The Stationarity of Internet Path Properties: Routing, Loss, and Throughput

Yin Zhang
Computer Science Department
Cornell University
Ithaca, NY
yzhang@cs.cornell.edu

Vern Paxson and Scott Shenker
AT&T Center for Internet Research at ICSI
International Computer Science Institute
Berkeley, CA
vern@aciri.org, shenker@icsi.berkeley.edu

May 2, 2000

Abstract

There is much interest in using network measurements for both modeling and operational purposes. In this paper we focus on the fundamental question of the stationarity of such measurements. That is, to what extent are past measurements a good predictor of the future? We used the NIMI infrastructure and a set of public traceroute servers to capture large measurement datasets of three quantities: routing, packet loss, and TCP throughput. We apply statistical tests to attempt to develop sound characterizations of the stationarity of these data sets, and discuss several types of nonstationarity.

1 Introduction

In recent years there has been a surge of interest in network measurements. These measurements have deepened our understanding of network behavior and led to more accurate and qualitatively different models of network traffic. Network measurements are also used operationally by various protocols to guide network usage. For instance, RLM [MJV96] and equation-based congestion control algorithms [PSC99] use network measurements to set transmission rates, and the caching of congestion information on a path can similarly be seen as an operational use of network measurements.

Measurements are inherently bound to the present; they can merely report the state of the network at the time of the measurement. However, the modeling and operational uses of these measurements are only successful if measurements are good predictors of the future: that is, are network measurements stationary? This is the question we address in this paper. We do so in the context of measurements of three quantities describing Internet paths: routes, packet loss, and throughput.

We say that a dataset of network measurements is *mathematically stationary* 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 stationarity 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 stationarity is the statement that the dataset is consistent with that set of parameters throughout the dataset.

One example of mathematical stationarity 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. An example of mathematical *nonstationarity* is packet delay distributions. Mukherjee found that packet delay along a particular Internet path is well-modeled using a shifted gamma distribution, but the parameters of the distribution vary from path to path and over the course of the day [Mu94]. Likewise, [PF95] found that session arrivals on longer time scales can still be well-modeled using Poisson processes, but only if the rate parameter is adjusted to reflect diurnal load patterns.

Testing for stationarity of the underlying mathematical model is relevant for modeling purposes, but is perhaps too severe a test for operational purposes because many nonstationarities are completely 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 stationary, 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 stationarity when addressing operational issues. The key criterion in operational, rather than mathematical, stationarity is whether an application (or other operational entity) would care about the changes in the dataset. We call a dataset *operationally stationary* if the quantities of interest remain within bounds considered operationally equivalent. Note that while it is obvious that operational stationarity does not imply mathematical stationarity, it is also true that mathematical stationarity does not imply operational stationarity. For instance, if the loss process is a stationary but highly bimodal process with a high degree of correlation, then the application will see sharp transitions from low-loss to high-loss regimes and back which, from the application's perspective, is highly nonstationary.

Another important distinction is between the concepts of *persistence* and *prevalence* [Pa97]. Persistence reflects how long one set of characteristics will remain unchanged if observed continuously. Prevalence, in contrast, quantifies the percentage of time the system will exhibit a particular set of characteristics if observed sporadically. The context usually determines which aspect of stationarity is more relevant. Also note that the two notions are orthogonal: you can have one and not the other, or both, or neither, depending on whether the property is short-lived or long-lived, and whether it tends to primarily manifest itself in many different ways or in just a few ways.

For the routing data we present, we focus mainly on prevalence. The motivating example is protocols which cache path information for the next time they access the same address; routing prevalence is a measure of how often an access to an address will travel the same route (and thus the cached information would be of use).

For the loss and throughput data, we concentrate on persistence by studying how long the observed loss and throughput characteristics remain unchanged. The motivation here is that the underlying transport protocols (or, in some cases, the application itself) is monitoring loss and throughput behavior and using past measurements to guide future behavior; nonstationarities in these quantities may lead to discontinuities in application performance.

The three different data categories—routes, loss, and throughput—represent three different levels of Internet behavior. The routing data represents the stability—or stationarity—of a very basic infrastructure. The dataset we collected, and our analysis of it, is quite similar to that by Paxson in [Pa97]. In addition to studying the prevalence of the current dataset (and briefly touching on its persistence), we present detailed comparisons with the results of the previous work to illuminate long-term trends in route stability.

While routes are typically invisible to higher layers, packet loss is an end-to-end path property quite visible to the transport layer. Correlation in packet loss was previously studied in [Bo93, Pa99a, 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. [Pa99a] 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 nonstationary samples—some of our results bear directly upon this previous work. In particular, 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.

Loss is visible to the transport layer, but it is typically hidden from applications. In contrast, throughput is precisely what many applications care about most. Throughput can be thought of as the application-relevant manifestation of the underlying loss and delay behavior on a path. Our third dataset thus addresses the stationarity of a quantity of direct relevance to applications. Previous work in this area found that available bandwidth as derived from timing patterns in TCP connections remained fairly steady for several hours [Pa99a], and that Web access to a large server exhibited significant temporal and spatial stability [BPSSK98].

Our focus is on stationarity, but to soundly assess stationarity 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 stationarity. Thus, while our goal is lofty—understanding stationarity—we necessarily devote considerable attention in our discussion to more mundane methodological issues. We view this study as an initial step: admittedly flawed, inherently limited, but also arguably useful in terms of uncovering some plausible generalities, and identifying some places to look in future studies.

The paper is organized as follows. We first describe the sources of data in Section 2. The routing data, and its stationarity analysis, is presented in Section 3. We discuss the loss and throughput data in Sections 4 and 5 respectively, and conclude in Section 6 with a brief summary of our results.

2 Sources of data

We gathered three basic types of measurements: routes, using the `traceroute` utility ([Ja89]; see [St94] for detailed discussion); Poisson packet streams, using the `zing` utility that comes with the NIMI infrastructure (see below); and bulk throughput, using 1 MB TCP transfers.

Most of our measurements were made using the NIMI measurement infrastructure [PMAM98]. NIMI is a follow-on to Paxson's NPD measurement framework [Pa97], in which a number of measurement platforms are deployed across the Internet and used to perform end-to-end measurements. NIMI attempts to address the limitations and resulting measurement biases present in NPD [Pa99a].

The infrastructure consisted of 31 hosts: 25 in the United States, 5 in Europe, and one in Asia. 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. For example, the top dozen domain names of the 1,301 interior routers appearing in traceroute measurements between the platforms are (in order): ucaid.edu (Abilene), calren2.net, alter.net, es.net, vbns.net, sprintlink.net, cw.net, teleglobe.net, sunet.se, ja.net, psi.net, and net.uk. We might plausibly argue then that our observations might apply fairly well to the general Internet of the not-too-distant future, if not today.

In addition, to gain a broader view of Internet routing behavior, we made use of a pool of 189 public traceroute servers, a third located within the United States, and the other two-thirds spread across 31 different countries. Data from these sources is not as clean as that from the NIMI infrastructure, because the collection process suffers from some of the same biases as the NPD framework (failure to connect to the measurement server may preclude an opportunity to measure a network problem). The data is nevertheless valuable due to its rich diversity.

We discuss the particulars of the measurements made from these sources in the subsequent sections analyzing those measurements. While we began preliminary data capture and analysis earlier, all of the data analyzed in this paper was captured during December, 1999, and January, 2000.

3 Routing stationarity

We gathered two sets of routing measurements, one from NIMI and one from the public traceroute servers mentioned above. For NIMI, we measured 36,724 routes, which included 707 of the 930 possible host pairs. Measurements were made at Poisson intervals with a mean of 10 minutes between measurements initiated by the same host. By using Poisson intervals, time averages computed using the measurements are unbiased [Wo82].

12,655 of the measurements were made by pairing the source host with a random destination host in the mesh each time a new measurement was made; these measurements assured broad coverage of the mesh. The remaining 24,069 paired a single source with the same destination over the course of a day. These measurements were made as part of the `zing` packet data discussed in Section 4 below. Thus, this dataset gives us a fairly detailed look at a smaller number of Internet paths.

Using the public servers, we made 287,206 route measurements (so for both datasets we have an average of over 1,000 routes measured from each host). Due to the size of the mesh, it was impractical to fully measure it in depth, so we split our measurements into one group scattered across the mesh, comprising 220,551 of the measurements, in an attempt to capture the breadth of routing anomalies, and another of 66,655 in-depth measurements of pairs, for assessing routing prevalence and persistence, similar to our NIMI measurements. The former set of measurements covered 97% of the mesh, with a median of 5 traceroutes per pair of hosts.

3.1 Routing pathologies

Following the approach used in Paxson's earlier Internet routing study [Pa97], we begin our routing analysis with characterizations of unusual or non-functioning routing behavior, i.e., "pathologies." We do so with three goals: first, as a sanity check on the data, to ensure it is not plagued with problems; second, so we can distinguish between ordinary routing fluctuations and apparent fluctuations that in fact are instead pathologies; and third, to form an impression on whether the quality of Internet routing has changed since Paxson's study, which was based on data 4–5 years older than ours.

To do so, we first categorize three different types of problems that we can associate with a traceroute measurement: measurement failures (the tool did not run or failed to produce useful output); connectivity problems (an end user would notice that there was some sort of problem); and eccentricities (unusual behavior, but not likely to affect end-to-end performance), which space does not permit us to further analyze here. These last two categories are somewhat blurred in Paxson's analysis, but his pathologies were dominated by outages (30 seconds or more of no connectivity), which we categorize as a connectivity problem.

For the NIMI data, we restrict our pathology analysis to the 12,655 traceroute measurements of the random mesh, as these reflect broad, even coverage of the different routes, rather than restricted, detailed coverage of a small subset of the routes. Of these, about 10% were marred by measurement errors occurring on the NIMI host themselves, so missing this data is unlikely to bias our samples. Of the remainder, 6% exhibit connectivity problems, with nearly all of these being connectivity outages. This figure is *double* that of Paxson's, even though the NIMI sites should enjoy better connectivity than the NPD sites due to the higher prevalence of university and research labs with high-quality Internet connections.

However, the NIMI pathologies are heavily skewed by two sites. If we remove these as outliers, then the total pathology rate falls to 3.2%, still dominated by outages, with the next most common pathology being unresolved routing loops, but these being 20 times as rare. The 3.2% figure is virtually unchanged from Paxson's 1995 data, which had a 3.3% pathology rate. Some pathologies are much more rare (persistent loops), others somewhat more common (30+ sec. outages). All in all, we would conclude that routing has not gotten significantly worse, but neither has it improved; furthermore, we had to discard a pair of our sites to get there, while Paxson did not need to resort to removing pathology outliers.

To attempt to assess the quality of broader Internet routing, we analyzed the public traceroute server data, as follows. First, we again restricted our analysis to the measurements made with random pairing (220,551 total). For those, we found that 11% of measurements completely failed, and another 4% were incomplete due to the connection to the server failing before it delivered all of its data. Of the remainder, we found that 4.3% suffered from a connectivity problem, the most common of which were outages (2.0%) and rapid route changes (1.4%).

This indicates that for the general Internet, routing is degraded compared to that measured in 1995. But this may not be a fair comparison, since the dataset is only one-third USA sites, while Paxson's data was about two-thirds USA sites. To assess the effects of this discrepancy, we repeated the analysis but limiting it to the 33,018 measurements made between two USA sites. Of these, 12.5% failed or were incomplete, and of the remainder, 2% exhibited a connectivity problem, with almost all (1.6%) of these being outages. From this we conclude that the evidence is solid that routing has neither improved nor degraded significantly since 1995, in terms of routing problems.

3.2 Routing prevalence

As noted above, in general we can think about two types of consistency for a network path property, its prevalence and its persistence. In this section, we characterize the prevalence of Internet routes as manifest in our datasets: that is, how often the most commonly occurring ("dominant") route is observed. The finding in [Pa97] concerning routing prevalence was that in general Internet paths were strongly dominated by a single route, though there was significant site-to-site variation.

To assess prevalence, we use the second type of measurements discussed above, namely repeated measurements between particular pairs of hosts. For the NIMI data, we had 50 or more successful, non-pathological measurements of 94 distinct paths (source/destination pairs), comprising a total of 17,627 measurements. For the public traceroute servers, we had 50 or more such measurements of 367 distinct paths, for a total of 52,872 measurements.

An important consideration when assessing routing stability (both prevalence and persistence) is how exactly to determine whether two routes are the same. The problem arises in two ways. First, traceroute measurements report IP addresses, and some routers do not always return the same address. For example, we have traceroutes in our datasets that differ only in one hop sometimes being reported as address 205.171.18.114 and other times as 205.171.5.129. However, both of these addresses have DNS entries as *sjo-core-01.inet.qwest.net*, and these do in fact refer to the same router. In addition, we sometimes have a similar situation in which the IP addresses resolve to *different*, but very similar, hostnames, such as *s8-0-0-14.nyc-bb5.cerf.net* and *s0-0-0-25.nyc-bb5.cerf.net*. These may be the same router, or they may in fact be different routers but ones which are co-located or at least functionally very similar. Accordingly, we might well argue that two routes that differ only by one of these two cases ought to be considered the same route, since either they are in fact the same route, or they at least should in many ways share the same properties. Accordingly, we would like to merge the two addresses into the equivalent of a single router prior to performing our analysis.

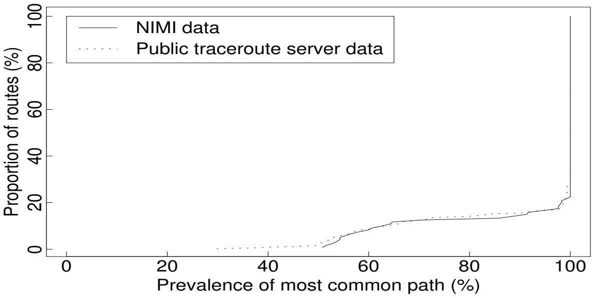

Figure 1: Routing prevalence in NIMI and public traceroute server datasets.

We identified addresses A and B as a pair to merge if they occurred in the same positions in adjacent traceroutes, and their hostnames were either identical or agreed both in domain and in whatever geographic clues were present in the hostname (e.g., *nyc*). For borderline cases we allowed as additional evidence of equivalence the fact that the next hop in both traceroutes was identical. For the NIMI data, we identified 64 equivalent addresses (out of 1,602 total), and for the public server data, 220 (out of 12,663 total)—enough in both cases to seriously skew our analysis were they not merged, since a number were frequently observed routers.

Figure 1 shows CDF's of the prevalence of the dominant route for the NIMI and public traceroute server datasets. For the NIMI routes, 78% always exhibited the same path, and 86% of the routes had a prevalence of 90% or higher. For the public servers, the corresponding figures are 73% and 85%, respectively.

These figures are considerably higher than those given by Paxson in [Pa97]. The difference may reflect that routing has changed such that today the dominance effect is even stronger than in 1995, or it may reflect differing measurement methodologies; in particular, Paxson's data was spread over more days than ours. However, for the most obvious way to exploit routing prevalence—caching path properties for future use—it is plausible that the primary concern is the validity of routing prevalence over time scales of minutes to perhaps hours, a regime well covered by our data. In addition, the striking agreement between the two distributions taken from very different datasets suggests that the finding is well-grounded and quite plausibly general. Finally, we observe that even for the 15% of routes for which the dominant path does not completely dominate, it still is almost always observed the majority of the time, so it remains useful to cache information about its properties.

3.3 Routing persistence

Our basic approach for assessing routing persistence is to look at how many consecutive traceroute measurements each ob-

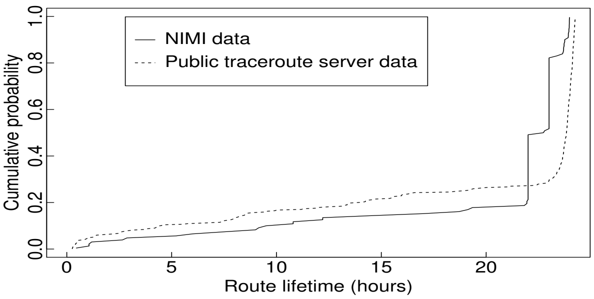

Figure 2: Routing persistence in NIMI and public traceroute server datasets, for routes not identified as exhibiting rapid changes.

served the same route. Because our measurements of the same route are made on average 10 min apart, this approach is sound except when the route is rapidly changing, in which case we may miss a change to another, short-lived route that then changed back, all between our two measurements. [Pa97] faced this problem too, and addressed it by first identifying paths with any evidence of rapidly changing routes and characterizing these separately. We follow the same approach.

For the NIMI data, there were 85 routes for which we had a successful series of day-long measurements. Of these, 8 exhibited rapid changes at some point, while for the public traceroute servers, 85 routes out of 383 did so.

Figure 2 gives the distribution of the route duration for the remaining routes. We see that very often routes persist for at least a day, the upper limit of what we can observe from our data. (The steps in the NIMI data reflects that some datasets were 22 or 23 hours long rather than a full 24 hours.) The long lower tail agrees with the finding for routing persistence in [Pa97], namely that most paths are persistent over time scales of many hours to days, but a fair number of paths are persistent only for quite shorter time scales.

From the figure, we see that about 10% of the commercial Internet routes have lifetimes of a few hours or less, and about 5% of the NIMI routes (highlighting that the routing between the NIMI infrastructure is considerably more stable than that of the Internet in the large). When we include the routes we had to factor out of our persistence assessment because they exhibited rapid changes at some point, we find good evidence that a total of about 1/3 of Internet routes in general, and 1/6 of the NIMI routes, are short-lived.

4 Loss stationarity

Our analysis of Internet packet loss is based on sets of measurements between pairs of NIMI hosts. Each host ran the "zing" measurement tool. 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.

Our methodology was to take day-long measurements during which the same two NIMI hosts were paired either for single hours, or for the entire day. We split the measurements into hour-long intervals to better cope with NIMI failures (and for some datasets we only measured every other hour). We configured zing on both hosts to send packets at Poisson intervals at an average rate of 10/sec. The packets were the default size of 256 bytes of payload carried in UDP datagrams. We also used zing's "round trip" option, meaning that each packet initially transmitted would elicit a packet in response from the receiver (but no further response is generated upon receiving the reply), to facilitate measuring round-trip time. In addition, we ran traceroute measurements between the two hosts at Poisson intervals with an average of one measurement in each direction every ten minutes. (This is the data we used above when assessing routing prevalence and persistence.)

We collected a total of 1,188 measurement hours, including 244 different host pairs. Thus, for the average host pair, we captured only 5 hour-long datasets, due to problems with the end hosts and (particularly) the measurement infrastructure. (There were no apparent time-of-day effects regarding which measurements were successful and which not, so the failures plausibly did not skew our measurements.) For 33 host pairs we captured 12 or more hours' worth of data.

In our measurement analysis, we discovered a deficiency of zing that biases our results somewhat: if the zing utility receives a "No route to host" error condition, then it terminates. This means that if there is a significant connectivity outage that results in the zing host receiving an ICMP unreachable message, then zing will stop running at that point, and we will miss a chance to further measure the problematic conditions. 47 of our measurement hours (roughly 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 our data is, regrettably, biased towards underestimating significant network problems, and how they correlate with non-stationarities.

The usable data comprised a total of 160 million packets. Packet loss was in general low, though it spanned a large range: 11% of the traces experienced no loss; 52% had some loss, but at a rate of 0.1% or less; 21% had loss rates of 0.1–1.0%; 15% had loss rates of 1.0–10%; and 1% had loss rates exceeding 10%.

Comparing these figures with those in [Pa99a] is somewhat tricky, as the latter were made measuring fairly short TCP connections, and the analysis divided connections into loss-free vs. lossy, which, with our much larger datasets, would be a somewhat oversimplified distinction. In addition, the TCP data packets have their loss rate inflated by the way in which TCP hunts for additional bandwidth. [Pa99a] attempted to account for this difference by analyzing the loss rate of ACK packets separately. We might then attempt to compare the two numbers, as follows. If a 100 KB TCP transfer such as used in [Pa99a] encounters no loss, then it will typically send either about 200 data packets of 512 bytes each, or about 70 packets of 1460 bytes, depending on the segment size. If delayed ACKs are used, then between 35 and 100 ACKs will sent.

Accordingly, loss-free and a loss rate of, say, 1% might be indistinguishable. If we then use a looser definition of "loss free" to mean "less than 1% loss," we find that 84% of our datasets were "loss-free." For the remainder, the average loss rate was 5.1%, but this drops to 4.1% if we eliminate the four data sets for which every single packet was lost.

The corresponding 1995 figures for USA sites in [Pa99a] are: 69% loss-free, and a 4.4% loss rate for lossy connections. In summary, it appears that times of loss have become less common among USA sites, but when loss does occur, the expected rate remains a bit above 4% (with, of course, wide variation). There is the additional question of the degree to which the NIMI sites are better connected than the NPD sites, which, unfortunately, appears difficult to further address with our data.

Both [Bo93] and [Pa99a] found that the conditional probability of a packet being lost given the loss of its predecessor was much higher than the unconditional loss probability. We likewise find this to be the case: after we remove traces for which every packet was lost, we find that the conditional loss probability over all of the traces was 27%,

Finally, 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 that, similar to as reported in [Pa99a], loss rates in a path's two directions are only weakly coupled, with a coefficient of correlation of 0.08. However, the logarithms of the loss rates are strongly coupled (0.56), 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, and it results in a discernible difference between the loss rate of requests and the loss rate of replies, the latter being about 90% of the former.

4.1 Pathologies: reordering and replication

As with routing, before analyzing stability patterns in packet loss, we first assess the presence of unusual packet behavior. We again do this both as a sanity check on the data, and to compare with [Pa99a] to see if we can discern significant changes since 1995.

Three types of pathologies are characterized in [Pa99a]: out-of-order delivery, replication (the delivery of multiple copies of a single packet), and corruption. As our measurements were made at user-level, and hence only recorded packet arrivals with good UDP checksums, we cannot accurately assess corruption. (`zing` packets include an MD5 checksum, which never failed for our data.)

We first needed to remove 354 traces from our analysis because clock adjustments present in the trace rendered facets of reordering ambiguous. On the remainder we then used the same definition of reordering as in [Pa99a]—that is, packets arriving with a sending sequence number lower than a packet that arrived previously are counted as "late"and hence an instance of reordering. We find that about 0.3% of the 136 million packets arrived out of order, and only 7% of our measured hours had no reordering at all. The highest reordering rate we observed sustained for one hour was 8.9%, and 25 datasets had rates exceeding 5% (three different sites dominated these). The largest reordering gap spanned 664 msec.

The 0.3% reordering rate is equal to the 1995 figure given in [Pa99a] of 0.3% of all data packets arriving out of order (and 0.1% for ACKs). However, we must be careful equating the agreement with a lack of change in reordering rates, because the data packets analyzed in [Pa99a] were often sent two back-to-back, due to TCP slow start and delayed acknowledgments acking every second packet, while our `zing` data was sent with an average of 50 msec between packets (a mean sending rate of 10/sec plus a mean reply rate to incoming `zing` packets of 10/sec), so the transit time difference to reorder our `zing` packets is quite high.

In agreement with [Pa99a], we find that reordering is dominated by just a few sites (the top three having seven times the median reordering rate), so another possible explanation of the relative increase in reordering we see is that it is simply due to chance in the selection of NIMI sites.

Also in agreement with [Pa99a], we find replication rare, with a total of 27 packets of the 160 million we studied arriving at the receiver more than once. This rate is very low (significantly lower than in [Pa99a]), and accordingly does not merit further characterization.

4.2 Periodicities

Because of the frequent use of timers in network protocols, and because such timers can sometimes synchronize in surprising ways [FJ94], it behooves us to analyze our loss data for periodicities, which form an important class of nonstationarities. We found a striking instance in the pattern of packet losses for packets sent to a particular NIMI site, "`nasa`". Figure 3 plots loss times for packets sent to `nasa`, as follows. The X-axis shows the packet's sending time during four 48-hour intervals, where we have compressed time between datasets taken several days apart by removing an integral number of days to facilitate plotting all the datasets together. The Y-axis plots the lost packet's sending time, too, except modulo 60 seconds. Finally, any time more than one packet was lost during the same second, we only plotted the first lost packet, to avoid over-cluttering the plot.

In the plot, a vertical line indicates a period of heavy loss, and any other line indicates periodic losses. If packets were lost exactly every sixty seconds, then we would see a horizontal line in the figure. If instead the line slopes away from horizontal, then its period is not quite exactly 60 sec. We see a number of such lines. The ones with the more gentle slope, such as in the lefthand side of the second plot, have a period

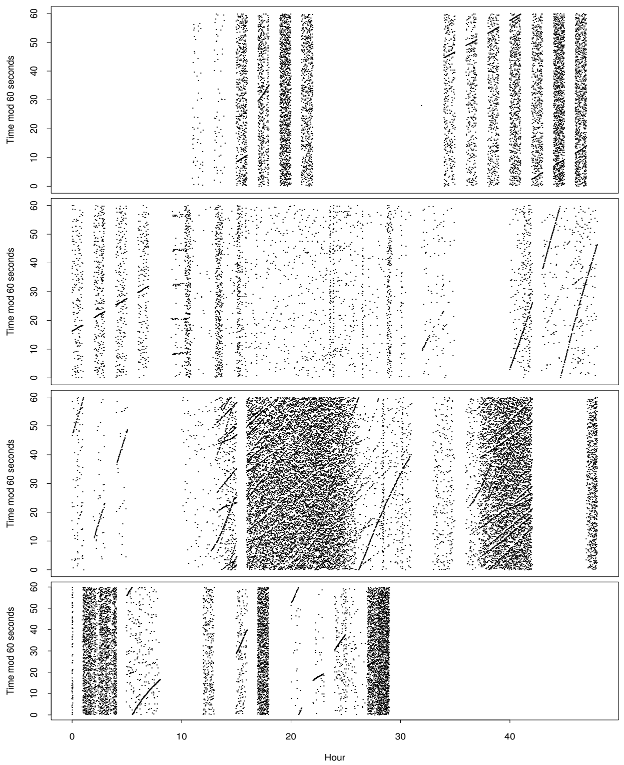

Figure 3: Time of loss for packets sent to `nasa` NIMI site, modulo one minute.

of around 60.035 sec, which we verified by plotting with that as the modulus instead of 60 sec, and observing that the lines then lay horizontal. The steeper lines, such as in the right-hand side of the same plot, have a period around 60.2 sec. They bring with them a cautionary tale, because if we analyze the data looking for periodicities at integral numbers of seconds, we find a clear indication of periodicity at exactly 43 sec. But it turns out that this frequency is merely a harmonic of 60.2 sec, and is evidenced because $43 = \frac{5}{7}60.2$. In addition, the clear steps in the second plot, left, do indeed have their own frequency, namely 12 sec. They correspond to a stronger loss process than do the others, since often after the initial loss, the loss continues into the next second. Finally, the various curves in the plot convey that the period of the losses often wanders over time.

We identified a particular router along the path to `nasa` that continues to exhibit periodic loss. We are working with the service provider (the router belongs to a commercial network) and the router vendor in an attempt to further identify the problem, and, in particular, to check whether it is indeed due to router synchronization as discussed in [FJ94]. (We note that the router is running the latest software version.)

Behavior such as this is important for assessing non-stationarity for several reasons. First, it introduces a strong non-stationarity due to its coupling with an external variable such as temporal phase. Second, we find it is easy to overlook this sort of behavior; for example, if we make the same plot but modulo 58 sec, the patterns go away completely. This speaks a cautionary note for those analyzing network data looking for periodicities. Finally, it highlights the utility of non-uniform sampling such as Poisson sampling, which can provide unbiased estimates of the degree of periodicity in a dataset, while uniform sampling runs the risk of missing or oversampling periodicities, and also of introducing periodic driving forces in the network.

Due to its persistent periodic loss behavior, we removed `nasa` from our subsequent loss-stationarity analysis. We also identified several other periodicities in our data, with cycle times of 60, 90, and 300 sec, and one set of traces with at least two periodic loss processes active at the same time. However, none of these periodicities was as strong or as pervasive as that for `nasa`, and we judged the traces could be kept for our stationarity analysis.

4.3 Individual loss vs. loss episodes

As noted above, the traditional approach for studying packet loss study is to examine the behavior of individual losses [Bo93, Pa99a, YMKT99]. These studies found correlation at time scales below 200–1000 ms, and left open the question of independence at larger time scales. In this section, 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. 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 a statistical tool sensitive to near-term correlations. The Box-Ljung Q statistic [LB78] works as follows. For a given time series with n elements, and a given lag k, the Box-Ljung Q is defined as a weighted sum of squares of autocorrelations from lag 1 to k. More precisely,

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

where r_i is the autocorrelation of given time series at lag i.

When n is large, under the null hypothesis that the time series is white noise, Q has a χ^2 distribution with k degrees of freedom. Thus, by comparing Q with the corresponding χ^2 distribution, we can test whether the autocorrelations of a given time series are significantly different from white noise.

In our study, we choose the maximum lag k to be 10, meaning ten consecutive losses or loss episodes. This is 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. 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 95% significance using the Box-Ljung Q statistic (i.e., for those 27% the hypothesis

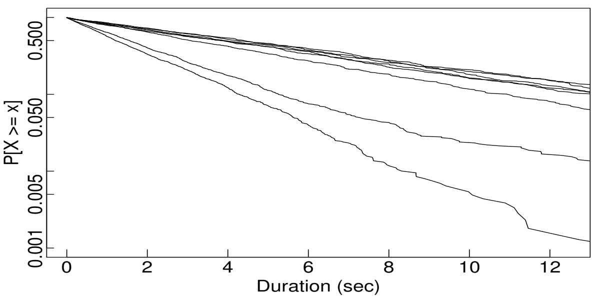

Figure 4: Example log-complementary distribution function plot of duration of loss-free runs.

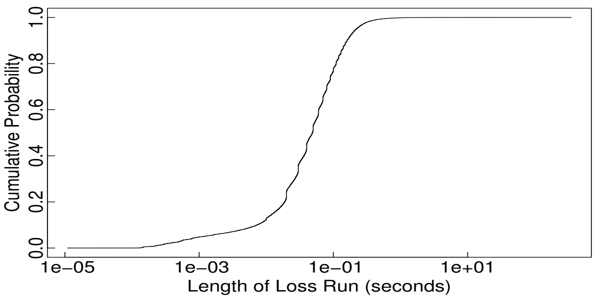

Figure 5: Distribution of loss run durations.

that the time series is white noise could not be rejected at 95% significance). The remaining traces show significant correlations at lags under 10, corresponding to time scales of 500–1000 ms. This is 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 (meaning, 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 4 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 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 (100 lags is 5 seconds on average), 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 our traces. It does not, however, indicate correlations on time scales of 10's of seconds (which we test for below, and find absent), but rather mixtures of exponentials arising from differing loss rates present at different parts of a trace.

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 related finding concerns the size of loss runs. Figure 5 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 msec 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: using the Q statistic to assess near-term correlations in the loss run size time series, we find that 94% of our traces are consistent with IID loss run sizes. We also confirm the finding in [YMKT99] that loss run lengths in packets often are well approximated with geometric distributions, though the larger loss runs do not fit this description.

If we instead consider the distribution *weighted* by the size of the interval—which we can interpret as giving the probability that any particular lost packet will be lost during a run of a particular size or smaller—then we find that, after removing traces for which every packet was lost, 60% of the packets are lost in runs of length 1, and 90% in runs of length 13 or less.

Thus, it would appear that we can formulate a descriptive model of Internet loss processes as loss episodes arriving according to a Poisson process with a fixed rate, with the size of each episode being drawn IID from the distribution given in Figure 5. Fully evaluating this conjectured model is beyond the scope of this paper, and clearly we may run into difficulties once we attempt to accommodate both wide ranges of network conditions and network paths more heavily loaded than the bulk of those in our study. But it does appear a promising avenue to pursue, and is potentially simpler than the k-th order Markov chain model proposed in [YMKT99].

4.4 Loss rate stationarity

We now turn to assessing patterns of stationarity for the incidence of loss episodes in our data. We begin with the statistical tests we will apply to test for stationarity. While we could use the Box-Ljung Q statistic further, for this analysis an exact test is available, because the time series is discrete.

Consider a series of random variables x_i, $i = 1, \ldots, n$, describing the number of instances of some event out of a total of y_i observations. One example, which we use below, is when the x_i are the number of lost packets in a minute and the y_i are the total number of packets sent in that minute. *Fisher's exact test* [Ri95] allows us to determine whether finding x_2 out of y_2 is consistent (within some significance level s, which we typically take to be 95%) with finding x_1 out of y_1, given the hypothesis that the observations are independent. An extension of this test, which we call *Nth Root Fisher* (NRF), tests whether the series (x_2, y_2), (x_3, y_3), $\ldots$, (x_n, y_n) are all consistent with (x_1, y_1) for a fixed n. NRF consists of testing the consistency of each (x_i, y_i) with (x_1, y_1) for $2 \leq i \leq n$ at a significance level $s^{\frac{1}{n}}$. If each pairwise test succeeds, then we know that they all simultaneously hold with probability s (by taking the product of probabilities). Note that this is not a test for complete internal consistency, which is computationally expensive to perform, but a weaker test only for consistency of each subsequent observation with the first observation.

The basic technique we use is to slide a window of size n across a time series of loss (episode) rates and for each new starting minute apply NRF for the given n to determine whether the n successive minutes are consistent with an IID model. We then compute the ratio ρ_n of the number of consistent intervals to the total number of tested intervals. ρ_n gives us a gauge as to the degree to which the time series is well-described as stationary on time scales of n. We can next vary n to see how stationarity behaves at differing time scales.

For our data, we used time series computed by binning packets into one-minute intervals and computing loss episode rates. (Tests of 10-second intervals find nearly ubiquitous stationarity.) We then vary n between 1 and 10 minutes. We find many different patterns of stationarity. Figure 6 shows a representative set of the most common ones. The top plot shows a trace for which the loss episode rate varied between 2.4–6.4% over the course of the hour, and yet the entire trace is consistent with an IID description for all n. We categorize such a trace as "completely stationary." Any variations in such a plot can be explained as simply reflecting stochastic fluctuations. (Note that if we test raw loss rates, as opposed to loss episode rates, then NRF detects two regions of mild non-stationarity, one starting at $T = 8$ min for $n = 1$, and one starting at $T = 38$ min for n ranging from 7 to 9—though the difference between the two time series is slight: the overall loss rate is 4.30%, while the overall loss episode rate is 4.06%.)

The trace in the second plot exhibits two loss "spikes". On this and subsequent plots we delimit windows of non-stationarity with brackets `[ ]`, which we draw for $n = 1$ lowermost and $n = 10$ uppermost. We see that for different n, as soon as the window includes the spike, NRF finds an inconsistency with the IID stationarity model.

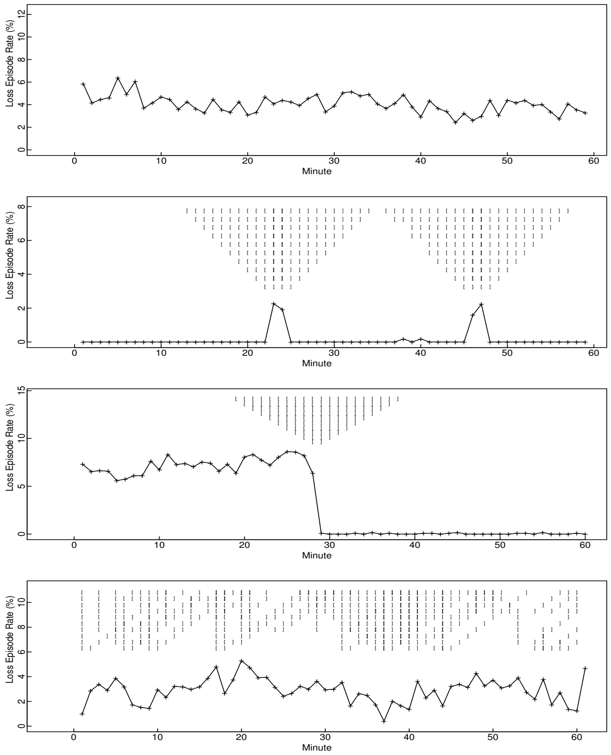

Figure 6: Fisher consistency regions for different loss patterns: strongly stationary, spikes, plateaus, messes.

The third plot shows the "plateau" pattern, in which the loss rate is stationary and basically level for quite a while, and then abruptly shifts to some other value. In this trace, the only region of non-stationarity is the transition between levels. This particular trace is additionally interesting because we might expect that the level shift coincided with a routing change, but traceroutes before and after the shift reveal the *same* route. However, they also reveal that prior to the level shift in loss episode rate, the fifth hop of the route (*195.atm11-0-0.br1.nyc1.alter.net*) had a latency of about 450 msec, while after the shift, it fell to 1 msec! Clearly, a major layer 2 property changed, and is likely the direct cause of the non-stationarity. This discontinuity highlights the numerous subtle effects, some virtually impossible to directly measure, which can come into play in determining the patterns of network dynamics.

The final plot shows a pattern that, for want of a better name, we term a "mess." That is, there are no persuasive patterns of stationarity other than on limited, fine time scales, and occasionally eked out on a larger time scale, but only for one or two window's worth. Such a trace defies description in terms of an IID process. Perhaps it can be modeled with a correlated process such as ARIMA, though the fact that it has variability across a number of time scales suggests that doing so will prove challenging.

Using these general categories, we can then assess their relative frequency in our data. We first note that, as discussed above, 11% of the traces contained no loss at all; obviously, these are stationary over the entire hour. In addition to those, another 62% are of the first type in Figure 6: some loss, but

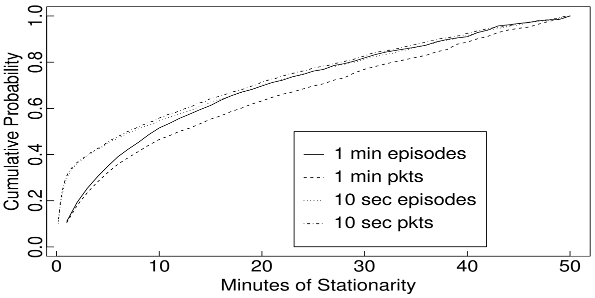

Figure 7: Operational stationarity for packet loss and loss episodes, conditioned on the stationarity lasting 50 minutes or less.

the entire hour consistent with IID loss at all test time scales (1–10 min). Another 12% are stationary at all time scales for at least 90% of the tested windows. These correspond to traces similar to the first type, with slightly more noise so that occasional, fleeting nonstationary regions are detected, indicating either slight deviations from the IID model, or perhaps simply stochastic fluctuations, since NRF is, after all, a 95% confidence test. Another 4% can be classified as spikes or plateaus, with plateaus being quite rare, and the remaining 11% are "messes," having no obvious pattern.

If, however, we restrict our analysis to traces with an overall loss episode rate of $\geq$ 1%, then the picture changes considerably. We find that 21% are stationary at all time scales at least 90% of the time, 4% are nonstationary due to spikes or plateaus, but the remaining 75% are nonstationary messes. These are dominated by a few particularly lossy paths, however, and more data will be required to discern between whether the "messes" are due to those paths in particular, or high loss rates in general.

4.5 Operational stationarity of loss rate

In this section, we leave behind mathematical modeling and conduct a brief assessment of the degree of stationarity in our datasets 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 7 plots the weighted CDF for four different loss series associated with each trace: 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. The CDF is weighted by the size of the stationarity interval; thus, we interpret the plot as showing the unconditional probability that at any given moment we would find ourselves in a stationarity interval of duration T or less. For example, about 50% of the time we will find ourselves in a stationarity interval of 10 min or less, if what we care about is the stationarity 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 min long or less. We characterize longer intervals separately, as these reflect entire datasets that were operationally stationary. Since our datasets spanned at most one hour, stationarity over the whole dataset provides a lower bound on the duration of stationarity, rather than an exact value, and hence differs from the distributions in Figure 7.

For the four loss series, the corresponding probabilities of observing a stationarity interval of 50 or more minutes are 71%, 57%, 25%, and 22%. We can interpret these as follows. If we only care about stationarity of loss viewed over 1-minute periods, then about two-thirds (57–71%) of the time, we will find we are in a stationarity 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 stationarity.

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 stationarity: 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 stationarity, then we find very similar curves. In particular, if we are not in a period of long-lived stationarity, 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 stationarity, regardless of whether we consider one-minute or 10-second stationarity, 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.

5 Throughput stationarity

The last facet of Internet path stationarity we study is end-to-end throughput. That is, to what degree is the throughput observed by an application consistent with that observed by subsequent instances of the application? Quantifying application throughput in a useful way can be very difficult, as different applications have different patterns of network use and different network performance requirements. In an attempt to balance between measuring a quantity sufficiently close to what applications do to not be completely irrelevant, and yet sufficiently application-independent as to retain a modicum of generality, we instrumented 1 MB TCP transfers at user-level. For each measurement instance, the sending NIMI host would open a TCP connection to a server activated by the receiving NIMI host, and, once the connection was established, send 1 MB to the other in a unidirectional bulk transfer. The receiver would time the elapsed interval between when the connection was established and when it terminated, and re-

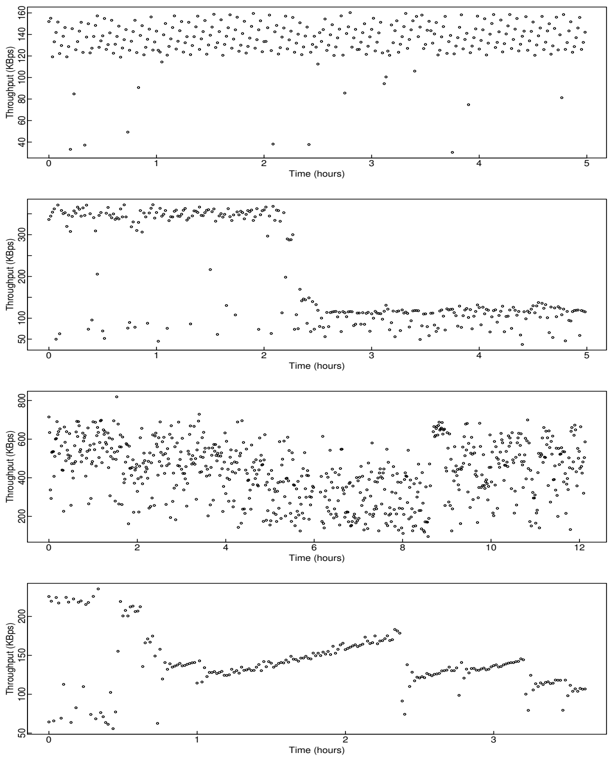

Figure 8: Different throughput patterns: IID, level shift, mess, trend.

port that figure as the total transfer time, very easy to convert to a throughput figure.

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 is fairly good for time periods up to several hours [Pa99a].

The data we analyzed consisted of 47 runs of either 5 or 12 hours (over the daytime busy period) during which the sending NIMI host would initiate one 1 MB TCP transfer to the receiving host every minute. Despite using socket calls to increase the TCP send and receive buffers to 200 KB, a number of the systems clamped the buffers at 64 KB, because the systems were configured to not activate the TCP window scaling option ([JBB92]; the *net.inet.tcp.rfc1323* `sysctl` variable for FreeBSD).

All in all, we successfully measured 17,878 TCP transfers.

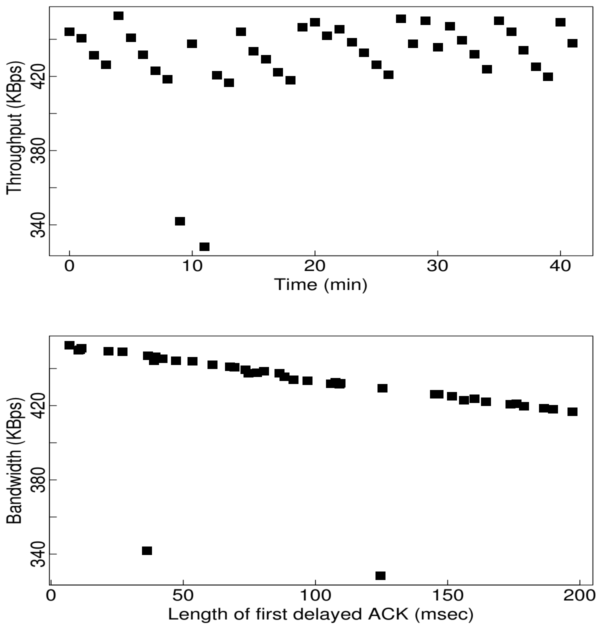

Figure 9: Throughput periodicity (top) and relationship between throughput and length of first delayed acknowledgment (bottom).

5.1 Throughput stationarity analysis

Figure 8 shows some of the different types of throughput dynamics we observed. The top shows a five hour dataset for which the entire run is well-modeled as IID (see further discussion below). The next plot shows a clear level shift from one throughput value to another. If the trace is split at the point of the shift, then both halves are well-modeled as IID, though without this the whole trace is not. The third plot shows a "mess"—throughput figures vary by a factor of five, with little apparent pattern other than a dip between hours 6 and 8. The final plot shows quite baffling behavior: a slow but steady climb in throughput from 120 KBps to 170 KBps over the course of more than an hour, followed by an abrupt return to 120 KBps, and another slow but steady rise.

Upon inspection, the top plot shows a striking pattern during the first two hours: a series of downward sloping lines that chart throughput varying from 160 KBps to 120 KBps. To investigate this behavior, we conducted additional measurements in which we used a packet filter to trace the individual packets of a series of 1 MB TCP transfers. The top half of Figure 9 shows a similar sawtooth pattern from such a trace, with throughput varying from 420–450 KBps.

The bottom half of the figure reveals the explanation for the pattern. When a TCP connection begins, after the SYN handshake the sender's congestion window is set to a single packet to begin "slow start." It sends this packet, but when it arrives, the receiver does not immediately acknowledge it but instead implements TCP's delayed acknowledgment mechanism, whereby it waits on a timer for additional data to arrive (though in this case, none can, since the congestion window does not permit it) prior to acknowledging. While waiting on the timer, the connection is completely stalled, because an ACK is required to advance the flow control and congestion windows to permit new data to be sent. Thus, each connection incurs a delay simply due to waiting on the delayed acknowledgment timer. Furthermore, it only incurs this penalty once, since as soon as two or more packets are in flight, the receiver will generate ACKs without any additional delay.

We have accordingly plotted the duration of the first delayed acknowledgment, i.e., the incurred penalty, along the X-axis, and the total throughput attained for the 1 MB transfer along the Y-axis. That the different measurements fall on a clear line demonstrates that the difference in measured throughput is entirely explained by the timer penalty—it is the only source of variability for these connections! (The two lower throughput figures reflect connections that incurred a retransmission.) Finally, the reason that the sawtooth pattern occurs is due to the use in many TCPs, including those in the NIMI infrastructure, of a "heartbeat" timer that chimes independently from the exact 200 msec timer interval the TCP would like. As the connection's initiation time moves in phase relative to the heartbeat, an increasingly short timer interval results, diminishing to 0 msec, until finally the interval wraps in phase and the delay penalty returns to 200 msec.

Understanding this effect is important, as otherwise we could erroneously conclude that there are complicated network dynamics that lead to various non-stationarities in our measurements. Unfortunately, this simple explanation does not suffice to explain the bottom plot in Figure 8. There, the difference in time between the connections at the low end of the throughput ramp and the high end is 2.2 sec, too much to be explained by a single timer. We are pursuing gathering additional measurements to diagnose this phenomenon.

We now turn to characterizing the degree to which throughput is well-modeled as stationary. We first note an important effect regarding the plateaus as seen in the second plot of Figure 8. From packet-level measurements, we confirmed that the highest such plateaus correspond simply to regions over which no packets in the TCP transfer were lost. This observation highlights the close coupling between the throughput process and the loss process, and suggests that they will likely have similar stationarity properties.

To test for the different throughput time series being well modeled as IID, the Box-Ljung test is again appropriate, since if the data is not IID, then we expect it to most likely be marred by short-term correlations. Doing so for 6 lags, we find that 6 of the 47 traces are well-modeled as IID over their entire duration; another 26 can be split at a level shift into two or three IID regions; 6 can be split into an IID half and a remaining "mess"; and 9 are messes for which we did not find a way to bifurcate the trace into at least one IID region. Thus, we find that throughput is often well-modeled as coming from a stationary IID process for periods of hours.

Note that this does not mean that the throughput does not vary over those hours. As noted in the Introduction, stationarity means that it is well described by a statistical process with fixed parameters, but one of those parameters can be the variance of the process, and, if large, the stationary process will exhibit significant fluctuations. It is here that we particularly run into the important distinction of stationarity in mathematical versus operational terms. We find a number of traces that are IID but exhibit large enough bandwidth fluctuations that operationally we would likely not want to consider stationary; we also find traces that operationally appear stationary, even though statistically they are not. Accordingly, we finish our analysis with an operational assessment of throughput stationarity.

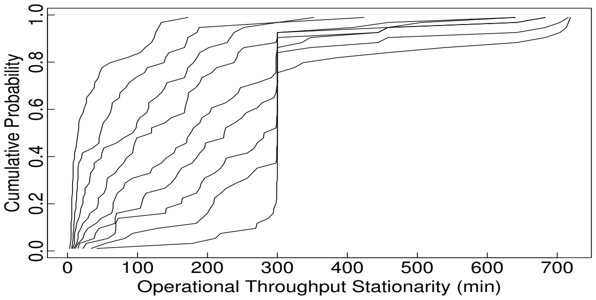

Figure 10: Distribution of maximal operational throughput stationarity regions for $P = \pm 10\%$ (leftmost), ..., $P = \pm 90\%$ (rightmost).

5.2 Operational stationarity of throughput

We adopt a simple notion of operational throughput stationarity, namely whether the observed bandwidth stays within $\pm P$ percent of a given value. For each measurement point in a trace, we see for how many successive points we can find a midpoint such that all the points are within $\pm P$ of one another. We then take the largest such run of points as defining the maximal stationarity region characterizing that trace's operational throughput. If this value is an appreciable fraction of the duration of the trace, then the trace exhibits good operational stationarity; if only a small portion of the trace, then the throughput consistently alternates over time.

Figure 10 shows the distribution of the size of the maximal stationary regions, for $P = \pm 10\%$ (leftmost) through $P = \pm 90\%$. We see that if our operational requirement is for bandwidth not to vary by more than $\pm 10\%$, then we will only have a few minutes of stationarity, but as P increases, so too does the maximal stationarity, fairly steadily, until for $P = \pm 50\%$ it is around 3 hours. (The abrupt increase at 300 minutes is artefact of some traces being only 5 hours long, while others were 12 hours.)

6 Summary

One of the modern tenets of Internet design is that applications (or lower level transport protocols) should be *network-conscious* and *adaptive*; that is, they should monitor network conditions and respond appropriately. This necessarily requires using measurements from the recent past to guide future behavior. The success of this strategy depends on the extent to which such measurements are good predictors of the future or, equivalently, are operationally stable.

In addition, analytic models of networks are very often couched in terms of stationary properties, such as IID packet losses. One particularly important example concerns self-similar models of network traffic [LTWW94], because undetected nonstationarities can easily be misinterpreted as self-similar behavior.

In this paper we have attempted to shed some light on the character of Internet stationarity. Our study has definite limitations, and we propose it as merely an initial effort to grapple with the issue. However, in the end, we also have a sufficient range of findings that it is appropriate to provide a summary (and necessarily oversimplified) answer to the basic question: to what degree is the Internet well described as stationary? As we have seen, the answer is complicated and depends on *where*, *what*, and *when*.

Where: The NIMI routing dataset is substantially more stable than the traceroute server dataset. In addition, some of the routes showed substantially more nonstationarity than others. These two findings indicate that portions of the Internet differ significantly in terms of stability. While we cannot make overly general statements about the Internet as a whole on the basis of our limited datasets, we suspect that the inhomogeneity of stability is a pervasive phenomenon.

What: Overall, routes appear to be very stable. Even for those routes with the least stationary behavior, there is a dominant route which is found the majority of the time. However, a significant minority of routes (1/6–1/3) at least sometimes exhibited rapid change. The loss and throughput data were considerably less stationary, overall, and the throughput data in particular highlighted the potentially significant differences between mathematical and operational stationarity.

When: Perhaps the most interesting aspect of our findings is that stationarity depends critically on the time scale on which one looks. For example, runs of consecutive losses result in nonstationarities on time scales of 100's of msec. On time scales of seconds to minutes, IID loss episode models apply very well. On time scales of 10's of minutes, stationarity is often lost, particularly during times with high loss rates, but we can also observe stationarity for hours on end, both due to times of very little loss, and sometimes due to sustained-but-consistent loss (one dataset maintains a 17% loss rate IID over an entire hour). In general, though, once we reach time scales of hours, then diurnal patterns can come into play, and the picture becomes more complicated still.

It is a difficult picture to untangle, and likely, at best, to resolve into a set of generalities that apply only to different modal regions such as busy hours and off hours. But the preliminary success of some of our IID models is encouraging, and suggests the intriguing possibility that, with sufficient additional work in this area, tractable stationarity models of wide applicability may emerge.

7 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; Mark Allman for the bulk-transfer measurement software; and Fred Baker and Greg Steele for their help with investigating the periodic loss phenomenon shown in Figure 3.

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.

[FJ94] S. Floyd, and V. Jacobson, "The Synchronization of Periodic Routing Messages," *IEEE/ACM Transactions on Networking*, 2(2), p. 122–136, April 1994.

[Ja89] V. Jacobson, "traceroute," *ftp://ftp.ee.lbl.gov/traceroute.tar.Z*, 1989.

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

[LTWW94] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, "On the Self-Similar Nature of Ethernet Traffic (Extended Version)," *IEEE/ACM Transaction on Networking*, 2(1), pp. 1-15, Feb. 1994.

[LB78] G. Ljung, and G. Box "On a Measure of Lack of Fit in Time Series Models," *Biometrika '65*, pp. 553–564, 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.

[Pa97] V. Paxson, "End-to-End Routing Behavior in the Internet," *IEEE/ACM Transactions on Networking*, 5(5), pp. 601–615, Oct. 1997.

[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.

[Pa99a] V. Paxson, "End-to-End Internet Packet Dynamics," *IEEE/ACM Transactions on Networking*, 7(3), pp. 277–292, June 1999.

[PSC99] The TCP-Friendly Website, Feb. 1999. http://www.psc.edu/networking/tcp_friendly.html

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

[St94] W.R. Stevens, "TCP/IP Illustrated, Volume 1: The Protocols," Addison-Wesley, 1994.

[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.