Chapter 17

Summary

We endeavored in this work to characterize a number of aspects of end-to-end Internet dynamics in general, meaningful ways. The Internet's great diversity makes this undertaking immensely challenging.

At the heart of our study lies the NPD measurement framework, in which a number of sites around the Internet run a specialized daemon that provides measurement services to authenticated users. The key scaling property of this framework is that, for N participating sites, it can probe $O(N^2)$ Internet paths. This scaling enabled us to probe over 1,000 Internet paths, due to the participation of 37 sites. Consequently, the data for our analysis is more than an order of magnitude richer than that available for previous end-to-end studies, and a serious argument can be made that we can indeed extrapolate our findings to conclusions about Internet paths in general.

17.1 The routing study

In Part I, we used the NPD framework to study the dynamics of end-to-end routing in the Internet, using two experimental runs, one at the end of 1994 and one at the end of 1995. The results were discussed in Chapter 2; here, we briefly summarize them.

We began by characterizing routing pathologies, as we must first identify anomalies before proceeding to analysis of more typical behavior, lest they skew our results. We cataloged a number of pathologies, including loops, outages, and flutter. Furthermore, the prevalence of pathologies significantly increased between the 1994 dataset and the 1995 dataset, indicating that routing degraded over the course of 1995.

We next analyzed routing stability, first developing a distinction between two orthogonal types of stability, routing “prevalence” and routing “persistence.” We found that most Internet paths are heavily dominated by a single dominant route, but that the length of time over which routes persist varies greatly, from seconds to many days.

We finished our look at routing with an assessment of routing symmetry. While asymmetries have little direct impact on end-to-end performance, they introduce significant measurement problems, because they cloud the accuracy of the easiest form of measurement, “sender-only” measurement, in which no receiver cooperation is required. We found that about half of all Internet routes exhibited a major asymmetry, in which at least one city differed between the route from A to B versus that from B to A.

17.2 The packet dynamics study

The goal of Part II of our study was to use the NPD framework to measure end-to-end Internet packet dynamics. We recorded over 20,000 TCP transfers at both sender and receiver, again in two experimental runs. Faced with such a large volume of data, we adopted the strategy of developing an analysis tool, `tcpanaly`, for automating the "micro-analysis" of individual connections.

Our goal was to develop meaningful characterizations of end-to-end packet delays. To do so required a great deal of preparatory work, to assure that the analysis rested upon sound measurements.

17.2.1 Measurement calibration and TCP behavior

We first needed to devise techniques for calibrating the measurement data, to assure that we did not misinterpret measurement artifacts for *bona fide* networking effects. The measurement process could fail in two basic ways: by misrecording which packets traversed the network, and by misrecording the times at which they appeared. We found that packet filters can: fail to record packets; record packets more than once; truncate the beginning or end of trace files; and rearrange the sequencing of packets. Accordingly, we developed tests so that `tcpanaly` can detect these events. We further developed the important notion of a packet filter's "vantage point," meaning where in the network path it observed the traffic. A filter's vantage point can introduce ambiguities in the apparent chain of cause-and-effect, which can only be removed with considerable care.

Hand-in-hand with calibrating the integrity of the traffic traces comes the problem of identifying the exact behavior of the TCP implementations used by the sending and receiving hosts. Often, the only way to accurately gauge the integrity of a traffic trace is by knowing in intimate detail how the TCPs participating in the connection behave and respond. Apparent deviations from this behavior then indicate a likely lack of integrity in the traffic trace, if the behavior has indeed been correctly characterized.

`tcpanaly` holds promise as a valuable tool for analyzing TCP behavior, useful both in its own right for diagnosing performance and congestion problems, and also as a way to account for the separate effects on a connection's dynamics of the behavior of the TCP endpoints versus that of the connection's Internet path. In the course of its development, we found a wide range of TCP behaviors, some of which have major, negative performance and stability implications for the associated TCPs. The most serious problems include excessive retransmissions and failures to correctly diminish the transmission rate during periods of congestion. Indeed, if some of these TCPs were ubiquitous in the Internet, the network would quite simply cease to function, due to "congestion collapse."

In the process of this analysis, we observed that the TCPs with the most serious problems were the only two in our study written independently from the "BSD-derived" implementations that directly benefited from much of the fundamental TCP research. To investigate this observation, we analyzed three additional implementations, finding a mid-level performance problem in one, a major performance problem in another (but one possibly due to use of a specific network interface card), and severe performance and stability problems in the third. Thus, our findings strongly argue that implementing TCP correctly is exceptionally difficult. Given that Internet stability *relies* on TCP correctness, it therefore behooves the Internet community to take energetic steps towards providing analysis tools and reference implementations to aid the efforts of implementors.

17.2.2 Timing calibration

Armed with the ability to detect inaccurate packet traces and to distinguish between TCP-induced effects and networking effects, we next turned to the difficult problem of calibrating the packet timings. The effort continued to be driven by the ultimate goal of analyzing end-to-end packet delays. To do so requires comparing pairs of unsynchronized clocks, namely those used by the tracing programs at the sender and receiver. We developed algorithms for (1) estimating clock resolution, (2) synchronizing clocks *post facto*, (3) detecting clock adjustments, and (4) detecting and removing relative clock skew. This last is particularly important because, if undetected, relative clock skew leads to variations in apparent packet delays quite similar to those of genuine networking effects. We found that it is fairly common for a pair of clocks to exhibit discernible relative skew. We also found that the fact that two clocks agree quite closely does *not* eliminate the possibility that the clocks suffer from problems such as adjustments and relative skew.

17.2.3 Network pathologies

With our measurements fully calibrated, we could then turn to analyzing packet dynamics. We began by characterizing packet-forwarding pathologies: out-of-order delivery, packet replication, and packet corruption. We found that the frequency with which packets arrive in a different order than sent varies enormously among Internet paths. While reordering often occurs in conjunction with the route “flutter” pathology, we also observed numerous instances in which it occurred in the absence of flutter, and some instances in which massive reordering events occurred due to “pauses” in router forwarding. Finally, the possibility of reordering limits how quickly a TCP sender can infer a packet loss using the “fast retransmission” mechanism. We investigated whether, based on our data, this mechanism could be altered to retransmit more efficiently. We found that we could only do so if we required changes at both the TCP sender and receiver. Consequently, we might as well instead change the sender and receiver to use the more sophisticated TCP “selective acknowledgement” extension, now being standardized [MMFR96].

We found that the curious phenomenon of packet replication—the network delivering a single packet more than once—does indeed occur, but it is exceptionally rare. On the other hand, our analysis of packet corruption suggests that, overall, about 1 Internet data packet in 5,000 arrives with data different than what was originally sent. This rate is high enough that, given TCP's 16-bit checksum, about one packet in 300,000,000 will be accepted with undetected errors. The Internet carries many more packets than this each day.

17.2.4 Estimating bottleneck bandwidth

We next turned to the problem of identifying a network path's *bottleneck bandwidth*. We needed to do so before analyzing packet loss and delay because the bottleneck bandwidth determines what we call the “self-interference time constant,” Q_b. Two data packets of size b sent less than an interval Q_b apart must necessarily queue at the bottleneck element of the network path. Thus, knowledge of Q_b enables us to determine which of our measurement probes were perforce correlated. It further plays a major role in assessing packet loss, because we want to distinguish between the loss of data packets that we know had to queue behind their predecessors (“self-interference”), versus those lost even though they did not have to queue on account of the connection's own loading of the network path.

We discussed how the main existing technique for estimating bottleneck bandwidth, “packet pair,” could produce incorrect estimates. These can occur in the presence of: excessive noise; packet reordering; changes in the bottleneck bandwidth; or network paths in which the bottleneck is comprised of multiple, separate channels or links. This last case is particularly interesting, because it leads to erroneously large bottleneck estimates even if the network is completely quiescent. The problem lies in the fundamental assumption made by packet pair that packets must queue behind one another at the bottleneck and be served by it one at a time. For a multi-channel or multi-link bottleneck, however, this assumption does not in fact apply, and a pair of packets can traverse the bottleneck without it altering the spacing between them.

These observations motivated us to devise a robust algorithm for estimating bottleneck bandwidth, based on “packet bunch modes” (PBM). By focussing on identifying multiple modes in the distribution of the estimated bottleneck bandwidth, PBM can accommodate errors introduced by noise, as well as detecting changes in bottleneck bandwidth and the presence of multi-channel links. By using receiver-based measurement, it also can cope with packet reordering, and with the possibility of asymmetries in the bottleneck bandwidths along the two directions of a network path.

We calibrated PBM by testing whether we could associate known, common link speeds with its estimates. We found that we could almost always do so. Once we had faith in PBM's accuracy, we could then test other estimation methods against PBM to see how well they perform. We found that receiver-based packet pair performs almost as well, if we can tolerate failing to detect shifts in bottleneck bandwidth or multi-channel links, both of which prove rare. Sender-based packet pair, however, does not perform nearly as well, due to the additional noise incurred by measuring timings that reflect the traversal of packets in both of a path's directions. Finally, we find that about 20% of the time, a path's two directions have *asymmetric* bottleneck bandwidths, but that, along a single direction, the bottleneck generally remains constant over lengthy periods of time.

One drawback with PBM is that it is ad hoc to an unsatisfying degree. It uses a considerable number of heuristics that can only be defended on the basis that they appear to work well in practice. We found this acceptable (if regrettable), because for our study bottleneck bandwidth estimation was fundamentally only a stepping stone to the later analysis, and not an end in itself. We hope, however, that the basic ideas underlying PBM—searching for multiple modes and interpreting the ways they overlap in terms of bottleneck changes and multi-channel paths—might be revisited in the future, in an attempt to develop them in a more systematic fashion.

17.2.5 Packet loss

We now could turn to analyzing patterns of packet loss in the Internet. We found that over the course of 1995, packet loss rates *nearly doubled*, indicating a marked degradation in service. However, these rates required further inspection to understand their implications. We first developed the notion of the network having two general states, “quiescent,” corresponding to periods of no loss, and “busy,” corresponding to periods in which connections observe at least one loss. The proportion of quiescent connections did not change appreciably during 1995; instead, the loss rate increases were due to higher levels of loss during busy periods.

We also distinguished between three different types of lost packets: “loaded” data packets, meaning those that necessarily had to wait at the bottleneck behind one or more of their predecessors; “unloaded” data packets, meaning those that did not have to queue behind predecessors, unless cross traffic arrived and delayed their predecessors; and acknowledgements.

We found that loaded packets are much more likely to suffer high loss rates than unloaded packets, which is not surprising, since they encounter not only the ambient network load but that of their predecessors; and that acks are more likely to be lost than unloaded packets (or even loaded packets, for high loss rates). We interpret these findings as reflecting the fundamental difference between data packets being sent at a rate that *adapts* in an effort to diminish packet loss, and acks being sent at a rate that does *not* adapt to the rate at which acks are lost. This finding highlights how the loss rates observed by a TCP connection's data packets *differ* from the unconditional loss rates along the path they traverse.

The last comparison between data packet and ack loss rates we made was to determine the degree of correlation between the two rates for a single connection. We found that the two are nearly uncorrelated, indicating that this fundamental property of a network path is *asymmetric*.

We next found that different major regions of the Internet—the United States, Europe, and connections from one to the other—experienced very different loss rates. Then, after showing that loss rates follow the well-known diurnal cycle reflecting working hours and off-work hours, we analyzed variations in the time of day during which our measurement apparatus succeeded in executing a measurement. For North American sites, these successes were uniformly spread over the 24 hours of each day. For European sites, though, the frequency of successes dipped to low points in patterns that closely matched the loss-rate cycle, indicating that our European measurements suffered from a discernible *bias* towards underestimating loss rates.

Another question we investigated was whether packet loss events are well-modeled as independent, since this assumption is sometimes made when theorizing about network behavior. We found that loss events are instead strongly correlated. Furthermore, the duration of loss “outages” exhibits infinite variance, which accords with a recent model of how individual connection behavior can give rise to “self-similar” aggregate traffic behavior [WTSW95].

We then looked at the question of *where* packets are lost along an Internet path. In particular, whether they are lost before or after the bottleneck element. From careful analysis of timing information we can sometimes distinguish between these two. We found that, while most losses occur at or before the bottleneck, a significant minority (roughly 25%) occur after.

We next evaluated how packet loss rates evolve over time, with an eye towards gauging the efficacy of caching packet loss statistics associated with a path in order to predict future path performance. We found that a path's state, in terms of “quiescent” or “busy,” is a good predictor of its future state for many hours, but a path's observed loss rate is *not* a good predictor of its future loss rate.

We then investigated how efficiently TCP implementations retransmit. We found that, for some implementations, the large majority of their retransmissions are unnecessary. Fixing these implementations and deploying the SACK extension would eliminate nearly all of the unnecessary retransmissions.

17.2.6 Packet delay

We finished our study with an analysis of end-to-end packet transit delays. We found that both round-trip times (RTTs) and one-way transit times (OTTs) exhibit great “peak-to-peak” variation. OTT variations for the most part are asymmetric. The only clear correlation occurs between the order-of-magnitude (logarithm) variation in the two directions. On the other hand, OTT variation is clearly correlated with packet loss rates, as we would expect. We further found that OTT variation is not a good predictor of future OTT variation, in accord with the finding that packet loss rates are not good predictors of future loss rates.

We then turned to an assessment of packet *timing compression*, in which a group of packets arrives at their receiver more closely spaced than when they were sent. We identify three types of compression: ack compression, data packet compression, and receiver compression. Each requires somewhat different assessment considerations. Overall, none of the three types occur frequently enough to pose a significant problem in terms of network performance and stability. Their presence does, however, complicate path measurement efforts, which must use judicious filtering to avoid mistaking compression events for different network effects, such as a temporary increase in bottleneck bandwidth.

We next investigated the *time scales* over which queueing occurs, by determining on which time scales we observed the maximum sustained and peak OTT variations. We found that both occur most frequently on time scales of about 100–1000 msec, though, as with many Internet phenomena, we also found a wide range of behavior beyond this region. (In particular, we sometimes found maximal queueing occurring on much longer time scales.)

The last aspect of packet delay we analyzed was the degree to which it reflects *available bandwidth*. We did this by studying the ratio between the delay a packet incurred due to its connection's own loading of the network path, versus the total delay it incurred. This ratio correlates well with the overall throughput achieved by a connection. However, we also showed that the accuracy of the ratio is diminished by the presence of errors in estimating the bottleneck bandwidth.

We observed a distinct decrease in available bandwidth over the course of 1995, though we also observed significant regional variation, with U.S. sites enjoying considerably more available bandwidth than European sites. Finally, we investigated how available bandwidth evolves over time. We found that a connection's available bandwidth is a fairly good predictor of future available bandwidth out to time scales of hours.

17.3 Future research

There are three general areas of future work suggested by our research. First, our original goal when proposing the research was to use end-to-end measurements to drive the development of new algorithms for how transport protocols can adapt to changing network conditions. We had to abandon this goal once the scope of analyzing the measurements themselves became apparent, but clearly an important potential benefit of end-to-end characterization such as we have undertaken is to better optimize how connections use the network.

Closely related to developing such new algorithms is the question of *fast* estimation of Internet path behavior. The algorithms we developed for calibrating network clocks (Chapter 12), estimating bottleneck bandwidth (Chapter 14), and assessing queueing time scales and available bandwidth (Chapter 16) all in their present form analyze entire connection traces. Yet, transport connections clearly need to make decisions based on path properties quickly, and cannot afford the luxury of analyzing the fate of several hundred packets. Our work, though, can play a key role in developing fast estimation techniques, because the algorithms we developed can then be used to calibrate the faster algorithms.

Finally, the NPD framework serves well to address the issue of capturing reasonably representative samples of a cross-section of Internet path behavior. Another important form of Internet heterogeneity, however, is how Internet traffic changes *with time*. Only longitudinal studies can address such "temporal" heterogeneity. We have attempted to touch on this issue by capturing two datasets spaced a year apart. Clearly, though, we need longer-term studies to develop solid conclusions about traffic trends. We believe this goal can be met in conjunction with the development of an Internet "measurement infrastructure," that is, large-scale deployment of NPD-like measurement platforms. We do not claim that the NPD framework can simply be scaled up to serve as this infrastructure; indeed, the problem of an infrastructure that *can* scale to the full Internet is the key research problem for the infrastructure. But, if accomplished, such an infrastructure could serve, through the accumulated archives of its measurements, as the basis for longitudinal studies; and, even more significantly, as a mechanism for assessing and improving the overall health of the network.

17.4 Themes of the work

Several themes emerge from our study:

- The N^2 scaling property of our measurement framework serves to measure a sufficiently diverse set of Internet paths that we might plausibly interpret the resulting analysis as accurately reflecting general Internet behavior.

- To cope with such large-scaled measurements requires attention to calibration using self-consistency checks; robust statistics to avoid skewing by outliers; and automated "micro-analysis," such as that performed by `tcpanaly`, that we might see the forest as well as the trees.

- With due diligence to remove packet filter errors and TCP effects, TCP-based measurement provides a viable means for assessing end-to-end packet dynamics.

- We find wide ranges of behavior, so we must exercise great caution in regarding any aspect of packet dynamics as "typical."

- Some common assumptions such as in-order packet delivery, FIFO bottleneck queueing, independent loss events, single congestion time scales, and path symmetries are all violated, sometimes frequently.

- The combination of path asymmetries and reverse-path noise renders sender-only measurement techniques markedly inferior to those that include receiver cooperation.

This last point argues that, when the measurement of interest concerns a unidirectional path—be it for measurement-based adaptive transport techniques such as TCP Vegas [BOP94], or general Internet performance metrics such as those in development by the IPPM effort [A+96, Pa96a]—the extra complications incurred by coordinating the sender and receiver are worth the effort.

Finally, we believe an important aspect of this work is how it might contribute towards developing a "measurement infrastructure" for the Internet: one that proves ubiquitous, informative, and sound.