# Empirically Derived Analytic Models of Wide-Area TCP Connections

Vern Paxson

*Abstract*—**We analyze 3 million TCP connections that occurred during 15 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. To assess these models we present a quantitative methodology for comparing their effectiveness with that of empirical models such as Tcplib [7]. Our methodology also allows us to determine which random variables show significant variation from site to site, over time, or between stub networks and internetwork gateways. Overall we find that the analytic models provide good descriptions, and generally model the various distributions as well as empirical models.**

## I. Introduction

IN THE LAST few years a number of papers have appeared giving statistical summaries of wide-area traffic on a per-protocol basis [2], [11], [4], [31], [6], an important first step to characterizing WAN traffic. The next step in understanding wide-area traffic is to form models for simulating and predicting traffic.

One such model, Tcplib [7], [8], 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. Two key questions are whether analytic models can describe the diverse phenomena found in wide-area traffic as well as empirical models, and whether either type of model faithfully captures the essential characteristics of the traffic.

In this paper we analyze 15 wide-area traffic traces gathered at seven different sites, five "stub" (end-point) 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. Our statistical methodology also allows us to determine which random variables show significant variation from site to site, over time, or between stub networks and internetwork gateways. Table I 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 the findings in the remainder of the paper.

Below, Section II presents an overview of the traces used in the study, and Section III gives a discussion of our statistical methodology. Section IV summarizes the models we developed and evaluates their effectiveness. Readers interested in particular protocols will find more detailed summaries and discussions in Sections V–VIII.

A longer, preliminary version of this paper is also available [24]. In the remainder of this paper we note where, in the interests of brevity, we have relegated details to that report instead.

## II. 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 seven 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.

### A. 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. We recorded our seven LBL traces using the *tcpdump* packet capture tool [13] running the Berkeley Packet Filter [17]. We used a *tcpdump* filter to capture only those TCP packets with SYN or FIN 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.

Manuscript received June 26, 1993; revised May 18, 1994; approved by the IEEE/ACM Transactions on Networking Editor J. Smith. This work was supported by the Director, Office of Energy Research, Scientific Computing Staff, of the U.S. Department of Energy under Contract DE-AC03-76SF00098.

The author is with the Lawrence Berkeley Laboratory and the EECS Division, University of California, Berkeley, CA 94720 USA; email: vern@ee.lbl.gov.

IEEE Log Number 9404832.

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

1063–6692/94$04.00 © 1994 IEEE

TABLE I
MAJOR FINDINGS

| |
|---|
| 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. |
| Network traffic varies significantly, both over time and more so from site-to-site, not only in traffic mix but in connection characteristics. We believe this variation is the basis for the success of the analytic models; there is enough variation that any model, empirical or analytic, must be a somewhat rough compromise. |
| The number of data bytes in bulk-transfer traffic (*FTPdata*, *SMTP*, and *NNTP*) 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. |
| The ratio between bytes sent by the computer-side of a *TELNET* connection and bytes sent by the user is about 20:1. |
| Of *FTP* sessions that are not "failures" (no data transferred), half transfer more than 32 KB, and a sixth transfer more than 500 KB. |
| The upper tail of *FTP* "bursts" (Section VIII—D) is so large that 2% of the bursts account for 50-80% of *all* the *FTP* data bytes. |

TABLE II
SUMMARY OF LBL DATASETS

| Dataset | Packets (days) | Start | End |
|---|---|---|---|
| LBL-1 | 124M (36) | 01Nov90 | 01Dec90 |
| LBL-2 | ? | 28Feb91 | 30Mar91 |
| LBL-3 | 207M (47) | 07Nov91 | 07Dec91 |
| LBL-4 | 210M (36) | 19Mar92 | 18Apr92 |
| LBL-5 | 337M (35) | 24Sep92 | 23Oct92 |
| LBL-6 | 447M (31) | 24Feb93 | 26Mar93 |
| LBL-7 | 560M (32) | 16Sep93 | 15Oct93 |

TABLE III
SUMMARY OF ADDITIONAL DATASETS

| Site | Starting Time | Duration |
|---|---|---|
| Bellcore (BC) | Tue 14:37 10Oct89 | 13 days |
| UCB (UCB) | Tue 10:30 31Oct89 | 24 hours |
| USC (USC) | Tue 14:24 22Jan91 | 26 hours |
| 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 |

Table II 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.[2] Each dataset was then trimmed to span exactly 30 days. Very few packets were dropped by the tracing program (always < 15 per million).

Since the LBL datasets span three 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 [22].

### *B. The Additional Traces*

As mentioned above, a number of colleagues generously provided access to traffic traces from other sites. The authors of [8] 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 [31] provided their traces of the UK–US academic network. The first four traces all originate from "stub" (endpoint) sites, while the latter two represent inter-network traffic (though the authors of [31] characterize the UK side of the UK—US traffic as similar to a large stub site since it comprises only a few hosts).

The additional datasets are summarized in Table III. Next to the site name we give in parentheses the abbreviation we will use to identify the dataset. In general the traces had no packet drops or an unknown number of drops (see [24] for specifics).

### *C. Filtering of Non-WAN Traffic*

Before proceeding with our analysis we filtered out nonwide-area traffic from the datasets: internal and transit traffic. The details are given in [24]. In addition, we removed from the LBL datasets all traffic between LBL and U.C. Berkeley.[3] 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.

### *D. 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 [10], the related [12], and [8] compare traffic from more than one institution. The first two papers found significant differences between their four traffic sites, which they attributed to the fact that the different sites engaged in different applications and had different hardware. The authors of [8] found that their three sites (which correspond to the USC and UCB datasets in 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.

Table IV 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 [8] because the latter reports *conversation* mixes, where multiple related connections have been combined into single conversations.[4]

[2] The statistics missing for the LBL-2 dataset are due to abnormal termination of the tracing program; this termination, however, did not imply any extra-ordinary loss of packets during the 30-day study period.

[3] Including *NNTP*, unlike [22], which keeps the *NNTP* traffic.

[4] The authors also used twenty-minute silences to delimit the end of connections, instead of FIN packets.

TABLE IV
PERCENTAGE CONNECTION MIXES FOR ALL DATASETS

| 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-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 |
| LBL-7 | 606,487 | 11 | 34 | 18 | 4 | 15 | 0.9 | 4 | 5 | 0.9 | 0.4 | 6 |
| 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 V
PERCENTAGE BYTE 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 |
| LBL-7 | 30,910 | 25 | 3 | 51 | 0.1 | 2 | 0.7 | 0.0 | 0.4 | 8 | 8 | 1.8 |
| 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 |

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). Some of the variation in the mix is due to periodic traffic. For example, the large spike in the LBL-4 *FINGER* connections, the large jump in *OTHER* connections at LBL-3, the increasing proportion of *FTPctrl* traffic (i.e., the interactive, control side of an *FTP* session), and the large number of *TELNET* connections in LBL-7, are all due to periodic traffic. [22] explores this phenomenon further.

Another factor affecting traffic mix over time (as seen in the LBL datasets) is the large variance of the *NNTP* mix, which is due to changes in LBL's *NNTP* peer servers and differences in the rate at which new network news arrives. Again, see [22] for a discussion.

Regarding the DEC datasets, DEC has a "firewall" in place which prohibits traffic other than *NNTP*, *SMTP*, *FTP*, and *DOMAIN*. The little remaining traffic due to other protocols originated on the outside of the firewall. Finally, the DEC-2 dataset includes part of the Thanksgiving holiday, accounting for the depressed number of connections.

Table V 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, explored further in [22].

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 IV).

We now turn to the development of the statistical methodology that we will use to characterize the individual connections making up the data shown in Tables IV and V.

## III. STATISTICAL METHODOLOGY

As noted in [21], one weakness of many network 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 [10] developed their own, apparently never-published goodness-of-fit measure; and in our own previous work [25] 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 shortcomings and to present a general statistical methodology that might serve future work as well.

### A. *Definitions and Conventions*

We will use a number of terms and concepts taken from statistics. In this section we define the terms and conventions used in the remainder of the paper.

*1) Random Variables:* For our purposes, we define a *random variable* as a quantity that each time it is measured takes on one of a range of values. Particular values occur with different probabilities. We refer to each separate measurement as an *instance* of the random variable.

An example of a random variable is the number of bytes transferred during an *FTP* session. Another (and closely related) random variable is the logarithm of this value.

By convention, $X$ represents a generic random variable and $x_i$ the $i$th instance of $X$. Unless otherwise stated, we assume there are a total of $n$ instances.

*2) Models:* We define a *model* of a random variable as a hypothesized distribution for what values the random variable might take, and with what probability. This definition of model is quite simple, as it assumes that instances of the random variable are independent and identically distributed. Often *correlations* between instances of a random variable are very important, meaning that instances of the random variable might be identically distributed, but not *independent*. For example, previous work has found that large *FTPdata* bursts tend to arrive in clusters [23]. While incorporating correlations into models can be very important, doing so is beyond the scope of our study, so we limit ourselves to simple correlation summaries [Section III-D)].

Another important aspect of modeling network connections is modeling the connection *arrival process*. Here we limit ourselves to briefly describing arrival phenomenon such as periodicity. See [23] for a more detailed look at the connection arrival processes.

In this paper we distinguish between *empirical* models and *analytic* models. An empirical model, such as Tcplib [7], [8], describes a random variable's distribution based on the observed distribution of an earlier sample of the variable. For example, the Tcplib models were constructed from the UCB dataset. An analytic model, on the other hand, attempts to capture a distribution in a simple mathematical form. We discuss the advantages of each type of model in Section III-D) below.

For each of the random variables modeled in this paper, we examine three models, one analytic and two empirical. The empirical models were constructed from the UCB and LBL-2 datasets. We refer to these three models as $A$, $U$, and $L$, respectively. As explained above, the $U$ model reflects the behavior of Tcplib.

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 analytic 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, except for UCB and LBL-2, since we used these to construct the $U$ and $L$ empirical models.

*3) Distributions:* We define the *distribution* of a random variable as:

$$F(x) = P(X \leq x).$$

That is, $F(x)$ is the probability that an instance of the random variable $X$ takes on a value less than or equal to $x$.

For our analytic models we draw upon a number of distributions commonly used in statistics. We assume that the reader is familiar with the normal and exponential distributions. Two other distributions we will use are the *extreme* distribution:

$$F(x) = \exp\left[-\exp\left(-\frac{(x-\alpha)}{\beta}\right)\right] \tag{1}$$

and the doubly-exponential *Pareto* distribution:

$$F(x) = 1 - (k/x)^{\alpha}. \tag{2}$$

The Pareto distribution is noteworthy for having a very heavy upper tail, an important property when considering *self-similarity* in network traffic [14].

We will be using variants of the normal and extreme distributions called *log-normal* and *log-extreme*; these are discussed in Section III-B.

*4) Estimating Parameters:* All of the distributions mentioned in the previous section are *parameterized* using one or more constants. A normal distribution is parameterized by a mean, $\bar{x}$, and a standard deviation, $\sigma_x$; an exponential distribution, by a rate $\lambda$; an extreme distribution, by $\alpha$ and $\beta$; and a Pareto distribution, by $\alpha$ and $k$ ($k$ is the lower bound of $X$).

Sometimes when using an analytic model, the parameterization constants are known in advance. Other times, they must be *estimated* from the same data that we are trying to describe using the analytic model. For a normal distribution, the mean is estimated using:

$$\hat{\bar{x}} = \sum_{i=1}^{n} x_i/n \tag{3}$$

and the standard deviation using:

$$\hat{\sigma}_x = \sqrt{\sum_{i=1}^{n} \frac{(x_i - \hat{\bar{x}})^2}{(n-1)}}. \tag{4}$$

We often use the $\hat{\bar{x}}$ and $\hat{\sigma}_x$ estimates to evaluate a distribution even if it is not normal.

For an exponential distribution,

$$\hat{\lambda} = 1/\hat{\bar{x}}.$$

For an extreme distribution, $\alpha$ and $\beta$ can be estimated using an iterative method [5], and $\alpha$ and $k$ for a Pareto distribution using a simple least-squares technique [3].

When estimating parameters, we will generally drop the "hat" notation (e.g., use $\bar{x}$ instead of $\hat{\bar{x}}$). An exception is when discussing the $\lambda^2$ discrepancy measure (Section III–E), where the fact that we are only estimating its value becomes an important consideration.

Finally, sometimes when estimating parameters we want to ignore part of the distribution we are using to compute the estimation. For example, below we model the number of bytes sent by a *TELNET* responder as a log-normal distribution, but the fit between the model and the dataset is only good for the upper 80% of the distribution [see Section V–C)]. If we compute $\hat{\bar{x}}$ from the entire dataset's distribution, we will spoil the upper-80% fit because of the disagreement between the model and the distribution in the lower 20%. Instead, we *censor* the distribution by ignoring its lower 20%, and then estimate $\hat{\bar{x}}$ and $\hat{\sigma}_x$ from the remainder using methods given in [5].

*5) The Notion of "Scaling":* When possible, we would like to avoid having to estimate parameterization constants for our models (discussed in the previous section). For example, consider the problem of modeling the bytes transferred during an *FTP* session. Suppose Model-1 describes the distribution of bytes transferred as log-normal with $\bar{x} = 15$ and $\sigma_x = 4$, but Model-2 simply describes the distribution as log-normal, with a note that the values of $\bar{x}$ and $\sigma_x$ must be estimated from each dataset to which we want to apply the model. We would prefer Model-1 because it tells us *in advance* what to expect; Model-2 only tells us the general *shape* of what to expect, but not the exact quantities. We can use Model-1 to make quantitative predictions of what we will measure in the future, but Model-2 can only make qualitative predictions.

We will refer to models like Model-1 as *unscaled*, and those like Model-2 as *scaled*. In general, we use the term *scaling* to refer to tailoring a model to fit a dataset by estimating parameters of the model from the dataset. If we find that an unscaled model gives us good fits to many different datasets, then we have reason to believe that the model captures an "invariant" distribution. Such models are particularly powerful because they allow confident prediction of future distributions. Sometimes, however, a scaled model gives us significantly better fits to different datasets than an unscaled version of the model. In this case, the distribution's general shape (e.g., log-normal) might be invariant, but the particulars of the shape vary. Scaled models are less powerful than unscaled models because they allow less complete predictions, but are still valuable because with them we can explore possible behavior given separate hypotheses as to the values of the model's parameters.

In Section IV below we discuss how we chose whether to use the unscaled or scaled version of each of our models.

While estimating parameters applies only to analytic models, we can perform an analogous operation on empirical models. Suppose we are modeling a random variable $X$ using an empirical distribution $Y$. If $\bar{x}$ is significantly different from $\bar{y}$, or $\sigma_x$ from $\sigma_y$, then there is little hope that the empirical model faithfully describes $X$'s distribution. But it may be that if $Y$ were adjusted to have the same mean and standard deviation as $X$, then it would also describe $X$'s distribution well. This adjustment is easy to make. If we define a new empirical distribution:

$$Y' = \frac{\sigma_x}{\sigma_y}(Y - \bar{y}) + \bar{x} \tag{5}$$

then $Y'$ keeps the same general shape as $Y$ (since we have merely applied a linear transformation to $Y$) but has the same mean and standard deviation as $X$. When discussing empirical models, we will use the term *scaling* to refer to the transformation given in (5).

When developing our unscaled analytic models, we picked for each model parameter a round value lying somewhere in the range the parameter exhibited in the LBL test datasets (see [24] for details regarding the parameter ranges). 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 not particularly exact.

*6) Comparing Estimates:* Suppose we have two estimated quantities, $\hat{a}$ and $\hat{b}$, and we want to compare them to see which is smaller. Because the quantities are estimates, we would rather not make the comparison on the basis of testing whether $\hat{a} < \hat{b}$, since perhaps the error in estimating $\hat{a}$ and $\hat{b}$ is large enough that the comparison will be misleading. We can make a more meaningful comparison if we have a quantitative possible-error associated with $\hat{a}$ and $\hat{b}$ [this becomes relevant in Section III–E)].

A natural measure of possible error in an estimate is a standard deviation. Suppose in addition to $\hat{a}$ and $\hat{b}$ we have standard deviations $\sigma_a$ and $\sigma_b$, which quantify the error in the estimates. Then we can define a comparison operator $<_\sigma$ as follows:

$$\hat{a} <_\sigma \hat{b} \text{ iff } \hat{a} + \sigma_a < \hat{b} - \sigma_b.$$

Using this comparison operator, we will find $a$ less than $b$ only when the difference between $\hat{a}$ and $\hat{b}$ is greater than what we can account for by the uncertainty in their estimates. Similarly, we define $\hat{a} >_\sigma \hat{b}$ iff $\hat{b} <_\sigma \hat{a}$.

If neither $\hat{a} <_\sigma \hat{b}$ nor $\hat{b} <_\sigma \hat{a}$ then we say that $a$ and $b$ are *unordered*, rather than "equal," to stress that one of them may in fact be smaller than the other, but we are unable to say so conclusively.

### *B. Logarithmic Transformations*

In this and the next section we discuss how we transformed the data prior to analysis, including dealing with outliers.

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 [20]. 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, which we will sometimes write as $\lg x$.

If the random variable $Y = \log X$ has a normal distribution, then $X$ is said to have a *log-normal* distribution. Similarly, if $Y$ has an extreme distribution, $X$ has a *log-extreme* distribution. (We will often write these distributions as $\log_2$-normal and $\log_2$-extreme as a reminder that all logarithms in this paper are taken base 2.)

With a random variable $X$ that has a large range of values, the computed mean (3) and standard deviation (4) are greatly skewed by the largest of the $x_i$. The mean and standard deviation of the transformed quantity $Y = \log_2 X$ do not have this problem, though, since the logarithmic transformation greatly reduces the range of values. With these types of random variables, it is generally more meaningful to analyze $y$ and $\sigma_y$ than $x$ and $\sigma_x$.

We can then attach interpretations to the quantities $2^y$ and $2^{\sigma_y}$. In particular, the *geometric mean* of $X$ is defined as:

$$\text{Geometric mean}(X) = \sqrt[n]{\prod_{i=1}^{n} x_i} \tag{6}$$

and it is easy to show that this is the same as $2^y$. This equivalence between the geometric mean and $2^y$ suggests an analogous definition for the *geometric standard deviation*, which we define as:

$$\text{Geometric std. dev.}(X) = 2^{\sigma_y}. \tag{7}$$

If $Y$ is normally distributed, then $\sigma_y$ characterizes the range of $Y$. For example, about 68% of the distribution of $Y$ will reside in the range $y \pm \sigma_y$. We then can interpret the geometric standard deviation of $X$ as giving an analogous range for $X$. If $X$ is log-normally distributed, then about 68% of the distribution of $X$ resides in the range $2^{\bar{y} \pm \sigma_y}$. More explicitly

$$(\frac{2^y}{2^{\sigma_y}}) \leq 68\% \text{ of the range of } X \leq (2^y \times 2^{\sigma_y}). \tag{8}$$

In the tables summarizing the different protocols below, when we report quantities such as $\bar{x}$ and $\sigma_x$, *they reflect* $2^y$ *and* $2^{\sigma_y}$; that is, they give values computed using (6) and (7) and not values computed using (3) and (4). As a reminder of this fact, we precede all $\sigma_x$ values with a "×" symbol, in keeping with their interpretation given in (8). So, for example, a value of $\sigma_x = \times 8$ indicates $\sigma_y = \sigma_{\lg x} = \lg 8 = 3$.

### *C. Dealing with Outliers*

When applying a logarithmic transformation to non-negative data, one immediately runs into the problem of what to do with data values equal to zero, since after a logarithmic transformation these become $-\infty$. 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. The appendices of [24] report the number of connections thus eliminated for each dataset; in the worst case they comprised 0.5% of the total connections.

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). Again, see [24] for a discussion of these outliers.

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

### *D. Empirical Versus Analytic Models*

In this section we look at the relative advantages of empirical and analytic models, which motivates our subsequent pursuit of analytic models.

The main advantage of empirical models is that they are known to fully reflect a portion of Real World behavior. If there are consistent spikes in a distribution or even subtle deviations from "smooth" behavior, the empirical model will capture these nuances if they were present in the dataset from which the model was derived. An analytic model might easily miss these characteristics.

There are, however, several advantages of analytic models compared to empirical models:

- 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 corresponding estimates for the analytic model's parameters [Section III-A4)].

While these advantages are certainly attractive, the crucial issue remains whether an analytic model truly 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 interest, and it becomes useless (or, even worse, 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 given dataset. One approach is to require that the distribution predicted by the analytic model and that actually measured from the dataset be indiscernable in a statistical sense. A large body of literature examines techniques for testing for such statistical exactness (an excellent reference is [5]).

In our earlier work we tried to find statistically exact models but failed (see [24] for details). This failure, however, is not surprising: it is well known in the statistics community that large datasets almost never have statistically exact descriptions [16].[5] The next section addresses how to deal with this failure.

### *E. Measuring Discrepancy*

Even if a model is not statistically exact, we can still attempt to gauge how *close* it is to the distributions it endeavors to describe. To do so, we turn to techniques for measuring *discrepancy*.

One widely-used technique for doing so is based on a modified $\chi^2$ test [19]. To understand it, we first review the $\chi^2$ test itself.

Suppose we have observed $n$ instances of a random variable $Y$ which we want to model using another model distribution $Z$. We partition the distribution $Z$ into $N$ *bins*. Each bin has a

[5] As well as giving a general discussion of this problem, [16] also analyzes an experiment in which 26,306 throws of 12 dice failed a $\chi^2$ test for an exact fit to the predicted binomial distribution.

probability $p_i$ associated with it, which is the proportion of the distribution $Z$ falling into the $i$th bin. Let $Y_i$ be the number of observations of $Y$ that actually fell into the $i$th bin. Then one computes the statistic

$$X^2 = \sum_{i=1}^{N} \frac{(Y_i - np_i)^2}{np_i}.$$

The $\chi^2$ test for a statistically-exact fit involves testing where the $X^2$ statistic falls in the range of a corresponding $\chi^2$ distribution (the exact details of the comparison are secondary to understanding the remainder of this section).

The $\chi^2$ discrepancy measure is then simply $X^2/n$. This is essentially the measure used in our preliminary work [24]. There is, however, a problem with using $X^2/n$ as a discrepancy measure [26]. In general, the optimal value of $N$ (the number of bins) to use when computing $X^2$ varies with the size of $Y$ and its standard deviation $\sigma_y$ [see Section III–F)]. The $X^2/n$ discrepancy measure, however, cannot be used to compare discrepancies for different values of $N$.

Pederson and Johnson [26] describe a related discrepancy measure, $\lambda^2$, which corrects the $X^2/n$ measure so that $\lambda^2$ can be used to compare discrepancies for different values of $N$. They also give a way to compute $\sigma_\lambda$, the standard deviation associated with estimating $\hat{\lambda}^2$ for a particular dataset $Y$ and model distribution $Z$. Knowing $\sigma_\lambda$ lets us use the $<_\sigma$ operator (Section III–A6) for comparing estimated discrepancies. Given two models, $Z_1$ and $Z_2$, we can then state in a meaningful way whether $Z_1$ is a better description of $Y$ than $Z_2$ (if we find that $\lambda^2{}_{Z_1} <_\sigma \lambda^2{}_{Z_2}$), or that $Z_2$ is better than $Z_1$ ($\lambda^2{}_{Z_2} <_\sigma \lambda^2{}_{Z_1}$), or that the two models are *unordered*, indicating that they are roughly equal.

We now summarize how to compute $\lambda^2$ for assessing the discrepancy between a random variable $Y$ and a model distribution $Z$. First, let $E_i = np_i$ be the expected count for the $i$th bin, and $D_i = Y_i - E_i$ be the discrepancy in the $i$th bin. Then define:

$$K = \sum_{i=1}^{N} \frac{D_i}{E_i}. \tag{9}$$

(Note that $K$ is quite similar to $X^2$ except the numerator of the summation is not squared.) We then define:

$$\hat{\lambda}^2 = \frac{X^2 - K - \mathrm{df}}{n - 1},$$

where "df" is the number of *degrees-of-freedom* in computing $X^2$ and $K$. For our purposes, df $= N - 1 -$ Est, where "Est" is the number of parameters estimated from $Y$ [0 for unscaled models, 1 or 2 for scaled models; see Sections III–A4) and 5)].

The *variance* associated with this estimate of $\lambda^2$ is given by

$$\hat{v}(\hat{\lambda}^2) = [2\mathrm{df} + 4n\hat{\lambda}^2 + 4n\hat{\lambda}^4 + 4T]/n^2,$$

where:

$$T = \sum_{i=1}^{N} [D_i^3 - 2D_iE_i + \frac{5}{2}D_i^2 + \frac{3}{2}(D_i + E_i)]/E_i^2.$$

(Reference [26] states that this expression for $T$ is not quite exact when the parameters of $Z$ are estimated from $Y$, but they found in practice the correction makes little difference).

The standard deviation associated with estimating $\lambda^2$ is then:

$$\sigma_\lambda = \sqrt{\hat{v}(\hat{\lambda}^2)}.$$

### *F. Considerations when Measuring Discrepancy*

There are a number of considerations when evaluating models using a discrepancy measure such as $\lambda^2$: how to pick the number of bins to use; how to capture significant discrepancies in the distribution's tails; what to do about significant "spikes" that fall into a single bin; and how to deal with dataset values not falling into any of the bins. We address each of these in turn.

When computing $\lambda^2$, we are forced to make a choice as to how many bins to use [i.e., the value of $N$ in (9)]. If $N$ is too small then we will be measuring discrepancies only on a gross scale, and similarly if $N$ is too large then we will be sensitive to quite small discrepancies which perhaps are of no real interest. Fortunately, statistics provides some guidance. Scott has shown that to minimize the mean-square error in approximating a distribution using fixed-sized bins, the bin-width should be:

$$w = 3.49\hat{\sigma}_x n^{-1/3} \tag{10}$$

where $\hat{\sigma}_x$ is the estimated standard deviation of the distribution (4), $n$ is the number of instances in the distribution, and $w$ the bin-width [30]. Given the range of the distribution, it is straightforward to compute the value of $N$ to use such that each bin has width $w$. Fortunately, the value of $w$ is not strict; any value close to $w$ provides a satisfactory estimate of the distribution, so we rounded $N$ to the nearest multiple of 5.

One important point we found, though, was to compute $\sigma_x$ *after* first applying the logarithmic transformation to $X$ [Section III–B)]. That is, we used for $\sigma_x$ the geometric standard-deviation (7) and not the value given in (4). In general, our untransformed data had such large ranges that using the untransformed values of $X$ in (10) resulted in a very large number of bins.

One other point is that when using (10) to compute $N$ for an empirical model, we had to decide between computing $w$ using $\hat{\sigma}_x$ and $n$ from the dataset being modeled, or from the empirical model. We chose whichever had the smaller $n$.

The number of bins tested ranged from 5 to 240; the average was around $N = 35$.

A related consideration with using the $\lambda^2$ discrepancy measure is that when using constant-sized bins the measure does not give any particular emphasis to the distribution's tails, which sometimes are the most important aspect of a distribution. We address this consideration in Section III–H).

A third consideration is that the discrepancy measure does not inform us of 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 consideration 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).

A final consideration is that since an empirical model has bounds on the range of values it allows for, the tested dataset may have values outside the range of any bin. We removed such values from the dataset prior to computing its fit to the model. We did, however, include these values when evaluating the model's tail discrepancy Section III–H).

### *G. Testing for Significant Differences*

The $\lambda^2$ discrepancy measure and the $<_\sigma$ operator allow us to determine whether, given two datasets and a model, the model is significantly better at modeling one of the datasets than the other. They also allow us to determine, given one dataset and two models, which of the models (if either) is significantly better at modeling the dataset.

In this section we build on these techniques to develop methods for comparing datasets and models in a general sense. The first method allows us to determine with high confidence whether a given model is better at describing one set of datasets than it is at describing a second set. The second method allows us to determine with high confidence whether one model is better than another model at describing a set of datasets.

*1) Method I:* In this section we describe Method I, a method for determining whether a given model is significantly better at modeling one set of datasets than at modeling another.

Suppose we have a model $Y$ and two sets of datasets, $D_1$ and $D_2$, with $m$ and $n$ datasets in each set, respectively. For example, $Y$ might be a model of the number of bytes transferred during an *SMTP* connection, $D_1$ the set of LBL datasets, and $D_2$ the set of non-LBL datasets. We would like to determine whether model $Y$ does substantially better at modeling $D_1$ than at modeling $D_2$, because we suspect that *SMTP* connections may show considerable variation between network sites. If our suspicion is correct, and if model $Y$ was derived from some of the LBL datasets, then we would expect to find that $Y$ is significantly better at modeling $D_1$ than $D_2$.

For each dataset in $D_1$ and $D_2$ we can compute the corresponding value of $\lambda^2_Y$, the discrepancy between the model and the dataset. It is not obvious, though, how to take this collection of discrepancies and reduce it to a simple statement that $Y$ does or does not appear to model $D_1$ significantly better than $D_2$.

The approach we take is as follows. We assume the null hypothesis that $Y$ performs equally well when modeling $D_1$ as when modeling $D_2$.[6] We compare each dataset $d_1$ in $D_1$ against each dataset $d_2$ in $D_2$. Let $l$ be the number of comparisons for which we found $d_1 <_\sigma d_2$, $g$ the number of times $d_1 >_\sigma d_2$, and $u$ the number of times two datasets were unordered. There are $m$ datasets in $D_1$ and $n$ in $D_2$, so we have $l + g + u = mn$. We now restrict ourselves to just the $t = l + g$ comparisons that were not unordered. If the null hypothesis is correct, then the probability that in $t$ tests we would find $l$ instances for which $d_1 <_\sigma d_2$ is:

$$\mathrm{P}(\text{In } t \text{ trials, } d_1 <_\sigma d_2 \text{ occurs } l \text{ times}) = \binom{t}{l} 2^{-t}.$$

We can easily generalize the above to find the probability that $d_1 <_\sigma d_2$ occurs at least $l$ times, rather than exactly $l$ times. For Method I, then, we pick a value of $k$ such that, given the null hypothesis, the probability that $d_1 <_\sigma d_2$ occurs at least $k$ times is $\leq 5\%$. We then test whether indeed $d_1 <_\sigma d_2$ occurs $k$ or more times. If so, then with 95% confidence we declare that the null hypothesis is incorrect and that $Y$ does in fact model $D_1$ better than $D_2$. If, however, $d_1 <_\sigma d_2$ occurs fewer than $k$ times, then the test is inconclusive, and we refrain from ruling out the null hypothesis that $Y$ is equally good (or bad) at modeling both $D_1$ and $D_2$.

One concern when applying Method I is: what if $u$, the number of times $d_1$ and $d_2$ were unordered, is large relative to $l+g$? For example, if $u = 95$, $l = 5$, and $g = 0$, then Method I will declare that $Y$ models $D_1$ better than $D_2$, even though it might be more reasonable to say that it basically models the two equally well. For our purposes, this did not turn out to be a problem: when Method I (or Method II; see below) declared a significant difference, we always had $u < \frac{4}{3}(l + g)$.

*2) Method II:* In this section we describe Method II, a method for determining whether one of two models is significantly better than the other at modeling a given set of datasets. It is quite similar to Method I.

Suppose we have a set $D$ of $n$ datasets, and two models, $Y$ and $Z$. For example, $D$ might be all of the datasets listed in Tables II and III, $Y$ might be an analytic model for bytes transferred in an *SMTP* connection, and $Z$ an empirical model of the same random variable. We would like to determine whether either of the models is significantly better than the other for describing $D$.

For each dataset in $D$ we can compute two discrepancy values, $\lambda^2_Y$ and $\lambda^2_Z$. We want to take this collection of discrepancies and reduce it to a statement that $Y$ or $Z$ (or neither) appears significantly better than the other at modeling the datasets in $D$.

We take a similar approach to that described for Method I. We assume the null hypothesis that $Y$ and $Z$ perform equally well when modeling $D$. Now, however, for each of the $n$ datasets in $D$, we compare the corresponding values of $\lambda^2_Y$ and $\lambda^2_Z$. Let $l$ be the number of times $\lambda^2_Y <_\sigma \lambda^2_Z$, $g$ the number of times $\lambda^2_Y >_\sigma \lambda^2_Z$, and $u$ the number of times the comparison is unordered. We consider just the comparisons that were not unordered. Let $t = l + u$. If the null hypothesis is correct, we have:

$$\mathrm{P}(\text{In } t \text{ trials, } \lambda^2_Y <_\sigma \lambda^2_Z \text{ occurs } l \text{ times}) = \binom{t}{l} 2^{-t}.$$

We again generalize the above to find the probability that $\lambda^2_Y <_\sigma \lambda^2_Z$ occurs at least $l$ times, and for Method II pick a corresponding value of $k$ such that, given the null hypothesis, the probability that $\lambda^2_Y <_\sigma \lambda^2_Z$ occurs at least $k$ times is $\leq 5\%$.

[6] Here "equally well" means that given datasets $d_1$ from $D_1$ and $d_2$ from $D_2$, if $Y$ models one better than the other (in a $<_\sigma$ sense), then the probability that it models $d_1$ better than $d_2$ is 1/2.

As before, this test, if successful, lets us state with 95% confidence that the null hypothesis is incorrect and that $Y$ does in fact model $D$ better than $Z$ does. If the test is inconclusive, however, we refrain from ruling out the null hypothesis that $Y$ and $Z$ are equally good (or bad) at modeling $D$.

*3) A Note on Methods I and II:* One important point regarding Method I and Method II is that they are fairly conservative. Their use of both the $<_\sigma$ comparison and 95% confidence levels assures that differences they uncover are very likely to be significant. It is quite possible for the tests to fail to declare two unequal models as being different, but it is not likely that they will erroneously declare two equal models as being different.

### H. Evaluating Deviation in the Tails

Often a distribution's behavior in its lower or upper tail can be crucially important. For example, as mentioned in Table I and developed in Section VIII–D), for the distribution of bytes in *FTPdata* "bursts," the upper 2% tail is so heavy that large-though-rare bursts will often completely dominate *FTP* traffic. As discussed in Section III–F), the $\lambda^2$ discrepancy measure does not give any special weight to agreement with a distribution's tails. In this section we present a simple way of qualitatively evaluating how well a model captures a distribution's tails.

Suppose we test the model against $n$ datasets. For the $i$th dataset, let $a_i$ be the number of instances predicted to lie in a given tail, and $b_i$ be the number actually found to do so. Define:[7]

$$\xi_i = \log_2 \frac{a_i}{b_i}.$$

Positive values of $\xi_i$ indicate that the model *overestimates* the tail, and negative values that it *underestimates* the tail.

With this definition, an underestimate by a factor of two ($a_i/b_i = 1/2, \xi_i = -1$) is considered just as bad as an overestimate by the same factor ($a_i/b_i = 2, \xi_i = 1$).

A value of $\xi_i = 0$ indicates that the model perfectly captures the $i$th dataset's tail. With this in mind, we then compute

$$\sigma_\xi = \sqrt{\sum_{i=1}^{n} \xi_i^2/n} \tag{11}$$

$\sigma_\xi$ then represents the standard deviation, from a mean of 0, of the model's accuracy in the tail. If we find $\sigma_\xi < 1$ then the model typically predicts the tail population within a factor of two; we deem this "acceptable." If $1 \le \sigma_\xi < \log_2 5$, then the model typically errs in predicting the tail by a factor between 2 and 5; we deem this "bad." Similarly, if $\sigma_\xi \ge 5$, then we evaluate the model's tail behavior as "very bad."

These definitions of "acceptable," "bad," and "very bad" appear quite generous; after all, one might wonder how a model could possibly misrepresent a dataset's tail by more than a factor of 5. Yet it turns out that for the extreme 1% tails, a fair number of our models are evaluated as "bad" or "very bad"; see Section IV.

One final note in evaluating tails. For models describing bytes transferred, we only evaluate the upper tails, as in these cases disagreement in the lower tails is of little consequence, 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. See [24] for more details concerning the models' tail-behavior.

### I. Evaluating Correlation

As noted in Section III–A2), it is often important to model the correlations between instances of a random variable. While doing so is beyond the scope of this paper, in Section IV we present a simple summary of each random variable's correlation, computed as follows.

The *autocorrelation function* (which we abbreviate as $\gamma_x(l)$, but do not further define here) of a random variable $X$ measures the degree to which instances of $X$ are correlated. For a given value of $l$ (called the "lag"), $\gamma_x(l)$ is a number ranging from -1 to 1. A value of $\gamma_x(l)$ close to 1 indicates that if $x_k$, the $k$th instance of $X$, is higher (lower) than $\bar{x}$, then $x_{k+l}$ will also tend to be higher (lower). A value of $\gamma_x(l)$ close to -1 indicates that $x_k$ and $x_{k+l}$ are anti-correlated; if $x_k$ is high then $x_{k+l}$ will tend to be low, and vice versa. A value of $\gamma_x(l)$ close to 0 indicates that $x_k$ and $x_{k+l}$ are not linearly correlated; to first order, knowing the value of $x_k$ does not help in predicting the value of $x_{k+l}$.

With many random variables, $\gamma_x(1)$ is a particularly significant value of $\gamma_x$, because if a random variable is correlated, often the correlation is greatest at a lag of 1. For example, a high value of $\gamma_x(1)$ indicates that successive instances of $X$ tend to have comparable values.

For our summary of correlation in Table VI, we computed $\gamma_x(1)$ for each random variable and dataset. We looked at how often $|\gamma_x(1)| > 2/\sqrt{n}$ (where $n$ is the number of instances in the dataset), because if $X$ is uncorrelated then $|\gamma_x(l)|$ will exceed this value only 5% of the time. If we found $|\gamma_x(1)| > 2/\sqrt{n}$ occurred for more than half of the datasets, then we considered the random variable to be correlated. (This happened for all but one of the random variables.) We then looked at the magnitude of the mean lag-1 autocorrelation, $|\bar{\gamma}_x(1)|$. If it was $\le 0.1$, we considered the random variable to be "weakly correlated," otherwise "significantly correlated." Finally, we looked at the range of values of $\gamma_x(1)$. If they were significantly positive (measured by a modified Method II test), we considered the correlation to be positive. Otherwise (since they were never significantly negative), we considered the correlation to be "undirected."

## IV. Summary and Evaluation of Models

Using the methodology described in Section III, we constructed analytic and empirical models of random variables associated with wide-area *TELNET*, *NNTP*, *SMTP*, and *FTP* TCP connections. As discussed in Section III-A2), we generically refer to the analytic model for a random variable as $A$, the UCB-derived empirical model (reflecting Tcplib; [7]) as $U$, and the empirical model derived from LBL-2 as $L$.

[7] One problem arises when using this definition of $\xi_i$: if $b_i$ is 0 then $\xi_i$ becomes undefined. We address this problem by replacing $b_i$ with 0.5.

TABLE VI
SUMMARY OF ANALYTIC MODELS OF CONNECTION CHARACTERISTICS

| Proto. | Variable | Abbr. | | Section | Model | Parameters | Corr. |
|---|---|---|---|---|---|---|---|
| *TELNET* | originator bytes | A | $T_{\mathrm{orig}}$ | V-B | $\log_2$-extreme ( 1); (Section III-B) | $\alpha \approx \log_2 100;\ \beta \approx \log_2 3.5$ | + |
| | responder bytes | B | $T_{\mathrm{resp}}$ | V-C | $\log_2$-normal, 80-100% | $\bar{x} = \log_2 4500;\ \sigma_x = \log_2 7.2$ | + |
| | duration secs. | C | $T_{\mathrm{dur}}$ | V-D | $\log_2$-normal | $\bar{x} = \log_2 240;\ \sigma_x = \log_2 7.8$ | + |
| | resp./orig. | D | $T_{\mathrm{ratio}}$ | V-E | $\log_2$-normal | $\bar{x} = \log_2 21;\ \sigma_x = \log_2 3.6$ | + |
| | resp./dur. | E | $T_{\mathrm{rd1}}$ | V-F | exponential, 0-90% resp. | $\lambda \approx 1/30$ | + |
| | resp./dur. | F | $T_{\mathrm{rd2}}$ | V-G | $\log_2$-normal, 90-100% resp. | $\bar{x} = 5.3;\ \sigma_x = 1.5;$ | ± |
| *NNTP* | originator bytes | G | $N_{\mathrm{orig}}$ | VI-B | $\log_2$-normal | $\bar{x} \approx 11.5;\ \sigma_x \approx 3;$ | ++ |
| *SMTP* | originator bytes | H | $S_{\mathrm{orig}}$ | VII-B | $\log_2$-normal + 300B, 0-80%<br>$\log_2$-normal + 300B, 0-100% | $\bar{x} \approx 10;\ \sigma_x \approx \log_2 2.75$<br>$\bar{x} \approx 8.5;\ \sigma_x \approx \log_2 3$ | ++ |
| *FTP* | connection bytes | I | $F_{\mathrm{conn}}$ | VIII-B | $\log_2$-normal | $\bar{x} \approx \log_2 3000;\ \sigma_x \approx 4$ | ++ |
| | session bytes | J | $F_{\mathrm{sess}}$ | VIII-B | $\log_2$-normal | $\bar{x} = 15;\ \sigma_x = 4$ | no |
| | burst bytes | K | $F_{\mathrm{burst}}$ | VIII-B | Pareto (2), 95-100% | $k \approx 10^{5.5}$ | ++, ± |

Table VI summarizes the random variables and the corresponding analytic models. The "Variable" column lists the random variable being modeled and the "Abbr." columns the label and short name we will use subsequently to identify the variable. The "Section" column lists the section in the paper that develops the model in detail.

The "Model" column lists the analytic models used to describe the random variable's distribution. Almost all first apply a $\log_2$ transformation to the data, as described in Section III-B). One model is $\log_2$-extreme, where the "extreme" distribution is defined by (1); one is exponential; one is Pareto (2); and the remainder are $\log_2$-normal.

Five of the models have restrictions. The *TELNET* responder bytes model describes only the upper 80% of the responses. The *TELNET* $E$ and $F$ models describe the ratio of the responder bytes to the connection's duration. The first such model does so for those connections whose number of responder bytes fell into the lower 90% of all *TELNET* connections. The second model describes this ratio for those connections in the upper 10% of all responses. The *SMTP* H model uses parts of two different $\log_2$-normal distributions in its description of the bytes transferred by the *SMTP* connection originator. The lower 80% of the originator distribution is modeled using the lower 80% of the first $\log_2$-normal distribution; similarly, the upper 20% is modeled using the upper 20% of the second $\log_2$-normal distribution. Finally, as developed in Section VIII–D), for *FTPdata* connection bursts it is crucial to accurately model the upper tail, so the *FTP* $K$ model describes only the upper 5% of the distribution of bytes in *FTP* bursts.

The "Parameters" column gives the parameters we used for the unscaled [Section III-A5)] version of the model, after applying $\log_2$ transformations. Parameters listed using "≈" instead of "=" correspond to models that performed significantly better when scaled than when unscaled (see below). For these models, the parameters given in Table VI should only be used with considerable caution.

The final column in Table VI, "Corr.," gives a simple summary of the correlations present in the datasets, using the methodology given in Section III-I. A "+" sign indicates weakpositive correlation, "++" indicates significant positive correlation, "±" indicates weak undirected correlation, and "no" indicates that the random variable does not appear to be correlated. Two values are given for the $F_{\mathrm{burst}}$ variable; the first reflects the correlation of all of the instances of the variable, and the second, the correlation of just the top 5% tail.

We see that successive *TELNET* connections are weakly correlated, perhaps due to small hour-to-hour variations in the characteristics of *TELNET* connections. Successive bulk-transfer connections, on the other hand, all tend to be significantly correlated, except that the size of complete *FTP* sessions is not correlated. These correlations are not hard to explain: *NNTP* connections will tend to be correlated as new network news first arrives in an inbound connection and then is propagated soon afterward in an outbound connection. *SMTP* connections can be correlated due to mailing lists expansions generating similar connections one after another, or inbound connections spawning outbound connections due to mail forwarding. *FTPdata* connection sizes will tend to be correlated because directory listings are substantially smaller than file transfers, so back-to-back directory listings will contribute to positive correlation. This effect will also contribute to correlation in the size of *FTP* bursts, if directory listings occur more than 4 seconds apart [Section VIII–D)] and thus constitute separate bursts. That the size of *FTP* sessions does not appear correlated suggests that *FTP* sessions are statistically independent, in line with the finding in [23] that *FTP* session arrivals appear well-modeled as Poisson processes.

Table VII summarizes our evaluation of the different models we constructed. A check (√) in one of the "Scaling" columns indicates that we found [using Method II, Section III–G2)] that the scaled version [Section III–A5)] of the corresponding model performed significantly better than the unscaled version. When this was the case, for the remainder of our evaluation we used the scaled version of the model; otherwise, we used the unscaled version. As noted in Section III–A5), a successful unscaled model suggests that the random variable being modeled is in some sense "invariant," and we can make strong predictions about future behavior using such a model.

The "Variation" columns summarize three tests for significant variation among the performance of the models. A check in the "site" column means that the $A$ and $L$ models did significantly better [as tested using Method I; Section III–G1)]

TABLE VII
EVALUATION OF MODELS

| Model | | Scaling | | | Variation | | | Ordering | Range | 1% Tail |
|---|---|---|---|---|---|---|---|---|---|---|
| | | $A$ | $U$ | $L$ | site | wide | time | | mean$\lambda^2$ | analytic dev. |
| A | $T_{\text{orig}}$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | $U < \{A, L\}$ | 0.05-0.09 | $u$: ++ |
| B | $T_{\text{resp}}$ | | ✓ | | ✓ | ✓ | ✓ | $\{A, U\} < L$ | 0.04-0.07 | $u$: + |
| C | $T_{\text{dur}}$ | | ✓ | | ✓ | ✓ | | none | 0.04-0.09 | $l$: +, $u$: ±± |
| D | $T_{\text{ratio}}$ | | | | ✓ | ✓ | ✓ | $U < \{A, L\}$ | 0.08-0.19 | |
| E | $T_{\text{rd1}}$ | ✓ | | | ✓ | ✓ | ✓ | $\{U, L\} < A$ | 0.02-0.10 | |
| F | $T_{\text{rd2}}$ | | | ✓ | ✓ | ✓ | | none | 0.06-0.10 | $l$: ±, $u$: ± |
| G | $N_{\text{orig}}$ | ✓ | | | * | | ✓ | $\{U, L\} < A$ | 0.36-2.00 | $u$: ++ |
| H | $S_{\text{orig}}$ | ✓ | ✓ | ✓ | ✓ | | | $U < \{A, L\}$ | 0.15-0.34 | |
| I | $F_{\text{conn}}$ | ✓ | ✓ | | † | † | | $U < A$ | 0.18-0.27 | $u$: + |
| J | $F_{\text{sess}}$ | | | | † | † | | none | 0.09-0.14 | $u$: + |
| K | $F_{\text{burst}}$ | ✓ | ✓ | ✓ | † | | | $L < \{A, U\}$ | 0.16-0.47 | $u$: +‡ |

in modeling the LBL datasets than in modeling the non-LBL datasets. From this we infer that the associated random variable shows significant site-to-site variation, as models derived from one site reflect that site significantly better than other sites.

A check in the "wide" column means that the $A$ and $L$ models did significantly better (using Method I) in modeling the "stub" sites than in modeling the internetwork gateway sites (the UK and NC datasets). From this we infer that there is significant variation in the corresponding random variable when modeling "very wide" traffic as opposed to "less wide" traffic, perhaps due to a richer degree of connection multiplexing.

A check in the "time" column means that the $A$ and $L$ models did significantly better (Method I) modeling the second fifteen days of the LBL-1, LBL-3, and LBL-4 datasets, than in modeling the complete LBL-5, LBL-6, and LBL-7 datasets. From this we infer that the corresponding random variable changes significantly over time, since the $A$ model was derived from the first fifteen days of the first four LBL datasets [the "test datasets"; Section III–A2)] and the $L$ model from the LBL-2 dataset, one of the early LBL datasets.

We have marked the $G$ variables' site variation with a "$*$" because while the Method I test did not indicate a significant difference between the LBL sites and the non-LBL sites, we believe this is simply because there is so much variation among the LBL datasets themselves (as indicated in part by the "time" checkmark) that it exceeds the considerable site-to-site variation. See Fig. 4 below for an illustration of the large site-to-site variation.

We marked the $I$, $J$, and $K$ variables' "site" and "wide" entries with "†"'s because of the following curious phenomenon: we found that the $A$ and $L$ models did significantly *worse* (as indicated by a Method I test) modeling the LBL datasets than modeling the non-LBL datasets. We do not have a firm explanation for this behavior. Evidently the LBL datasets are substantially "noisier" than the non-LBL datasets, perhaps because there are unusually high variations or spikes in the size of files transferred by scientists, or perhaps because the 30 day length of the LBL datasets allows more opportunity for rare phenomena to skew the distribution.

The "ordering" column gives the results of using Method II to compare the effectiveness of the $A$, $U$, and $L$ models. An entry like "$U < \{A, L\}$ indicates that the Method II test found that the $U$ model performed significantly less well than both the $A$ and $L$ models, but that the $A$ and $L$ models are unordered. An entry of "none" indicates that all three models were unordered. We note that except for variable $B$, the $A$ model does as well as or better than both of the empirical models, indicating that analytic models can perform as well as empirical models for describing wide-area connection characteristics.

While the "ordering" column gives an indication as to the relative performance of the three models, the "Range" column gives an indication as to absolute performance. Here we list the range of $\bar{\lambda}^2$, the average value of $\lambda^2$ for each of the three models. For example, for variable $A$, the best model (which must have been either $A$ or $L$) had an average $\lambda^2$ value of 0.05, and the worst ($U$), 0.09. Because $\lambda^2$ has the property that it can be meaningfully compared for different models, the variation in the "Range" column tells us a good deal about how each model performs. For example, we see that of the four protocols, *TELNET* is consistently modeled the most successfully, even by the worst-performing of the three models. *NNTP*, on the other hand, is the most poorly modeled, not at all surprising given the irregular distributions of *NNTP* originator bytes (see Section VI–B) Fig. 4). *SMTP* is not well-modeled either, though considerably better than *NNTP*. Finally, *FTP* connection and burst sizes are modeled about as well as *SMTP* originator sizes, but *FTP* session sizes appear as well modeled as *TELNET* connections.

The final column in Table VII summarizes the deviation of the $A$ model when describing the lower and upper 1% tails of the random variable's distribution. Here we have used the methodology discussed in Section III–H). A blank entry indicates that the $A$ model's description of the upper (and lower, if appropriate) 1% tail is "acceptable." A "+" indicates that the description is "bad" and that the $A$ model consistently *overestimates* the tail. A "++" indicates that the description is "very bad" and also consistently an overestimation. "±" and "±±" are analogous to "+" and "++" except the model sometimes overestimates the tail and sometimes underestimates it. Finally, the K variable's tail is marked with a "‡" as a reminder that the $F_{\text{burst}}$ model only models the upper 5% of *FTPdata* connection bursts, so the upper 1% tail of this model reflects only on the upper 0.05% tail of the entire distribution.

TABLE VIII
SUMMARY OF TELNET CONNECTIONS

| Dataset | # Conn | # Rej | $\bar{x}_{\rm orig}$ | $\sigma_{\rm orig}$ | $\max_{\rm orig}$ | $\bar{x}_{\rm resp}$ | $\sigma_{\rm resp}$ | $\max_{\rm resp}$ | $\bar{x}_{\rm dur}$ | $\sigma_{\rm dur}$ | $\max_{\rm dur}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 | 15,437 | 52 | 242B | ×4.5 | 777KB | 5.7KB | ×7.3 | 14.0MB | 270 s | ×7.7 | 102.9 h |
| LBL-7 | 17,998 | 106 | 235B | ×4.2 | 651KB | 5.7KB | ×6.9 | 3.10MB | 252 s | ×7.5 | 172.8 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 |

While we have only reported the tail of the $A$ model, the $U$ and $L$ models have similar sorts of problems. Of the three models, the $A$ model is not, overall, especially good or bad in describing the tails.

We also tested the 10% tails using the same methodology. We found that all of the 10% tails were "acceptable."

The next four sections develop in greater detail the models summarized in Tables VI and VII above. One aspect of wide-area TCP connections we do not discuss in this paper is the connection arrival process (other than to note the presence of periodic patterns). Instead, we refer the reader to [23].

## V. TELNET

We now turn to analyzing the characteristics of individual protocols and developing models to describe them. We begin with *TELNET*.[8]

### A. Overview of TELNET Connections

Table VIII 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 [Section III–C)]; [24] details the rejected connections. As discussed in [22], the LBL-6 and LBL-7 *TELNET* traffic included a large number of connections due to periodic traffic. We removed those connections prior to our analysis, and they do not appear in Table VIII.

The fourth through sixth 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 (6), the geometric standard deviation (7), and the maximum. As noted in Section III–B), we applied $\log_2$ transformations to the data prior to analysis.

The seventh through ninth columns give the same summary for the number of bytes transmitted by the responder (remote computer), and the 10th through 12th columns summarize 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 [12] reported average connection lengths for local logins of 17 to 34 minutes, and [1] gives a local-login median of 20 minutes and a mean of 45-50 minutes. Jackson and Stubbs inferred that connection time "may be considerably reduced by providing a high-speed channel from the computer to the user," so we appear to be seeing Jackson and Stubbs' effect 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, even if we adjust for the fact that the LBL datasets span more days and hence give an opportunity to detect long-lived connections missed by the other datasets. We conclude that, at least with regard to mean bytes transferred and duration, the LBL *TELNET* traffic is significantly different from that at other sites (and this is what is shown in Table VII above).

We also note an apparent trend over the LBL datasets towards increasing values of $\bar{x}_{orig}$ and $\bar{x}_{resp}$, 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 (again reflected in Table VII).

Finally, we note that the data provide support for the observation in [8] that "interactive applications can generate 10 times more data in one direction than the other," and actually suggest the factor is around 20:1. 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 [15].

In Section V–E), we present a model for this ratio.

### B. TELNET Originator Bytes (A)

With the bulk transfer protocols examined in subsequent sections, we usually are only interested in modeling the number of bytes transferred. 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 (generally a human typing at a keyboard). The best fit we found to the LBL *TELNET* test datasets came using the $\log_2$-extreme distribution [Section III–B)].

[8] [24] presents a similar overview for *RLOGIN* traffic, along with results of modeling it using the *TELNET* models developed in this section.

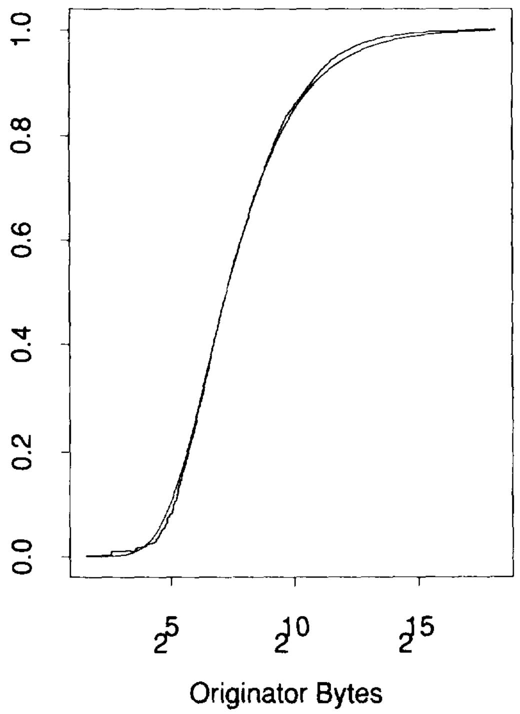

Fig. 1. TELNET originator-bytes model for LBL-2: $\log_2$-extreme distribution

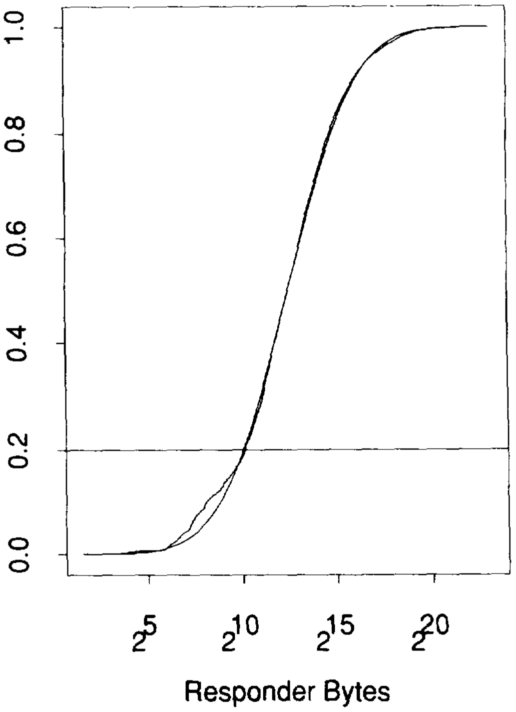

Fig. 2. Censored $\log_2$-normal fit to upper 80% of LBL-4 TELNET responder bytes.

Fig. 1 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, where, as indicated in Table VII, the upper 1% tail is grossly overestimated.

One important point regarding this model is that it is easy to assume that the number of *bytes* generated by the *TELNET* originator equates to the number of *packets* generated. As mentioned in [23], this equation is erroneous. Often an originator packet holds more than a single keystroke. [23] finds that the number of *packets* generated by the originator appears better modeled using a log-normal distribution; as we did not have any intra-connection information, we were unable to test this finding.

## *C. TELNET Responder Bytes (B)*

We next model the bytes transferred by the *TELNET* responder. As shown in Fig. 2, the upper 80% of the distribution is well-modeled using a $\log_2$-normal distribution, but 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. No doubt this roughness is due to the varying sizes of log-in dialogs and message-of-the-day greetings. Fortunately, exactly modeling the lower tail is of little importance, so we limit ourselves to modeling just the upper 80% [in doing so, we apply data *censoring*; see Section III–A)].

## *D. TELNET Duration (C)*

We model *TELNET* connection durations using a simple $\log_2$-normal distribution. An alternative is to simply not model the duration at all, for the following reason: Paxson and Floyd show that by using Poisson arrivals, log-normal originator connection sizes (in packets), and the Tcplib packet interarrival distribution, one can synthesize *TELNET* traffic which reproduces the same burstiness at different time scales as observed in actual traffic traces [23]. The duration of their *TELNET* connections is simply the sum of the packet interarrivals. That their model is successful without using a separate distribution for the connection duration suggests that to simulate *TELNET* traffic with realistic durations, it is sufficient to use the Tcplib interarrival distribution.

## *E. TELNET Responder/Originator Ratio (D)*

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 $\log_2$-normal distribution. 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 unscaled model uses a mean ratio of 21:1.

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

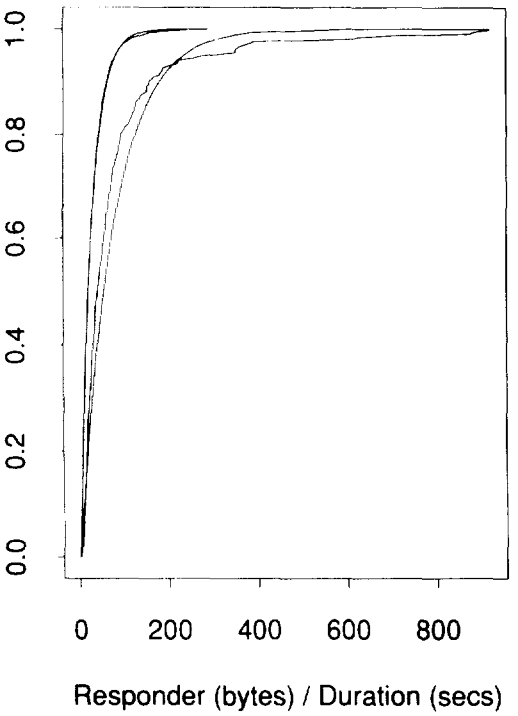

Fig. 3. Responder/duration distributions for LBL-1: exponential fits.

difference arises because while both the responder bytes and the ratio are $\log_2$-normal distributed, the originator bytes are $\log_2$-*extreme* distributed. Multiplying the originator byte's log-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 a log-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.

### *F. TELNET Responder/Duration Ratio (E and F)*

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 (though see the last paragraph of this section).

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.

Fig. 3 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. 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 transferred) occur over relatively short durations: while $\bar{x}_{\rm resp}$ in these large connections is 45 times that of the smaller (lower 90%) connections, $\bar{x}_{\rm dur}$ is only 16 times that of the smaller connections. This phenomenon was also observed by the authors of [29], 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 upper 10% of the responders we compared considerably fewer datasets. Our requirement that each dataset include at least 100 measurements [Section III–C)] ruled out any dataset with fewer than 1000 *TELNET* connections, leaving just the LBL and NC datasets. The fit remains good, though.

The use of two separate models for the responder/duration ratio is not wholly satisfying, but was the best we could find. One other somewhat successful model for relating *TELNET* connection size and duration was the ratio between the duration and the originator size. We found that when this ratio was $\geq 0.5$ (about 80% of the connections), then it was well modeled using a Pareto distribution (2). But when the ratio was $< 0.5$, we found no simple yet accurate analytic description of the marginal distribution. Unlike when modeling *TELNET* responder bytes, for the duration/originator ratio both the lower and upper parts of the distribution are important; we cannot simply model the upper 80% of the connections and ignore the remainder. So we chose to use models E and F above instead.

## VI. NNTP

### *A. Overview of NNTP connections*

Table IX summarizes *NNTP* connections. As *NNTP* is non-interactive, the connection duration is not of much interest and has been omitted. [24] 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: 1) a server contacts a peer and is informed that the peer presently cannot talk to the server; 2) the server offers the peer news articles but the peer already has the articles; 3) 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

TABLE IX
SUMMARY OF NNTP CONNECTIONS

| Dataset | # Conn | # Rej | % Failures | $\bar{x}_{\text{orig}}$ | $\sigma_{\text{orig}}$ | $\max_{\text{orig}}$ | $\bar{x}_{\text{resp}}$ | $\sigma_{\text{resp}}$ | $\max_{\text{resp}}$ |
|---|---|---|---|---|---|---|---|---|---|
| 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 |
| LBL-7 | 70,842 | 143 | 7% | 41.7KB | ×7.1 | 10.8MB | 1032B | ×2.2 | 2.6MB |
| 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 |

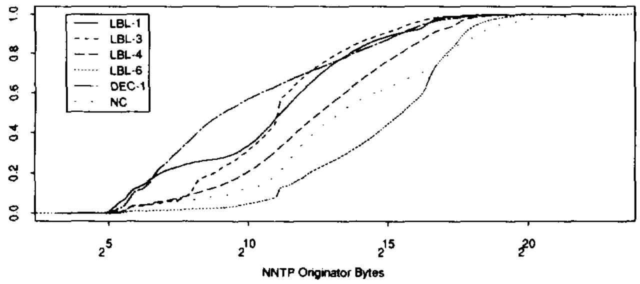

Fig. 4. Distribution of NNTP originator bytes.

the originating peer, it sends a message with response code 400 ("service discontinued") as per [28]. When the originating peer then replies with "QUIT" followed by a carriage-return and a line-feed, it will have sent 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 [8]. 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. 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 nonfailure 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-7 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 is most likely also a reflection of the global growth in USENET *NNTP* traffic, which increases in volume about 75%/year [22].

### *B. NNTP Originator Bytes (G)*

Fig. 4 shows the distributions of bytes sent by the originator in non-failure *NNTP* connections at LBL, DEC, and coNCert. The distributions show a huge degree of variance (recall that the X-axis is scaled logarithmically).

Given the great variation in originator bytes transferred, we decided to simply use a $\log_2$-normal model to describe the connections, with the caveat that we do not expect the model to perform well (we also do not expect empirical models to do well). Indeed, as shown in Table VII, none of the models do well.

One final important point regarding modeling *NNTP* originator bytes is that the distribution is not stationary but changes over the course of a day. Fig. 5 shows the hourly $\bar{x}_{\text{orig}}$ for LBL-1 and LBL-4 non-failure *NNTP* connections. We see considerable but not consistent variation. The peak-to-peak differences for both datasets is about a factor of 3.4; but LBL-1's connections tended to be largest in the middle of the night, with secondary peaks during "prime-time" work hours. LBL-4's connections peaked during working hours and were lowest at precisely the time when LBL-1's were highest.[9]

The variation in the daily pattern may be due to the influence of key *NNTP* gateways either propagating news as soon as it comes in (consistent with the LBL-4 case) or waiting till the late-night hours to take advantage of minimal loads (LBL-1).

### *C. NNTP Responder Bytes*

As seen in Table IX above, there is in general much less variation in the bytes sent by an *NNTP* responder than by the originator. For the majority of the datasets, the responder sent fewer than 1500 bytes in 80% or more of the connections. Thus we decided not to model *NNTP* responder bytes, as in general the datasets do not show interesting variations.

[9] The test datasets also showed a weekly pattern, with LBL-1 and LBL-4 (and to a lesser extent LBL-2) having minimal $\bar{x}_{\text{orig}}$ during weekends, while LBL-3 had a *maximum* $\bar{x}_{\text{orig}}$ on Saturdays.

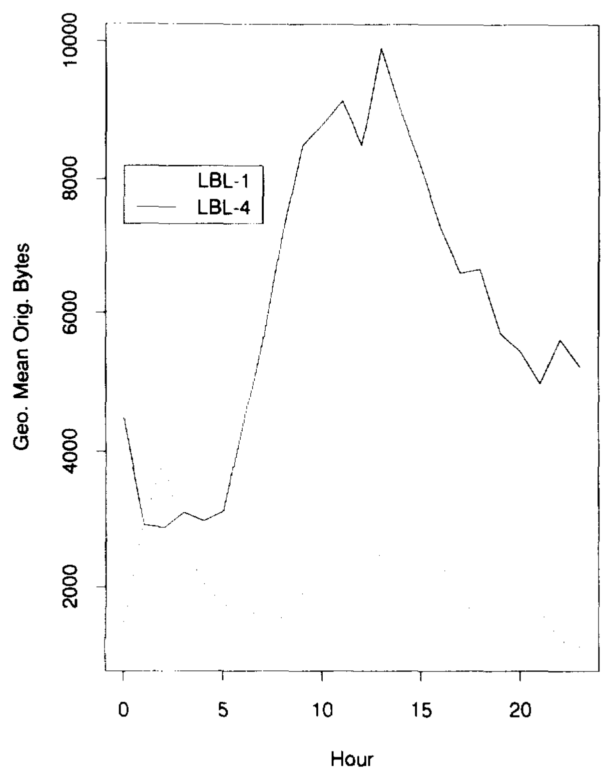

Fig. 5. Daily variation in $\log_2$-mean of LBL NNTP originator bytes.

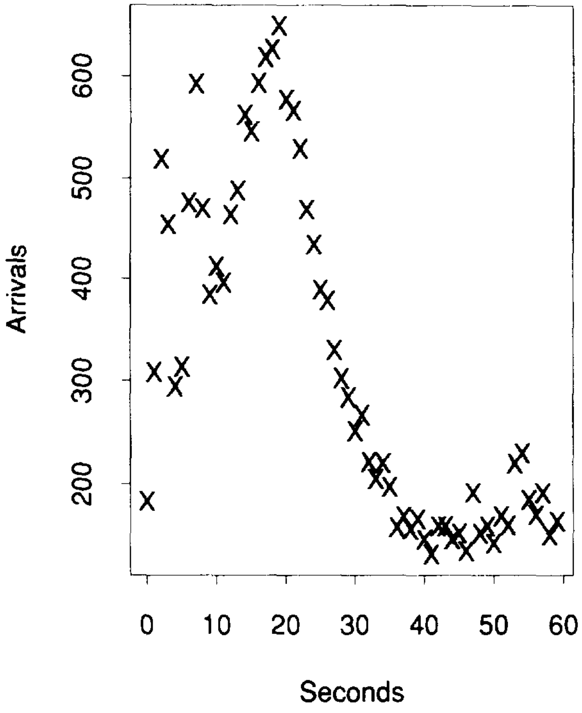

Fig. 6. One-minute variation in DEC-2 NNTP arrivals.

### D. NNTP Duration

Since *NNTP* is a bulk-transfer protocol and not interactive, we do not model connection durations, because these are presumably dominated by networking latencies and not a fundamental aspect of the *NNTP* protocol. Similarly, below we do not model *SMTP* or *FTP* durations.

### E. NNTP Arrival Patterns

*NNTP* arrivals have a definite one-minute periodicity about them.[10] Fig. 6 shows the number of DEC-2 *NNTP* connections that arrived during each second (i.e., ignoring minutes and larger units of the arrival time). Clearly, arrivals tended to show up at about 19 s past the minute, though some tended to arrive about 7 s past. All of the *NNTP* datasets show this pattern to varying degrees except for LBL-3; LBL-4 shows two distinct spikes. Sometimes the spike is quite sharp. With the other datasets, it is broad, like in Fig. 6. In general, periodicity such as this can lead to global synchronization of network processes; see [9].

## VII. SMTP

### A. Overview of SMTP Connections

Table X summarizes the *SMTP* connections. Again, [24] summarizes the reasons for removing the connections marked as rejects. Based on the values for $\max_{\mathrm{orig}}$ it is clear that *SMTP* is sometimes used to transfer quite large files.

There is quite a bit of variation in $\bar{x}_{\mathrm{orig}}$ (and just about none in $\bar{x}_{\mathrm{resp}}$). In [31] the authors note that the UK *SMTP* data show a substantially higher (arithmetic) $\bar{x}_{\mathrm{orig}}$ than for the LBL-1 and LBL-2 datasets reported in [25]. They attribute this difference to the fact that since the U.K. academic network (JANET) was not at that time fully connected to the Internet, U.K. users were more likely to use *SMTP* to transfer files. The large UK $\sigma_{\mathrm{orig}}$ variance supports their hypothesis. The DEC traffic has similar $\sigma_{\mathrm{orig}}$ values, and Mogul also states that an "FTP-by-mail" facility is responsible for about 150 rather lengthy *SMTP* messages at DEC-WRL each day [18].

Another explanation is that perhaps the NC and UK traffic tends to make more *SMTP* "hops," each of which adds a `Received` header to the mail message [27], pushing up the average number of bytes.[11] One would expect the greater number of hops to be correlated with "wider" wide-area traffic, presumably a property of the NC and UK traffic, as these sites are at inter-network gateways.

We see a definite trend in the LBL data indicating larger and larger mail messages. As discussed in [22], LBL's wide-area traffic did become "wider" during the three year period spanned by the LBL datasets, in agreement with the "hops overhead" explanation.

### B. SMTP Originator Bytes (H)

When modeling the number of bytes sent by the *SMTP* originator, we found that nearly all connections transferred more than 300 bytes, while the connections transferring fewer bytes showed sporadic distributions. We hypothesize that the

[10] We also found three-, five-, 15-, and 20-minute patterns in various datasets.

[11] A check of one of the author's mail folders revealed an average `Received` header length of more than 100 bytes.

TABLE X
SUMMARY OF SMTP CONNECTIONS

| Dataset | # Conn | # Rej | $\bar{x}_{\text{orig}}$ | $\sigma_{\text{orig}}$ | $\max_{\text{orig}}$ | $\bar{x}_{\text{resp}}$ | $\sigma_{\text{resp}}$ | $\max_{\text{resp}}$ |
|---|---|---|---|---|---|---|---|---|
| LBL-1 | 38,481 | 286 | 1.4KB | ×2.8 | 2.1MB | 331B | ×1.2 | 1.9KB |
| LBL-2 | 51,240 | 572 | 1.5KB | ×2.9 | 7.2MB | 334B | ×1.2 | 6.5KB |
| LBL-3 | 75,418 | 333 | 1.6KB | ×2.6 | 1.6MB | 334B | ×1.2 | 2.9KB |
| LBL-4 | 92,694 | 1583 | 1.7KB | ×3.0 | 1.2MB | 335B | ×1.3 | 2,980KB |
| LBL-5 | 123,741 | 446 | 1.7KB | ×2.9 | 2.4MB | 320B | ×1.3 | 8.0KB |
| LBL-6 | 207,485 | 6,567 | 1.9KB | ×3.0 | 37.0MB | 321B | ×1.3 | 9.5KB |
| LBL-7 | 205,668 | 6,306 | 1.9KB | ×2.9 | 8.0MB | 314B | ×1.4 | 16.6KB |
| BC | 8,428 | 121 | 1.3KB | ×2.8 | 1.1MB | 324B | ×1.3 | 10.2KB |
| UCB | 16,929 | 61 | 1.3KB | ×3.0 | 0.5MB | 334B | ×1.3 | 2.0KB |
| USC | 3,498 | 3 | 1.4KB | ×2.3 | 0.1MB | 337B | ×1.2 | 1.6KB |
| DEC-1 | 25,160 | 19 | 2.0KB | ×3.1 | 2.5MB | 340B | ×1.2 | 4.7KB |
| DEC-2 | 10,777 | 5 | 2.1KB | ×3.5 | 4.9MB | 341B | ×1.2 | 4.7KB |
| DEC-3 | 31,631 | 70 | 2.0KB | ×3.2 | 5.1MB | 338B | ×1.2 | 3.5KB |
| NC | 26,161 | 511 | 1.9KB | ×2.9 | 1.8MB | 340B | ×1.4 | 10.6KB |
| UK | 10,729 | 129 | 1.9KB | ×3.3 | 4.6MB | 319B | ×1.3 | 6.0KB |

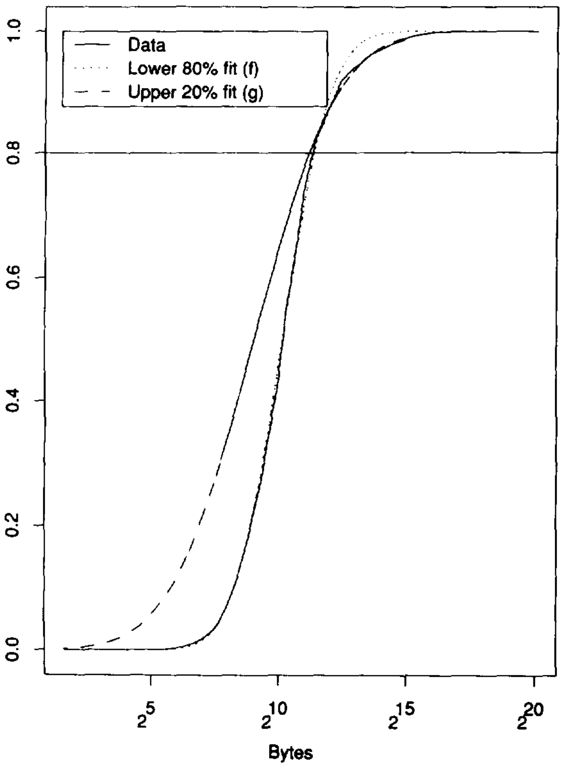

Fig. 7. Bimodal $\log_2$-normal fit to LBL-3 SMTP originator bytes.

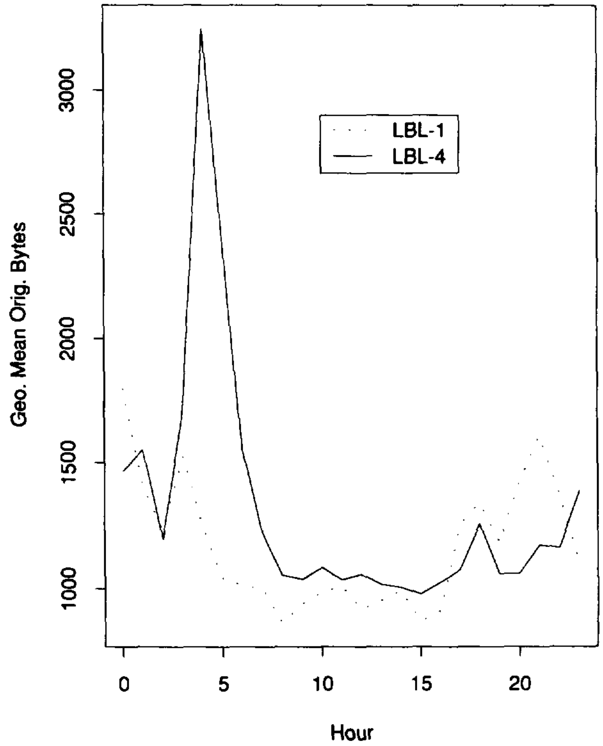

Fig. 8. Daily variation in $\log_2$-mean of LBL SMTP originator bytes.

first 300 bytes of these connections constitute a more-or-less fixed overhead, and that connections with fewer total originator bytes correspond to "failures": either invalid email addresses or busy remote machines unable to accept mail at the moment. In constructing our models we therefore removed any connections of ≤ 300 bytes (anywhere from 0.6 to 2.3% of all connections) and subtracted 300 bytes from the remaining connections.

We found the distribution of *SMTP* originator bytes to be bimodal, not surprisingly given that *SMTP* is sometimes used to transfer files. We model the distribution using two $\log_2$-normal distributions, one (called $f$ here) for the lower 80% of the data, and one for the remaining 20% ($g$). Fig. 7 shows this model's fit to the LBL-3 test data after removing failures and subtracting 300 bytes; the horizontal line indicates the dividing line between using distribution $f$ (below the line) and $g$ (above).

As was the case for *NNTP*, for *SMTP* we found that the originator bytes distribution is not stationary. Fig. 8 shows the hourly $\bar{x}_{\text{orig}}$ for LBL-1 and LBL-4 *SMTP* connections. Unlike *NNTP*, which suffered from inconsistent variations, here the pattern is more stable: connection sizes peak during off-hours, and reach minima during peak working hours. We conjecture that uses of *SMTP* to transfer files tend to happen off-hours and cause this pattern.

TABLE XI
SUMMARY OF FTP DATA CONNECTIONS

| Dataset | # Conn | # Rej | Get | $\bar{x}_{\text{bytes}}$ | $\sigma_{\text{bytes}}$ | $\max_{\text{bytes}}$ |
|---|---|---|---|---|---|---|
| LBL-1 | 23,555 | 287 | 80% | 2.3KB | ×15.3 | 54MB |
| LBL-2 | 27,917 | 335 | 92% | 2.4KB | ×17.4 | 124MB |
| LBL-3 | 39,552 | 349 | 91% | 3.3KB | ×17.7 | 62MB |
| LBL-4 | 65,860 | 335 | 86% | 3.8KB | ×14.7 | 67MB |
| LBL-5 | 66,411 | 344 | 80% | 4.5KB | ×16.0 | 177MB |
| LBL-6 | 86,464 | 464 | 91% | 2.1KB | ×14.9 | 292MB |
| LBL-7 | 105,821 | 468 | 94% | 2.8KB | ×15.2 | 223MB |
| BC | 5,199 | 58 | 97% | 2.5KB | ×12.6 | 16MB |
| UCB | 4,529 | 77 | 96% | 1.0KB | ×13.5 | 22MB |
| USC | 1,870 | 29 | 93% | 1.3KB | ×14.5 | 5MB |
| DEC-1 | 7,970 | 6 | 100% | 2.2KB | ×16.5 | 5MB |
| DEC-2 | 4,013 | 13 | 100% | 1.3KB | ×17.1 | 7MB |
| DEC-3 | 6,775 | 25 | 99% | 1.9KB | ×16.7 | 13MB |
| NC | 19,076 | 183 | 98% | 1.8KB | ×19.0 | 44MB |
| UK | 10,018 | 58 | 97% | 3.4KB | ×14.2 | 7MB |

TABLE XII
SUMMARY OF FTP CONTROL CONNECTIONS

| Dataset | # Conn | # Rej | 0 xfer | $\bar{x}_{\text{xfers}}$ | $\sigma_{\text{xfers}}$ | $\max_{\text{xfers}}$ | $\bar{x}_{\text{bytes}}$ | $\sigma_{\text{bytes}}$ |
|---|---|---|---|---|---|---|---|---|
| LBL-1 | 3,757 | 51 | 19% | 3.3 | ×2.9 | 1,006 | 28KB | ×15.2 |
| LBL-2 | 5,312 | 72 | 25% | 3.2 | ×2.8 | 388 | 27KB | ×17.0 |
| LBL-3 | 6,916 | 90 | 21% | 3.1 | ×2.9 | 612 | 30KB | ×18.4 |
| LBL-4 | 7,941 | 189 | 17% | 3.3 | ×3.0 | 1,951 | 33KB | ×17.6 |
| LBL-5 | 9,968 | 1,227 | 26% | 3.0 | ×3.0 | 975 | 31KB | ×16.7 |
| LBL-6 | 12,470 | 535 | 24% | 3.1 | ×2.9 | 2,996 | 31KB | ×16.8 |
| LBL-7 | 17,556 | 319 | 27% | 3.0 | ×2.9 | 1,666 | 34KB | ×16.8 |
| BC | 669 | 19 | 32% | 3.3 | ×2.7 | 426 | 13KB | ×14.2 |
| UCB | 756 | 19 | 26% | 3.9 | ×2.6 | 350 | 12KB | ×14.9 |
| USC | 272 | 6 | 22% | 3.8 | ×2.8 | 133 | 20KB | ×14.5 |
| DEC-1 | 727 | 8 | 26% | 5.4 | ×3.2 | 961 | 36KB | ×15.6 |
| DEC-2 | 491 | 8 | 13% | 5.0 | ×3.0 | 106 | 36KB | ×17.8 |
| DEC-3 | 811 | 17 | 25% | 4.8 | ×2.9 | 232 | 36KB | ×15.3 |
| NC | 2,500 | 59 | 31% | 5.0 | ×2.9 | 392 | 26KB | ×18.6 |
| UK | 1,733 | 35 | 24% | 3.4 | ×3.0 | 368 | 22KB | ×16.0 |

### *C. SMTP Responder Bytes*

We did not model the distribution of the responder bytes in *SMTP* connections, as the responder's role shows little variation. For the LBL test datasets, in about 75% of all connections the responder sent between 300 and 400 bytes, and in every dataset more than 97% of the connections sent between 100 and 1000 bytes. While reference [8] finds that *SMTP* connections are bidirectional, this finding must be interpreted with the rather fixed nature of the *SMTP* responder in mind.

## VIII. FTP

### *A. Overview of FTP Connections*

Table XI summarizes *FTPdata* connections. Each connection is unidirectional, with sometimes data flowing from the connection originator to the responder (corresponding to an *FTP* `get` command) and sometimes in the other direction (a `put` command). The "Get" column shows the percentage of connections that were `get` commands; the remainder were `put` commands. The next three columns show the (geometric) mean, standard deviation, and maximum for the number of bytes transferred. As before, [24] gives details regarding the connections we rejected.

A considerable portion of the UCB, LBL-5, LBL-6, and LBL-7 *FTP* connections were due to periodic traffic, as discussed in [22]. As with periodic *TELNET* connections, we eliminated these prior to analysis.

There clearly is quite a range in $\bar{x}_{\text{bytes}}$, and the uniformly large values of $\sigma_{\text{bytes}}$ shows that in general file sizes vary widely.

Table XII summarizes the *FTPctrl* connections. We have not shown statistics for bytes transferred and duration of the *FTPctrl* connections themselves since the primary use of *FTPctrl* connections is to spawn *FTPdata* connections, either for file transfer or to list remote directories. Instead, we grouped with each *FTPctrl* connection its associated *FTPdata* connections. We considered an *FTPdata* connection to belong to a *FTPctrl* connection if it occurred during the span of the *FTPctrl* connection and was between the same two hosts (see [24] for details). We refer to such a collection of an *FTPctrl* connection and its associated *FTPdata* connections as an *FTP session*.

TABLE XIII
SUMMARY OF FTP BURSTS

| Dataset | # Bursts | $\bar{x}_{\rm bytes}$ | $\sigma_{\rm bytes}$ | $\max_{\rm bytes}$ | 2% Tail | 0.5% Tail | $\alpha$ | $k$ |
|---|---|---|---|---|---|---|---|---|
| LBL-1 | 13,055 | 2.0KB | ×17.6 | 102MB | 70% | 49% | 1.00 | 375KB |
| LBL-2 | 16,111 | 1.8KB | ×19.2 | 124MB | 70% | 44% | 1.01 | 445KB |
| LBL-3 | 22,388 | 2.3KB | ×19.0 | 96MB | 68% | 41% | 1.03 | 624KB |
| LBL-4 | 27,084 | 2.5KB | ×18.8 | 83MB | 68% | 41% | 1.05 | 717KB |
| LBL-5 | 30,358 | 2.8KB | ×19.1 | 754MB | 79% | 61% | 0.93 | 620KB |
| LBL-6 | 39,740 | 2.7KB | ×16.7 | 465MB | 77% | 58% | 1.06 | 595KB |
| LBL-7 | 48,542 | 2.8KB | ×17.1 | 200MB | 67% | 44% | 1.16 | 748KB |
| BC | 2,077 | 1.4KB | ×13.9 | 16MB | 64% | 41% | 1.21 | 290KB |
| UCB | 2,804 | 0.9KB | ×13.2 | 12MB | 66% | 42% | 1.11 | 148KB |
| USC | 830 | 1.1KB | ×19.1 | 5MB | 53% | 31% | 1.37 | 315KB |
| DEC-1 | 4,487 | 1.3KB | ×13.9 | 32MB | 70% | 48% | 1.06 | 248KB |
| DEC-2 | 2,743 | 1.3KB | ×13.5 | 12MB | 66% | 37% | 1.18 | 374KB |
| DEC-3 | 4,276 | 1.3KB | ×13.2 | 15MB | 67% | 45% | 1.09 | 244KB |
| NC | 13,086 | 1.3KB | ×17.7 | 44MB | 57% | 37% | 1.34 | 308KB |
| UK | 5,837 | 1.9KB | ×14.6 | 8MB | 54% | 29% | 1.40 | 375KB |

The fourth through seventh columns in Table XII summarize the number of *FTPdata* connections that occurred during each *FTP* session. The "0 xfer" column lists the percentage of all *FTP* sessions that did not have *any* associated *FTPdata* connections, presumably due to failed attempts to provide login information. These numbers are somewhat lower than the 42.9% reported in [6], but still surprisingly high.

The $\bar{x}_{\rm xfers}$ and $\sigma_{\rm xfers}$ columns give the geometric mean and standard deviation for the number of files transferred, given that at least one file was transferred. That the mean is substantially higher than one is not surprising since we classify remote directory listings as file transfers (both result in an *FTPdata* connection), and probably the most common use of *FTP* is to connect to a remote archive site, do several listings to find the file or files of interest, and then transfer those files.

The $\bar{x}_{\rm bytes}$ and $\sigma_{\rm bytes}$ columns show the geometric mean and standard deviation for the total number of bytes transferred via *FTPdata* connections during each *FTP* session (for those connections with at least one *FTPdata* transfer). We note that these means are 5-10 times greater than those for individual *FTPdata* connections, an increase larger than that due simply to the multiplying effect of $\bar{x}_{\rm xfers}$. We suspect that this disparity is due to a typical *FTP* session including at least one true file transfer. As files will tend to be significantly larger than directory listings, the mean number of transferred bytes during an *FTP* session will approach the mean file size, and not be held down, as are the *FTPdata* connection summaries, by a large number of smaller directory listings. The $\sigma_{\rm bytes}$ values are quite large, again showing a wide range in transfer sizes.

One important point regarding the *FTPdata* connections occurring during an *FTP* session is that they frequently arrive in *bursts* [23]. An *FTPdata* connection burst is defined as one or more *FTPdata* connections belonging to the same session that are spaced less than 4 s apart. That is, each connection in the burst begins less than 4 s after the previous connection ended. A burst can be due to a "multiple-get" transfer, or to a user doing a remote directory listing and shortly after it completes, fetching a file. A key finding in [23] regarding bursts is that the distribution of the number of bytes in a burst has an extremely heavy tail: just a handful of the largest bursts carry the majority of *all* of the *FTP* data bytes.

Table XIII summarizes the *FTPdata* bursts. We see from the second column that each dataset had roughly half as many bursts as *FTPdata* connections (Table XI). The third through fifth columns summarize the number of bytes transferred per burst of *FTPdata* connections. The values of $\bar{x}_{\rm bytes}$ are surprising—they are lower than the corresponding values for *FTPdata* connections! This appears paradoxical, because each *FTPdata* connection burst is made up of at least one *FTPdata* connection, so we would expect the bursts on average to be at least as large as the individual connections. The key to understanding this discrepancy is that *large FTPdata connections tend to arrive together in single bursts*. This means that the upper tail of the distribution of the number of bytes per burst is heavier than the corresponding upper tail for *FTPdata* connections; there are fewer big bursts, but those few are very large. Because $\bar{x}_{\rm bytes}$ is a *geometric* mean (6), and the geometric mean is relatively insensitive to outliers, we find $\bar{x}_{\rm bytes}$ becomes *lower* when we shift distribution weight higher into the upper tail.

The next two columns explore the tail-weight further. The "2% Tail" column gives the percentage of all *FTP* burst data bytes due to the 2% largest bursts, and similarly for the "0.5% Tail" column. We see that the 2% upper tail in all cases accounts for more than 50% of all of the data bytes! Thus, *FTP* traffic is heavily dominated by a few rare but huge bursts. As stated in [23], this finding means that modeling *FTP* traffic should concentrate heavily on the upper tails of *FTPdata* bursts [as is done in Section VIII–D)]. Note that *FTPdata* connections alone do *not* have nearly as heavy a tail. For example, in the DEC-1 dataset the upper 2% of the *FTPdata* connections holds about 25% of the data bytes, versus 70% for the bursts. It is the fact that large *FTPdata* connections tend to arrive together that leads to the very heavy *FTP* burst tail.

In Section VIII–D), we model the upper 5% tail of the *FTPdata* burst distribution using a Pareto distribution (2). The Pareto distribution has an extremely heavy tail, heavier than that of any of the other distributions discussed in Section III–A3) or their logarithmic versions except for log-extreme

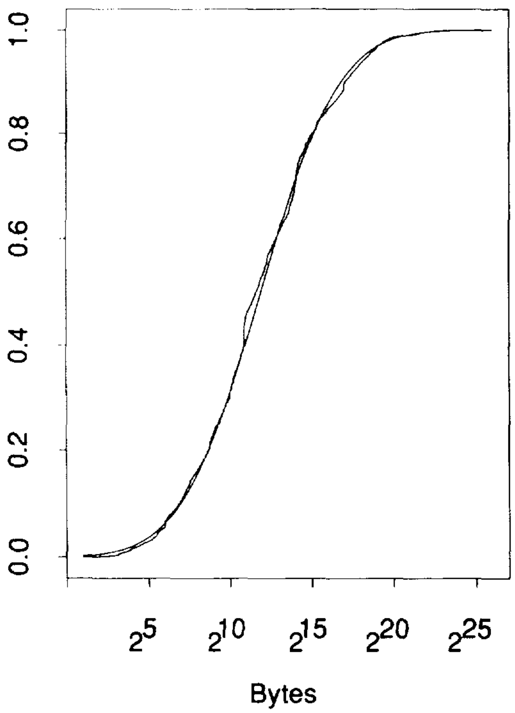

Fig. 9. Log$_2$-normal fit to LBL-4 FTP data bytes.

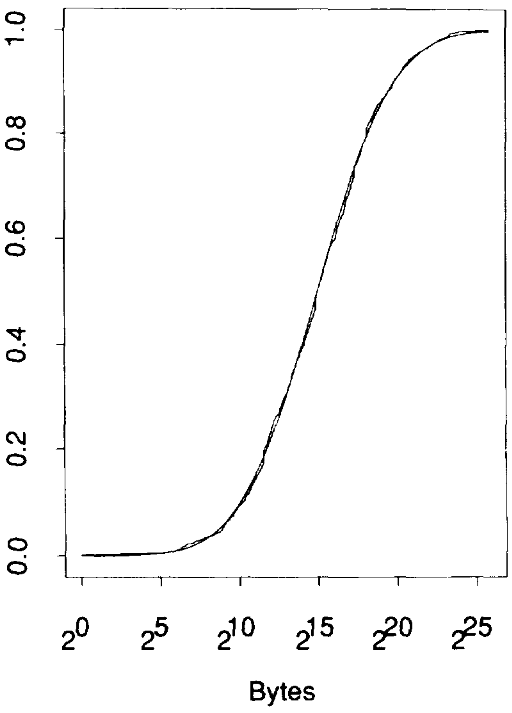

Fig. 10. Log$_2$-normal fit to bytes in LBL-1 FTP sessions.

[Section III–B)]. To this end, the final columns in Table XIII gives the estimated values of $\alpha$ and $k$ [corresponding to (2)] for each dataset. That the values of $\alpha$ are smaller for the LBL datasets than for the others, and the values for $k$ larger, indicates that the LBL *FTP* bursts are significantly different than those found in the other datasets. In particular, the LBL bursts have heavier tails. Probably this difference is due to the prevalence of LBL users exchanging large scientific datasets.

### *B. FTP Connection Bytes (I)*

We model the bytes transferred during an *FTPdata* connection using a log$_2$-normal distribution. Fig. 9 shows this model fitted to the first half of the LBL-4 dataset. While the model appears to match the overall shape, a number of clumps and spikes make the actual distribution irregular. For example, LBL-4 has a spike of 1269 connections, each transferring 1856 bytes. For the most part, unfortunately, these spikes do not occur in predictable locations, making it difficult to incorporate them into our analytic model. Such unpredictability also impairs the ability of empirical models to fit other datasets. One spike stands out, however, being present in all the DEC datasets, the NC dataset, LBL-4, LBL-5, and LBL-7. This spike occurs at 524 288 bytes ($= 2^{19}$), a size often used when splitting a large distribution archive into manageable pieces.

### *C. FTP Session Bytes (J)*

Fig. 10 shows an example of the distribution of the total number of bytes transferred during *FTP* sessions, for the LBL-1 test dataset, again fitted to a log$_2$-normal model. In this case the fit is visually fairly satisfying.

The authors of [8] reported that 80% of *FTP* sessions transfer less than 10 KB. But once we remove the 20-30% of sessions that do not transfer any data, half of the remainder transfer more than 32 KB, and a sixth transfer more than 500 KB. Thus if a file transfer session is not a "failure," it should not be assumed small.

### *D. FTP Bursts (K)*

As mentioned in Section VIII–A), modeling *FTPdata* bursts is particularly important. Not only does the distribution of bytes per burst have an extremely large tail, but because the interconnection spacing within a burst is (by definition) $< 4$ seconds, from a link-level (or queueing) viewpoint, there is little difference between a burst of *FTPdata* connections and a single large connection transferring the same total number of bytes.

Because the upper tail of this distribution is so dominant, we decided to concentrate on modeling the size of the largest 5% of the bursts. We found that upper-tail burst size is well-modeled using a Pareto distribution [doubly-exponential; (2)]. As discussed further in [23], that *FTP* burst sizes are Pareto-distributed suggests a mechanism by which *FTP* traffic might contribute to the presence of *self-similarity* ([14]) in wide-area network traffic.

## IX. Summary

We have presented a number of analytic models for describing the characteristics of *TELNET*, *NNTP*, *SMTP*, and *FTP* connections, drawn from wide-area traces collected from seven different sites, comprising more than 3 million connections. While these models are inexact in a statistical sense, we developed a methodology for comparing their effectiveness to that of other models. We found that in general the analytic models reflect the connection characteristics as well as or better than two empirical models, one corresponding to the Tcplib library [7] and one corresponding to a one-month trace of traffic at the Lawrence Berkeley Laboratory. We also found that wide-area connection characteristics exhibit significant variation from site to site and over time.

The essence of the argument presented in this paper is that while wide-area traffic cannot be modeled exactly in a statistical sense, we can usually construct simple analytic models that are a good approximation. Furthermore, these analytic models are as accurate as empirical models, meaning we can reap the benefits of using analytic models without losing accuracy in the process. We believe the methodology presented in this paper will prove beneficial for developing future analytic models and for gauging their effectiveness.

## Acknowledgment

This work would not have been possible without the support, guidance, and patience of V. Jacobson and D. Ferrari. The author is also much indebted to P. Danzig and his coauthors for the UCB, USC, and BC datasets; J. Mogul for the DEC datasets; I. Wakeman and J. Crowcroft for the UK dataset; W. Sung for the NC dataset; and especially to R. Cáceres and S. Jamin, who between them made all of the non-LBL datasets available.

The LBL traces were gathered with the help of C. Leres and S. McCanne. C. Leres was also helpful in understanding *NNTP*-related phenomena. The Bellcore traces were gathered by D. V. Wilson.

The author would also like to thank T. Speed and particularly S. Floyd for valuable discussions on both modeling the data and presenting the results, and K. Raatikainen, J. Crowcroft, and the referees for many helpful comments on earlier drafts of this paper.

## References

[1] G. E. Bryan, "JOSS: 20,000 hours at a console," in *Proc. Fall 1967 AFIPS Conf.*, vol. 31.

[2] R. Cáceres, "Measurements of wide area internet traffic," Comput. Sci. Div., Univ. of California, Berkeley, CA, Rep. UCB/CSD 89/550, 1989.

[3] W. Cook and D. Mumme, "Estimation of Pareto parameters by numerical methods," in *Statistical Distributions for Scientific Work*, C. Taillie *et al.* Eds. Boston, MA: D. Reidel, 1980, pp. 127-132.

[4] J. Crowcroft and I. Wakeman, "Traffic analysis of some UK-US academic network data," in *Proc. INET'91*, Copenhagen, June, 1991.

[5] R. B. D'Agostino and M. A. Stephens, Eds., *Goodness-of-Fit Techniques.* New York: Marcel Dekker, 1986.

[6] P. Danzig, R. Hall, and M. Schwartz, "A case for caching file objects inside internetworks," in *Proc. SIGCOMM '93*, San Francisco, CA, Sept. 1993.

[7] P. Danzig and S. Jamin, "Tcplib: A library of tcp internetwork traffic characteristics," Comput. Sci. Dep., Univ. of Southern California, Rep. CS-SYS-91-01, 1991.

[8] P. Danzig, S. Jamin, R. Cáceres, D. Mitzel, and D. Estrin, "An empirical workload model for driving wide-area TCP/IP network simulations," *Internetworking: Res. Exper.*, vol. 3, no. 1, pp. 1-26, 1992.

[9] S. Floyd and V. Jacobson, "The synchronization of periodic routing messages," in *Proc. SIGCOMM '93*, Sept. 1993, pp. 33-44. Also in *Trans. Networking*, vol. 2, pp. 122–136, Apr. 1994.

[10] E. Fuchs and P. E. Jackson, "Estimates of distributions of random variables for certain computer communications traffic models," *Commun. ACM*, vol. 13, no. 12, pp. 752-757, Dec. 1970.

[11] S. Heimlich, "Traffic characterization of the NSFNET national backbone," in *Proc. 1990 Winter USENIX Conf.*, Washington, DC.

[12] P. E. Jackson and C. D. Stubbs, "A study of multiaccess computer communications," in *Proc. Spring 1969 AFIPS Conf.*, vol. 34, 1969.

[13] V. Jacobson, C. Leres, and S. McCanne, *tcpdump*, available via anonymous FTP to ftp.ee.lbl.gov, June, 1989.

[14] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, "On the self-similar nature of Ethernet traffic," in *Proc. SIGCOMM '93*, pp. 183-193, Sept. 1993.

[15] W. T. Marshall and S. P. Morgan, "Statistics of mixed data traffic on a local area network," *Comput. Networks and ISDN Syst.*, vol. 10, nos. 3, 4, pp. 185-194, 1985.

[16] P. Martin-Löf, "The notion of redundancy and its use as a quantitative measure of the discrepancy between a statistical hypothesis and a set of observational data," *Scandinavian J. Statist.*, vol. 1, no. 1, pp. 3-18, 1974.

[17] S. McCanne and V. Jacobson, "The BSD packet filter: A new architecture for user-level packet capture," in *Proc. 1993 Winter USENIX Conf.*, San Diego, CA, 1993.

[18] J. C. Mogul, "Observing TCP dynamics in real networks," in *Proc. SIGCOMM '92*, Baltimore, MD, Aug. 1992.

[19] D. Moore, "Measures of lack of fit from tests of chi-squared type," *J. Statist. Planning and Inference*, vol. 10, no. 2, pp. 151-166, 1984.

[20] F. Mosteller and J. W. Tukey, *Data Analysis and Regression.* Reading, PA: Addison-Wesley, 1977.

[21] P. Pawlita, "Two decades of data traffic measurements: A survey of published results, experiences and applicability," in *Teletraffic Science for New Cost-Effective Systems, Networks and Services, ITC-12*, M. Bonatti, Ed. New York: Elsevier Science, 1989.

[22] V. Paxson, "Growth trends in wide-area TCP connections," in *IEEE Network*, vol. 8, pp. 8–17, July/Aug. 1994.

[23] V. Paxson and S. Floyd, "Wide-area traffic: The failure of Poisson modeling," in *Proc. SIGCOMM '94*, London, England, 1994..

[24] V. Paxson, "Empirically-derived analytic models of wide-area TCP connections: Extended report," Lawrence Berkeley Lab., Rep. LBL-34086, May, 1993. Available as papers/ WAN-TCP-models.prelim.1.ps.Z and papers/WAN-TCP-models.prelim.2.ps.Z via anonymous FTP to ftp.ee.lbl.gov.

[25] ——, "Measurements and models of wide area TCP conversations," Rep. LBL-30840, Lawrence Berkeley Lab., 1991.

[26] S. Pederson and M. Johnson, "Estimating model discrepancy," *Technometrics*, vol. 32, no. 3, pp. 305-314, Aug. 1990.

[27] D. Crocker, "Standard for the format of ARPA internet text messages," RFC 822, Network Inform. Cen., SRI Int., Menlo Park, CA, 1982.

[28] B. Kantor and P. Lapsley, "Network news transfer protocol," RFC 977, Network Inform. Cen., SRI Int., Menlo Park, CA, 1986.

[29] A. Schmidt and R. Campbell, "Internet protocol traffic analysis with applications for ATM switch design," *Comput. Commun. Rev.*, vol. 23, no. 2, pp. 39-52, Apr. 1993,

[30] D. Scott, "On optimal and data-based histograms," *Biometrika*, vol. 66, no. 3, pp. 605-610, 1979.

[31] I. Wakeman, D. Lewis, and J. Crowcroft, "Traffic analysis of transatlantic traffic," in *Proc. INET'92*, Kyoto, Japan, 1992.

**Vern Paxson** received the M.S. degree from the University of California at Berkeley.

He is a Staff Scientist at the Lawrence Berkeley Laboratory, where he is a member of the Network Research Group, and a doctoral student at UCB under the direction of Prof. D. Ferrari. His main research interests are wide-area networking and software buses for distributed systems.