present, the cumulative rank sums s_i should increase roughly linearly with i. Indeed, suppose we form the adjusted sum:

$$s'_i = |s_i - \overline{s}_i|$$

as the difference between s_i, and its presumed mean $\overline{s}_i = i(n+1)/2$ assuming no change-point to be present. Then s'_i should stay close to zero. If, however, a change-point is present, higher ranks should predominate in either the earlier or later part of the set, and hence s'_i will climb to a maximum before decreasing to zero at $i = n$. We identify the maximizing index i_0 for s'_i and i running over $\{1, \ldots, n\}$ as a candidate change-point.

In the second stage, to test equality of two sets $X^- = \{x_1, \ldots, x_{i_0-1}\}$ and $X^+ = \{x_{i_0+1}, \ldots, x_n\}$, $\mathcal{CP}/Bootstrap$ uses the bootstrap analysis procedure outlined in [Ta00], while $\mathcal{CP}/RankOrder$ uses the Fligner-Policello Robust Rank-Order Test [SC88].

- *Bootstrap analysis* (used in $\mathcal{CP}/Bootstrap$). The bootstrap analysis procedure outlined in [Ta00] uses S_{diff}, defined as $(\max s_i - \min s_i)$, to estimate the magnitude of the change at the candidate change-point. It determines the confidence level of change by testing how often the bootstrap difference S^0_{diff} of a bootstrap sample $\{x^0_i\}$—a random permutation of $\{x_i\}$—is less than the original difference S_{diff}.
- *Fligner-Policello Robust Rank-Order Test* (used in $\mathcal{CP}/Rank-Order$). The test statistic is constructed as follows. For $x \in X^+$ define r^+_x as the rank of x in $X^+ \cup X^-$ minus the rank of x in X^+, with rank ties handled by assigning the average rank to all members of a tie set. Define rank mean $r^+ = \sum_{x\in X^+} r^+_x/n^+$ where $n^+ = \#X^+$, and sums of squared differences $v^+ = \sum_{x\in X^+}(r^+_x - r^+)^2$. Define n^-, r^-, and v^- symmetrically. Then the test statistic:

$$z = \frac{n^+r^+ - n^-r^-}{2\sqrt{r^+r^- + v^+ + v^-}}$$

has, asymptotically as $n \to \infty$, a standard normal distribution. Thus we can associate a significance level with the candidate change point i_0 in the usual manner. By choosing a significance level ℓ (we use 5% throughout this thesis) we specify our acceptable probability ℓ of incorrectly rejecting the null hypothesis. The test accepts the null hypothesis (in a two-sided test) if $F(|z|) < 1 - \ell/2$ where F is the cumulative distribution function of the standard normal distribution. (However, note that the large n asymptotic is not sufficiently accurate when i_0 and $n - i_0 \le 12$; in this case Table K in Appendix I of [SC88] should be used.) In some cases we shall use this test on binary data, in which case it reduces to a test of the equality of the expectations corresponding to binary states on either side of the candidate change-point.

The above can be extended to the identification of multiple change points, as follows [La96], [Ta00]. First, choose a significance level. Second, apply the above method recursively to the two segments $\{1, \ldots, i_0\}$ and $\{i_0 + 1, \ldots, n\}$ until no more change points are found at the chosen significance level. Third, apply backward elimination to reinspect the set of candidate change points in order to eliminate false detections, as follows. Let there be m change-point candidates $j_1 < \cdots < j_m$. Let j_0 and j_{m+1} be 0 and n respectively. Starting with the first identified candidate, call it j_{k_0} $(1 \le k_0 \le m)$, reinspect for change-points on the set $\{j_{k_0-1} + 1, \ldots, j_{k_0+1}\}$, and adjust or delete non-significant change-points. Repeat for all candidates in order of identification. Repeat backward elimination until the number of change-points is stable. By reestimating each change-point using only the data between the two surrounding change-points, backward elimination avoids the contamination caused by the presence of multiple change-points at the time of recursion and consequently helps to reduce the rate of false detections.

B. Testing for independence

We assess independence using the Box-Ljung test [LB78]. For a time series with n elements, the Box-Ljung statistic Q_k is a weighted sum of squares of measured autocorrelations r_i from lags 1 up to k:

$$Q_k = n(n+2)\sum_{i=1}^{k} \frac{r_i^2}{n-i}.$$

Under the null hypothesis that the process comprises independent Gaussian random variables, the distribution of Q_k converges, for large n, to a χ^2 distribution with k degrees of freedom. Thus by comparing the test statistic Q_k with the $1 - \ell$

quantile of the appropriate χ^2 distribution, we can test whether the autocorrelations of the time series differ at significance level ℓ from those of independent Gaussian random variables. In fact, as remarked in [LB78], the test is relatively insensitive to departures from the Gaussian hypothesis in the underlying process. This is because the measured autocorrelations r_i are asymptotically Gaussian provided the marginal distribution of the underlying process has finite variance. (While infinite variance (heavy tails) abound in networking behavior, the time series we consider here are generally well bounded, and certainly have finite variance.)

C. Testing for exponential distributions

An exploratory test for an exponential distribution of inter-event times is to plot the log-complementary distribution function; for an exponential distribution this is linear with slope equal to the negative of the reciprocal of the mean. A statistical test is that of Anderson-Darling. This test has been found to be more powerful than either the Kolmogorov-Smirnov or the χ^2 tests, i.e., its probability of correctly rejecting the null hypothesis (that the distribution is exponential) is greater; see [DS86]. This is, in part, due to the fact that the Anderson-Darling test employs the full empirical distribution (rather than binning, as in a χ^2 test), allowing it to give more weight to larger sample values whose presence can lead to a violation of the null hypothesis.

For a set of n rank-ordered inter-event times $t_1 < \cdots < t_n$, the appropriate Anderson-Darling statistic is:

$$A^2 = -n - \frac{1}{n}\sum_{i=1}^{n}(2i-1)\left\{\log(1-e^{-t_i/\bar{t}}) - t_{n+1-i}/\bar{t}\right\}$$

where $\bar{t} = n^{-1}\sum_{i=1}^{n} t_i$ is the empirical mean inter-event time. We reject the null hypothesis at significance level ℓ if the test statistic exceeds the tabulated values appropriate for that level; see, e.g., Table 4.11 in [DS86]. We note the importance of using the table appropriate to the present case in which the mean is estimated from the sample, rather than being specified in advance. Moreover, the table explicitly takes into account the effect of a finite sample size n.

References

[BPSSK98] H. Balakrishnan, V. Padmanabhan, S. Seshan, M. Stemm and R. Katz, "TCP Behavior of a Busy Web Server: Analysis and Improvements," *Proc. IEEE INFOCOM '98*, Mar. 1998.

[Bo93] J-C. Bolot, "End-to-End Packet Delay and Loss Behavior in the Internet," *Proc. SIGCOMM '93*, pp. 289–298, Sept. 1993.

[CPB93] K. Claffy, G. Polyzos and H-W. Braun, "Measurement Considerations for Assessing Unidirectional Latencies," *Internetworking: Research and Experience*, 4 (3), pp. 121–132, Sept. 1993.

[DS86] R.B. D'Agostino and M.A. Stephens, Eds., *Goodness-of-Fit Techniques*, Marcel Dekker, New York, 1986.

[FHPW00] S. Floyd, M. Handley, J. Padhye and J. Widmer, "Equation-Based Congestion Control for Unicast Applications," *Proc. SIGCOMM '00*, pp. 43–56, Aug. 2000.

[Gi60] E. Gilbert, "Capacity of a Burst-Noise Channel," *Bell Systems Technical Journal*, 39(5), pp. 1253–1265, September 1960.

[ITU96] International Telecommunication Union, "One-way Transmission Time," *ITU Recommendation G.114*, Feb. 1996.

[Ja88] V. Jacobson, "Congestion Avoidance and Control," *Proc. SIGCOMM '88*, pp. 314-329, Aug. 1988.

[JBB92] V. Jacobson, R. Braden, and D. Borman, "TCP Extensions for High Performance," RFC-1323, May 1992.

[JS00] W. Jiang and H. Schulzrinne, "Modeling of Packet Loss and Delay and Their Effect on Real-Time Multimedia Service Quality," *Proc. NOSSDAV 2000*, June 2000.

[La96] J. Lanzante, "Resistant, robust and non-parametric techniques for the analysis of climate data: theory and examples, including applications to historical radiosonde station data," *Int. J. Climatol.*, 16 (11), 1197-1226, 1996.

[LB78] G. Ljung, and G. Box "On a Measure of Lack of Fit in Time Series Models," *Biometrika '65*, pp. 297–303, 1978.

[MJV96] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven Layered Multicast," *Proc. SIGCOMM '96*, pp. 117–130, Aug. 1996.

[Mu94] A. Mukherjee, "On the Dynamics and Significance of Low Frequency Components of Internet Load," *Internetworking: Research and Experience*, Vol. 5, pp. 163–205, December 1994.

[PF95] V. Paxson, and S. Floyd, "Wide-Area Traffic: The Failure of Poisson Modeling," *IEEE/ACM Transactions on Networking*, 3(3), pp. 226-244, June 1995.

[PMAM98] V. Paxson, J. Mahdavi, A. Adams, and M. Mathis, "An Architecture for Large-Scale Internet Measurement," *IEEE Communications*, 36(8), pp 48–54, Aug. 1998.

[Pa99] V. Paxson, "End-to-End Internet Packet Dynamics," *IEEE/ACM Transactions on Networking*, 7(3), pp. 277–292, June 1999.

[Ri95] J. Rice, "Mathematical Statistics and Data Analysis," 2nd edition, Duxbury Press, 1995.

[SCK00] H. Sanneck, G. Carle, and R. Koodli, "A framework model for packet loss metrics based on loss run length," *Proc. SPIE/ACM SIGMM Multimedia Computing and Networking Conference*, January 2000.

[SC88] S. Siegel and N. Castellan, *Non-parametric statistics for the behavioral sciences*, McGraw–Hill, New York, 1988.

[Ta00] W.A. Taylor, "Change-Point Analysis: A Powerful New Tool For Detecting Changes", preprint, available as http://www.variation.com/cpa/tech/changepoint.html

[Wo82] R. Wolff, "Poisson Arrivals See Time Averages," *Operations Research*, 30(2), pp. 223–231, 1982.

[YMKT99] M. Yajnik, S. Moon, J. Kurose and D. Towsley, "Measurement and Modeling of the Temporal Dependence in Packet Loss," *Proc. IEEE INFOCOM '99*, Mar. 1999.

[ZPS00] Y. Zhang, V. Paxson and S. Shenker, "The Stationarity of Internet Path Properties: Routing, Loss, and Throughput," ACIRI Technical Report, May 2000. http://www.aciri.org/vern/papers/stationarity-May00.ps.gz

[Zh01] Y. Zhang, "Characterizing End-to-End Internet Performance," Ph.D. Thesis, Cornell University, Aug. 2001. http://www.cs.cornell.edu/yzhang/papers/thesis.ps.gz