Empirically-Derived Analytic Models of Wide-Area TCP Connections: Extended Report

Vern Paxson
Lawrence Berkeley Laboratory and
EECS Division, University of California, Berkeley
1 Cyclotron Road
Berkeley, CA 94720
vern@ee.lbl.gov

LBL-34086

June 15, 1993

Abstract

We analyze 2.5 million TCP connections that occurred during 14 wide-area traffic traces. The traces were gathered at five "stub" networks and two internetwork gateways, providing a diverse look at wide-area traffic. We derive analytic models describing the random variables associated with *telnet*, *nntp*, *smtp*, and *ftp* connections, and present a methodology for comparing the effectiveness of the analytic models with empirical models such as *tcplib* [DJ91]. Overall we find that the analytic models provide good descriptions, generally modeling the various distributions as well as empirical models and in some cases better.

1 Introduction

Though wide-area networks have been in use since the early 1970's, until recently we have known virtually nothing about the characteristics of the individual connections of different protocols. In the last few years a number of papers have appeared giving statistical summaries of traffic on a per-protocol basis [Càceres89, Heimlich90, CW91, EHS92, WLC92], an important first step. The next step in understanding wide-area traffic is to form models for simulating and predicting traffic.

One such model, *tcplib* [DJ91, DJCME92], is now available. *tcplib* is an *empirical* model of wide-area traffic: it models the distribution of the random variables (e.g., bytes transferred, duration) associated with different protocols by using the distributions actually measured for those protocols at an Internet site.

Ideally we would like to have *analytic* traffic models: simple mathematical descriptions rather than empirical distributions. Such models are easier both to convey and to analyze. The key question is whether analytic models can describe the diverse phenomena found in wide-area traffic as well as empirical models. Our previous work [Paxson91] offered such models but suffered in part from flawed statistical methodology.

In this paper we analyze 14 wide-area traffic traces gathered at seven different sites, five "stub" networks and two internetwork gateways. We derive analytic models describing the random variables associated with *telnet*, *nntp*, *smtp*, and *ftp* connections, and present a methodology for comparing the effectiveness of the analytic models with *tcplib* and with another empirical model constructed from one of the datasets. Table 1 summarizes our main results. Overall we find that the analytic models provide good descriptions, generally modeling the various distributions as well as the empirical models and in some cases better. We develop each of these findings in the remainder of the paper.

In the next section we give an overview of the 14 traffic traces. We describe the gross characteristics of the traces including their traffic mix, and discuss how we filtered the traffic to remove anomalous connections.

The following section presents our statistical methodology. We discuss how we transformed the data and dealt with outliers; our unsuccessful attempts to find "statistically valid" models; the metric we devised for comparing the fit of two different models to a dataset; and our methodology for modeling connection interarrivals, which is more complex than modeling the other random variables associated with a connection.

We then present one section each on modeling *telnet*, *nntp*, *smtp*, and *ftp*. These sections can be read independently if the reader is more interested in one protocol than another, except that the first section describes how to read the plots used in all four sections to compare models. By reading the text accompanying Figures 4, 5, 13, and 30, the reader can if desired skip the remainder of the *telnet* section.

Random variables associated with wide-area network connections can be described as well by analytic models as by empirical models.
When using either type of model, caution must be exercised due to frequent discrepancies in the upper 1% tails.
While in general the analytic models do not match the observed distributions identically in a statistical sense, often a random subsample of hundreds of data points does result in a statistically valid fit, indicating that the analytic models are often *close* though not *exact*.
Bulk-transfer traffic (*ftpdata*, *smtp*, *nntp*, and *telnet* response) is best modeled using log-normal distributions.
Bulk-transfer traffic is not strongly bidirectional; the responses to bulk transfers show little variation relative to the variation in the size of the transfer.
Network traffic varies significantly, both over time and more so from site-to-site, not only in traffic mix but in connection characteristics.
Scaling usually helps significantly in modeling the bytes transferred by *nntp*, *smtp*, *rlogin*, and individual *ftpdata* connections, but is usually not necessary for adequate fits to *telnet* connections and full *ftp* conversations.
Except for *nntp*, connection interarrivals are well modeled using nonhomogeneous Poisson processes with fixed hourly rates.

Table 1: Major Findings

In the last section we summarize the different analytic models and discuss findings in addition to those listed in Table 1. We also include appendices summarizing how we filtered the data prior to analysis, and exploring the effectiveness of modeling *rlogin* traffic using the *telnet* models.

2 Overview of Network Traffic Traces

To develop and then evaluate our models we acquired a number of traces of wide-area traffic. Our main data were from six month-long traces of all wide-area TCP connections between the Lawrence Berkeley Laboratory (LBL) and the rest of the world. With the help of colleagues we also were able to study traces from Bellcore, the University of California at Berkeley, the University of Southern California, Digital's Western Research Laboratory, the United Kingdom–United States academic network link, and traffic between the coNCert[1] network and the rest of the world. We discuss the general characteristics of each of these datasets in turn and then provide summaries of their TCP traffic.

2.1 The LBL Traces

All off-site communication at LBL funnels through a group of gateways that reside on a network separate from the rest of the Laboratory. The first two datasets were taken using a Sun 3/50 residing on the gateway network, using the *tcpdump* packet capture tool [JLM89] running the Berkeley Packet Filter [MJ93]. The Sun 3/50 had kernel modifications to gain a clock resolution of 10 msec. These are the traces discussed in [Paxson91]. When we took the last four traces the monitor workstation had been upgraded to a Sun SLC with a consequent improvement of clock resolution to 1 microsecond.

We used a *tcpdump* filter to capture only those TCP packets with SYN, FIN, or RST flags in their headers, greatly reducing the volume and rate of data (but at the cost of no analysis of intra-connection dynamics). From SYN and FIN packets one can derive the connection's TCP protocol, connection duration, number of bytes transferred in each direction (excluding TCP/IP overhead), participating hosts, and starting time. In principle we could derive the same information using RST packets instead of FIN packets, but we found that often the sequence numbers associated with RST packets were erroneous. Since we could not derive reliable byte counts from RST-terminated connections we excluded them from subsequent analysis.

With this packet capture scheme there are two mechanisms by which packets can be lost. The first is that, if a packet arrives at the Ethernet controller and the controller has run out of kernel memory to buffer the packet, it drops the packet and sets a bit indicating that this event occurred. The Ethernet driver subsequently reads the bit and increments a corresponding counter. It is possible that more than one packet will be dropped before the driver is able to read the bit, so the actual number of dropped packets is unknown but at least as large as the driver's counter.

The second packet-drop mechanism occurs when the kernel determines that the packet filter accepts a packet, but has no

[1] Communications for North Carolina Education, Research and Technology.

more buffer space for saving the packet (due to the user-level program failing to consume previously accepted packets). In this case the kernel drops the packet and increments a counter. Values reported by this counter thus correspond to exactly the number of acceptable packets (in our case, SYN/FIN/RST packets) dropped.

Dataset	Packets (days)	Start	End	Drops
LBL-1	124M (36)	01Nov90	01Dec90	$0 + 0$
LBL-2	?	28Feb91	30Mar91	0+?
LBL-3	207M (47)	07Nov91	07Dec91	$9 + 24$
LBL-4	210M (36)	19Mar92	18Apr92	$6 + 233$
LBL-5	337M (35)	24Sep92	23Oct92	$8 + 1808$
LBL-6	447M (31)	24Feb93	26Mar93	$3 + 0$

Table 2: Summary of LBL Datasets

Table 2 summarizes the LBL datasets. The second column gives the total number of network packets received by the kernel for each dataset, along with the number of days spanned by the entire trace. (The statistics missing for the LBL-2 dataset are due to abnormal termination of the tracing program.) Each dataset was then trimmed to span exactly 30 days, beginning at midnight on a Thursday and ending at midnight on a Saturday (i.e., just after 11:59PM Friday night), except for LBL-6, which begins on a Wednesday and ends on a Friday midnight. The "Drops" column gives the drop count reported by the Ethernet driver followed by the drop count reported by *tcpdump*; this last value represents dropped SYN/FIN/RST packets.[2]

Finally, since the LBL datasets span 2.5 years at roughly regular intervals, they provide an opportunity to study how a site's wide-area traffic evolves over time. Such a study is reported in [Paxson93].

2.2 The Additional Traces

As mentioned above, a number of colleagues generously provided access to traffic traces from other sites. The authors of [DJCME92] provided their traces of traffic from Bellcore, U.C. Berkeley, and U.S.C.; Jeffrey Mogul provided traces from DEC-WRL; Wayne Sung provided traces of traffic to/from the coNCert network in North Carolina; and the authors of [WLC92] provided their traces of the UK–US academic network. The first four traces all originate from "stub" sites, while the latter two represent inter-network traffic (though the authors of [WLC92] characterize the UK side of the UK–US traffic as similar to a large stub site since it comprises only a few hosts).

Site	Starting Time	Duration	Drops
Bellcore (BC)	Tue 14:37 10Oct89	13 days	0
UCB (UCB)	Tue 10:30 31Oct89	24 hours	0
USC (USC)	Tue 14:24 22Jan91	26 hours	$\approx 0.6\%^*$
DEC (DEC-1)	Tue 16:46 26Nov91	24 hours	?
DEC (DEC-2)	Wed 17:55 27Nov91	24 hours	?
DEC (DEC-3)	Mon 15:02 02Dec91	24 hours	?
coNCert (NC)	Wed 09:04 04Dec91	24 hours	?
UK-US (UK)	Wed 05:00 21Aug91	17 hours	0

Table 3: Summary of Additional Datasets

The additional datasets are summarized in Table 3. Next to the site name we give in parentheses the abbreviation we will use to identify the dataset. The drop rates for the first three datasets correspond to those listed in [DJCME92]; for the last dataset, to that listed in [WLC92]; and the drop rates for the remaining datasets were unavailable. The USC dataset's drop rate is marked because we found our copy of the trace plagued throughout by "blackouts" of missing packets, occurring almost exactly a minute apart and each blackout lasting roughly ten seconds.[3] Because of these blackouts, we exclude the USC dataset from our interarrival models.

2.3 Filtering of non-WAN traffic

Before proceeding with our analysis we filtered out non-wide-area traffic from the datasets: internal and transit traffic. The details are given in Appendix A. In addition, we removed from the LBL datasets all traffic between LBL and U.C. Berkeley[4]. While traffic with the University forms a significant fraction of LBL's off-site traffic (20-40% of all connections), it is atypical wide-area traffic due to the close administrative ties and the short, high-speed link between the institutions.

2.4 Traffic Overview

We now turn to characterizing the different datasets in order to gauge their large-scale similarities and differences. Of previous traffic studies, only [FJ70], the related [JS69], and [DJCME92] compare traffic from more than one institution. The first two papers found significant differences between their four traffic sites, which they attribute to the fact that the different sites engaged in different applications and had different hardware. The authors of [DJCME92] found that their three sites (which correspond to the USC and UCB datasets in

[2] In the LBL-4 dataset we observed the heated exchange of nearly 400,000 RST packets sent between a lone remote host and three LBL hosts, separated by a geometric mean of 1.3 msec. In LBL-5 we observed the exchange of nearly 120,000 RST packets between a single pair of hosts, virtually all occurring during a 98 second period, separated by a geometric mean of 400 μsec. LBL-6 did not include any RST bursts.

The LBL-5 RST bursts correspond to enough traffic to consume $>$ 500 kbit/sec. We suspect that the RST bursts are the cause of the relatively large number of dropped SYN/FIN/RST packets in LBL-4 and LBL-5.

[3] These blackouts do not correspond to network outages; sequence numbers of TCP connections spanning outages show jumps.

[4] Including *nntp*, unlike [Paxson93], which keeps the *nntp* traffic.

Dataset	# Conn	*nntp*	*smtp*	*ftpdata*	*ftpctrl*	*telnet*	*rlogin*	*finger*	*domain*	*X11*	*shell*	*other*
LBL-1	146,209	40	26	16	3	4	1	4	4	0.2	0.5	0.5
LBL-2	170,718	34	30	16	3	4	1	5	4	0.2	0.2	0.7
LBL-3	229,835	20	33	17	3	4	1	4	11	0.4	0.3	5
LBL-4	449,357	16	21	15	3	2	1	32	5	0.4	0.2	4
LBL-4*	312,429	23	30	21	4	3	1	3	8	0.5	0.3	5
LBL-5	370,397	14	34	22	5	4	1	6	8	0.9	0.2	5
LBL-6	528,784	11	40	23	6	3	0.8	5	5	0.7	0.4	4
BC	17,225	2	49	30	4	4	2	5	0.1	0.1	0.5	2
UCB	37,624	18	45	18	2	2	0.9	12	0.1	0.02	0.2	0.8
USC	13,097	35	27	14	2	3	1	11	2	0.09	0.3	3
DEC-1	72,821	33	35	11	1	0.08	0.05	0.1	20	0	0.001	0.8
DEC-2	49,050	38	22	8	1	0.04	0.06	0.2	29	0	0.02	1
DEC-3	73,440	26	43	9	1	0.07	0.07	0.2	19	0	0.003	1
NC	62,819	1	42	30	4	5	0.3	5	0.8	0.03	0.3	5
UK	25,669	0.02	42	39	7	4	0.4	0.9	1	0.02	0.02	4

Table 4: Percentage Connection Mixes for All Datasets

Dataset	MB	*nntp*	*smtp*	*ftpdata*	*ftpctrl*	*telnet*	*rlogin*	*finger*	*domain*	*X11*	*shell*	*other*
LBL-1	2,852	19	5	65	0.2	6	0.8	0.1	1	3	1	0.1
LBL-2	3,785	14	6	67	0.2	5	1	0.1	0.9	1	3	2
LBL-3	6,710	7	4	67	0.1	4	1	0.1	0.7	3	11	1
LBL-4	11,398	21	4	52	0.1	4	0.9	0.0	0.6	6	10	1
LBL-5	19,269	17	3	57	0.1	3	0.7	0.1	0.4	11	8	1
LBL-6	22,076	22	5	57	0.2	2	0.7	0.1	0.5	8	3	0.8
BC	346	4	8	78	0.3	4	2	0.2	0.1	0.1	2	2
UCB	318	23	16	50	0.3	4	3	0.9	0.0	0.2	0.6	1
USC	362	62	3	18	0.1	2	0.9	0.3	0.3	5	7	2
DEC-1	981	43	17	38	0.2	0.1	0.2	0.0	0.7	0.0	0.0	1
DEC-2	819	54	14	30	0.1	0.0	0.2	0.1	0.6	0.0	0.0	2
DEC-3	1,379	52	16	30	0.1	0.1	0.2	0.1	0.6	0.0	0.0	1
NC	1,553	9	8	68	0.3	5	0.3	0.1	0.3	0.1	0.3	8
UK	625	0.5	11	80	0.4	3	0.5	0.0	0.3	0. 1	0.5	4

Table 5: Percentage Byte Mixes for All Datasets

this paper, as well as part of the BC dataset) had quite different mixes of traffic, but that the characteristics of any particular protocol's traffic were very similar (though they did not quantify the degree of similarity).

Table 4 shows the "connection mix" for each of the datasets. The second column gives the total number of connections recorded, and the remaining columns the percentage of the total due to particular TCP protocols. The mixes for BC, UCB, and USC differ from those given in [DJCME92] because the latter reports *conversation* mixes, where multiple related connections have been combined into single conversations. (The authors also used twenty-minute silences to delimit the end of connections, instead of FIN packets.)

From the Table it is immediately clear that traffic mixes for all protocols vary substantially, both from site-to-site and over time (for LBL). There are also a number of anomalies which merit comment:

- The huge spike in the LBL-4 *finger* connections, the large jump in *other* connections at LBL-3, and the increasing proportion of *ftpctrl* traffic (i.e., the control side of an *ftp* conversation), are all due to the use of background scripts to automate periodic network access. Reference [Paxson93] explores this phenomenon further. LBL-4* shows the LBL-4 connection mix with the periodic *finger* connections removed, as they significantly skew the mix profile.
- The large variance of LBL's *nntp* mix is due to changes in LBL's *nntp* peer servers and differences in the rate at which new news arrives. Again, see [Paxson93] for a

discussion.

- DEC has a "firewall" in place which prohibits traffic other than *nntp*, *smtp*, and *ftp*, and *domain*. The little remaining traffic due to other protocols originated on the outside of the firewall.

- The DEC-2 dataset includes part of the Thanksgiving holiday, accounting for the depressed number of connections.

- As mentioned in [WLC92], the United Kingdom receives its network news from Holland, hence the very low proportion of *nntp* connections.

Table 5 shows the total number of data megabytes transferred (in either direction) for each of the datasets, along with the "byte mix"—the percentage of the total bytes due to each protocol. The LBL datasets show striking growth over time, which we explore further in [Paxson93]. The LBL datasets naturally total more bytes than the others because they span 30-day periods, as opposed to about 1 day for all the other datasets except BC (see Table 3).

We see immediately that, much as with the connection mix, the byte mix also varies considerably both from site-to-site and over time. Some sites (the first three LBL datasets, BC, NC, and UK) are wholly dominated by *ftp* traffic, while others (the last three LBL datasets, UCB, and the DEC datasets) show more of a balance between *nntp* and *ftp* traffic; and USC is dominated by *nntp* traffic. For some sites (UCB, DEC), *smtp* traffic contributes a significant volume, and for others (LBL, USC), traffic due to *X11* and *shell* far outweighs the almost negligible proportion of connections due to those protocols (see Table 4).

We now turn to the development of the statistical methodology that we will use to characterize the individual connections that make up the data shown in Tables 4 and 5.

3 Statistical Methodology

As noted in [Pawlita89], one weakness of many traffic studies to date has been in their use of statistics. Often the studies report only first or perhaps second moments, and distributions are summarized by eye. Frequently they omit discussion of dealing with outliers, and rarely do they report goodness-of-fit methodologies and results. The few cases where goodness-of-fit issues have been discussed are somewhat unsatisfying (the authors of [FJ70] developed their own, apparently never-published goodness-of-fit measure; and in our own previous work [Paxson91] we used the Kolmogorov-Smirnov goodness-of-fit test as a goodness-of-fit *metric*, an inferior choice). We endeavor in this work to address these statistical shortcomings and to present a general statistical methodology that might serve future work as well.

Our initial goal was to develop "statistically valid" analytic models of the characteristics of wide-area network use. By statistically valid we mean models whose distributions for random variables could not be distinguished in a statistical sense from the actual observed distributions of the variables. In this attempt we failed. Most of the models we present do not reflect the underlying data in a statitistically valid sense; that is, we cannot say that our analytic distributions do indeed precisely give the distributions of the random variables they purport to model. We discuss our failure in Section 3.8 below, and then in Section 3.9 develop a "metric" for determining which of two statistically invalid models better fits a given dataset. But first we discuss the value of statistically valid analytic models and our methodology for developing them, as these issues remain fundamental to putting our results in perspective.

3.1 Analytic vs. Empirical Models

For our purposes we define an analytic model of a random variable as a mathematical description of that variable's distribution. Ideally the model has few bound parameters (making it easy to understand) and no free parameters (making it predictive), in which case it fully predicts the distribution of similar random variables derived from datasets other than the ones used to developed the model. But typically the model might include free offset and scale parameters, in which case it predicts the general shape of future distributions but not the exact form. If those parameters are known for a future dataset, then the model becomes fully predictive for that dataset.

In contrast, an empirical model such as *tcplib* describes a random variable's distribution based on the observed distribution of an earlier sample of the variable. The empirical model includes a great number of bound parameters, one per bin used to characterize the variable's distribution function; it may be predictive but not easy to understand.

There are a number of advantages of an analytic model compared to an empirical model for the same random variable:

- analytic models are often mathematically tractable, lending themselves to greater understanding;

- analytic models are very concise and thus easily communicated;

- with an analytic model, different datasets can be easily compared by comparing their fitted values for the model's free parameters.

A key question, though, is whether an analytic model fully captures the essence of the quantity measured by a random variable. An empirical model perfectly models the dataset from which it was derived; the same cannot be said of an analytic model. If the analytic model strays too far from reality, then, while the above advantages remain true, the model no longer applies to the underlying phenomena of primary interest, and becomes useless (or misleading, if one does not recognize that the model is inaccurate).

The key question then is how to tell that an analytic model accurately reflects reality as represented by a dataset of samples. One approach is to require that the random variable distributions predicted by the model and those actually observed be indiscernable in a statistical sense. To test for such agreement we turn to goodness-of-fit techniques.

3.2 Goodness-of-fit Tests

The random variables we model (amount of data transferred, connection duration, interarrival times, and ratios of these quantities) all come from distributions with essentially unbounded maxima. Furthermore, these distributions are either continuous or, in the case of data transferred, continuous in the non-negative integers. As such the values of the variables do not naturally fall into a finite number of categories, which makes using the well-known chi-squared test less than ideal because it requires somewhat arbitrary choices regarding binning [Knuth81, DS86].

The goodness-of-fit test commonly used with continuous data is the Kolmogorov-Smirnov test. The authors of [DS86], however, recommend the Anderson-Darling (A^2) test [AD54] instead. They state that A^2 is often much more powerful than either Kolmogorov-Smirnov or chi-squared, and that A^2 is particularly good for detecting deviations in the tails of a distribution, often the most important to detect. We followed their recommendation and, in attempting to develop statistically valid models, always used A^2 in assessing goodness-of-fit.

3.3 Logarithmic Transformations

When analyzing data drawn from distributions unbounded in one direction and bounded in the other, often it helps to re-express the data by applying a logarithmic transformation [MT77]. We found that for many of our models logarithmic transformations were required to discern patterns in the large range of values in the data.

For convenience we developed and tested our models using a $\log_2 x$ transformation. Note that, when converting from logarithmic models back to untransformed models, arithmetic means of transformed values become *geometric* means of the untransformed values, and standard deviations become *factors* instead of additive values. For example, a log-normal model with $\bar{x} = 4.0$ and $\sigma = 2.5$ specifies that any observation within a factor of 5.66 ($2^{2.5}$) of 16 ($2^{4.0}$) lies within one standard deviation of the geometric mean. Thus, 2.83 ($= 16/5.66$) and 90.56 ($= 16 \times 5.66$) are the boundaries of values lying within one standard deviation of the geometric mean, which is 16.

3.4 Dealing with Outliers

When applying a logarithmic transformation to non-negative data, one immediately runs into the problem of what to do with data equal to zero. Fortunately for us, in our data such values are rare (and confined to values representing number of data bytes transferred), so we decided to eliminate any connections in which the number of bytes transferred in either direction was zero. We report in Appendix B the number of connections thus eliminated for each dataset; in the worst case they comprised 0.5% of the total connections. An alternative approach would have been to bias our logarithms, by using $\log_2(x+1)$ rather than $\log_2 x$; we rejected this approach as being error-prone when converting to and from the logarithmic models.

Some of our datasets also exhibited values so anomalously large that we removed their associated connections from our study. These outliers were much rarer than those discussed above. Often the values were clearly due to protocol errors (for example, connections in which the sequence numbers indicated $2^{32} - 1$ bytes transferred). We discuss these outliers also in Appendix B.

Finally, we restricted our analysis to datasets with at least 100 connections of interest, to prevent small, anomalous datasets from skewing our results.

3.5 Censored Data

Some of our models describe only a portion of the distribution of a random variable (such as the upper 80% of the distribution). Reference [DS86] discuss modified goodness-of-fit tests (including A^2) to use with such *censored* distributions, in which a known fraction of either tail has been removed from the measurements prior to applying the test. In addition, they describe a method (due to Gupta [Gupta52]) for estimating the mean and variance of such a censored distribution, which can be used to derive estimated parameters of a model from censored data.

3.6 Deriving Model Parameters from Datasets

Often a model has free parameters that must be estimated from a given dataset before testing the model for validity in describing that dataset. For example, a log-normal model may require that the geometric mean and standard deviation be estimated from the dataset. The authors of [DS86] make the important point that estimating free parameters from datasets alters the significance levels corresponding to statistics such as A^2 computed from the fitted model. They then provide both methods to estimate free parameters from datasets, and the required modifications for interpreting the significance of the resulting A^2 (and other) statistics. We followed their approach.

3.7 Model Development vs. Testing

To know if a model is truly predictive, we must test it on data other than that used to develop the model. To this end, we developed all of our models using the first half of the LBL-1 through LBL-4 datasets. We refer to these below as the "test datasets". We then tested the models against the second half of these LBL datasets along with the entirety of the remaining datasets (including all of LBL-5 and LBL-6).

Below we compare our analytic models with two empirical models: one derived from the UCB dataset, which is essentially the same as the *tcplib* model, and one derived from all of LBL-2. Thus, in keeping with our goal of testing models only on data other than that used to develop them, we do not report results for fits to these datasets. An exception is for our interarrival models, which in general we do not compare to the empirical models (see Section 3.11 below).

3.8 Failure to Find Statistically Valid Models

Using the methodology described above, we attempted to develop models for a number of random variables for TCP connections of various protocols. While we often could find fairly simple analytic models that appeared to the eye to closely match the distributions of the random variables for a given dataset, these models rarely proved valid at a significance level of 5%, or even 1%, when tested against other datasets.[5]

What we found tantalizing, though, is that often, when we subsampled the dataset, we *did* find valid fits to the smaller sample. This pattern held whether the subsamples were constructed randomly or chronologically (for example, testing each day in the LBL datasets separately). We tested whether the pattern was due to daily variations in the model's parameters by using autocorrelation plots. We found such patterns only in the arrival process and bytes transferred of *nntp*, and bytes transferred by *smtp* connections. We discuss these findings below in Sections 5.2 and 6.2. We did not find any consistent patterns in the LBL *telnet* or *ftp* test datasets, ruling out simple hourly, daily, or weekly patterns in the parameters.

These findings are consistent with our models being *close* to describing the distributions but not statistically *exact*. In such a case it will take a large number of sample points for a goodness-of-fit test to discern a difference between the distributions. When we subsample we present the test with fewer points and the fit is then more likely to be found valid.

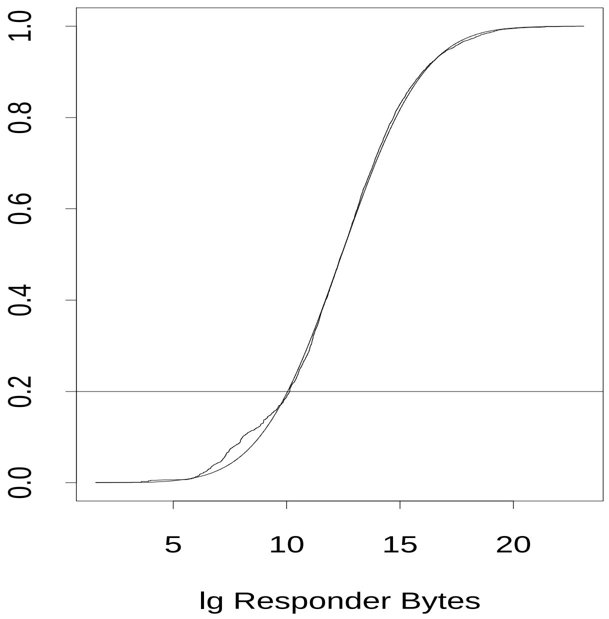

Figure 1: Censored Log-Normal Fit to Upper 80% of LBL-4 TELNET Responder Bytes

Figure 1 illustrates the problem. Here we see the distribution of $\log_2$ of the bytes sent by the *telnet* responder (i.e., not the host that began the connection) for the first half of the LBL-4 dataset. Fitted against the distribution is our responder-bytes model, which uses a normal distribution for the upper 80% of the data (and ignores the lower 20%). The horizontal line indicates the 20th percentile; the goodness-of-fit test applied only to the agreement above this line. While judging visually we might be tempted to call the fit "good", it fails the A^2 test even at the 1% level.

This sample consisted of 5,448 points. We then subsampled 1,000 points randomly, tested the validity of the model's fit to the subsample, and repeated the process 100 times. Of these 100 tests, 79 were valid at the 1% level and 55 at the 5% level. Thus we feel confident that the model is *close*, though we know it is not exact.

3.9 Comparing Analytic and Empirical Models

While we must abandon our initial goal of producing statistically valid, "exact" models, we still can produce useful analytic models by building on the work of [DJ91, DJCME92] in the following way. In those papers the authors argue that their empirical models are valuable because the variation in traffic characteristics from site-to-site and over time is fairly small. Therefore the *tcplib* models, which were taken from the UCB dataset, faithfully reproduce the characteristics of wide-area TCP connections. If we can develop analytic models that fit other datasets as well as *tcplib* does, then the analytic models are just as good at reproducing the characteristics of wide-area TCP connections; a network researcher is just as well off using either set of models, and may prefer the analytic descriptions for the advantages discussed in Section 3.1.

The question then remains how to compare an analytic

[5] A significance level of 5% indicates a 5% probability that the A^2 test erroneously declares the analytic model to *not* fit the dataset. A 5% test is more stringent than a 1% test; it errs more often because it demands a closer correspondence between the model and the dataset before declaring a "good fit." See [Ross87, pp. 205-206] for further discussion of significance levels.

model with an empirical one. Rather than a goodness-of-fit *test*, we need some sort of goodness-of-fit *metric*. While under certain conditions one can apply tests like A^2 as metrics [DS86], they are not appropriate metrics for measuring the fit of an empirical model; the tests are designed for comparing a continuous distribution (an analytic one) with an empirical distribution.

We chose as our metric a measure of "bin" frequencies, similar to a chi-squared test. A chi-squared test computes:

$$X^2 = \sum_{i=1}^{M} \frac{(N_i - np_i)^2}{np_i}$$

where M is the number of bins, p_i the fraction of all observations predicted to fall into the ith bin, n the total number of observations, and N_i the number of observations actually falling into the ith bin. We make one important change, though. If a chi-squared test is used to compare non-identical distributions, then the resulting X^2 increases with n, making it difficult to compare X^2 values when testing a distribution against different-sized datasets to see which it more closely matches. If two distributions are different, then for large values of n, $N_i/(np_i)$ will approach some fixed factor ρ_i, and the squared term in the X^2 computation approaches $(\rho_i - 1)(np_i)$. We then see that the metric:

$$K^2 = \sum_{i=1}^{M} \frac{(N_i - np_i)^2}{(np_i)^2}$$

remains invariant with increasing n. If the bins have equal width, then we have:

$$K^2 = \sum_{i=1}^{M} (\rho_i - 1)^2$$

which allows us to compute μ, the "average deviation" in each bin:

$$\mu = \sqrt{K^2/M} \qquad (1)$$

We interpret μ as follows: the value of K^2 we observed is consistent with what we would observe if in *each* bin the proportion of observations deviates from the predicted proportion by μ, i.e., $|N_i - np_i|/(np_i) = \mu$. While in general the deviation will vary from bin to bin, we can use μ to summarize the "average" deviation.

We are faced with several problems when using this metric:

- Similar to the problems using chi-squared tests mentioned in Section 3.2 above, we are forced to make a somewhat arbitrary choice as to how many bins to use. We chose to use ten equal-sized bins, so as to measure the deviation from the predicted distribution within each 10th percentile.[6]

- The metric does not inform us of deviations in the distribution tails, often the most important type of deviation. We address this shortcoming in the next section.

- The metric does not inform of us interesting, localized spikes or clumps. Within a single bin we may miss considerable departure from a model; the danger is particularly acute when testing analytic models, since their continuous nature does not usually allow for clumping. Empirical models, on the other hand, may exactly predict the clumping.

  We do not believe this problem to be major because in our studying of the LBL test datasets to form our models we rarely encountered consistent clumping (we make mention below of those occasions when we did). We also note that if clumping exists and is not accompanied by nearby sparseness, then the clump will "pull" more values into the bin than a model without the clump would predict, which will raise the μ value. So a major clump may be detected as an overall poor fit by the model.

- An empirical model does not always allow us to create equal-sized bins. It may be that the model has a single-valued spike straddling a bin boundary (for an exaggerated example, suppose that the lower 20% of an empirical distribution are all equal and we want to create bins 10% wide). We deal with this case by placing the entire spike in the lower bin and adjusting the bin widths accordingly. If the spike is substantial and not aberrant, then this procedure will aid the fit of the empirical model more than that of an analytic model.

- Since an empirical model has bounds on the range of values it allows for, the tested dataset may have values not corresponding to any bin. We removed such values from the dataset prior to computing its fit to the model. We did, however, include these values in the summary of deviation in the tails (see Section 3.10 below).

We use the μ metric to gauge how closely the distributions of different models match that of a particular observed distribution. We deem the model distribution with the lowest μ value as corresponding to the best-fitting model for the observed distribution. In general we tested each dataset against three model distributions: one produced by our analytic model, one produced using the empirical distributions found in the UCB dataset, and one drawn from the LBL-2 dataset. As mentioned in Section 2.4, the distributions in *tcplib* come from the UCB dataset, with some minor differences in the data reduction. Thus, how well the UCB dataset fits the other datasets should closely match the fit of *tcplib* to those datasets. If the analytic models fit the datasets as well or better, then we argue that the analytic models provide as good or better an overall model. Finally, to guard against the possibility that the UCB dataset is atypical and that better

[6] In one case below we use nine bins, to accommodate censored data.

empirical models might exist, we also constructed and tested an empirical model consisting of the entire LBL-2 dataset.

We developed and settled on this metric prior to observing the values it gave for the different models. We tested two versions of each model. In the first version all parameters were fixed; none were derived from the dataset being tested. When developing our analytic models we picked for each free parameter a round value lying somewhere in the range the parameter exhibited in the LBL test datasets. We chose round values as reminders that there is in general considerable range in the possible values of the parameters, and that our choice was therefore somewhat arbitrary (nearby choices would work just as well).

In the second version of each model we derived the model's free parameters from the dataset being tested. For empirical models we applied a linear transformation to the empirical distribution so that its mean and standard deviation matched that computed for the tested dataset. We refer to this second type of model as *scaled*.

3.10 A Metric for Deviation in the Tails

We summarize each model's fit to the extreme tails as follows. Suppose we test the model against n datasets. For the ith dataset, let x_i be the number of observations predicted to lie in the tail, and y_i be the number actually found to do so. Define:

$$\xi = \frac{1}{n}\sum_{i=1}^{n} \log \frac{x_i}{y_i} \tag{2}$$

ξ then gives the mean of the natural logarithm of the proportion by which the model overestimates the population of the tail. Positive values of ξ indicate that the model overestimates the tail, either consistently or in a few cases grossly. Similarly, negative values indicate the model underestimates the tail.

With this definition, an underestimate by a factor of two ($x_i/y_i = 1/2$) is just as bad as an overestimate by the same factor ($x_i/y_i = 2$), though if the two occur in different datasets they will cancel out one another. Values of ξ close to 0.0 indicate that either the model consistently does well in modeling the tail, or overestimates for some datasets and underestimates for others. In the latter case there probably is great diversity in the distribution's tail across the different datasets, and the model's estimate of the tail is a good compromise.

One problem arises when using this definition of ξ: if y_i is 0 then ξ becomes undefined. We address this problem by replacing x_i/y_i with 100 in these cases.

In comparing models we summarize how well each model does in the 10% and 1% tails. For models describing bytes transferred, we only summarize the upper tails, as in these cases disagreement in the lower tails is a matter of predicting a few bytes too many (or few) in small connections, while disagreement in the upper tails can result in large connections that are megabytes too big or small. For other models we summarize both the upper and lower tails.

3.11 Modeling Interarrivals

The final aspect of our methodology is how we model connection interarrivals. Our hope was to successfully model interarrivals as Poisson processes, as these have many attractive properties and a natural interpretation (uncorrelated, memoryless arrivals).

We cannot hope for much success, though, if we simply model the interarrival distribution directly: we expect that the arrival process will vary over the course of each day, since computer users tend to work during daylight hours, take lunch breaks, and so on; we do not expect a *homogeneous* Poisson arrival process. Instead we first look at the relative rate of connection arrivals over the course of a day in order to develop a nonhomogeneous Poisson model.

Figure 2 shows the mean, normalized, hourly connection rate for the test datasets. For each hour we plot the fraction of the entire day's connections that occurred during that hour. We see, for example, that *telnet* connections are particularly prominent during the 8AM-6PM working hours, with a lunch-related dip at noontime; this pattern has been widely observed before. *ftp* file transfers have a similar hourly profile, but they show substantial renewal in the evening hours, when presumably users take advantage of lower networking delays. The *nntp* traffic hums along at a fairly constant rate, only dipping somewhat in the early morning hours (but the mean size of each connection varies over the course of the day; see Section 5.2). The *smtp* traffic is interesting because it shows more of a morning bias than either *telnet* or *nntp*. To explore this bias we have also plotted the hourly rates for the BC dataset's *smtp* connections. Here we see a significant afternoon bias. As LBL lies on the west coast of the United States and Bellcore to the east, three time zones away, we can interpret this difference as being due to cross-country mail: mail sent by east-coast users arrives early in the day for west-coast users, and mail sent by west-coast users late in the day for east-coast users.

We can then use this data to attempt to model interarrivals as Poisson processes. First we compress datasets consisting of more than one day into a single "superday" by grouping together all connections beginning during each hour of the day. For example, all connections arriving between 9:00AM and 9:59AM are placed in one 9AM "superhour", regardless of during which day the connection arrived. The hope is that the daily variations are considerably less than the hourly variations, which is true in general except for weekends, during which much less traffic is generated. But because weekends have many fewer arrivals, the effect of aggregating them with weekday connections of the same hour is small.

Next we predict the number of connections occurring dur-

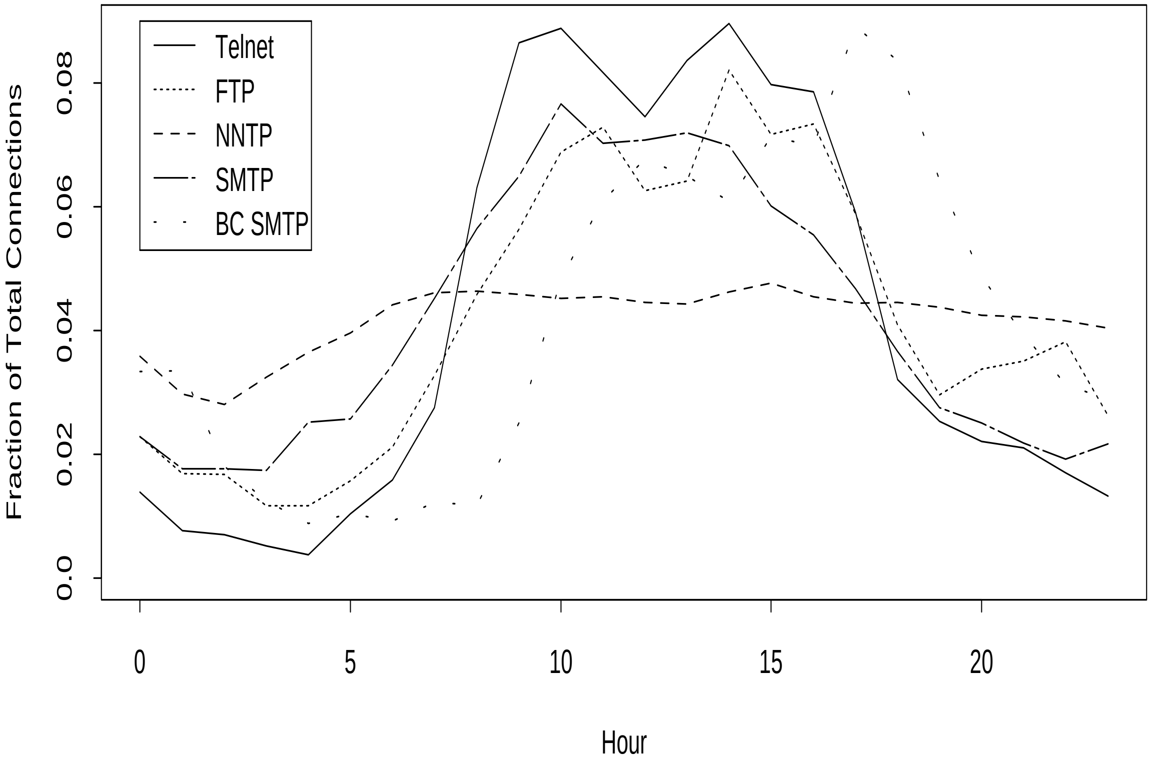

Figure 2: Mean Daily Variation in the Test Dataset Connection Rate

ing each hour by multiplying that hour's fraction as given in Figure 2 by the total number of connections during the superday. Call this quantity a_h, for the number of arrivals during hour h. If we have a_h arrivals from a Poisson process during a single hour, then we expect the mean interarrival time in seconds to be

$$\overline{m}_h = 3600\text{sec}/a_h \qquad (3)$$

and if we divide the interarrival times by $\overline{m}_h$, then they should be exponentially distributed with a mean of 1. Now that each hour's interarrivals have been normalized to the same mean, we test the distribution of all of the superday's normalized arrivals together against that predicted by an exponential model with mean 1.

We can also test a "scaled" version of this model which does not rely on the rates given by Figure 2. Instead of computing $\overline{m}_h$ as given in Equation 3, we simply compute each superhour's interarrival mean directly and divide by that value, guaranteeing a resulting mean of 1.

tcplib does not presently include empirical models for interarrivals, probably because creating such empirical models requires a fair amount of transformation to the raw interarrival times. We therefore do not compare the performance of the analytic interarrival model against that of empirical models, but instead compare the scaled version of the model against the unscaled. If we find that for both versions μ is quite low, then the analytic model is successful and the rates given by Figure 2 are widely applicable. If μ is only low for the scaled model, then the arrivals are indeed from a nonhomogeneous Poisson process, but with rates different from those given in Figure 2. If μ is high for both versions, then the arrivals are not from a Poisson process with a fixed hourly rate. If μ were to be high for the scaled model but low for the unscaled model, then we would be left with a puzzle, but fortunately this never happened.

Note that we do not model the arrival of a site's inbound and outbound connections separately, though the two might well have different hourly rates; nor do we model the correlations between inbound and outbound arrivals. We leave these important refinements to future work.

4 TELNET

We now turn to analyzing the characteristics of individual protocols and developing models to describe them. We begin with *telnet*.[7]

[7] Appendix C presents a similar overview for *rlogin* traffic, along with results of modeling it with the *telnet* models developed in this section.

Dataset	# Conn	# Rej	$\bar{x}_{\text{orig}}$	σ_{orig}	$\max_{\text{orig}}$	$\bar{x}_{\text{resp}}$	σ_{resp}	$\max_{\text{resp}}$	$\bar{x}_{\text{dur}}$	σ_{dur}	$\max_{\text{dur}}$
LBL-1	5,734	9	199B	×4.4	207KB	4.2KB	×7.9	1.9MB	266 s	×6.8	90.5 h
LBL-2	7,582	12	199B	×4.6	282KB	4.3KB	×7.5	3.2MB	237 s	×6.8	78.2 h
LBL-3	9,607	23	214B	×4.7	537KB	4.1KB	×7.6	5.5MB	226 s	×6.9	167.9 h
LBL-4	10,897	58	237B	×4.3	613KB	5.3KB	×7.4	86.6MB	271 s	×6.8	270.0 h
LBL-5	14,922	81	237B	×3.9	215KB	5.2KB	×6.8	19.3MB	248 s	×7.1	386.8 h
LBL-6	17,425	52	147B	×7.3	777KB	3.8KB	×8.7	14.0MB	256 s	×6.9	102.9 h
LBL-6*	15,437	52	242B	×4.5	777KB	5.7KB	×7.3	14.0MB	270 s	×7.7	102.9 h
BC	744	2	145B	×4.1	9.7KB	2.9KB	×8.7	0.6MB	193 s	×6.4	8.1 h
UCB	655	4	155B	×4.7	27KB	2.5KB	×9.1	0.7MB	166 s	×6.9	7.9 h
USC	405	0	184B	×4.3	12KB	4.1KB	×7.2	0.6MB	168 s	×6.5	5.5 h
NC	3,023	34	112B	×3.9	146KB	2.6KB	×10.6	3.4MB	106 s	×7.4	6.8 h
UK	962	35	143B	×3.6	30KB	2.5KB	×9.3	0.7MB	175 s	×5.2	7.2 h

Table 6: Summary of TELNET Connections

4.1 Overview of TELNET Connections

Table 6 summarizes some basic statistics of the datasets' *telnet* connections. The Table is read as follows.

The second column gives the number of "valid" connections recorded for the dataset and the third column the number of "rejected" connections; Appendix B details the rejected connections. As discussed in [Paxson93], the LBL-6 *telnet* traffic included 1,988 connections due to periodic traffic. LBL-6* summarizes the LBL-6 traffic with these connections removed. For the remainder of this section we use LBL-6* instead of LBL-6.

The 4th through 6th columns summarize the number of data bytes transmitted by the originator (the user end of the remote-terminal connection). The values given are the geometric mean, the geometric standard deviation, and the maximum. As noted in Section 3.3, except for interarrival times we applied logarithmic transformations to the data prior to analysis. This transformation is also important for summary statistics such as those presented in this Table, because arithmetic means and standard deviations are quickly dominated by upper-tail outliers; compare the figures given in this paper with those of our previous work [Paxson91]. The latter tend to be much larger.

The 7th through 9th columns give the same summary for the number of bytes transmitted by the responder (remote computer), and the 10th through 12th columns the same for the duration of the connections, with 's' used to indicate seconds and 'h' for hours.

We note that the geometric mean duration of *telnet* connections ranges from 2 to 4 minutes, while Jackson and Stubbs [JS69] reported average connection lengths for local logins of 17 to 34 minutes, and [Bryan67] gives a local-login median of 20 minutes and a mean of 45-50 minutes. Jackson and Stubbs infer that connection time "may be considerably reduced by providing a high-speed channel from the computer to the user", so we might suspect the difference between their measurements and the *telnet* data is due to the higher communication speeds of today's computers. More recently (1985), Marshall and Morgan found that local-area remote logins had an average duration of 45 minutes [MM85], and non-network logins had an average duration of 150 minutes. Thus the distance between the user and the computer appears inversely correlated with the login duration. Since bandwidth usually decreases with distance, we appear to be seeing Jackson and Stubbs' effect but rescaled to reflect today's range of communication speeds.

The LBL *telnet* connections were on average substantially longer and consisted of more bytes than those at other sites. We would expect slightly longer average durations for LBL connections since the datasets span several weeks, giving an opportunity to detect long-lived connections that would be missed by the short spans of the other datasets (except for BC, which spans 13 days and has the next highest average). But this effect is small: if we eliminate from LBL-1 all connections spanning more than one day (i.e., crossing midnight), then $\bar{x}_{\text{orig}}$ drops to 197B, $\bar{x}_{\text{resp}}$ drops by 53B, and $\bar{x}_{\text{dur}}$ drops to 260 s. Given the difference in these parameters even after this adjustment, we are forced to conclude that, at least with regard to mean bytes transferred and duration, the LBL *telnet* traffic is significantly different from that at other sites.

We also note a definite trend over the LBL datasets towards increasing values of $\bar{x}_{\text{orig}}$, and a similar though less convincing trend in $\bar{x}_{\text{resp}}$, too, indicating that *telnet* connections are growing larger with time. Connection durations, on the other hand, are not growing longer, suggesting that higher network bandwidths are enabling users to engage in more work during each session.

Finally, we note that the data provide support for the observation in [DJCME92] that "interactive applications can generate 10 times more data in one direction than the other," and actually suggest the factor is around 20:1. The observation that the computer end of a terminal session generates an order-of-magnitude more data than the user end can be found as far back as reference [JS69], though [Bryan67] found the ratio to be 2.85:1 on a line-by-line basis (the author also states, however, that the studied system was substantially different from a general-purpose, on-line, time-shared system). Marshall and Morgan found ratios as high as 35:1 for teletypewriters in technical use, with half that being a representative average, and as low as 3:1 for teletypewriters used for word processing [MM85].

In Section 4.5 below we present a model for this ratio.

4.2 TELNET Originator Bytes

With the bulk transfer protocols we examine in subsequent sections, we usually are only interested in modeling the number of bytes transferred and the connection interarrival process. With interactive applications, on the other hand, we not only are interested in the bytes transferred in both directions but also the connection duration and the relationships between these variables.

We begin by modeling the number of bytes sent by the originator of a *telnet* connection (typically a human typing at a keyboard). The best fit we found to the LBL *telnet* test datasets came using the *extreme* distribution:

$$F(x) = \exp\left[-\exp\left(-\frac{(x-\alpha)}{\beta}\right)\right] \quad (4)$$

Reference [DS86] gives a procedure for estimating α and β for a given dataset. For our originator-bytes model, x in Equation 4 is $\log_2$ of the number of bytes transmitted by the connection originator.

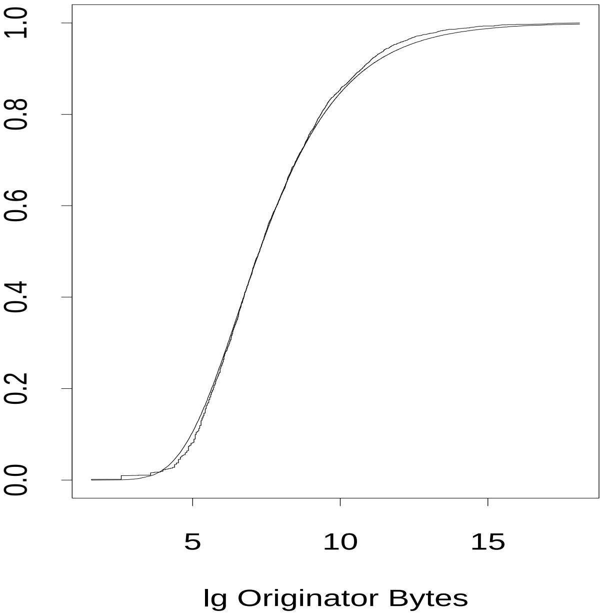

Figure 3: TELNET Originator-Bytes Model for LBL-2: Log-Extreme Distribution

Figure 3 shows the distribution for the first half of the LBL-2 dataset, along with the fitted model. We see apparently good agreement except in the tails, but when tested with A^2 the fit fails to be valid; the same holds for the other LBL half-datasets.

For the four test datasets, α varied from 6.55 to 6.93; we chose $\alpha = \log_2 100 \approx 6.64$. β varied from 1.74 to 1.92. For our fixed model we chose $\beta = \log_2 3.5 \approx 1.81$.

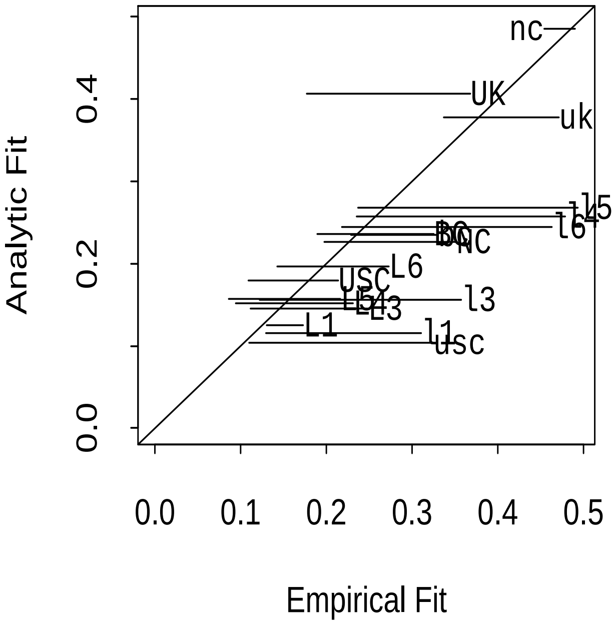

Figure 4: Empirical vs. Analytic Models for TELNET Originator Bytes

Figure 4 shows the computed values of μ for this analytic model plotted against both the UCB and LBL-2 empirical models, where μ is defined as in Equation 1. The X axis gives the value of μ corresponding to one of the empirical models, and the Y axis the value corresponding to the analytic model.

We read the plot as follows. Each point on the plot is labeled with the name of the corresponding dataset. “L1” through “L6” represent the LBL datasets and “D1” through “D3” the DEC datasets (not present in this particular plot). Labels written in lower case (e.g., “usc”) reflect μ values for *unscaled* models; that is, using the raw UCB or LBL-2 data for an empirical model, and the fixed version (no fitted parameters) for the analytic model. Labels in upper case (“USC”) reflect the *scaled* models.

We plot the text label at the point corresponding to comparing the UCB empirical model, on the X axis, with the analytic model, on the Y axis. We then draw a line from that point to the corresponding point comparing the LBL-2 empirical model with the analytic model. This line is always horizontal because the two comparisons share the same μ value, for the analytic model, on the Y axis.

Thus for each dataset four different points are plotted: the unscaled analytic model vs. the UCB empirical model (e.g., “usc”); the scaled version of the same (“USC”); the unscaled analytic model vs. the LBL-2 empirical model (the line drawn from “usc”); and the scaled version of the same (line drawn from “USC”).

For example, the lowest pair of points indicate that the UCB empirical model had $\mu \approx 0.3$; the LBL-2 model, $\mu \approx 0.1$; and the analytic model, $\mu = 0.1$. Since the line drawn from “usc” goes to the left, the LBL-2 empirical model provided a better fit to the unscaled USC dataset than did the UCB empirical model. In general, if the lines head to the left of the labels then the LBL-2 empirical model surpasses the UCB model; and vice versa if the lines go to the right.

The diagonal line indicates where $\mu_{\text{analy}} = \mu_{\text{emp}}$, i.e., where the analytic and empirical models yield the same closeness-of-fit metric. Points below and to the right of this line indicate datasets for which the analytic model fitted better than the empirical model; points above and to the left, where the empirical model fitted better. For example, from this plot we see that the unscaled analytic fit to the LBL-5 dataset was much better than that of the UCB empirical model (“l5”) but about the same as that of the LBL-2 empirical model.

We see in this plot that the LBL-2 empirical model almost always does better than the UCB empirical model, and that the analytic model performs comparably. The points tend to lie either just above the diagonal, indicating a slightly better empirical fit, or a bit further away from and below the diagonal, indicating a better analytic fit. For this model, scaling sometimes results in a big improvement (NC, LBL-4), no improvement (BC), or an improved empirical model but worsened analytic model (USC, UK). Thus in this case it makes sense to scale the empirical models when predicting traffic, but not the analytic model.

Closer observation reveals that for every dataset except UK and NC (two extreme cases), the analytic model fits the dataset better than the UCB model, while the LBL-2 model fits best in every case except for a few points very close to the line of equality. Thus we can order the models: the LBL-2 empirical model is better than the analytic model, which in turn is better than the UCB model.

The overall fit of the model to the datasets does not tell the entire story, however. As is generally the case with bytes-transferred models, for *telnet* originator bytes the models’ fits to the upper tail are much more important than fits to the lower tail. Figure 5 summarizes the upper-tail fits. The plot is labeled with “a” for the unscaled analytic model, “u” for the unscaled empirical UCB model, and “l” for the unscaled empirical LBL-2 model. The upper-case versions of these letters correspond to the tails for the scaled versions of these models. The X axis gives the ξ value for the upper 10% tail, and the Y axis the value for the upper 1% tail, where ξ is computed as given in Equation 2. In this plot we see that scaling had little effect on fitting the upper tails, as all of the uppercase letters are near their lowercase counterparts.

A letter close to the origin, such as “u”, indicates excellent modeling in both the upper 10% tail and the upper 1% tail. That all of the models are clustered around the Y-axis indicates they all model the upper 10% tail well. But we see that both the analytic model and the LBL-2 model have high values of ξ for the upper 1% tail. As explained in Section 3.10 above, this indicates that those models *overestimate* the distribution in the 1% tail. That is, they tend to predict more values in the 1% tail than were actually present in the datasets. As the axes are scaled logarithmically, the deviations shown are quite large. Indeed, the unscaled analytic model overestimates the upper 1% tail for every single dataset, and for all except the LBL and NC datasets not a single observation actually resided in the predicted tail. Thus a value of $\xi \approx 2$ corresponds to exceptionally poor tail fitting.

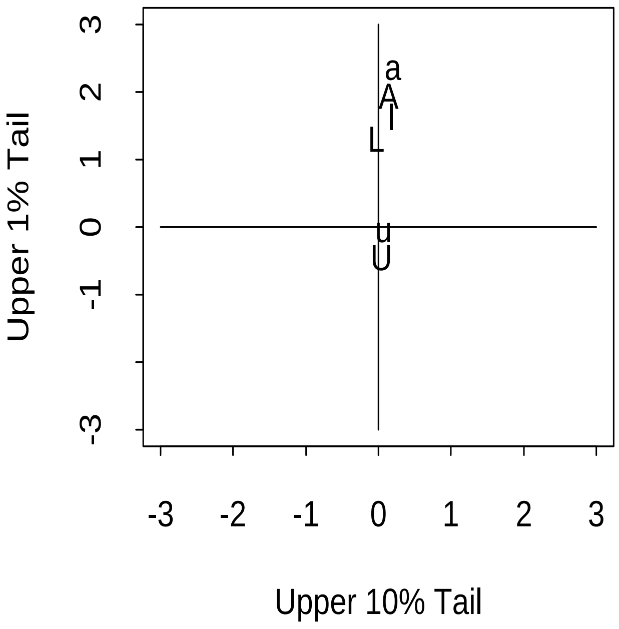

Figure 5: Tail Summary for TELNET Originator Bytes

While the UCB empirical model does poorly versus the other models in fitting the datasets over the entire distribution of originator bytes, it is the obvious champion when it comes to fitting the upper 1% tail. Thus predicting *telnet* originator bytes leaves us in a quandary: we must decide which is more important to us, the overall fit to the distribution, in which case LBL-2 or the analytic model is recommended, or the upper 1% tail, in which case UCB is recommended. If fitting just the upper 10% tail well is adequate, then either the analytic model or LBL-2 is recommended.

In the interest of conserving space, for the remaining models we relegate their outlier summaries to Appendix D.

4.3 TELNET Responder Bytes

We next turn to modeling the bytes transferred by the *telnet* responder. Figure 6 shows a log-normal fit to the upper 80%

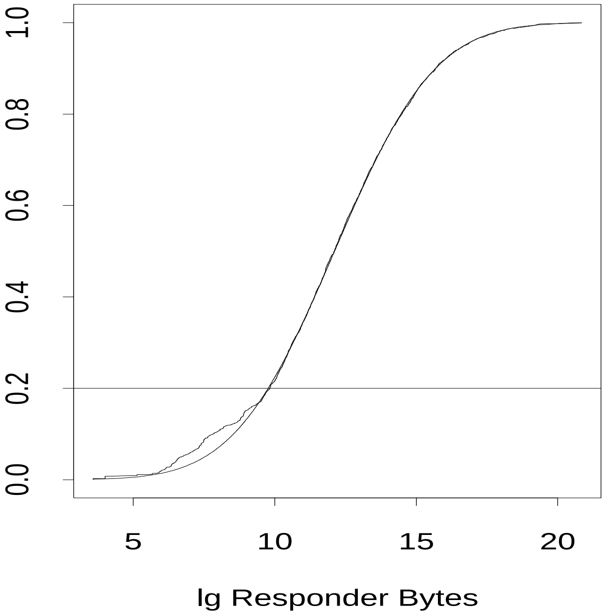

Figure 6: TELNET Responder-Bytes Model for LBL-2: Log-Normal Fit to Upper 80%

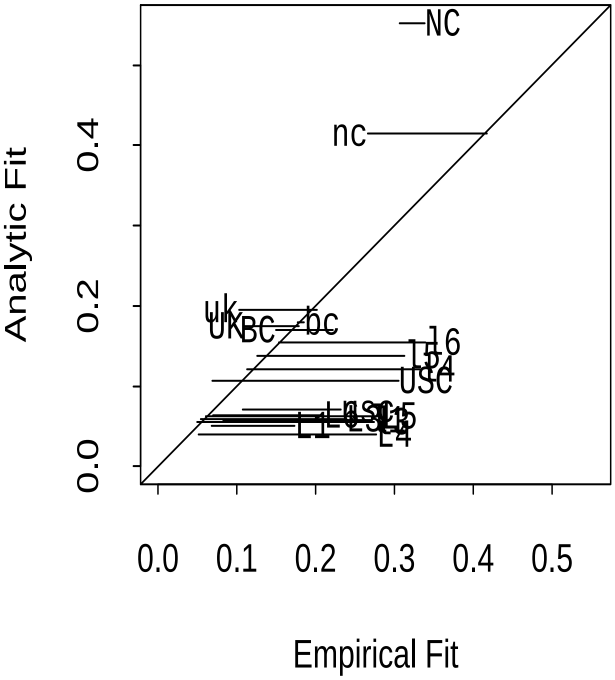

Figure 7: Empirical vs. Analytic Models for TELNET Responder Bytes

of the responder bytes in the LBL-1 test dataset. This fit is excellent; it passes the A^2 test at the 25% significance level (compare with Figure 1, which shows the same fit for the LBL-4 dataset and fails A^2 even at 1% significance). We see, however, that the lower 20% (below the horizontal line, corresponding to less than 1 KB transferred) is not smoothly distributed, making it unlikely we might find a simple analytic model encompassing it. We speculate that this roughness is due to the varying sizes of log-in dialogs and message-of-the-day greetings. Fortunately the lower tail is the least important part of this distribution.

We found in the test datasets that the log-mean ($\bar{x}$) varied from 12.0 to 12.4, generally closer to 12.0, and we chose for our fixed model $\bar{x} = \log_2 4500 \approx 12.1$. σ_x varied from 2.79 to 2.89; we chose $\sigma_x = \log_2 7.2 \approx 2.85$.

For this one model we evaluated the metric μ using 9 bins, from 0.2 to 1.0, instead of 10 bins (0.1 to 1.0), because the analytic model only fits the upper 80% of the data and it did not seem worthwhile to develop a separate model for the lower 20%. Figure 7 summarizes the fits. Except for NC, the analytic model uniformly performs well, with μ always ≤ 0.2. The LBL-2 model also fares quite well, while the UCB model is not as good except for UK and BC. Scaling these models does not always improve things (USC in particular) but in general helps. Figure 8 explains the terrible performance fitting NC: the distribution suffers from two large clusterings, one between 240 and 265 bytes, and the other between 400 and 425 bytes. The first consumes 13% of all the connections, the second 5%. A single host originated virtually all of the connections in the first cluster, but to a number of different

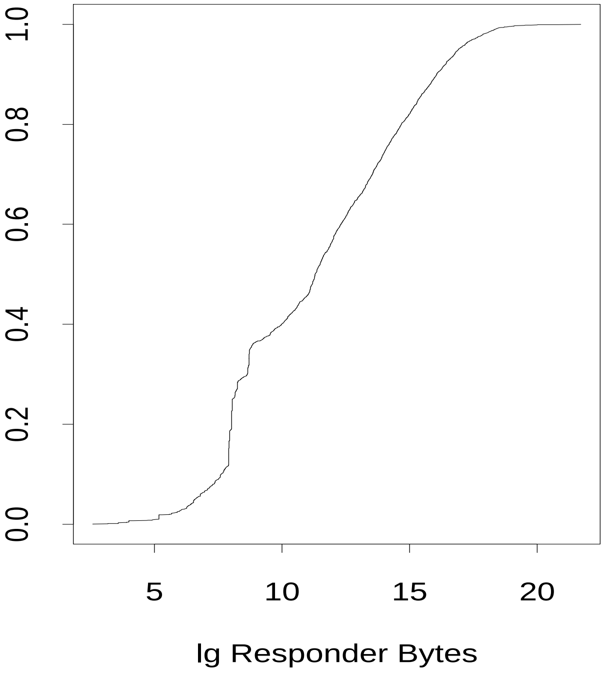

Figure 8: Distribution of NC TELNET Responder Bytes

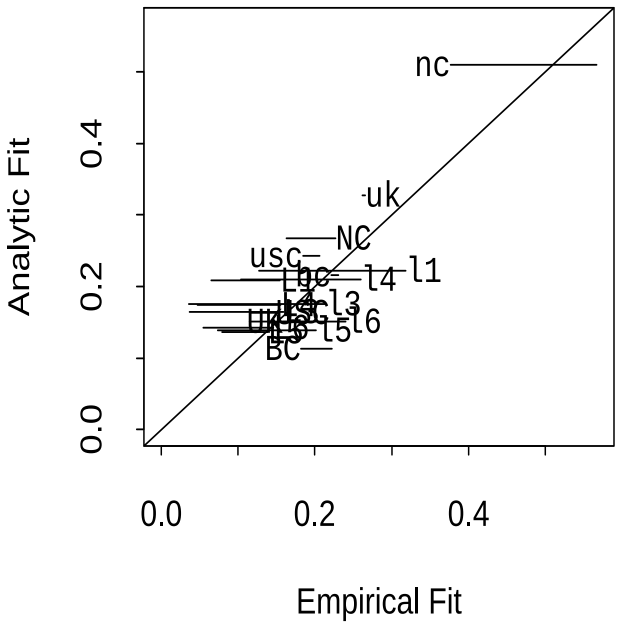

Figure 9: Empirical vs. Analytic Models for TELNET Duration

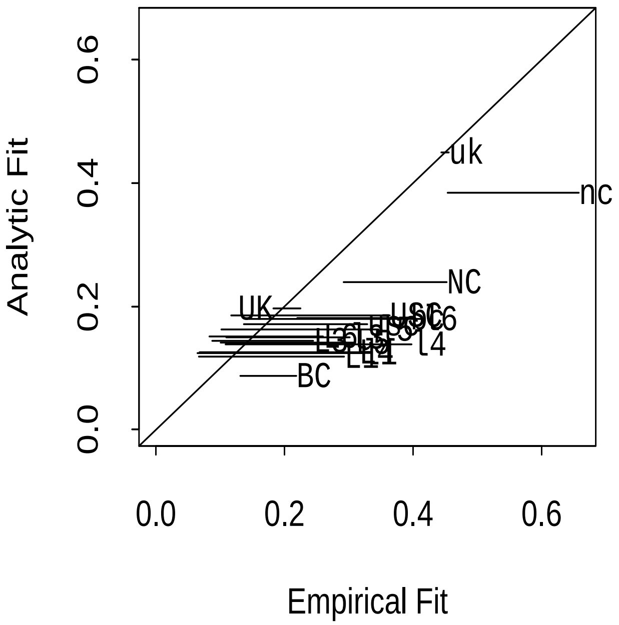

Figure 10: Empirical vs. Analytic Models for TELNET Resp./Orig. Ratio

hosts, and two other hosts originated almost all of the connections in the second cluster, primarily to two remote hosts. We were unable to find obvious patterns in the interarrivals (see Figure 18 below for an example of clear one-minute patterns in connection arrivals); therefore, unlike many of the spikes discussed in [Paxson93], the connections were probably not generated by background scripts. Perhaps they correspond to cracking attempts, or more benign searches. Overall they remain puzzling.

Figure 28 in Appendix D shows the performance of the models with regard to the upper tails. Each model except for unscaled UCB does well in the upper 10% tail. All of the models overestimate the upper 1% tail somewhat; the unscaled UCB model surprisingly doing the best. On the basis of these plots we would prefer the empirical models if the upper 1% tail is important to us; otherwise either the analytic model or LBL-2 is preferable. LBL-2 provides the best overall model.

4.4 TELNET Duration

We model *telnet* connection durations using a simple log-normal distribution. For the test datasets we found $\bar{x}$ ranging from 7.67 to 8.03 and chose $\bar{x} = \log_2 240 \approx 7.91$. σ_x ranged from 2.83 to 3.02; we chose $\sigma_x = \log_2 7.8 \approx 2.96$.

Figure 9 shows the fits for the duration models. In general the models are fairly good, with the metric μ falling between 0.1 and 0.3. NC again proves troublesome, though not so when scaled. No model emerges a clear winner, and, while the analytic model appears to do worst, it is not considerably worse than the other two.

Figure 28 in Appendix D summarizes the tail performance of the models. In general the models do well in the upper 10% tail, though the unscaled analytic and LBL-2 models overestimate somewhat. In the upper 1% tail these same models do quite poorly, while the UCB models are excellent in both tails. Because the UCB model did well in the general fitting shown in Figure 9, its good performance here makes it the model of choice for *telnet* duration.

4.5 TELNET Responder/Originator Ratio

If we wish to use these models to generate or predict *telnet* traffic, then we also need models giving the relationships between the various distributions. In particular, we would like to know how many responder bytes to expect given a particular number of originator bytes, and how long a connection will last given how many bytes it transfers.

We model the ratio between the number of responder bytes and originator bytes using a simple log-normal distribution. For the test datasets we found $\bar{x}$ ranged from 4.17 to 4.46, tending toward the high end, and σ_x from 1.77 to 1.89, also tending to the larger value. For the fixed model we chose $\bar{x} = \log_2 21 \approx 4.39$ and $\sigma_x = \log_2 3.6 \approx 1.85$.

Figure 10 shows the performance of each model. Other than the unscaled UK and NC datasets, the analytic model does quite well, with $\mu \leq 0.2$ except for the scaled NC, with $\mu = 0.25$. In general the LBL-2 empirical model does a little better than the analytic model, and almost always better than UCB. Scaling improves some fits considerably and has only marginal effect on others. The overall success of the unscaled analytic model gives solid evidence that the ratio between the bytes generated by the computer in a remote login session and those generated by the user is about 20:1, since the fixed model uses a ratio of 21:1.

For the responder/originator ratio we are interested in agreement in both the upper and lower tails, as disagreement in either could result in skewed predictions when the number of originator bytes is large. Figure 28 in Appendix D shows the performance for the upper and lower tails. All of the models do fairly well for the upper tails except for the unscaled UCB model, which underestimates both upper tails. The analytic model does best. With the LBL-2 model, scaling trades off better performance in the 10% tail for worse in the 1% tail.

In the lower tail for both the analytic and LBL-2 models scaling helps the 10% tail but worsens the 1% tail, indicating that the 1% tail is distributed differently than the other 99%. The UCB model does well in the lower tails, though. All in all we are left with no clear best model, and none of the models is really bad.

One might wonder whether the responder/originator ratio's distribution itself varies according to the number of bytes transferred; for example, perhaps when many originator bytes are transferred, the ratio tends to be low, because relatively speaking not so many responder bytes are transferred. For the test datasets we found that the correlation coefficient between $\log_2$ of the originator bytes and $\log_2$ of the responder/originator ratio varied from 0.07 to 0.10, indicating at most a mild positive correlation.

When using the responder/originator ratio to generate *telnet* traffic, a subtle point arises: one can either derive the originator bytes and the ratio, and multiply to obtain the responder bytes, or one can proceed in the opposite fashion, generating the responder bytes and the ratio, and dividing to obtain the originator bytes. While these two approaches appear equivalent, they are not, and the former (deriving the responder bytes from the originator) is preferable. The difference arises because while both the responder bytes and the ratio are log-normal distributed, the originator bytes are *extreme* distributed. Multiplying the originator byte's extreme distribution by the ratio's log-normal distribution yields a distribution close to log-normal; but dividing the responder byte's log-normal distribution by the ratio's log-normal distribution yields *exactly* a log-normal distribution (since the difference of two normal distributions is a normal distribution), and not an extreme distribution. Alternatively, we can think of the originator bytes as having a somewhat skewed log-normal distribution. Multiplying this distribution by another log-normal distribution smears out the deviations, and the result is close to log-normal; but chances are dividing two log-normal distributions will never reproduce the skewed distribution.

Thus, to generate traffic we should begin by generating the number of originator bytes and the responder/originator ratio, and then multiply to derive the responder bytes. This approach is not ideal, however, because it ignores the responder-bytes model we outlined above, which is more successful than the originator-bytes model.

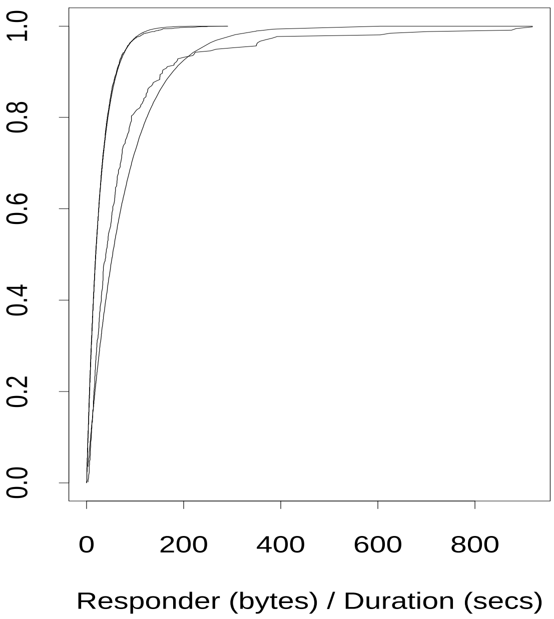

Figure 11: Responder/Duration Distributions for LBL-1: Exponential Fits

4.6 TELNET Responder/Duration Ratio

Just as we want a way to relate the originator bytes sent with the responder bytes, we also would like to relate these random variables to the connection duration. We investigated analytic models for three different ratios: originator bytes to duration, responder bytes to duration, and total bytes to duration. We found the best fits came using the responder/duration model.

For most connections the responder/duration ratio was well modeled by an exponential distribution, but "large" connections—those whose responder bytes were in the upper 10% of all connections—had a different distribution. For these, the ratio was fairly well modeled by a log-normal distribution.

Figure 11 shows the responder/duration ratio for both the lower 90% of the LBL-1 connections (in terms of responder bytes) and the upper 10%. The distribution on the left is for the lower 90%; though it is hard to tell due to scaling, an exponential with the same mean has been drawn and lies squarely on top of it. This fit is very good; it passes A^2 at the 5% level. To the right we show the distribution of the upper 10%, plotted with an exponential with the same mean. We see that the distribution is qualitatively different, and the corresponding exponential not a good fit.

We find the bimodality shown in this figure a bit puzzling. It says that very large connections (in terms of bytes trans-

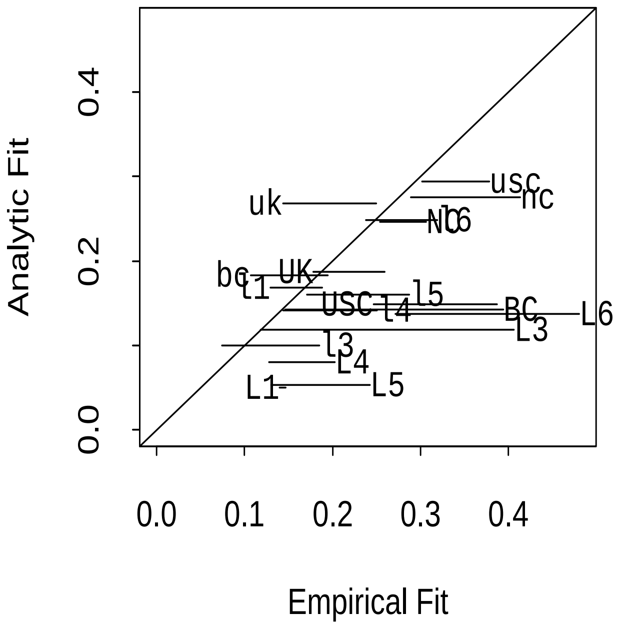

Figure 12: Empirical vs. Analytic Models for TELNET Resp./Duration Ratio

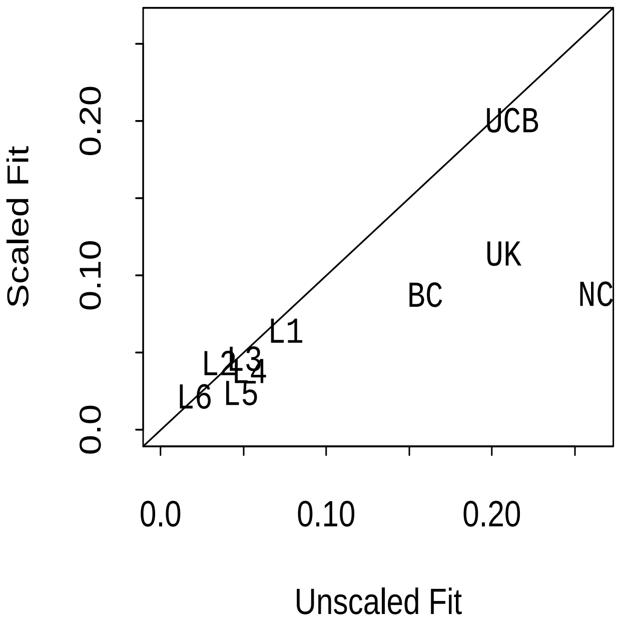

Figure 13: Interarrivals for TELNET

ferred) occur over relatively short durations: while the geometric mean of the responder bytes in these large connections is 45 times that of the smaller (lower 90%) connections, the geometric mean of their durations is only 16 times that of the smaller connections. This phenomenon was also observed by the authors of [SC92], who found that "users transmitting large amounts of data over a link tend to transmit that data within 15 minutes." We do not have a good explanation for this phenomenon.

For the lower-90% model, the test datasets gave $\bar{x}$ ranging from 27 to 33 for the responder/duration ratio; we chose $\bar{x} = 30$. For the upper-10% model, $\bar{x}$ ranged from 5.19 to 5.41 and σ_x from 1.38 to 1.61; we chose $\bar{x} = 5.3$ and $\sigma_x = 1.5$.

Figure 12 shows the fit of the models for the lower 90% of the responders. The analytic fit is good, with $\mu \leq 0.3$ and often $\mu \leq 0.2$; in general it fits better than either empirical model.

For the upper 10% of the responders we compared considerably fewer datasets. Our requirement that each dataset include at least 100 measurements ruled out any dataset with fewer than 1,000 *telnet* connections, leaving just the LBL and NC datasets. The fit remains good, though: the analytic model does well, with $\mu \leq 0.3$ except for the unscaled NC dataset (where $\mu \approx 0.6$ for all three models), quite a bit better than the UCB model and about equal to the LBL-2 model.

Figure 28 in Appendix D summarizes the upper and lower tail distributions for the fit to the lower 90% of the responders. In the upper tails the analytic model does best, only mildly underestimating the upper 1% tail; only the scaled LBL-2 model is roughly comparable. In the lower tails the scaled analytic model does very well, with the unscaled version overestimating the 1% tail somewhat. Again the empirical models do considerably worse and the scaled UCB model is completely inadequate, though the unscaled model is acceptable.

For the models of the upper 10% of the responders, every model underestimates the upper 1% tail somewhat, with the analytic models and the scaled LBL-2 model about the same at $\xi \approx -0.5$. The unscaled empirical models fare poorly in the 10% tail, too, considerably underestimating it, while analytic models and the scaled empirical models match the 10% tail well. In the lower tails the unscaled models do fairly well with the 10% tail, and the scaled models do quite well. Except for the scaled UCB and analytic models, though, the lower 1% tail is considerably underestimated.

4.7 TELNET Interarrivals

We now turn to modeling *telnet* interarrivals, using the methodology discussed in Section 3.11 above. Figure 13 compares the μ values for the unscaled and scaled arrival models. As explained in Section 3.11, instead of comparing the analytic model to the empirical models, we compare the analytic model's scaled version with its unscaled version. We plot μ for the scaled analytic model on the Y axis vs. μ for the unscaled model on the X axis. Also, as mentioned in Section 2.2, we omit the USC dataset from our interarrival models because of the trace's periodic blackouts.

As expected, the scaled model in general does uniformly better, but we note that even for the unscaled model, $\mu \leq 0.25$, which, when compared to the fits of other models above, we see is quite good. The arrivals are thus well-

Dataset	# Conn	# Rej	% Failures	$\bar{x}_{\text{orig}}$	σ_{orig}	$\max_{\text{orig}}$	$\bar{x}_{\text{resp}}$	σ_{resp}	$\max_{\text{resp}}$
LBL-1	57,898	2	38 %	2.0KB	×9.2	4.2MB	305B	×2.0	923KB
LBL-2	57,997	1	36 %	2.4KB	×7.8	1.1MB	328B	×2.1	584KB
LBL-3	46,167	6	19 %	2.4KB	×6.2	1.9MB	384B	×1.9	128KB
LBL-4	73,179	39	2 %	6.0KB	×8.5	5.6MB	398B	×2.2	1.4MB
LBL-5	50,969	161	8 %	14.5KB	×8.5	16.5MB	633B	×2.9	9.5MB
LBL-6	55,176	1048	8 %	28.4KB	×6.8	15.7MB	888B	×2.2	1.3MB
BC	345	116	25 %	15.5KB	×6.2	2.4MB	1005B	×3.0	81KB
UCB	6,899	0	1 %	2.1KB	×7.2	720KB	307B	×2.0	1.7MB
USC	4,615	15	4 %	11.5KB	×10.3	3.6MB	709B	×2.3	74KB
DEC-1	23,864	5	2 %	1.1KB	×11.6	5.8MB	264B	×2.2	75KB
DEC-2	18,819	88	3 %	1.3KB	×11.7	26MB	292B	×2.4	356KB
DEC-3	19,244	7	7 %	2.2KB	×14.1	18MB	339B	×2.7	223KB
NC	904	206	9 %	12.9KB	×12.3	12MB	1182B	×4.5	3.2MB

Table 7: Summary of NNTP Connections

modeled as a non-homogeneous Poisson process with hourly rates given by Figure 2. This finding is at odds with that of [MM85], who found that "user interarrival times look roughly lognormal". Perhaps the discrepancy is due to the authors characterizing all interarrivals lumped together, rather than postulating separate hourly rates.

Figure 30 in Appendix D summarizes the tail distributions for the scaled and unscaled arrival models. (See the text in Appendix D for an explanation of the symbols in the figure.) Note the range shown in the figure: even the worst fits have $|\xi| \leq 0.25$. Thus both the unscaled and scaled models do quite well, and the scaled model does exceptionally well.

5 NNTP

5.1 Overview of NNTP Connections

Table 7 summarizes *nntp* connections. As *nntp* is non-interactive, the connection duration is not of much interest and has been omitted. Appendix B discusses the connections we rejected due to protocol errors.

We expect *nntp* connections to show considerable variation because they can come in at least three modes: a server contacts a peer and is informed that the peer presently cannot talk to the server; the server offers the peer news articles but the peer already has the articles; the server offers articles and the peer does not have the articles. Each of these modes will result in significantly different distributions of the bytes transferred during the connection. Furthermore, the second and third modes are somewhat indistinct, since the remote peer may have some but not all of the offered articles.

The first mode is easy to detect. If upon initially being contacted a responder peer is unable to communicate with the originating peer, it sends a message with response code 400 ("service discontinued") as per [RFC977]. When the originating peer then replies with "QUIT" followed by a carriage-return and a line-feed, it will send a total of 6 bytes during the connection. Indeed, we find large spikes of 6 originator bytes in the *nntp* datasets, as did the authors of [DJCME92]. Thus we can recognize a connection in which the originating host sent 6 bytes as a "failure".

Not surprisingly, the failure rate varies greatly from site to site and from time to time, since it is often due to transient phenomena such as full disks. These failure rates are given in the "% Failures" column. Note that even over a period of 7 days, the DEC failure rate moved from 2% to 7%. To compute the remaining statistics in the Table, we first removed all failure connections from the datasets.

Not only can the failure rate vary significantly, but so can the bytes transferred during non-failure connections. For example, as can be seen by the large increase in $\bar{x}_{\text{orig}}$ between LBL-3 and LBL-4, the LBL *nntp* server became much more effective in propagating news over a five month period. LBL-5 and LBL-6 continue the impressive growth in $\bar{x}_{\text{orig}}$. A similar effect can be seen between DEC-1 and DEC-3, only a week apart. Such changes can be due in part to circumstances wholly outside of the local site. Whether the articles a server attempts to propagate to its peers are accepted depends on whether those peers already have the articles; a subtle change in the *nntp* peer topology can swing a server's position from one of holding mostly "stale" news to holding mostly "fresh" news. The steadily increasing $\bar{x}_{\text{orig}}$ value for the last four LBL datasets, though, is most likely simply a reflection of the global growth in USENET *nntp* traffic, which increases in volume about 75%/year (see [Paxson93]).

5.2 NNTP Originator Bytes

Figure 14 shows the distributions of bytes sent by the originator in non-failure *nntp* connections at LBL, DEC, and coNCert. The distributions show a large degree of variance