Why We Don't Know How To Simulate The Internet

Sally Floyd and Vern Paxson[*]
AT&T Center for Internet Research at ICSI (ACIRI)
Berkeley, CA 94704 USA
http://www.aciri.org/

October 11, 1999

Abstract

This paper discusses both the importance and the dangers of simulations in network research. Simulating how the global Internet data network behaves is an immensely challenging undertaking because of the network's great heterogeneity and rapid change. The heterogeneity ranges from the individual links that carry the network's traffic, to the protocols that interoperate over the links, to the "mix" of different applications used at a site, and the levels of congestion (load) seen on different links. We discuss two key strategies for developing meaningful simulations in the face of these difficulties: searching for invariants and judiciously exploring the simulation parameter space. We finish with a brief look at a collaborative effort within the research community to develop a common network simulator.

1 Introduction

The design of the Internet continues to evolve, and many aspects of its behavior are poorly understood. Due to the network's complexity, simulation plays a vital role in attempting to characterize how different facets of the network behave and how proposed changes might affect the network's different properties. Yet simulating different aspects of the Internet is exceedingly difficult. In this paper we discuss the fundamental role of simulation in Internet research (§2), and explore the underlying difficulties (§3–§6), which are rooted in the network's immense heterogeneity and the great degree to which it changes over time. We then discuss some strategies for accommodating these difficulties (§7). We finish with a brief look at a collaborative effort within the research community to develop a common network simulator (§8).

2 The Role Of Simulation

While measurement and experimentation are crucial for exploring the "real world", simulations and analysis are restricted to exploring a constructed, abstracted model of the world. In some fields the interplay between measurement, experimentation, simulation, and analysis may be obvious, but Internet research introduces some unusual additions to these roles, in part because of the large scale and rapid evolution of the subject area (i.e., the global Internet).

Measurement is needed for a crucial "reality check." It often serves to challenge our implicit assumptions. Indeed, while we personally have conducted a number of measurement studies, we have never conducted one that failed to surprise us in some fundamental fashion.

Experiments are crucial for dealing with implementation issues—which can at first sound almost trivial, but often wind up introducing unforeseen complexities—and for understanding the behavior of otherwise intractable systems. Experimentation also plays a key role in exploring new environments before finalizing how the Internet protocols should operate in those environments.

However, measurement and experimentation have limitations in that they can only be used to explore the existing Internet; while they can be used to explore particular new *environments*, they cannot be used to explore different possible *architectures* for the future Internet. (There is no instantiation of a "future Internet", on the relevant scale and with the relevant range of "future" applications, for our measurement and experimentation.)

One problem Internet research suffers, absent from most other fields, is the possibility of a "success disaster"—designing some new Internet functionality that, before the design is fully developed and debugged, escapes into the real world and multiplies there due to the basic utility of the new functionality. Because of the extreme speed with which software can propagate to endpoints over the network, it is not at all implausible that the new functionality might spread to a million computers within a few weeks. Indeed, the HTTP protocol used by the World Wide Web is a perfect example of a success disaster. Had its designers envisioned it in use by

[*] This work was supported by the Director, Office of Energy Research, Office of Computational and Technology Research, Mathematical, Information, and Computational Sciences Division of the United States Department of Energy under Contract No. DE-AC03-76SF00098, and by ACIRI. An earlier version of this paper appeared in the *Proceedings of the 1997 Winter Simulation Conference*, Atlanta, GA, 1997.

the entire Internet—and had they explored the corresponding consequences with analysis or simulation—they would have significantly altered its design, which in turn would have led to a more smoothly operating Internet today.

Analysis provides the possibility of exploring a model of the Internet over which one has complete control. The role of analysis is fundamental because it brings with it greater understanding of the forces at play. It carries with it, however, the risk of using a model simplified to the point where key facets of Internet behavior have been lost, in which case any ensuing results could be useless (though they may not appear to be so!). Even in light of this risk, as scientists we need to recognize the fundamental role analysis plays in providing the bedrock on which to build our understanding of the Internet. Furthermore, while the network is immensely complex and dauntingly difficult to encompass, we can still make progress (often incremental) towards building this understanding.

Simulations are complementary to analysis, not only by providing a check on the assumptions of the model and on the correctness of the analysis, but by allowing exploration of complicated scenarios that would be either difficult or impossible to analyze. (Simulations can also play a vital role in helping researchers to develop intuition.) In particular, the complexities of Internet topologies and traffic, and the central role of adaptive congestion control, make simulation the most promising tool for addressing many of the questions of Internet traffic dynamics.

In this paper we develop the argument that, due to the heterogeneity and rapid change in the Internet, there does not exist a single suite of simulation scenarios sufficient to demonstrate that a proposed protocol or system will perform well in the future evolving Internet. Instead, simulations play the more limited role of examining particular aspects of proposed changes or of Internet behavior, and of adding to our understanding of the underlying dynamics.

For some topics, such as the division of bandwidth among competing TCP connections with different roundtrip times, the simplest scenario that illustrates the underlying principles is often the best. In this case the researcher can make a conscious decision to abstract away all but the essential components of the scenario under study. As the research community begins to address questions of *scale*, however, these small, simple simulation scenarios will be less useful. It becomes more critical for researchers to address questions of topology, traffic generation, and multiple layers of protocols, and to pay more attention to the choices made in picking the underlying models to be explored.

Along with the strengths, simulation as a tool of network research has its share of dangers and pitfalls. In addition to the problems described in the rest of this paper of defining the relevant model, there can be considerable difficulty in verifying that your simulator in fact accurately implements the intended model. It is generally easier to verify the correctness of a mathematical analysis than it is to verify the correctness of the software implementation of an extensive and complex underlying model.

For these reasons, Internet simulations are most useful as a tool for building understanding of dynamics, or to illustrate a point, or to explore for unexpected behavior. Internet simulations are more treacherous, in our opinion, when used simply to produce numbers that are taken at face value (e.g., that protocol A performed 23% better than protocol B). Not only are there all of the questions of whether a small change in the model could have resulted in a large change in the results; there is, in addition, the question of whether the results would have been affected by the change in a detail of the simulator's software *implementation* of the underlying model.

That said, we note that different communities are likely to have different requirements of network simulators. For more immediate development work, where there is a reasonably well-defined question of whether Alternative A or Alternative B performs best in Environment X, it could be feasible to carefully define the underlying model, verify the simulator, and indeed to use simulation results to show that Alternative A performs 23% better than Alternative B.

For longer-term research, where the question is whether Alternative A or Alternative B is likely to be a better choice for the Internet architecture five years in the future, a different approach is required, and possibly a different simulator. That is, the most useful simulator for this purpose would be one that not only incorporated one's own proposed protocols for the future Internet architecture, but also the proposed protocols from other researchers as well. This would allow some investigation of the potential interactions between these various proposed protocols (which might be implemented at different places in the network or at different layers of the protocol stack).

Whether simulations are used to obtain quantitative results, or to explore more general relationships between network parameters and network dynamics, simulations are most useful (and taken seriously by other researchers) if other researchers can confirm for themselves that slight changes in the network scenario do not radically change the results, and that the simulation results are not actually due to errors in the implementation of the simulator. One of the best ways to address these issues of simulation validation is for researchers to make their simulator and scripts publicly available, so that other researchers can easily check for themselves the effect of changing underlying assumptions of the network scenario. This also allows other researchers to use the simulator, and to explore for themselves the correctness of the implementations in the simulator.[1]

3 An Immense Moving Target

The Internet has several key properties that make it exceedingly hard to characterize, and thus to simulate. First, its

[1] One of the recommendations from the 1999 DARPA/NIST Network Simulation Validation Workshop [NSVW99] is that researchers make their simulation scripts publicly available, for exactly this reason.

great success has come in large part because the main function of the Internet Protocol (IP) architecture is to unify diverse networking technologies and administrative domains. IP allows vastly different networks administered by vastly different policies to seamlessly interoperate. However, the fact that IP masks these differences from a *user*'s perspective does not make them go away! IP buys uniform *connectivity* in the face of diversity, not uniform *behavior*. Indeed, the greater IP's success at unifying diverse networks, the harder the problem of understanding how a large IP network behaves.

A second key property is that the Internet is big. It included more than 55 million computers as of June 1999 [Tel99]. Its size brings with it two difficulties. The first is that the range of heterogeneity mentioned above is very large: if only a small fraction of the computers behave in an atypical fashion, the Internet still might include thousands of such computers, often too many to dismiss as negligible.

Size also brings with it the crucial problem of *scaling*: many networking protocols and mechanisms work fine for small networks of tens or hundreds of computers, or even perhaps "large" networks of tens of thousands of computers, yet become impractical when the network is again three orders of magnitude larger (today's Internet), much less five orders of magnitude (the next decade's Internet). Large scale means that rare events *will* routinely occur in some part of the network, and, furthermore, that reliance on human intervention to maintain critical network properties such as stability becomes a recipe for disaster.

A third key property is that the Internet changes in *drastic* ways over time. For example, we mentioned above that in June 1999, the network included more than 55 million computers. But in Jan. 1997, two years earlier, it comprised only 16 million computers [Lot97], reflecting growth of 65%/year. This growth then begs the question: how big will it be in two more years? 5 years? One might be tempted to dismiss the explosive growth between 1997 and 1999 as surely a one-time phenomenon, reflecting the sudden public awareness of the Web. But Figure 1 belies this conclusion. It plots time on the X-axis and the volume of traffic through USENET (the Internet's main bulletin board system) in bytes/day on the Y-axis, which is logarithmically scaled.

The excellent (for real data) straight-line fit to the USENET traffic's growth over time corresponds to exponential growth of 80%/year. But the data plotted go back to 1984! Clearly, the Internet has sustained *major exponential growth* for well over a decade, with *no sign of slowing down*. Accordingly, we *cannot* assume that the network's current, fairly immense size indicates that its growth must surely begin to slow.

Figure 2 shows a considerably different growth statistic. Here we have plotted the number of connections made by the Lawrence Berkeley National Laboratory (LBNL) each day from Jan. 1, 1997, to March 5, 1999, with the Y-axis again log-scaled. Thus, we are no longer viewing an aggregate Internet growth statistic but one specific to a particular site; but we again see sustained exponential growth, this time at a rate of about 54%/year. (See [Pax94a] for more discussion of this particular site's growth characteristics.)

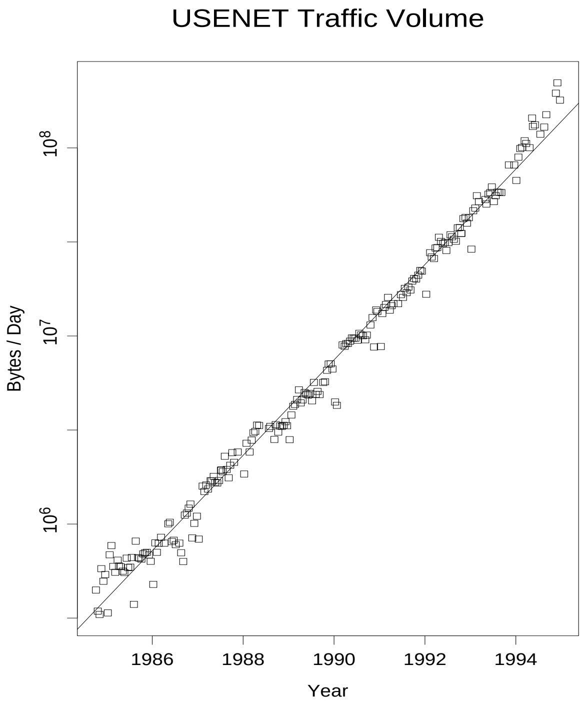

Figure 1: Bytes per day sent through the USENET bulletin board system, averaged over two-week intervals (taken from [Pax94a]). The growth rate corresponds to exponential increase of 80%/year. Data courtesy of Rick Adams.

Unfortunately, growth over time is not the only way in which the Internet is a moving target. Even what we would assume must certainly be solid, unchanging statistical properties can change in a brief amount of time. For example, in Oct. 1992 the median size of an Internet FTP (file transfer) connection observed at LBNL was 4,500 bytes [Pax94b]. The median is considered a highly *robust* statistic, one immune to outliers (unlike the mean, for example), and in this case was computed over 60,000 samples. Surely this statistic should give us some solid predictive power in forecasting future FTP connection characteristics! Yet only five months later, the same statistic computed over 80,000 samples yielded 2,100 bytes, less than half what was observed before.

Again, it might be tempting to view this variation as a one-time fluke. But repeating the same analysis seven years later, we find that in March, 1998, the median connection size was 10,900 bytes, while nine months later, it fell again by a factor of two, this time to 5,600 bytes.

Thus, we must exercise great caution in assuming that observations made at a particular point in time tell us much about properties at other points in time.

For Internet engineering, however, the growth in size

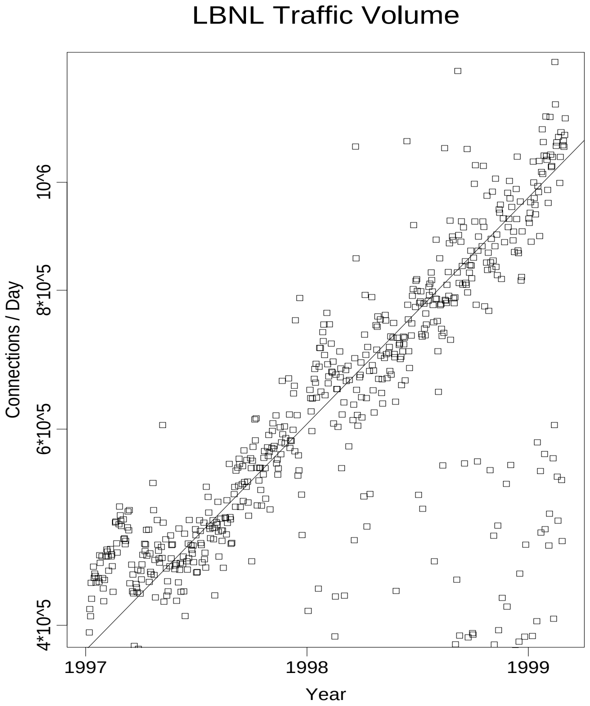

Figure 2: Internet connections per day at the Lawrence Berkeley National Laboratory. The growth rate corresponds to exponential increase of 54%/year.

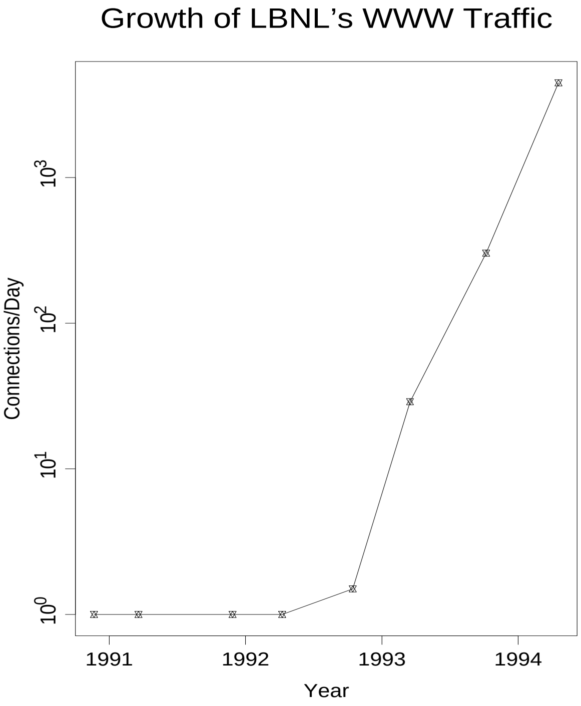

Figure 3: World Wide Web (HTTP) connections per month at the Lawrence Berkeley National Laboratory (taken from [Pax94a]). The growth rate corresponds to doubling every 7–8 weeks.

and change in connection characteristics in some sense pale when compared to another way in which the Internet is a moving target: it is subject to major changes in *how* it is used, with new applications sometimes virtually exploding on the scene and rapidly altering the lay of the land.

Figure 3 plots the number of HTTP connections made per day for 8 datasets recorded at LBNL, with a log-scaled Y-axis. We see that the Web was essentially unknown until late 1992 (and other traffic dominated). Then, a stunning pattern of growth set in: the site's Web traffic began to *double every 7–8 weeks*, and continued to do so for *two full years*. Clearly, any predictions of the shape of future traffic made before 1993 were hopelessly off the mark by 1994, when Web traffic wholly dominated the site's activities.

Furthermore, such explosive growth was not a one-time event associated with the paradigm-shift in Internet use introduced by the Web. For example, in Jan. 1992 the MBone—a “multicast backbone” for transmitting audio and video over the Internet [Eri94]—did not exist. Three years later, it made up 20% of all of the Internet data bytes at Digital's Western Research Lab; 40% at LBNL; and more than 50% at a Bellcore. It too, like the Web, had exploded. In this case, however, the explosion abated, and today MBone traffic is overshadowed by Web traffic (it made up 12% of LBNL's wide-area traffic in December, 1997). How this will look tomorrow, however, is anyone's guess.

In summary: the Internet's technical and administrative diversity, sustained growth over time, and immense variations over time regarding which applications are used and in what fashion, all present immense difficulties for attempts to simulate it with a goal of obtaining “general” results.

4 Heterogeneity Any Which Way You Look

Even if we fix our interest to a single point of time, the Internet remains immensely heterogeneous. In the previous section we discussed this problem in high-level terms; here, we discuss two specific areas in which ignoring heterogeneity can completely undermine the strength of simulation results.

4.1 Topology and link properties

A basic question for a network simulation is what topology to use for the network being simulated—the specifics of how the computers in the network are connected (directly or indirectly) with each other, and the properties of the links that foster the interconnection.

Unfortunately, the topology of the Internet is difficult to characterize. First, it is constantly changing. Second, the topology is engineered by a number of competing entities, not all of whom are willing to provide topological information. Because there is no such thing as a “typical” Internet topology, simulations exploring protocols that are sensitive to topological structure can at best hope to characterize how the protocol performs over a range of topologies.

On the plus side, the research community has made significant advances in developing topology-generators for Internet simulations [CDZ97]. Several of the topology generators can create networks with locality and hierarchy loosely based on the structure of the current Internet.

The next problem is that while the properties of the different types of links used in the network are generally known, they span a very large range. Some are slow modems, capable of moving only hundreds of bytes per second, while others are state-of-the-art fiber optic links with bandwidths a million times faster. Some are point-to-point links directly connecting two routers (this form of link is widely assumed in simulation studies); others are broadcast links that directly connect a large number of computers (fairly common in practice). These two types have quite different properties: broadcast links have contention in which multiple transmitting computers must resolve which of them gets to use the link when (so traffic on broadcast links becomes *correlated* in a fashion that is different from when using point-to-point links). However, broadcast links can also make some protocols much more efficient, by directly facilitating one-to-many communication. An additional consideration is that some links are multiaccess without being true broadcast. For example, a wireless radio link might include locations where some of the users of the link can hear *some* but not *all* of the other users of the link.

Another type of link is that provided by connections to satellites. If a satellite is in geosynchronous orbit, then the latency up to and back down from the satellite will be on the order of hundreds of milliseconds, much higher than for most land-based links. On the other hand, if the satellite is in low-earth orbit, the latency is quite a bit smaller, but *changes* with time as the satellite crosses the face of the earth.

Another facet of topology easy to overlook is that of dynamic routing. In the Internet, routes through the network can change on time scales ranging from seconds to days [Pax96], and hence the topology is not fixed. If route changes occur on fine enough time scales (per-packet changes are not unknown), then one must refine the notion of "topology" to include multi-pathing. Multi-pathing immediately brings other complications: the latency, bandwidth and load of the different paths through the network might differ considerably.

Finally, routes are quite often *asymmetric*, with the route from computer A to computer B through the network differing in the hops it visits from the reverse route from B to A. Routing asymmetry can lead to asymmetry in path properties such as latency and bandwidth (which can also arise from other mechanisms). An interesting facet of routing asymmetry is that it often only arises in large topologies: it provides a good example of how *scaling* can lead to unanticipated problems.

4.2 Protocol differences

Once all of these topology and link property headaches have been sorted out, the researcher conducting a simulation study must then tackle the specifics of the protocols used in the study. For some studies, simplified versions of the relevant Internet protocols may work fine. But for other studies that are sensitive to the details of the protocols (it can sometimes be hard to tell these from the former!), researchers and engineers face some hard choices. While conceptually the Internet uses a unified set of protocols, in reality each protocol has been implemented by many different communities, often with significantly different features (and of course bugs). For example, the widely used Transmission Control Protocol (TCP) has undergone major evolutionary changes (see [Ste96] for a "family tree" showing the lineages associated with one evolutionary branch). A study of eleven different TCP implementations found distinguishing differences among nearly all of them [Pax97a], and major problems with several [PADR99]. More recently, techniques for "fingerprinting" different TCP implementations (based on analyzing their distinct behavior in response to a wide range of inputs) have been used to identify (at last count) 200 different implementations and versions, based on their idiosyncracies [Fyo99].

Consequently, researchers must decide which real-world features and peculiarities to include in their study, and which can be safely ignored. For some simulation scenarios, the choice between these is clear; for others, determining what can be ignored can present considerable difficulties. After deciding which specific Internet protocols to use, they must then decide which *applications* to simulate using those protocols. Unfortunately, different applications have major differences in their characteristics; worse, these characteristics can vary considerably from site to site, as does the "mix" of which applications are predominantly used at a site [Pax94a]. Again, researchers are faced with hard decisions about how to keep their simulations tractable without oversimplifying their results to the point of uselessness.

5 Traffic Generation

For many Internet simulations, a basic problem is how to introduce different traffic sources into the simulation. The difficulty with synthesizing such traffic is that no solid, abstract description of Internet traffic exists. At best, some (but not all) of the salient characteristics of such traffic have been described in abstract terms, a point we return to in §7.1. Researchers often attempt to address this problem by simulating the cross-traffic, too. Doing so, however, then drags into the simulation the difficulties described above with selecting topologies, protocols, and applications, which might otherwise have been absent, or at least only present on a more modest, tractable scale.

Trace-driven simulation might appear at first to provide a cure-all for the heterogeneity and "real-world warts and all" problems that undermine abstract descriptions of Internet traffic. If only one could collect enough diverse traces, one could in principle capture the full diversity. This vision fails for a basic, often unappreciated reason. One crucial property of much of the traffic in the Internet is that it uses adaptive congestion control. Each source transmitting data over the network reacts to the progress of the data transfer so far. If it detects signs that the network is under stress, it cuts the rate at which it sends data, in order to do its part in diminishing the stress [Jac88]. Consequently, the timing of a connection's packets as recorded in a trace intimately reflects the conditions in the network at the time the connection occurred. Furthermore, these conditions are *not* readily determined by inspecting the trace. Connections adapt to network congestion anywhere along the end-to-end path between the sender and the receiver. So a connection observed on a high-speed, unloaded link might still send its packets at a rate much lower than what the link could sustain, because somewhere else along the path insufficient resources are available for allowing the connection to proceed faster.

We refer to this phenomenon as resulting in traces that are *shaped*. Shaping leads to a dangerous pitfall when simulating the Internet, namely the temptation to use trace-driven simulation to incorporate the diverse real-world effects seen in the network. The key point is that, due to rate adaptation from end-to-end congestion control, we cannot safely reuse a trace of a connection's packets in another context, because the connection would not have behaved the same way in the new context! This problem is insidious because there are often no overt indications that the connection's behavior in the new context (different topology, level of cross-traffic, link capacities) is incorrect; the simulation simply produces plausible but inaccurate results.

Traffic shaping does *not* mean that, from a simulation perspective, measuring traffic is fruitless. Instead, we must shift our thinking away from trace-driven *packet-level* simulation and instead to trace-driven *source-level* simulation. That is, for most applications, the volumes of data sent by the endpoints, and often the application-level pattern in which data is sent (request/reply patterns, for example), are not shaped by the network's current properties; only the lower-level specifics of exactly *which* packets are sent *when* are shaped. Thus, if we take care to use traffic traces to characterize *source behavior*, rather than packet-level behavior, we can then use the source-level descriptions in simulations to synthesize plausible traffic. See [DJCME92, Pax94b, CBC95] for examples of source-level descriptions, and [ITA, MOAT] for some on-line repositories of traffic traces.

Finally, we note that not all sources can be reliably characterized by traffic traces. For example, remote login users faced with heavy congestion may terminate their sessions earlier than they otherwise would, or only issue commands that generate modest output. A more general class of exceptions come from applications that are inherently *adaptive*, such as some forms of Internet video (see [MJV96], for example). These applications not only have their packet-level characteristics shaped by current traffic, but also their application-level behavior. For example, instead of the current congestion level simply determining the rate of transmission for a fixed amount of video data, it might instead determine the content or level of detail for the transmitted video. One might still be able to determine from a traffic trace a higher-level description of the original source characteristics—what properties it must have had prior to adapting—and then use this description plus the application-level adaptation algorithms in a simulation. But this will not be easy; the transformation is considerably more complex than reconstructing simple source properties such as data volumes.

An alternative approach to deriving source models from traffic traces is to characterize traffic sources in more abstract terms, such as representing large data transfers as continually sending a steady stream of packets. Depending on the simulation scenario of interest, doing so can easily bring with it the risk of oversimplifying the traffic, leading to misleading simulation results. Even if the abstractions are indeed appropriate descriptions in some contexts, the Internet's pervasive heterogeneity raises the question: *which* set of abstractions should be used? Is the traffic of interest dominated by, for example, the aggregate of thousands of small connections (Web "mice"), or a few extremely large, one-way, rate-adapting bulk transfers ("elephants"), or long-running, high-volume video streams multicasted from one sender to multiple destinations, or bidirectional multimedia traffic generated by interactive gaming?

A final dimension that must be explored is: to what level should the traffic congest the network links? Virtually all degrees of congestion, including none at all, are observed with non-negligible probability in the Internet [Pax99]. More generally, variants on the above scenarios—many small connections vs. a few large ones, unidirectional vs. bidirectional transfers, multicast vs. unicast, real-time vs. adaptive bulk transfer—all occur in different situations frequently enough that they cannot be dismissed out of hand.

However, while it is impossible to define a "typical" Internet traffic mix or a "typical" level of congestion, we can still use simulations to explore network behavior as a function of the traffic, topology, link properties, and so on. The crucial point is to keep in mind that we must consider a spectrum of scenarios, rather than one particular scenario. Unfortunately, this also significantly increases the burden of work required for sound simulation.

Similarly, using source models of individual connections to generate aggregated cross-traffic for simulations can also present scaling issues. If the intent is to simulate highly aggregated cross-traffic, then doing so by simulating each individual source can be prohibitively expensive in terms of processing time, because a highly-aggregated Internet link consists (today) of many thousands of simultaneous connections [TMW97]. Solid, high-level descriptions of aggregate traffic, and simulation models of aggregate traffic that faithfully reproduce the response of the aggregate to individual packet drops (or to other indications of congestion), would be a great help to researchers in exploring large-scale simulations. But, so far, such abstractions are beyond the start of the art.

6 Today's Network Is Not Tomorrow's

A harder issue to address in a simulation study is how the Internet might evolve in the future. For example, all of the following might or might not happen within the next few years:

- **Pricing structures:** New pricing structures are set in place, leading users to become much more sensitive to the type and quantity of traffic they send and receive.

- **Scheduling:** The Internet routers switch from the common FIFO scheduling for servicing packets to methods that attempt to more equably share resources among different connections (such as Fair Queueing, discussed by [DKS90]).

- **Wireless:** A network link technology not widely used in the Internet today catches on and becomes a much more common method for how millions of Internet users access the network. An example comes from wireless techniques such as radio or infrared. These technologies have some significantly different characteristics than those of links widely used today, such as being much more prone to packet damage during transit, and having considerably different broadcast properties.

- **Native multicast:** Native multicast becomes widely deployed, enabling an explosion in the level of multicast audio and video traffic. Presently, Internet multicast is built on top of unicast tunnels, so the links traversed by multicast traffic are considerably different from those that would be taken if multicast were directly supported in the heart of the network.

- **Differentiated service:** Mechanisms for supporting different classes and qualities of service [ZDESZ93, BBCDWW98] become widely deployed in the Internet. These mechanisms would then lead to different connections attaining potentially much different performance than they presently do, with little interaction between traffic from different classes.

- **Ubiquitous web-caching:** For many purposes, Internet traffic today is dominated by World Wide Web connections. (This is one of the relatively few epochs in the Internet's history for which a single application clearly dominates use of the network.) Although the use of the global web-caching infrastructure is growing [BBMRS97], web traffic is still dominated by wide-area connections that traverse geographically and topologically large paths through the network.

  As the web-caching infrastructure matures, and as clients and servers both become more caching-friendly as a way of reducing access times seen by the end users, this could increase both the fraction of web content that is cacheable, and the fraction of cacheable web content that is in fact accessed from caches rather than from an origin or replicated server. Such a shift to a traffic pattern that makes more use of web caches could entail a corresponding shift from traffic dominated by wide-area connections to traffic patterns with locality and less stress of the wide-area infrastructure. On the other hand, there are possible counter-trends towards real-time and customized web content that is more difficult to cache, though may prove amenable to web *replication* servers rather than cache servers.

- **A new "killer app":** A new "killer application" comes along. While Web traffic dominates today, it is vital not to then make the easy assumption that it will continue to do so tomorrow. There are many possible new applications that could take its place (and surely some unforeseen ones, as was the Web only a few years ago), and these could greatly alter how the network tends to be used. Real-time applications such as telephony and video are one possibility. Another example sometimes overlooked by serious-minded researchers is that of multi-player gaming: applications in which perhaps thousands or millions of people use the network to jointly entertain themselves by entering into elaborate (and bandwidth-hungry) virtual realities.

Some of these changes might never occur, and others that do occur might have little effect on a researcher's simulation scenarios. However, a fundamental difficulty in modeling the swiftly-evolving Internet is that the protocols and mechanisms that we are designing now will be called upon to perform not in the current Internet, but in the Internet of the future (i.e., next month, or next year, or N years from now). This is unlike a simulation in particle physics, where the underlying structure of physical reality that is being modeled can be presumed not to undergo radical changes during the course of our research.

Accordingly, it is of high value to attempt to direct our simulations towards understanding the fundamental underlying dynamics in packet networks, rather than exploring specific performance in particular environments. Such an understanding can serve as the bedrock to build upon as the specifics of the Internet infrastructure evolve. However, some of our simulation research must of course be directed towards evaluating specific protocols or proposed new mechanisms; this evaluation requires some consideration of how the Internet infrastructure is evolving, and how this evolution is likely to affect our proposed protocol or mechanism. Striking the right balance between these is fundamentally difficult; there are, unfortunately, no easy answers for how to anticipate the evolutionary path of the Internet architecture.

7 Coping Strategies

So far we have focused our attention on the various factors that make Internet simulation a demanding and difficult endeavor. In this section we discuss some strategies for coping with these difficulties.

7.1 The search for invariants

The first observation we make is that, when faced with a world in which seemingly everything changes beneath us, any *invariant* we can discover then becomes a rare point of stability upon which we can then attempt to build. By the term "invariant" we mean some facet of Internet behavior which has been *empirically* shown to hold in a very wide range of environments. We note that we should not allow ourselves to trust alleged invariants posited on theoretical grounds—Internet measurement has all too often wholly undermined these, jettisoning the misleading theory in the process—hence the emphasis on deriving invariants from empirical observations.

From a modeling perspective, the search for invariants becomes the search for *parsimonious* models: just as the great heterogeneity of the Internet makes it extremely difficult to construct realistic simulations, for want of knowing how to set all the parameters, so too do traditional, analytic models of Internet behavior often founder for lack of utility, because they require more parameters than a practitioner has hope of being able to set in some plausible fashion. Thus, for an analytic model to prove successful, it is vital that it not require many parameters.

Thinking about Internet properties in terms of invariants has received considerable informal attention, but to our knowledge has not been addressed systematically (though see [WP98] for a related discussion). We therefore undertake here to catalog what we believe are promising candidates:

- **Diurnal patterns of activity:** It has been recognized for more than thirty years that network activity patterns follow daily patterns, with human-related activity beginning to rise around 8-9AM local time, peaking around 11AM, showing a lunch-related noontime dip, picking back up again around 1PM, peaking around 3-4PM, and then declining as the business day ends around 5PM (see, for example, [JS69, Kle76, Pax94a]). The pattern often shows renewed activity in the early evening hours, rising around say 8PM and peaking at 10-11PM, diminishing sharply after midnight. Originally, this second rise in activity was presumably due to the "late night hacker" effect, in which users took advantage of better response times during periods of otherwise light load. Now, the effect is presumed largely due to network access from users' homes rather than their offices.

  A related invariant is the presence of diminished traffic on weekends and holidays. Indeed, in Figure 1 we can discern activity dips 12 months apart, corresponding to the end-of-year holidays.

  There are significant variations in diurnal patterns, such as: different patterns for different protocols, especially those that are not human-initiated such as NNTP traffic between Network News peers [PF95]; different patterns for the same protocol, such as work-related Web surfing during the work day versus leisure-related surfing off-hours; and geographic effects due to communication across time zones. But often for a particular subclass of traffic, one can devise a plausible diurnal pattern, so we consider such patterns as collectively comprising an invariant.

- **Self-Similarity:** Longer-term correlations in the packet arrivals seen in aggregated Internet traffic are well described in terms of "self-similar" (fractal) processes. To those versed in traditional network theory, this invariant appears highly counter-intuitive. The traditional modeling framework (termed Poisson or Markovian modeling) predicts that longer-term correlations should rapidly die out, and consequently that traffic observed on large time scales should appear quite smooth. Nevertheless, a wide body of empirical data argues strongly that these correlations remain non-negligible over a large range of time scales [LTWW94, PF95, CB96].

  "Longer-term" here means, roughly, time scales from hundreds of milliseconds to tens of minutes. On longer time scales, non-stationary effects such as diurnal traffic load patterns (see previous item) become significant. On shorter time scales, effects due to the network transport protocols—which impart a great deal of structure on the timing of consecutive packets—appear to dominate traffic correlations [FGHW99]. Still, time scales from many msec to many minutes are often highly relevant for simulation scenarios.

  In principle, self-similar traffic correlations can lead to drastic reductions in the effectiveness of deploying buffers in Internet routers in order to absorb transient increases in traffic load [ENW96]. However, we must note that it remains an open question whether in very highly aggregated situations, such as on Internet backbone links, the correlations, while present, have little actual effect because the variance of the packet arrival process is quite small. In addition, in the Internet transient increases in traffic load are heavily affected by the presence (or absence) of end-to-end congestion control, which basic self-similar models do not include. That self-similarity is still finding its final place in network modeling means that a diligent researcher conducting Internet simulations must not *a priori* assume that its effects can be ignored, but must instead consider how to incorporate self-similarity into the traffic models used in a simulation.

Accurate synthesis of aggregate self-similar traffic remains an open problem. A number of algorithms exist for synthesizing exact or approximate sample paths for different forms of self-similar processes [GW94, LEWW95, Pax97b, RL95, Fla92]. These, however, solve only one part of the problem, namely how to generate a specific instance of a set of longer-term traffic correlations. The next step—how to go from the pure correlational structure, expressed in terms of a time series of packet arrivals per unit time, to the details of exactly *when* within each unit of time each individual packet arrives—has not been solved, particularly given that these finer time scales corresponding to individual arrivals are *not* well modeled as self-similar, though appear well-modeled as *multi-fractal* [FGW98]. The lack of these timing specifics in turn can lead to significant uncertainties in simulation results. Even once addressed, we still face the difficulties of packet-level simulation vs. source-level simulation discussed in §5. In this regard, we note that [WTSW95] presents one approach for unifying link-level self-similarity with specific source behavior, based on sources that exhibit ON/OFF patterns with durations drawn from distributions with heavy tails (see below).

- **Poisson session arrivals:** Network user "session" arrivals are well-described using Poisson processes. A user session arrival corresponds to the time when a human decides to use the network for a specific task. Examples are remote logins, the initiation of a file transfer (FTP) dialog, and the beginning of Web-surfing sessions. Unlike the packet arrivals discussed above, which concern when individual packets appear, session arrivals are much higher level events; each session will typically result in the exchange of hundreds of packets. [PF95] examined different network arrival processes and found solid evidence supporting the use of Poisson processes for user session arrivals, providing that the rate of the Poisson process is allowed to vary on an hourly basis. (The hourly rate adjustment relates to the diurnal pattern invariant discussed above.) That work also found that slightly finer-scale arrivals, namely the multiple network connections that comprise each session, are *not* well described as Poisson, so for these we still lack a good invariant on which to build. This in turn points up a subtle requirement in source modeling: a source model at the level of individual connections would miss the Poisson nature of the arrival of individual sessions.

- **Log-normal connection sizes:** A good rule of thumb for a distributional family for describing connection sizes or durations is log-normal, i.e., the distribution of the logarithm of the sizes or durations is well-approximated with a Gaussian distribution. [Pax94b] examined random variables associated with measured connection sizes and durations and found that, for a number of different applications, using a log-normal with mean and variance fitted to the measurements generally describes the body of the distribution as well as previously recorded empirical distributions (likewise fitted to the mean and variance of the measurements). This finding is beneficial because it means that by using an analytic description, we do not sacrifice significant accuracy over using an empirical description; but, on the other hand, the finding is less than satisfying because [Pax94b] also found that in a number of cases, neither model (analytic or empirical) fit well, due to the large variations in connection characteristics from site-to-site and over time.

- **Heavy-tailed distributions:** When characterizing distributions associated with network activity, expect to find heavy tails. By a heavy tail, we mean a Pareto distribution with shape parameter $\alpha < 2$. These tails are surprising because for $\alpha < 2$ the Pareto distribution has infinite variance. (Some statisticians argue that infinite variance is an inherently slippery property—how can it ever be verified? But then, independence can never be proven in the physical world, either, and few have difficulty accepting its use in modeling.)

  The evidence for heavy tails is widespread, including CPU time consumed by Unix processes [LO86, H-BD96]; sizes of Unix files [Irl93], compressed video frames [GW94], and World Wide Web items [CB96]; and bursts of Ethernet [WTSW95] and FTP [PF95] activity.

  Note that the log-normal distribution discussed in the previous item is not a heavy-tailed distribution, yet these two invariants are not in conflict, because the log-normal invariant refers to the *body* of the size distribution, while this invariant refers only to the *upper tail* (i.e., the distribution of extreme values).

- **Invariant distribution for Telnet packet generation:** Danzig and colleagues found that the pattern of network packets generated by a user typing at a keyboard (e.g., using a Telnet application) has an invariant distribution [DJCME92]. Subsequently, [PF95] confirmed this finding and identified the distribution as having both a Pareto upper tail and a Pareto body, in sharp contrast to the common assumption that keystrokes can be modeled using the much tamer exponential distribution.

- **Invariant characteristics of the global topology:** We described in §4.1 how properties of the underlying network topology such as link bandwidth and propagation delay can change over time. While many of the properties of the global topology are likely to change dra-

> matically over time, there are a few invariant characteristics of the global topology, namely that the Earth is divided into continents, and that the speed of light does not change. In other words, it will always be 5,850 km from New York to Paris,[2] and it will always take a signal at least 20 ms to travel between these two points. This gives a lower bound of 40 ms for the roundtrip time for a connection from New York to Paris. The Earth's geography in terms of continents and how the human population is spread among them changes only extremely slowly, and the logistics of intercontinental communication (and, in general, the fact that "farther costs more") will remain an important invariant, from which we can infer that there will always be significant structure to the global Internet topology. Earth-based Internet hosts will remain distributed mostly on the continents, with significant communication delays between continent pairs.

Some of these invariants make network analysis easier, because they nail down the specifics of behavior that otherwise might be open to speculation. Others make analysis difficult—for example, mathematical models of self-similar processes, while concise, are often very difficult to solve exactly. For simulation, however, the key is that the invariants help reduce the parameter space that must be explored. Using the invariants then serves as a step towards ensuring that the results have widespread applicability.

7.2 Carefully exploring the parameter space

A second strategy for coping with the great heterogeneity and change in the Internet architecture is to explore network behavior as a function of changing parameters. Exploring network behavior for a fixed set of parameter values can be useful for illustrating a point, or for determining whether the simulated scenario exhibits a show-stopping problem, but not for generalizing to the wider space. As one Internet researcher has put it, "If you run a single simulation, and produce a single set of numbers (e.g., throughput, delay, loss), and think that that single set of numbers shows that your algorithm is a good one, then you haven' t a clue." Instead, one must analyze the results of simulations for a wide range of parameters.

Obviously, it is rarely feasible to explore the entire parameter space. The challenge is to figure out which parameters to modify, and in what combinations. A useful approach is to hold all parameters fixed except for one element, in order to gauge the sensitivity of the simulation scenario to the single changed variable. The changed variable could relate to protocol specifics, router queue management or packet scheduling, network topologies and link properties, traffic mixes, or congestion levels. One rule of thumb is to consider orders of magnitude in parameter ranges (since many Internet properties are observed to span several orders of magnitude). In addition, because the Internet includes non-linear feedback mechanisms, and often subtle coupling between its different elements, sometimes even a slight change in a parameter can completely change numerical results (see [FJ92] for a discussion of one form of traffic phase effects).

In its simplest form, this approach serves to identify elements to which a simulation scenario is sensitive. Finding that the simulation results do *not* change as the parameter is varied does not provide a definitive result, since it could be that with altered values for the other, fixed parameters, the results *would* indeed change. On the other hand, careful examination of *why* we observe the changes we do may lead to insights into fundamental couplings between different parameters and the network's behavior. These insights in turn can give rise to new invariants, or perhaps "simulation scenario invariants," namely properties that, while not invariant over Internet traffic in general, are invariant over an interesting subset of Internet traffic.

8 The VINT Project

The difficulties with Internet simulation discussed in this paper are certainly daunting. In this section we discuss an ongoing collaborative effort to provide some shared simulation resources for the networking research community.

The UCB/LBNL network simulator NS is a multi-protocol simulator that implements unicast and multicast routing algorithms, transport and session protocols, (including both reliable and unreliable multicast protocols), reservations and integrated services, and application-level protocols such as HTTP [McC99a]. In addition, NS incorporates a range of link-layer topologies and scheduling and queue management algorithms. This level of multi-protocol simulation is fostered by contributions from many different researchers incorporated into a single simulation framework. Taken together, these contributions enable research on the interactions between the protocols and the mechanisms at the various network layers.

VINT is a joint project between researchers at a number of institutions to build a multi-protocol simulator based on NS [BBEF99], with a key goal to facilitate studies of scale and protocol interaction, and to provide a common simulation platform that can incorporate contributions from many areas of the network research community. The project also incorporates libraries of network topology generators [CDZ97] and traffic generators, the network animator NAM [McC99b], and an emulation interface to allow the NS simulator to interact with real-world traffic [F99].

One goal of the VINT project has been to study protocol interaction and behavior at large scales. The emphasis on how to deal with scaling is not on parallel simulation techniques to speed up the simulations (though these would be helpful, too), but to allow the researcher to use *different lev-*

[2] Well, until communication through the Earth's interior is possible, and great-circle distances no longer apply!

els of abstraction for different elements in the simulation.

The NS simulator has been used by a wide range of researchers to share research and simulation results and to build on each other's work. If this use of NS as *lingua franca* is expanded to other areas, then NS could make an important contribution to the advent of Internet research.

We emphasize that there is a role for many different simulators in the network research community, and that no simulator eliminates the difficulties inherent in Internet simulation. Researchers still have to take care to use the tool of simulation properly, understand the abstractions they are making, and recognize the limitations of their findings. Shared and publicly-available network simulators in the network research community make it easier for researchers to create simulations, but the researchers themselves remain responsible for making their use of simulation relevant and insightful rather than irrelevant or misleading.

9 Final Note

We hope with this discussion to spur, rather than discourage, further work on Internet simulation.

In many respects, simulating the Internet is fundamentally harder than simulation in other domains. In the Internet, due to scale, heterogeneity and dynamics, it can be difficult to evaluate the results from a single simulation or set of simulations. Researchers needs to take great care in interpreting simulation results and drawing conclusions from them. A researcher using simulation must also rely on other tools, which include measurements, experiments and analysis, when possible, as well as intuition and good judgement.

The challenge, as always, is to reap sound insight and understanding from simulations, while never mistaking simulation for the real world.

10 Acknowledgments

Our thanks to Lee Breslau and John Heidemann for feedback on this paper, and to Kinh Tieu for his careful reading of an earlier draft.

References

[BBMRS97] Baentsch, M., Baum, L., Molter, G., Rothkugel, S., and Sturm, P. World Wide Web Caching: The Application-Level View of the Internet. *IEEE Communications Magazine*, June 1997.

[BBEF99] Bajaj, S., Breslau, L., Estrin, D., Fall, K., Floyd, S., Haldar, P., Handley, M., Helmy, A., Heidemann, J., Huang, P., Kumar, S., McCanne, S., Rejaie, R., Sharma, P., Varadhan, K., Xu, Y., Yu, H., and Zappala, D., Improving Simulation for Network Research. Technical Report 99-702, University of Southern California, March, 1999.

[BBCDWW98] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., Weiss, W. An Architecture for Differentiated Services, RFC 2475, December 1998.

[CDZ97] Calvert, K., M. Doar, and E. W. Zegura. 1997. Modeling Internet topology. *IEEE Communications Magazine* 35:160–163.

[CB96] Crovella, M. and A. Bestavros. 1996. Self-similarity in World Wide Web traffic: evidence and possible causes. *Proceedings of SIGMETRICS '96*.

[CBC95] Cunha, C., A. Bestavros, and M. Crovella. 1995. Characteristics of WWW client-based traces. Technical Report TR-95-010, Boston University Computer Science Department.

[DJCME92] Danzig, P., S. Jamin, R. Cáceres, D. Mitzel, and D. Estrin. 1992. An empirical workload model for driving wide-area TCP/IP network simulations. *Internetworking: Research and Experience* 3:1–26.

[DKS90] Demers, A., S. Keshav, and S. Shenker. 1990. Analysis and simulation of a fair queueing algorithm. *Internetworking: Research and Experience* 1:3–26.

[Eri94] Eriksson, H. MBone: The Multicast Backbone. Communications of the ACM. August 1994, pp. 54-60.

[ENW96] Erramilli, A., O. Narayan, and W. Willinger. 1996. Experimental queueing analysis with long-range dependent packet traffic. *IEEE/ACM Transactions on Networking* 4:209–223.

[F99] Fall, K., Network Emulation in the Vint/NS Simulator, To appear in Proc. ISCC'99. URL “http://www.cs.berkeley.edu/ kfall/papers/iscc99.ps”.

[FGW98] Feldmann, A., A. Gilbert, and W. Willinger. 1998. Data networks as cascades: Explaining the multifractal nature of Internet WAN traffic. Proceedings of ACM/SIGCOMM.

[FGHW99] Feldmann, A., A. Gilbert, P. Huang, and W. Willinger. 1999. Dynamics of IP Traffic: A Study of the Role of Variability and the Impact of Control. Proceedings of ACM/SIGCOMM.

[Fla92] P. Flandrin, “Wavelet Analysis and Synthesis of Fractional Brownian Motion,” *IEEE Transactions on Information Theory*, 38(2), pp. 910-917, March 1992.

[FJ92] Floyd, S., and V. Jacobson. 1992. On traffic phase effects in packet-switched gateways. *Internetworking: Research and Experience* 3:115–156.

[Fyo99] Fyodor. 1999. *nmap*. http://www.insecure.org/nmap/.

[GW94] M. Garrett and W. Willinger, "Analysis, Modeling and Generation of Self-Similar VBR Video Traffic," *Proceedings of SIGCOMM '94*, pp. 269-280, September, 1994.

[H-BD96] M. Harchol-Balter and A. Downey, "Exploiting Process Lifetime Distributions for Dynamic Load Balancing," *Proc. SIGMETRICS '96*, Philadelphia, May 23-26, 1996.

[ITA] The Internet Traffic Archive. http://www.acm.org/sigcomm/ITA/.

[Irl93] G. Irlam (gordoni@netcom.com), "ufs' 93 [Updated file size survey results]," USENET newsgroup comp.os.re-search, message 2ddp3b$jn5@darkstar.UCSC.EDU, Nov. 29, 1993.

[JS69] P. E. Jackson and C. D. Stubbs, "A study of multiaccess computer communications," Proc. of the Spring 1969 AFIPS Conference, Vol. 34.

[Jac88] Jacobson, V. 1988. Congestion avoidance and control. *Proceedings of SIGCOMM '88*, pp. 314–329.

[Kle76] L. Kleinrock, "Queueing Systems, Volume II: Computer Applications," John Wiley & Sons, 1976.

[LEWW95] W-C. Lau, A. Erramilli, J. Wang, and W. Willinger, "Self-Similar Traffic Generation: The Random Midpoint Displacement Algorithm and Its Properties," *Proceedings of ICC '95*, 1995.

[LO86] W. Leland and T. Ott, "Load-balancing Heuristics and Process Behavior," *PERFORMANCE '86 and ACM SIGMETRICS 1986 Joint conference on Computer Performance Modelling, Measurement and Evaluation*, North Carolina State University, pp. 54-69, May 1986.

[LTWW94] Leland, W., M. Taqqu, W. Willinger, and D. Wilson. 1994. On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Transactions on Networking* 2:1–15.

[Lot97] *http://www.nw.com/zone/WWW/top.html*.

[MJV96] McCanne, S., V. Jacobson, and M. Vetterli, Receiver-driven Layered Multicast, ACM SIGCOMM, August 1996, Stanford, CA, pp. 117-130.

[McC99a] McCanne, S. UCB/LBNL Network Simulator—ns (version 2), http://www-mash.cs.berkeley.edu/ns/.

[McC99b] McCanne, S. Network Animator - nam, http://www-mash.cs.berkeley.edu/nam/.

[MOAT] NLANR Measurement and Operations Analysis Team. http://moat.nlanr.net/Traces/.

[NSVW99] DARPA/NIST Network Simulation Validation Workshop, May, 1999. URL "http://www.dyncorp-is.com/darpa/meetings/nist99may/".

[Pax94a] Paxson, V. 1994a. Growth trends in wide-area TCP connections. *IEEE Network* 8:8–17.

[Pax94b] Paxson, V. 1994b. Empirically-derived analytic models of wide-area TCP connections. *IEEE/ACM Transactions on Networking* 2:316–336.

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

[Pax96] Paxson, V. 1996. End-to-end routing behavior in the Internet. *Proceedings of SIGCOMM '96*, pp. 25–38.

[Pax97a] Paxson, V. 1997. Automated packet trace analysis of TCP implementations. *Proceedings of SIGCOMM '97*.

[PADR99] V. Paxson, M. Allman, S. Dawson, W. Fenner, J. Griner, I. Heavens, K. Lahey, J. Semke, B. Volz, Known TCP Implementation Problems, RFC 2525, Informational, March 1999. URL "ftp://ftp.isi.edu/in-notes/rfc2525.txt".

[Pax97b] Paxson, V. 1997. Fast, Approximate Synthesis of Fractional Gaussian Noise for Generating Self-Similar Network Traffic. Computer Communication Review, 27(5), pp. 5-18, Oct. 1997.

[Pax99] Paxson, V. 1999. End-to-End Internet Packet Dynamics. *IEEE/ACM Transactions on Networking* 7(3), pp. 277–292.

[RL95] Ryu, B., and S. Lowen. 1995. Modeling, Analysis, and Simulation of Self-Similar Traffic using the Fractal-Shot-Noise-Driven Poisson Process. Proc. IASTED International Conference on Modeling and Simulation' 95. Pittsburgh, PA.

[Ste96] Stevens, W. R. 1996. TCP/IP Illustrated, Volume 3: TCP for Transactions, HTTP, NNTP, and the UNIX Domain Protocols. Addison-Wesley.

[Tel99] Telcordia Netsizer, Internet Growth Forecasting Tool. 1999. URL "http://www.netsizer.com/".

[TMW97] K. Thompson, G. Miller, and R. Wilder, Wide Area Internet Traffic Patterns and Characteristics, IEEE Network, Nov. 1997. http://www.vbns.net/presentations/papers/MCItraffic.ps

[WTSW95] Willinger, W., M. Taqqu, R. Sherman, and D. Wilson. 1995. Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level. *Proceedings of SIGCOMM '95*, pp. 100–113.

[WP98] Willinger, W., and V. Paxson. Where Mathematics meets the Internet. Notices of the American Mathematical Society. Vol.45, No.8, pp. 961-970, August 1998.

[ZDESZ93] Zhang, L., S. Deering, D. Estrin, S. Shenker, and D. Zappala. 1993. RSVP: a new resource reservation protocol. *IEEE Network*. 7:8-18.