

FTP Traffic Generator

Joseph Ishac
jai5@po.cwru.edu

Department of Electrical Engineering and Computer Science
Case Western Reserve University

Technical Report

January 10, 2001

Abstract

Network simulation is an important tool used to increase the understanding of network services and protocols. A key to conducting compelling simulation studies is to use a realistic traffic pattern. However, many generators oversimplify traffic patterns. This paper presents a tool that has been created to capture the main characteristics of the file transfer protocol (FTP) and generate sessions based on those characteristics.

1 Introduction

The FTP Application-Level Traffic Generator (ALTGen-F) is a tool that can be used to describe realistic FTP traffic across network paths. ALTGen-F is a stand-alone tool that generates a traffic profile, which can be used in conjunction with various applications such as simulators or measurement tools used in real networks.

The traffic profile produced by ALTGen-F is based on mathematical models which incorporate both observations seen across real networks and the reasoning behind those observations [Pax94]. In comparison, empirical models are inflexible and rehash what was seen without endeavoring into the reasons behind the data. Also, simple mathematical models tend to fail to capture any complexity within the data.

In the area of network simulation, the process of generating traffic is often overlooked [PF97]. Often, traffic consists of a stream of uniform data or is generated from an oversimplified model. Such representations fail to capture the key characteristics of the protocols. This generator was created in order to produce traffic that better models the main characteristics and traits of the FTP protocol.

2 FTP Traffic Generation

2.1 Terminology and details

The following terms are explained below and are used throughout this report:

Flow: A network path which hosts a set of FTP sessions.

Flows are independent of each other.

Session: All occurrences within a specific invocation of a FTP application.

Connection: A single data transfer within a session. All connections utilize the Transmission Control Protocol (TCP).

Inter-connection time: Time between the end of one data connection and the start of the next within the same session.

Delay between connections: Same as Inter-connection time.

Burst: A series of connections whose inter-connection times are at most 4 seconds [Pax94].

Load: The number of sessions to generate.

Using this vocabulary set, an example FTP session is shown in Figure 1 below.

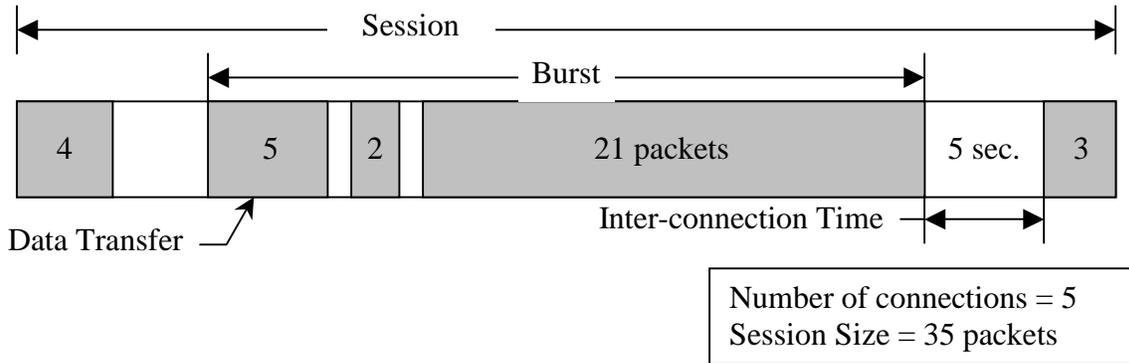


Figure 1: Example FTP session

2.2 Distributions

Aside from the standard distributions, several other distributions are used as follows:

Pareto Distribution

A skewed, heavy-tailed distribution whose distribution function is defined as

$$F(x) = 1 - (k/x)^\alpha, \quad (1)$$

where α is the shape of the distribution and k is a scaling factor.

Log₂-Normal Distribution

A log₂-normal distribution refers to a random variable which follows a normal distribution after first applying a logarithmic transformation (in this case log base 2). Thus if random variable $Y = \log_2(X) \sim \text{Normal}(\mu, \sigma)$, then $X \sim \text{Log}_2\text{-Normal}(\mu, \sigma)$.

2.3 Algorithm

The following algorithm describes the steps used to generate a FTP traffic profile:

- I) Preprocessing
 - Preprocessing validates the program's parameters and writes a small header to the traffic profile.
- II) Generate the Traffic
 - 1) Flow Identification
 - Flow identification is used to identify which sessions belong to which flow. Flows range in ID from zero to one less than the total number of flows. The ID is incremented for each session and follows the following formula in general:

For N flows,
 session[k] corresponds to flow[$k \bmod N$]
 where $k=0, 1, 2, \dots, (\text{Load} - 1)$

- 2) Session Start Time Calculation
 - No session will start at time zero, unless the initial calculation yields an initial increment of zero. The time between successive session starts for a particular flow is based on a Poisson distribution with a user-specified mean. A smaller mean will yield more condensed traffic, while a larger value will yield more

sparse traffic. [PF95] shows that the time between FTP sessions is well modeled as a Poisson distribution.

$$\begin{aligned} \text{time}_i(\text{Session}[k]) &= \text{time}_i(\text{flow}[k \bmod N]) \\ &= \text{time}_{i-1}(\text{flow}[k \bmod N]) + \text{Poisson}(\text{mean start time}) \end{aligned}$$

3) Session Size Calculation

The current session size in bytes is calculated using a Log_2 -Normal distribution [Pax94]¹. Thus,

$$\text{Session Size} = 2^R, \text{ where } R \sim \text{Normal}(\mu_{\text{session}}, \sigma_{\text{session}})$$

4) Generation of Connections within a Session

Any given connection is defined by a Log_2 -Normal distribution.

$$\text{Connection, } C_i = 2^R, \text{ where } R \sim \text{Normal}(\mu_{\text{connection}}, \sigma_{\text{connection}})$$

A connection, C_i , is calculated, and its size in bytes is added to the cumulative sum of all previous connections for that session. If the total is less than the size of the session, this step is repeated. If it is not, the connection is truncated so that the total number of bytes in the connections is equal to the session size. This introduces a slight bias to the connection size calculation. While the shape of the generated distribution remains the same, the average is shifted downward slightly as a result of the truncations. Also, due to memory considerations for both the simulator and generator, the total number of connections may be restrained and an appropriate warning message displayed.²

5) Calculation of Bursts

The calculation of bursts is a complicated procedure that relies on two different distributions. First, a Log_2 -Normal distribution is used for the top 95 percentile. The remaining bursts are characterized by a Pareto distribution. The procedure (see Figure 2) goes as follows:

For every connection in the session (in order of occurrence) :

If the current connection is the first connection or the previous connection is marked as non-burst, calculate a new burst size, B , based off of the two part distribution. If the difference between the current connection size and the burst size is greater than or equal to zero, then the current connection is marked as burst, and the difference between the current connection size and the burst size becomes the new B . Move to the next connection and repeat.

¹ Values for the parameters in both the session, connection, and burst size calculations are those used in [Pax94]. However, they are easily changeable within the code if adjustment to the model becomes necessary.

² The restraint is specified within the program, but can be modified if necessary.

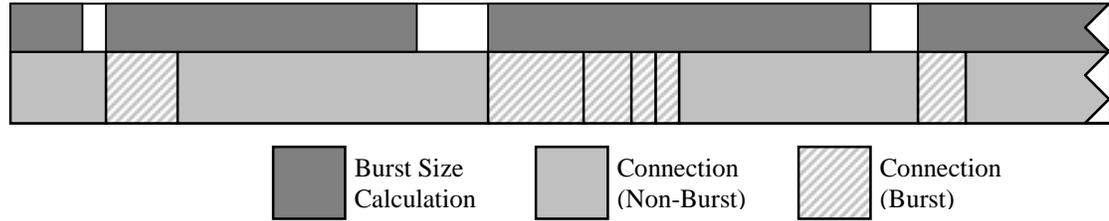


Figure 2: Example of burst calculation

6) Calculation of Delays between Connections

The calculation of delay between data connections contains models not discussed in [Pax94] and is discussed in greater detail in § 3.

A burst by definition is a series of connections separated by less than four seconds [Pax94]. Thus, for any two consecutive connections within a burst a time between zero and four is generated. For any other combination of connections, a time greater than four is generated. The model used in generating these values consists of three different distributions - Log_2 -Normal, exponential, and uniform - and are listed in the table below. Finally, if a delay is observed to be zero, it is adjusted by one nanosecond since zero delays cannot exist in many simulators.

Bursts	0 - 2.5 seconds	Exponential (0.616)
	2.5 - 4 seconds	Uniform (2.5, 4)
Non-Bursts	4 - 180 seconds	Log_2 -Normal (3.27, 2.16)

7) Generate the Report for the Session

A report for the session is generated, where the start time is an offset in seconds from the start of the simulation and not the time between sessions. All sends are the number of packets to transmit, which correspond to the number of bytes that were calculated previously and the packet size used in the simulation. Finally, all delays are in seconds. The format of the report is shown below.

Line/Session Number	Flow ID	Start Time	Initial Send	[Subsequent Sends ...]	[Delays ...]	Number of Connections
---------------------	---------	------------	--------------	--------------------------	----------------	-----------------------

Figure 3: Report format

8) Reset sizes and Repeat

Step two is repeated for the desired number of sessions.

3 Modeling Delays Between Connections

One of the characteristics absent from the models in [Pax94] was a detailed analysis of the time between connections within a session. Thus, independent analysis was done using the same LBL-7 dataset used in [Pax94]. Since the times extracted from the data set were extremely skewed, a base two logarithmic transformation was applied to the data. The result (see Figure 4) showed a distinct bi-modularity to the data, with the split occurring roughly around 2.5 seconds. The data above this split fit very well to a normal distribution and resulted in an overall Log_2 -Normal distribution. The data below the split was less behaved and seemed dependent on factors such as round trip times and process scheduling.

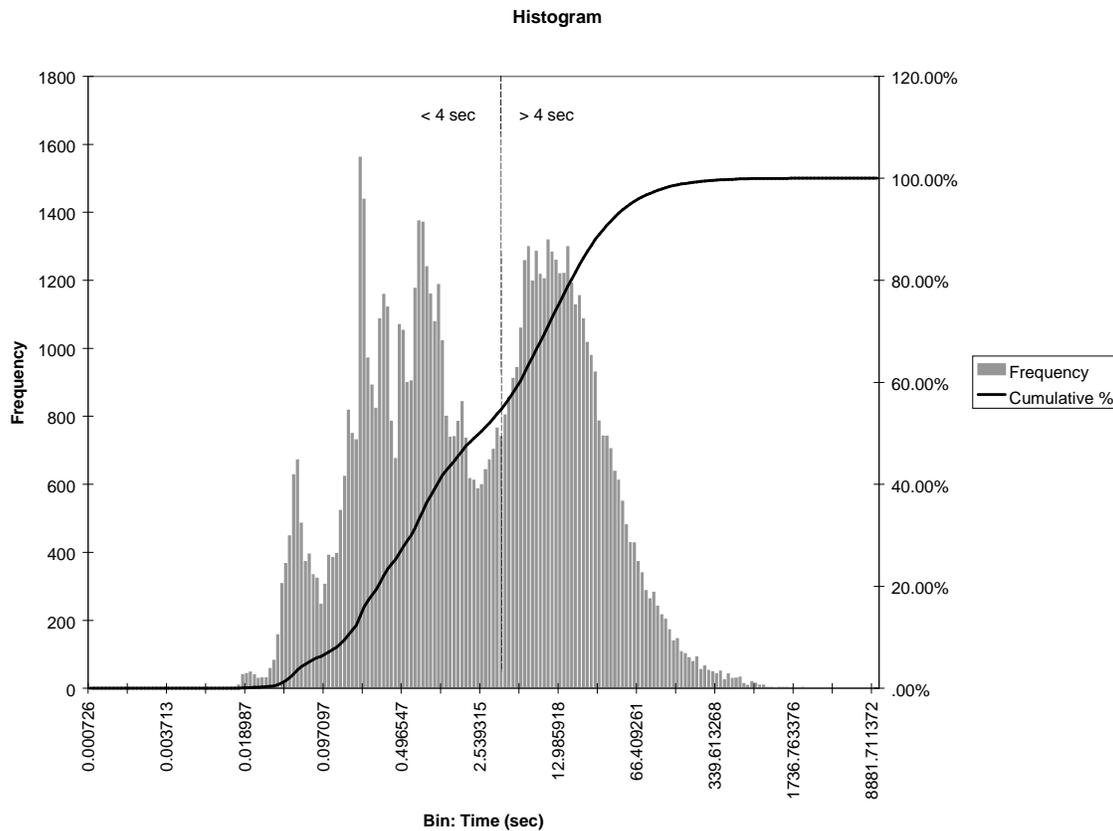


Figure 4: Time between connections after log base 2 transformation (LBL-7 Dataset)

After some analysis a model consisting of two parts was derived using the data before the transformation. An exponential model was used for data less than 2.5 seconds, and a uniform model was used for data between 2.5 and 4 seconds. A simulation was then run and the results of the simulation were plotted against the cumulative distribution function (CDF) of the original data set, as shown in Figure 5 and Figure 6.

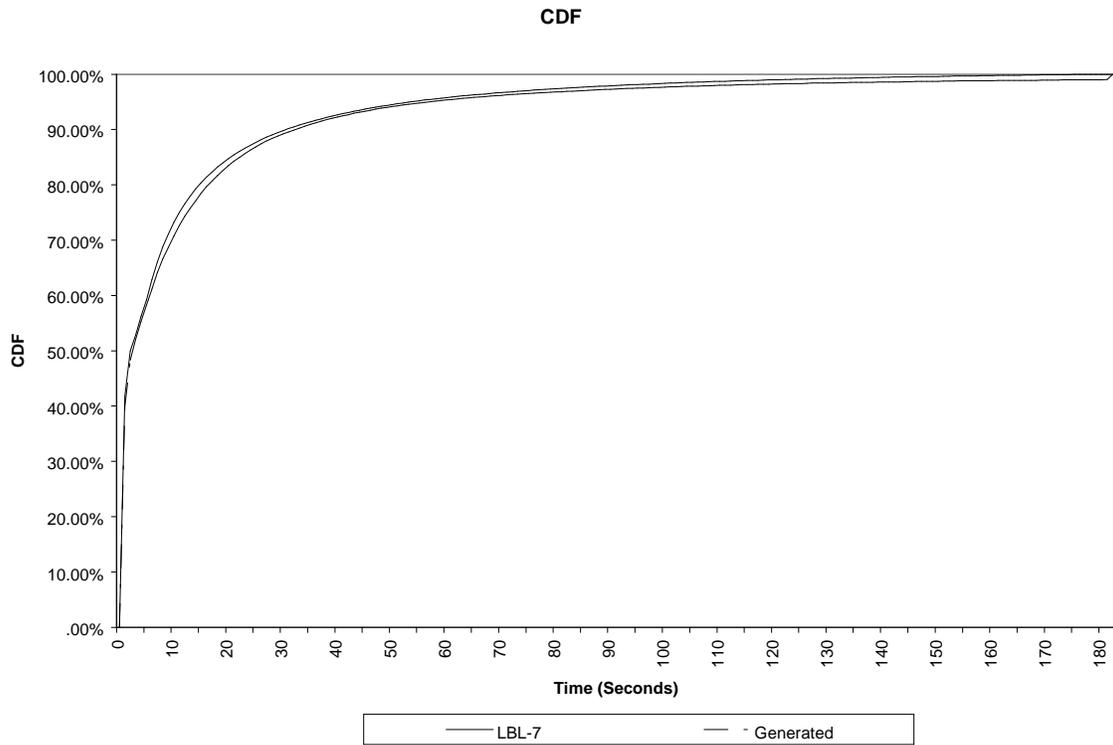


Figure 5: CDFs of the time between connections for both the LBL-7 and Generated datasets

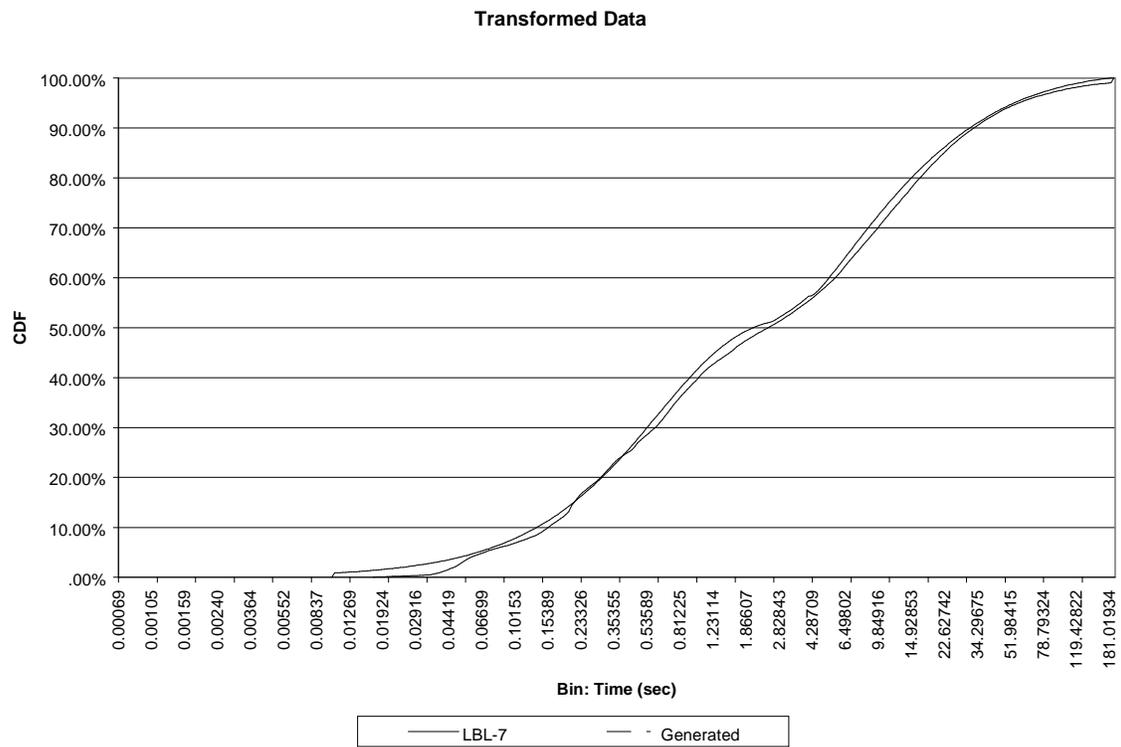


Figure 6: CDFs of the times after a log base 2 transformation

It should be noted that the generated model tracks the original well³ but is smoother due to the use of mathematical models.

4 Future Work

While this tool currently supports FTP traffic only, it was written with the hope of extending it to other applications such as HTTP. Thus, the design loosely supports the addition of other traffic models. Also, while analysis of the inter-connection model is based primarily on the LBL-7 dataset, it would be valuable to test the current model on additional, more recent traces. Another interest is the effect that round trip times have on inter-connection times. Such an analysis would require a packet level trace of network activity, which was unavailable at the time of the analysis.

5 Conclusion

This tool generates FTP sessions based on models which incorporate the key characteristics of the protocol. By generating traffic that behaves off characteristics found at the application level, this tool will allow for a better analysis of other network issues. Finally, the traffic profile generated can be used alongside a variety of applications and tools.

6 Acknowledgements

I would like to thank Mark Allman for his continuous support and guidance with this work. Also, I would like to thank Vern Paxson for both the LBL-7 dataset and for taking time to discuss the work, and Mark Allman and Funda Ergun for commenting on earlier drafts of this report.

This work was supported by the NASA Glenn Research Center in conjunction with Case Western Reserve University under award number NAG3-2391.

References

- [Pax94] V. Paxson, "Empirically-Derived Analytic Models of Wide-Area TCP Connections", *IEEE/ACM Transactions on Networking*, 4(2), pp. 316-336, August 1994
- [PF97] V. Paxson and S. Floyd, "Why We Don't Know How to Simulate the Internet", *Proceedings of the 1997 Winter Simulation Conference*, Atlanta, Georgia, December 1997
- [PF95] V. Paxson and S. Floyd, "Wide-Area Traffic: The Failure of Poisson Modeling", *IEEE/ACM Transactions on Networking*, 3(3), pp. 226-244, June 1995

³ Deviations are no more than +2.5% to -0.5% at any given point.