CHAPTER 8

Conclusions and Future Work

> ***real**: (2b3) existing as a physical entity and having properties that deviate from an ideal, law, or standard.*
>
> — Merrian–Webster English Dictionary

> *There are sadistic scientists who hurry to hunt down errors instead of establishing the truth.*
>
> — Marie Curie (1867–1934)

This dissertation proposed and evaluated a new approach for generating realistic traffic in networking experiments. Our construction relied on several components to form a coherent solution to this problem:

1. The a-b-t model of source-level behavior, which provides a *generic but detailed* way of describing source-level behavior that is applicable to any Internet application.
2. An *efficient measurement* method for accurately translating the packet header trace of any arbitrary TCP connection into its a-b-t connection vector, even in the presence of packet reordering and retransmission.
3. The source-level trace replay method for generating traffic in a closed-loop manner, which provides a way of introducing fully *reproducible* synthetic traffic in networking experiments.
4. The ability to directly compare original traffic and its source-level replay, after incorporating network parameters also derived from packet header analysis. Such a comparison enables us to *assess the realism* of the synthetic traffic.
5. A method for resampling a-b-t connection vectors that supports both the introduction of *controlled variability* in the generated traffic and the *predictable scaling* of the offered load.

The rest of this chapter discusses these components[1], highlighting some concrete contributions and open

---

[1] Also known as the five pillars of Abtism. [*sic*]

questions, which could be the subject of future work. Our focus is on the larger scheme of things, so we refer the reader to the summaries of each chapter for additional findings and possible refinement of our methodology.

8.1 Empirical Modeling of Traffic Mixes

The main problem solved by our approach is generating closed-loop traffic consistent with the behavior of the entire set of applications in modern traffic mixes. Unlike earlier approaches, which described individual applications in terms of the specific semantics of each application, we proposed to describe the source behavior driving each connection in a generic manner using the a-b-t model. This is consistent with the view of traffic from TCP, which does not concern itself with application semantics, but only with sending and receiving Application Data Units (ADUs) as demanded by the applications. The a-b-t model provides an intuitive but detailed way of describing source behavior. It also satisfies a crucial property: given a packet header trace collected from an arbitrary Internet link, we can algorithmically infer the source-level behavior driving each connection, and cast it into the notation of the a-b-t model.

Section 3.3 described our inference algorithm, whose asymptotic cost is is $O(sW)$, where s is the number of segments in a connection and W is the maximum size of the TCP receiver window (in segments). The foundation of the analysis is the *logical data order* that can be established between segments of the same connection. This order corresponds to the application-layer order of the data carried in each segment. From this order, we can accurately identify individual ADUs *without* any timing analysis. Furthermore, the handling of retransmission and reordering becomes very generic, eliminating the need to handle the many possible cases one by one. Our validation using traffic from synthetic applications with known source behavior demonstrated the robustness of our analysis to segment loss and reordering, and to the way in which endpoints use sockets (*i.e.*, using different sizes and timings of I/O operations).

Overall, our algorithmic approach enables us to model traffic in an automated manner in a question of hours. This addresses a major difficulty with earlier efforts targeted at individual applications, which required months to be completed and were hardly ever updated. One future direction is develop an online implementation of the algorithm, which would enable us to model traffic mixes in real time. The $O(sW)$ cost of our analysis makes this online processing feasible. Efficient memory management is the main challenge, since each connection would require separate state during its lifetime. It seems possible to restrict this per-connection state to the current ADU in each direction, which is much more efficient than keeping track of entire connection vectors. Real-time modeling has several benefits. First, the set of a-b-t connection vectors is between tens and hundreds of times smaller than packet header traces from which it derives. This would enable researchers to study traffic at the source-level for much longer periods that it is possible nowadays. Second, real-time modeling can remain active indefinitely, which makes it possible to observe unusual but important phenomena, such as flash crowds, BGP failures, *etc.* To satisfy storage constraints, uninteresting traffic can be periodically thrown away.

In our study, we identified a fundamental dichotomy between applications that exchange ADUs in a sequential manner and those that do it concurrently. Sequential communication follows an alternating sequence of ADUs sent in opposite directions, where ADUs from one endpoint usually play the role of requests and ADUs from the opposite endpoint play the role of responses. One important property of this pattern is that each ADU exchange must necessarily take one round-trip time. As a consequence, the duration of sequential connections often has little to do with the amount of transferred data, being dominated by the number of request/response pairs. For this reason, sequential connections usually show far lower throughputs than one would expect from their total number of bytes. SMTP provides a good example of this phenomenon, since most SMTP connections carry little data but take rather long to complete. As illustrated in Figure 3.3, this is mostly due the substantial number of control ADUs required by this protocol.

Concurrent communication supports the sending of ADUs from both endpoints at the same time. This is the natural model both for applications without requests and responses, and for applications that are able to pipeline their requests and responses. Pipelining eliminates the need to spend one full round-trip time to complete each request/response exchange, which can substantially increase throughput. The analysis of our collection of traces revealed that the number of connections that exhibit concurrent data exchanges is small (0.9-3.6%), but that they account for a far larger fraction of the total bytes in the traces (12.1-31.9%). This is consistent with the observation that concurrency can increase overall throughput, so application protocol designers are more compelled to use concurrency in applications that exchange large amounts of data. BitTorrent is a prominent example of data concurrency, where we can observe simultaneously natural concurrency (both endpoints send and received requests and file pieces), and pipelining (multiple requests and file pieces can be outstanding at any point in time). Figure 3.9 showed one example of this behavior.

Our measurement algorithm can determine whether a connection exhibits sequential or concurrent data exchanging by examining only the sequence and acknowledgment numbers in the segments of a connection, without analyzing of segment arrival times. The basis of our technique is again the logical data order among TCP segments, which is a total order for sequential connections, and a partial order for concurrent ones. The inequalities presented in Section 3.3.2 formalized this idea, providing a method for identifying data exchange concurrency without false positives.

8.2 Refining and Extending our Modeling

Our methodology strongly relies on *non-parametric modeling*. Parametric models are far more compact and can often provide deeper insights than non-parametric ones. However, their use has little to do with the quality of synthetic traffic. A non-parametric model can result in traffic as realistic or more than a parametric model, without the risk of oversimplification. In any case, our a-b-t connection vectors offer a good foundation for building a parametric model of Internet traffic mixes. Our analysis of the relationship between ADU sizes and numbers of epochs in Section 3.5.1 uncovered substantial complexity and a striking lack of consistency among the different links considered in our study. Techniques like Hidden Markov Modeling could perhaps provide the right approach.

Our own related work explored the possibility of attacking this complex modeling problem by decomposing traffic mixes in to a set of fundamental pattern of communication [HCNSJ05]. The idea was to use statistical clustering to find applications that behave in a similar manner, *i.e.*, that follow the same "communication pattern", and to separately model each of the identified *traffic clusters*. For example, interactive applications such as telnet and SSH are very different from file-sharing applications such as Kazaa or Gnutella, so it seems much easier to develop separate models for "interactive applications" and "file-sharing applications" than a single model to encompass both of them. In our exploratory study, we followed a two step process to find traffic clusters. First, we computed a vector of features for each connection, which included statistics such as the median size of the ADUs in the connection, a measure of the directionality of the data exchanges, and the correlation between the sizes of a-type and b-type ADUs. Feature vectors provide a way to compare connections, even if their a-b-t connection vectors have very different forms, and use a distance metric to quantify the similarity between the source behaviors in two connections. Second, we used a hierarchical clustering algorithm to construct a taxonomy of traffic classes based on the similarity among connections. The results of our analysis demonstrated that some clear and intuitive traffic clusters emerged when this procedure was applied to sets of connection vectors derived from real traces. We believe this type of approach can simplify the modeling of traffic mixes. Furthermore, it can also provide a more flexible way of resampling traces, where the fraction of connection vectors from each of the traffic clusters can be changed at will (*e.g.*, increasing of decreasing the fraction of file-sharing-like traffic).

There are other open questions in the modeling of Internet traffic mixes, and their solution is complicated by the need to devise better measurement methods. We can cite the following examples:

- Our modeling of concurrent connections employs two separate connection vectors, one for each direction, eliminating any dependencies among ADUs flowing in opposite direction. This dependencies are certainly present in some cases, at least when concurrency is used to implement pipelining. A refined version of the a-b-t model where the causality between ADUs is specified using an acyclic graph could capture this type of structure. The analysis of sequence and acknowledgment numbers can provide a starting point for understanding ADU dependencies. However, such an approach would result in a substantial number of spurious dependencies that were not really part of the application behavior.
- The a-b-t model has no mechanism to specify dependencies between ADUs in different connections. While more complex forms of the model are possible, there is again great difficulty in determining when these dependencies exists. By analyzing ADU arrival times for the same endpoint, we could hypothesize a dependency. We could further strengthen such an analysis by requiring several instances of the same dependency pattern, *i.e.*, only accepting a timing dependency when several pairs of connections with “similar” ADU sizes and number of epochs are observed.
- A important problem that has received very limited attention in the source-level modeling literature is the possibility of changes in user behavior as a function of network conditions. Such possibility would break the assumption of network independence in source-level models. Our work in this area [PHCM$^+$06] revealed phenomenal difficulties in measuring such dependencies. Even a simple question such as whether users with higher access bandwidths tended to download larger objects was statistically problematic. Our results showed that this trend does not appear to be present in the UNC trace. While substantial differences exists in the access bandwidth of different UNC endpoints (*e.g.*, between wireless and wired end hosts), the number of endpoints with severely limited bandwidth is very small (*e.g.*, few endpoints were behind a modem).

These three problems are unlikely to have straightforward solutions. We also believe that their impact on the quality of synthetic traffic is small, or even insignificant, in empirical studies focusing on large traffic aggregates.

A final question is how to combine source-level modeling and unwanted traffic modeling. Our analysis in Section 4.2.1 showed the need to carefully separate connections with regular data exchanges, for which the a-b-t model is applicable, and other types of connections (*i.e.*, failed connection establishments attempts, port and network scans, *etc.*). While our filtering for regular connections removed only a tiny fraction of the bytes in the traces, the number of individual connections was very large, which may be detrimental for certain studies. In addition, we did not consider how to generate malicious traffic. Our literature review discussed some relevant efforts on this topic. However, they tend to be open-loop. Since malicious traffic can have dramatic effect on the network conditions, understanding its impact on source behavior seems critical. We know of no study that considered this question.

8.3 Assessing Realism in Synthetic Traffic

The result of our packet header processing is a collection of a-b-t connection vectors, which can then be replayed in software simulators and testbed experiments to drive network stacks. Such a replay generates synthetic traffic that fully preserves the feedback loop between the TCP endpoints and the state of the network, which is essential in experiments where network congestion can occur. By construction, this type of traffic generation is fully reproducible, providing a solid foundation for networking experiments where two or more network mechanisms must be compared under similar conditions.

Our experimental work demonstrated the high quality of the generated traffic, by directly comparing traces from real Internet links and their source-level trace replay. This comparison is both a rigorous way of validating the a-b-t model and its measurement methods, and a challenging exercise where each connection vector must be replayed in a TCP connection whose original network conditions are preserved in the experiments. If these network conditions were not preserved, it would be very difficult to determine whether differences between an original trace and its source-level trace replay are due to shortcomings of the a-b-t model or to a lack of realistic network parameters. For this reason, we devote substantial effort to the accurate measurement, purely from packet header traces, of three important network parameters: round-trip times, maximum receiver window sizes, and loss rates. These three parameters have a major impact on the throughput that a TCP connection can achieve. In addition, the testbed experiments in our evaluation of the approach carefully reproduce these parameters, using an extended version of *dummynet* to efficiently simulate per-connection round-trip times and loss rates.

It is important to note that the inclusion of open-loop loss rates in some of our experiments is only a means to achieve a more fair validation of the a-b-t model. A substantial loss rate has a dramatic effect on the characteristics of a connection, so comparing such a connection in the original trace and in a replay without a simulated loss rate tells us very little about the accuracy of the source-level characterization. In general, we always conduct source-level trace replay experiments both with and without simulated loss rate, and compare their results. This type of analysis allowed us to conclude that source-level behavior had a more substantial impact on our traces than losses, but that neither of them can be ignored when trying to understand the characteristics of network traffic. One interesting finding from our experimental work is that simplistic source-level models substantially exacerbate the impact of losses, which may substantially change the conclusions from certain empirical studies.

Our results demonstrated that source-level trace replay can closely approximate the characteristics of real traffic traces. By comparing synthetic traffic with and without detailed source-level structure, we showed that more complete source-level modeling makes synthetic traffic closer or far closer to real Internet traffic. In particular, the largest difference was observed for the time series of packet throughput, the body of the packet throughput marginal and the time series of active connections. Other metrics did not show consistent improvement when detailed source-level modeling is used. However, in these cases, it is often difficult to determine whether the difference between real and synthetic traffic comes from the shortcomings of the source-level model or from the lack of certain network-level parameters. This is the main difficulty with our approach: while providing the most stringent way of evaluating synthetic traffic, it also requires to deconstruct the factors that shape traffic very carefully. While some factors are well understood and can be measured accurately, others are not. In this regard, our work complements current efforts to further understand traffic, provides a way to verifying new theories using an elaborate experimental approach.

One important future direction for our work is to expand the set of metrics used to evaluate the quality of synthetic traffic. At a low level, the distribution of packet sizes provides a good avenue to understand the effect of source behavior on packetization. At a higher level, the distribution of connection goodputs is a particularly good (and demanding) metric to study how closely the modeling (of sources and network parameters) reproduces TCP performance. We could study goodput either by looking at the distribution of connection goodputs directly, or by comparing each replayed connection with its original version and computing relative errors of some sort. Another important high-level metric is response time, which can be easily defined as the duration of epoch for sequential connections. Many studies rely on response time to examine the performance of network mechanism, so it is desirable to validate its experimental reproduction. However, there are several difficulties with this metric. It requires to identify request and response pairs, which are not necessarily the pair formed by ADUs a_i and b_i. The server side initiates the connection in some protocols, while other protocols do not have clearly-defined roles as client and server for their endpoint. It is very difficult to distinguish among these situations purely from packet header analysis. Also, there is no simple definition of response time for concurrent connections. As an alternative, we can use connection duration as a metric, which is always well-defined, but it has far lower resolution.

8.4 Incorporating Additional Network-Level Parameter

While our methods to measure and simulate network parameters appear sufficiently accurate in our experimental evaluation, there are several directions in which this part of the work can be refined. Path round-trip times are not fixed for each connection, but follow a distribution of delays. It seems possible to refine our measurement to incorporate this fact, at least to some extent, into our approach, although the lack of samples for most connections greatly complicates this problem. It is also unclear whether this refinement would have any significant impact on the generated traffic. Improving the measurement and simulation of losses could have a more substantial effect. Figure 4.18 already revealed some level of inaccuracy, and our experimentation revealed the need to take into account pure acknowledgment losses and not just data segment losses. More importantly, the assumption of independent losses and their simulation using random dropping seems unrealistic, which explains some of the differences between original and synthetic traffic.

There are other network parameters that could be taken into account. In general, we believe that only two of them would have a significant impact on the quality of synthetic traffic: maximum segment sizes and path capacity. Maximum segment sizes are straightforward to measure, and their incorporation into the experiments would improve the realism of packetization in the generated traffic. Its implementation in a network testbed experiments requires some careful handling of resources, since maximum segment sizes are often a machine-wide constant. The impact of this refinement is not expected to be dramatic, given that most connections are known to use the same maximum segments size (the one derived from Ethernet's MTU, which we employed in our experiments).

Path capacity presents a much more difficult measurement problem, both when defined as bottleneck capacity and as available bandwidth. Recent work by Huang and Dovrolis [JD04] provides a useful foundation. While it is only applicable with confidence to connections with large amounts of data, “bulk connections”, this is precisely the type of connection whose throughput could be dominated by capacity limits. Throughput in connections with small amounts of data is mostly a function of round-trip time. As discussed in Section 3.3, most connections are in this case. However, bulk connections are responsible for a large fraction of the bytes, so their accurate replay is important. We also believe that combining our ADU analysis with the Huang and Dovrolis approach can provide less noisy samples, improving the accuracy of the method. In the case of capacity, the implementation in the experiment is not difficult by making use of *dummynet* ’s per-connection capacity.

Besides these concrete specific network parameters, we believe that a better understanding of the impact of traffic shapers and end host bandwidth quotas can help to explain some of the differences between source-level trace replay experiments and original traffic. This seems specially relevant for UNC, where the impact of losses appeared rather different from the ones in other sites. We hypothesized that the presence of a major data and software repository known to use bandwidth constraints was behind our finding. Another important factor in traffic characteristics is the growing impact of wireless networks. Our large-scale measurement effort in this area [HCP05], showed an insignificant increase of end-to-end losses in this environment (thanks to link-layer retransmission) but substantial increases in the magnitude and variability of round-trip times.

8.5 Flexible Traffic Generation

The final problem that we considered in this work was how to introduce controlled variability in network experiments, *i.e.*, how to derive from a trace of connection vectors a new trace that still “resembles” the original one. Our solution involves resampling entire connection vectors, fully preserving observed source-level behavior, and assigning them new start times. We gave two methods for this assignment: sampling from an exponential distributions, which results in Poisson connection arrivals, and sampling blocks of connections, which preserves the long-range dependence in the connection arrival process that we encountered in our traces. The first method, Poisson Resampling, is analytically appealing, and supported by empirical data, since the marginal distribution of connection inter-arrival is consistent with an exponential distributions. Block Resampling provides a non-parametric alternative, which is more realistic with regards to the dependency structure of the connection arrival process. This structure did not show any effect on packet and byte arrivals, but it seems important for mechanisms that require per-connection state.

We also showed that our resampling methods can be carefully directed to produce a new trace of connection vectors whose offered traffic load matches an arbitrary target very closely. Such trace scaling is a common requirement in suites of experiments that must expose a network mechanism to a range of traffic loads. The key to our solution is to count the total amount of data in the resamplings, which was shown to be strongly correlated to offered load. On the contrary, our results clearly showed that the number of connections is only weakly correlated to offered load, and cannot be used for accurate scaling of resamplings. While this result is an intuitive consequence of the heavy-tailness in the amount of data carried by connections, the issue has been poorly understood in earlier models, where the parameters that can be controlled to tune offered load were associated with the number of connections. This is for example the case for the number of user equivalents in web traffic models. The traffic load offered by this type of "connection-driven" models can never match a target offered load as accurately as our "byte-driven" resamplings of connection vector traces.

Our work on trace resampling can be extended in several directions. First, there is some need to refine our handling of the packetization overhead, which would result in even more accurate load scaling. Second, our methods only manipulate one trace at a time. Being able to combine multiple traces would provide an even more flexible framework. While it seems straightforward to extend our methods to support this operation, demonstrating the validity of the results appears difficult. It represents a departure from measured traffic into a hypothetical traffic that may or may not be realistic, and it can introduce non-stationarities. Third, developing a broader model of network traffic, either parametric or non-parametric, could provide a better way to guide the resampling process. In this direction, a better understanding of the main patterns of source-level behavior would provide more flexible way of creating hypothetical scenarios. Our work on traffic clusters described above is a step in this direction, since combining clusters support the exploration of a wide range of traffic generation scenarios. The possibility of succinctly describing the range of patterns in a cluster, *e.g.*, file-sharing applications with symmetric bulk transfers and concurrency, is specially useful for exploring future scenarios where applications that only represent a small fraction of the traffic become increasingly important.

BIBLIOGRAPHY

[AA01] Masaki Aida and Tetsuya Abe. Pseudo-address generation algorithm of packet destinations for Internet performance simulation. In *Proceedings of IEEE Infocom*, pages 1425–1433. IEEE, 2001.

[AKM04] G. Appenzeller, I. Keslassy, and N. McKeown. Sizing router buffers. In *Proceedings of ACM SIGCOMM*, August 2004.

[AKSJ03] Jay Aikat, Jasleen Kaur, F. Donelson Smith, and Kevin Jeffay. Variability in TCP round-trip times. In *Proceedings of ACM SIGCOMM Internet Measurement Conference*, pages 279–284. ACM Press, 2003.

[APS99] M. Allman, V. Paxson, and W. Stevens. RFC 2581: TCP congestion control, 1999.

[AV98] P. Abry and D. Veitch. Wavelet analysis of long-range-dependent traffic. *IEEE Transactions on Information Theory*, 44(1):2–15, 1998.

[AW95] Martin F. Arlitt and Carey L. Williamson. A synthetic workload model for Internet Mosaic traffic. In *Summer Computer Simulation Conference*, pages 24– 26, July 1995.

[Bar00] S. Barber. RFC 2980: Common NNTP Extensions, October 2000. Status: INFORMATIONAL.

[BBBC99] Paul Barford, Azer Bestavros, Adam Bradley, and Mark Crovella. Changes in web client access patterns: Characteristics and caching implications. *World Wide Web*, 2(1-2):15–28, 1999.

[BC98] Paul Barford and Mark Crovella. Generating representative web workloads for network and server performance evaluation. In *Proceedings of ACM SIGMETRICS*, pages 151–160, 1998.

[BC99] Paul Barford and Mark Crovella. A performance evaluation of hyper text transfer protocols. In *Proceedings of ACM SIGMETRICS*, pages 188–197. ACM Press, 1999.

[BD99] Gaurav Banga and Peter Druschel. Measuring the capacity of a web server under realistic loads. *World Wide Web*, 2(1-2):69–83, 1999.

[BEF$^+$00] Lee Breslau, Deborah Estrin, Kevin Fall, Sally Floyd, John Heidemann, Ahmed Helmy, Polly Huang, Steven McCanne, Kannan Varadhan, Ya Xu, and Haobo Yu. Advances in Network Simulation. *IEEE Computer*, 33(5):59–67, May 2000.

[BHCKS04] A. Budhiraja, F. Hernández-Campos, V. G. Kulkarni, and F. D. Smith. Stochastic differential equation for tcp window size: Analysis and experimental validation. *Probab. Eng. Inf. Sci.*, 18(1):111–140, 2004.

[BHK$^+$91] Mary G. Baker, John H. Hartman, Michael D. Kupfer, Ken W. Shirriff, and John K. Ousterhout. Measurements of a distributed file system. In *Proceedings of the ACM Symposium on Operating Systems Principles (SOSP)*, pages 198–212, New York, NY, USA, 1991. ACM Press.

[BLFF96] T. Berners-Lee, R. Fielding, and H. Frystyk. RFC 1945: Hypertext Transfer Protocol — HTTP/1.0, May 1996. Status: INFORMATIONAL.

[Bra65] P.T. Brady. A technique for investigating on-off patterns of speech. *The Bell System Technical Journal*, pages 1–22, January 1965.

[Bra89] R. T. Braden. RFC 1122: Requirements for Internet hosts — communication layers, October 1989.

[CB96] Mark E. Crovella and Azer Bestavros. Self-similarity in world wide web traffic: evidence and possible causes. In *Proceedings of ACM SIGMETRICS*, pages 160–169, New York, NY, USA, 1996. ACM Press.

[CBC95] Carlos Cunha, Azer Bestavros, and Mark Crovella. Characteristics of WWW client-based traces. Technical report, Boston University, 1995.

[CCG$^+$04] J. Cao, W. S. Cleveland, Y. Gao, K. Jeffay, F. D. Smith, and M. Weigle. Stochastic Models for Generating Synthetic HTTP Source Traffic. In *Proceedings of IEEE Infocom*, 2004.

[CDJM91] Ramón Cáceres, Peter B. Danzig, Sugih Jamin, and Danny J. Mitzel. Characteristics of wide-area TCP/IP conversations. In *Proceedings of ACM SIGCOMM*, pages 101–112. ACM Press, 1991.

[CHC$^+$04a] Yu-Chung Cheng, Urs Hoelzle, Neal Cardwell, Stefan Savage, and Geoffrey M. Voelker. Monkey see, monkey do: A tool for tcp tracing and replaying. In *USENIX Annual Technical Conference*, June 2004.

[CHC$^+$04b] Yu-Chung Cheng, Urs Hölzle, Neal Cardwell, Stefan Savage, and Geoffrey M. Voelker. Monkey see, monkey do: A tool for TCP tracing and replaying. In *USENIX Annual Technical Conference*, pages 87–98, 2004.

[CJOS00] Mikkel Christiansen, Kevin Jeffay, David Ott, and F. Donelson Smith. Tuning RED for web traffic. In *Proceedings of ACM SIGCOMM*, pages 139–150, 2000.

[CL97] Mark Crovella and Lester Lipsky. Long-lasting transient conditions in simulations with heavy-tailed workloads. In *Proceedings of the Winter Simulation Conference*, pages 1005–1012, 1997.

[CM99] P. Chaudhuri and J.S. Marron. SiZer for exploration of structures in curves. *Journal of the American Statistical Association*, 94:807–823, 1999.

[Coh03] B. Cohen. Incentives build robustness in BitTorrent. In *Conference Workshop on Economics of Peer-to-Peer Systems*, May 2003.

[Con04] Internet 2 Consortium. Internet2 netflow weekly reports. `http://netflow.internet2.edu/weekly`, 2004.

[Cor06] Cisco Corporation. Netflow. `http://www.cisco.com/netflow`, 2006.

[DJ91] Peter B. Danzig and Sugih Jamin. tcplib: A library of TCP/IP traffic characteristics. *USC Networking and Distributed Systems Laboratory TR CS-SYS-91-01*, October, 1991.

[Dow01a] Allen B. Downey. Evidence for long-tailed distributions in the internet. In *Proceedings of ACM SIGCOMM Internet Measurement Workshop*, pages 229–241, New York, NY, USA, 2001. ACM Press.

[Dow01b] Allen B. Downey. The structural cause of file size distributions. In *Proceedings of the ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, page 361, Washington, DC, USA, 2001. IEEE Computer Society.

[ENW96] Ashok Erramilli, Onuttom Narayan, and Walter Willinger. Experimental queueing analysis with long-range dependent packet traffic. *IEEE/ACM Transactions on Networking*, 4(2):209–223, 1996.

[ET93] B. Efron and R. Tibshirani. *An Introduction to the Bootstrap.* Chapman & Hall, 1993.

[EV03] Cristian Estan and George Varghese. New directions in traffic measurement and accounting: Focusing on the elephants, ignoring the mice. *ACM Transactions on Computer Systems*, 21(3):270–313, 2003.

[Fel88] D. Feldmeier. Improving gateway performance with a routing-table cache. In *Proceedings of IEEE Infocom*, pages 27–31, March 1988.

[Fel00] Anja Feldmann. Characteristics of TCP connection arrivals. In Kihong Park and Walter Willinger, editors, *Self-Similar Network Traffic and Performance Evaluation*. Wiley-Interscience, 2000.

[FGHW99] Anja Feldmann, Anna C. Gilbert, Polly Huang, and Walter Willinger. Dynamics of IP traffic: A study of the role of variability and the impact of control. In *sigcomm*, pages 301–313, 1999.

[FGM$^+$97] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, and T. Berners-Lee. RFC 2068: Hypertext Transfer Protocol — HTTP/1.1, January 1997. Status: PROPOSED STANDARD.

[FJ93a] Sally Floyd and Van Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, 1993.

[FJ93b] Sally Floyd and Van Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, August 1993.

[FK03] Sally Floyd and Eddie Kohler. Internet research needs better models. *ACM Computer Communication Review*, 33(1):29–34, 2003.

[FP01] Sally Ford and Vern Paxson. Difficulties in simulating the internet. *IEEE/ACM Transactions on Networking*, 9(4):392–403, 2001.

[FTD03] Chuck Fraleigh, Fouad Tobagi, and Christophe Diot. Provisioning ip backbone networks to support latency sensitive traffic. In *Proceedings of IEEE Infocom*. IEEE, 2003.

[GcC02] Kartik Gopalan and Tzi cker Chiueh. Improving route lookup performance using network processor cache. In *Proceedings of the ACM/IEEE Conference on Supercomputing*, pages 1–10, Los Alamitos, CA, USA, 2002. IEEE Computer Society Press.

[GDS$^+$03] P. Krishna Gummadi, Richard J. Dunn, Stefan Saroiu, Steven D. Gribble, Henry M. Levy, and John Zahorjan. Measurement, modeling, and analysis of a peer-to-peer file-sharing workload. In *Proceedings of the ACM Symposium on Operating Systems Principles (SOSP)*, pages 314–329, 2003.

[GM01] Liang Guo and Ibrahim Matta. The war between mice and elephants. In *Proceedings of the IEEE International Conference on Network Protocols (ICNP)*, page 180, Washington, DC, USA, 2001. IEEE Computer Society.

[GMP97] Ian Graham, Jed Martens, and Murray Pearson. The dag: an atm measurement board. In *4th Electronics New Zealand Conference*, 1997.

[GW94] Mark W. Garrett and Walter Willinger. Analysis, modeling and generation of self-similar vbr video traffic. In *Proceedings of ACM SIGCOMM*, pages 269–280, New York, NY, USA, 1994. ACM Press.

[HCJS$^+$01] F. Hernández-Campos, K. Jeffay, F.D. Smith, J. S. Marron, and A. Nobel. Methodology for developing empirical models of TCP-based applications, June 2001. unpublished manuscript.

[HCJS03] F. Hernández-Campos, K. Jeffay, and F. D. Smith. Tracking the evolution of web traffic: 1995-2003. In *Proceedings of the ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, October 2003.

[HCMSS04] F. Hernández-Campos, J. S. Marron, Gennady Samorodnitsky, and F. D. Smith. Variable heavy tails in internet traffic. *Perform. Eval.*, 58(2+3):261–261, 2004.

[HCNSJ05] Félix Herández-Campos, Andrew B. Nobel, F. Donelson Smith, and Kevin Jeffay. Understanding patterns of tcp connection usage with statistical clustering. In *Proceedings of the ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 35–44, 2005.

[HCP05] Félix Hernández-Campos and Maria Papadopouli. Assessing the real impact of 802.11 WLANS: A large-scale comparison of wired and wireless traffic. In *14th IEEE Workshop on Local and Metropolitan Area Networks*, 2005.

[HCSJ04] F. Hernández-Campos, F.D. Smith, and K. Jeffay. Generating realistic TCP workloads. In *Proceedings of Computer Measurement Group (CMG) Conference*, December 2004.

[HNHC02] Peter Hall, Andrew B. Nobel, and Félix Hernández-Campos. Block bootstrap approach to simulating Internet traffic. Unpublished Manuscript, 2002.

[HV03] Nicolas Hohn and Darryl Veitch. Inverting sampled traffic. In *Proceedings of ACM SIGCOMM Internet Measurement Conference*, pages 222–233, New York, NY, USA, 2003. ACM Press.

[HVA02] N. Hohn, D. Veitch, and P. Abry. Does fractal scaling at the IP level depend on TCP flow arrival processes. In *Proceedings of ACM SIGCOMM Internet Measurement Workshop*, 2002.

[Inc] NetIQ Software Inc. Chariot performance evaluation platform. `http://www.netiq.com/products/chr/default.asp`.

[Int] Internet Traffic Archive. `http://ita.ee.lbl.gov`.

[Jai90] R. Jain. Characteristics of destination address locality in computer networks: a comparison of caching schemes. *Computer Networks and ISDN Systems*, 18(4):243–254, 1990.

[Jai91] R. Jain. *The Art of Computer Systems Performance Analysis*. Wiley, 1991.

[JBB92] V. Jacobson, R. Braden, and D. Borman. RFC 1323: TCP extensions for high performance, May 1992. Status: PROPOSED STANDARD.

[JD02] Hao Jiang and Constantinos Dovrolis. Passive estimation of TCP round-trip times. *ACM Computer Communication Review*, 32(3):75–88, 2002.

[JD04] Hao Jiang and Constantinos Dovrolis. The effect of flow capacities on the burstiness of aggregated traffic. In *Passive and Active Measurement*, pages 93–102, 2004.

[JRF$^+$99] Y. Joo, V. Ribeiro, A. Feldmann, A. Gilbert, and W. Willinger. On the impact of variability on the buffer dynamics in IP networks. In *Allerton Conference on Communication, Control and Computing*, September 1999.

[JRF$^+$01] Youngmi Joo, Vinay Ribeiro, Anja Feldmann, Anna C. Gilbert, and Walter Willinger. Tcp/ip traffic dynamics and network performance: a lesson in workload modeling, flow control, and trace-driven simulations. *ACM Computer Communication Review*, 31(2):25–37, 2001.

[KBBkc03] Thomas Karagiannis, Andre Broido, Nevil Brownlee, and Michalis Faloutsos kc claffy. File-sharing in the Internet: A characterization of p2p traffic in the backbone. Technical report, UC Riverside, November 2003.

[KcLH$^+$02] Purushotham Kamath, Kun chan Lan, John Heidemann, Joe Bannister, and Joe Touch. Generation of high bandwidth network traffic traces. In *Proceedings of the ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 401–410, Fort Worth, Texas, USA, October 2002. USC/Information Sciences Institute, IEEE.

[KHR02] Dina Katabi, Mark Handley, and Charles Rohrs. Internet congestion control for future high bandwidth-delay product environments. In *Proceedings of ACM SIGCOMM*, 2002.

[KL86] B. Kantor and P. Lapsley. RFC 977: Network news transfer protocol: A proposed standard for the stream-based transmission of news, February 1986. Status: PROPOSED STANDARD.

[KLPS02] Eddie Kohler, Jinyang Li, Vern Paxson, and Scott Shenker. Observed structure of addresses in IP traffic. In *Proceedings of ACM SIGCOMM Internet Measurement Workshop*, pages 253–266. ACM Press, 2002.

[KP88a] P. Karn and C. Partridge. Improving round-trip time estimates in reliable transport protocols. In *Proceedings of ACM SIGCOMM*, pages 2–7, New York, NY, USA, 1988. ACM Press.

[KP88b] P. Karn and C. Partridge. Improving round-trip time estimates in reliable transport protocols. In *Proceedings of ACM SIGCOMM*, pages 2–7, New York, NY, USA, 1988. ACM Press.

[KZ97] Edward W. Knightly and Hui Zhang. D-bind: an accurate traffic model for providing qos guarantees to vbr traffic. *IEEE/ACM Transactions on Networking*, 5(2):219–231, 1997.

[LAJS03] Long Le, Jay Aikat, Kevin Jeffay, and F. Donelson Smith. The effects of active queue management on web performance. In *Proceedings of ACM SIGCOMM*, pages 265–276, New York, NY, USA, 2003. ACM Press.

[LH02] Kun-Chan Lan and John Heidemann. Rapid model parameterization from traffic measurements. *ACM Trans. Model. Comput. Simul.*, 12(3):201–229, 2002.

[LTWW93] Will E. Leland, Murad S. Taqq, Walter Willinger, and Daniel V. Wilson. On the self-similar nature of Ethernet traffic. In Deepinder P. Sidhu, editor, *Proceedings of ACM SIGCOMM*, pages 183–193, San Francisco, California, 1993.

[Mah97] Bruce A. Mah. An empirical model of HTTP network traffic. In *Proceedings of IEEE Infocom*, volume 2, pages 592–600, 1997.

[MDG01] Joerg Micheel, Stephen Donnelly, and Ian Graham. Precision timestamping of network packets. In *Proceedings of ACM SIGCOMM Internet Measurement Workshop*, pages 273–277, New York, NY, USA, 2001. ACM Press.

[MGT00] Vishal Misra, Wei-Bo Gong, and Don Towsley. Fluid-based analysis of a network of aqm routers supporting tcp flows with an application to red. In *Proceedings of ACM SIGCOMM*, pages 151–160, New York, NY, USA, 2000. ACM Press.

[MH00] A. Mena and J. Heidemann. An empirical study of Real Audio traffic. In *Proceedings of IEEE Infocom*, March 2000.

[MHCS02] J. S. Marron, F. Hernández-Campos, and F. D. Smith. Mice and elephants visualization of Internet traffic. In *Proceedings of 15th Conference on Computational Statistics*, August 2002.

[Mit04] M. Mitzenmacher. A brief history of generative models for power law and lognormal distributions. *Internet Mathematics*, 1(2):226–251, 2004.

[MJ93] Steven McCanne and Van Jacobson. The bsd packet filter: A new architecture for user-level packet capture. In *Proceedings of the Winter USENIX Technical Conference*, pages 259–269, January 1993.

[MJ98] David Mosberger and Tai Jin. httperf: a tool for measuring web server performance. *SIGMETRICS Perform. Eval. Rev.*, 26(3):31–37, 1998.

[MSM97] M. Mathis, J. Semke, and J. Mahdavi. The macroscopic behavior of the TCP congestion avoidance algorithm. *ACM Computer Communication Review*, 27(3), 1997.

[Nag84] J. Nagle. RFC 896: Congestion control in IP/TCP internetworks, January 1984. Status: UNKNOWN.

[NIS06] NIST/SEMATECH. e-handbook of statistical methods. `http://www.itl.nist.gov/div898/handbook`, 2006.

[nlaa] National Laboratory for Applied Network Research (NLANR). `http://www.nlanr.net`.

[nlab] NLANR's Passive Measurement and Analysis (PMA) project. `http://pma.nlanr.net`.

[NSSW02] Carl Nuzman, Iraj Saniee, Wim Sweldens, and Alan Weiss. A compound model for TCP connection arrivals for LAN and WAN applications. *Computer Networks*, 40(3):319–337, 2002.

[Ost] David Ostermann. Tcptrace: a TCP connection analysis tool. `http://www.tcptrace.org`.

[Pax94] Vern Paxson. Empirically derived analytic models of wide-area TCP connections. *IEEE/ACM Transactions on Networking*, 2(4):316–336, 1994.

[Pax97] V. Paxson. Fast, approximate synthesis of fractional gaussian noise for generating self-similar network traffic. *ACM Computer Communication Review*, 27(5):5–18, October 1997.

[PF95] Vern Paxson and Sally Floyd. Wide area traffic: the failure of Poisson modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244, 1995.

[PFTK98] Jitendra Padhye, Victor Firoiu, Don Towsley, and Jim Kurose. Modeling TCP throughput: a simple model and its empirical validation. In *Proceedings of ACM SIGCOMM*, pages 303–314, New York, NY, USA, 1998. ACM Press.

[PHCL$^+$] C. Park, F. Hernández-Campos, L. Le, J. S. Marron, J. Park, V. Pipiras, F. D. Smith, R. L. Smith, M. Trovero, and Z. Zhu. Long-range dependence analysis of Internet traffic.

[PHCM$^+$06] Cheolwoo Park, Félix Hernández-Campos, J. S. Marron, Kevin Jeffay, and F. D. Smith. Correlations of size, rate, and duration in tcp connections: the case against. In submission, 2006.

[PHCMS05] Cheolwoo Park, Félix Hernández-Campos, J. S. Marron, and F. Donelson Smith. Long-range dependence in a changing internet traffic mix. *Computer Networks*, 48(3):401–422, 2005.

[Pos81] J. Postel. RFC 793: Transmission control protocol, September 1981.

[Pro] The DAG Project. `http://dag.cs.waikato.ac.nz`.

[PW00] Kihong Park and Walter Willinger, editors. *Self-Similar Network Traffic and Performance Evaluation*. Wiley-Interscience, 2000.

[RDFS04] Andy Rupp, Holger Dreger, Anja Feldmann, and Robin Sommer. Packet trace manipulation framework for test labs. In *Proceedings of ACM SIGCOMM Internet Measurement Conference*, pages 251–256. ACM Press, 2004.

[Riz97] Luigi Rizzo. Dummynet: a simple approach to the evaluation of network protocols. *ACM Computer Communication Review*, January 1997.

[SB04] Joel Sommers and Paul Barford. Self-configuring network traffic generation. In *Proceedings of ACM SIGCOMM Internet Measurement Conference*, pages 68–81, New York, NY, USA, 2004. ACM Press.

[SBDR05] Joel Sommers, Paul Barford, Nick Duffield, and Amos Ron. Improving accuracy in end-to-end packet loss measurement. In *Proceedings of ACM SIGCOMM*, August 2005.

[SGG02] S. Saroiu, P. Gummadi, and S. Gribble. A measurement study of peer-to-peer file sharing systems. In *Proceedings of Multimedia Computing and Networking*, 2002.

[SHCJO01] F. Donelson Smith, Félix Hernández-Campos, Kevin Jeffay, and David Ott. What TCP/IP protocol headers can tell us about the web. In *Proceedings of ACM SIGMETRICS*, pages 245–256, 2001.

[SRB01] S. Sarvotham, R. Riedi, and R. Baraniuk. Connection-level analysis and modeling of network traffic. In *Proceedings of ACM SIGCOMM Internet Measurement Workshop*, November 2001.

[Ste94] R. W. Stevens. *TCP/IP Illustrated, Volume 1: The Protocols.* Addison-Wesley, 1994.

[SYB04] Joel Sommers, Vinod Yegneswaran, and Paul Barford. A framework for malicious workload generation. In *Proceedings of ACM SIGCOMM Internet Measurement Conference*, pages 82–87. ACM Press, 2004.

[tcpa] Tcpdump public repository. `http://www.tcpdump.org`.

[tcpb] Tcpreplay: Pcap editing and replay tools for *nix. `http://tcpreplay.sourceforge.net`.

[Wal99] J. Walker. *A primer on wavelets and their scientific applications.* CRC Press, 1999.

[WP98] Walter Willinger and Vern Paxson. Where mathematics meets the Internet. *Notices of the American Mathematical Society*, pages 961–970, September 1998.

[WTSW97] Walter Willinger, Murad S. Taqqu, Robert Sherman, and Daniel V. Wilson. Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level. *IEEE/ACM Transactions on Networking*, 5(1):71–86, 1997.

[YVIB05] Tao Ye, Darryl Veitch, Gianluca Iannaccone, and Supratik Bhattacharyya. Divide and conquer: PC-based packet trace replay at OC-48 speeds. In *Tridentcom*, Trento, Italy, February 2005.

[ZRMD03] Z.-L. Zhang, V. Ribeiro, S. Moon, and C. Diot. Small-time scaling behaviors of Internet backbone traffic: An empirical study. In *Proceedings of IEEE Infocom*, San Francisco, March 2003.