A Thesis entitled

A New Model for the Marginal Distribution of HTTP Request Rate

Submitted to the
University of Wollongong
Australia
in fulfilment of the requirements for the degree of
Doctor of Philosophy

John Thomas Judge, BE

School of Electrical, Computer and Telecommunications Engineering
September 2004

Abstract

This thesis proposes a new model for the marginal distribution of HTTP request rate. The model applies to aggregate network traffic generated by a population of users accessing the Web on the Internet. The new model is relatively simple and allows for both the accurate estimation of peak HTTP request rate and the development of two new rules of thumb concerning HTTP request rate. Previous models of HTTP request rate have generally been single user models of a form that are both complex to transform into a model of aggregate traffic and apply to the estimation of peak aggregate HTTP request rate. One comparable model of aggregate HTTP traffic models HTTP request inter-arrival time rather than HTTP request rate and is shown to over estimate peak HTTP request rate. There are few existing rules of thumb concerning HTTP request rate. The two rules proposed here are the first for the estimation of either standard deviation or peak HTTP request rate at the second time scale.

The new model for the marginal distribution of aggregate per second HTTP request rate is based on the Pólya-Aeppli probability distribution. The selection of the Pólya-Aeppli distribution can be justified from observed distributions of HTTP request rate of individual Web users and the number of active users per second in a population of Web users.

The results are based on the analysis of five independent traces of Web traffic. One trace, collected by the candidate, is of per-user Web traffic generated in a postgraduate research laboratory at the University of Wollongong (UOW) between 1994 and 1997. The other four traces are large independent traces of aggregate Web traffic collected between 1996 and 2002.

CERTIFICATION

I, John Thomas Judge, declare that this thesis, submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, is wholly my own work unless otherwise referenced or acknowledged. This document has not been submitted for qualifications at any other academic institution.

Signed

John Judge

7 September 2004

Acknowledgments

I would like to thank a variety of people and their respective organisations for access to Web traffic log files. Making these log files available required their time and effort, either in preparation of data specifically for me or in making it available to the entire research community, and I am grateful. Specifically I would like to thank and acknowledge; Jeff Mogul and Digital Western Research Lab (now HP Western Research Lab) for access to his published data set and correspondence concerning the data set, Steve Gribble and the University of California, Berkeley for the Home IP data set and correspondence concerning the data set, Adam Radford, Chris Merrigan and Keith Burston, University of NSW (UNSW) for access to their Web cache log files, Aidan Williams and National ICT Australia for access to their PolyMix-4 traffic traces and the postgraduate research students at the Switched Networks Research Centre UOW for their trust in allowing me to log their Web traffic.

I would like to thank my supervisor, Joe Chicharo, for his friendship and support. I would also like to thank Peter Beadle, formally of the University of Wollongong, for his support which continued long after he left the university.

On a personal note I would like to thank Keith Burston, the late Gary Anido, Joe Chicharo, Peter Beadle and Roger Kermode for the employment opportunities that made it financially possible for me to undertake a research PhD. Thanks to Aidan Williams, Peter Beadle and Joe Chicharo for proof reading various drafts.

Finally, I would like to thank my partner Kathlene and my daughter Jessie. Without Kathlene's love, support and patience none of this would have been possible. Jessie, a delightful and beautiful three year old girl, gets thanked for simply being.

Table Of Contents

List of Tables

List of Figures

List of Abbreviations

ARIMA	Autoregressive Integrated Moving Average
ASCII	American Standard Code for Information Interchange
CDF	Cumulative Distribution Function
CERN	European Laboratory for Particle Physics
FSD	Fractional Sum-Difference
GOF	Goodness-of-Fit
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
ISP	Internet Service Provider
LAN	Local Area Network
LRD	Long Range Dependence
Mb	Megabit
MIME	Multipurpose Internet Mail Extensions
ML	Maximum Likelihood
MOM	Method of Moments
NICTA	National ICT Australia
PDF	Probability Distribution Function
PMF	Probability Mass Function
PP Plot	Probability-Probability Plot
PSTN	Public Switched Telephone Network
QQ Plot	Quantile-Quantile Plot
SNRC	Switched Network Research Centre, University of Wollongong
TCP	Transmission Control Protocol
UNSW	University of New South Wales
URL	Uniform Resource Locator
UOW	University of Wollongong
W3C	World-Wide Web Consortium
WCA	Web Characterisation Activity
Web	World-Wide Web
XHTML	Extensible Hypertext Markup Language

1. Introduction

This dissertation looks at the characterisation of hypertext transfer protocol (HTTP) traffic as generated by users of the World-Wide Web (Web) application on the Internet. Of interest is the traffic generated by multiple (aggregate) users on access networks. This traffic is examined by analysis of proxy log file data collected from various sources over a total of eight years. The primary contributions are a model that describes the marginal distribution of aggregate per second hypertext transfer protocol (HTTP) request rate and two associated rules of thumb for HTTP request rate.

This work was prompted by the extraordinary growth in the popularity of the Internet and the Web since the early 1990s. The connection of end users to the Internet via access networks, such as those run by an Internet Service Providers (ISP), is now big business. Web traffic is a dominant component of Internet traffic and consideration of Web traffic load is an important aspect in the design and dimensioning of these access networks. The effort expended by access network operators to manage Web traffic load is evidenced by the availability of large capacity proxy caches. By servicing a percentage of user requests locally, through use of a Web proxy cache, the access network operator can substantially reduce the network traffic traversing the network link to the Internet. What is missing are models of Web traffic that are useful in the design and dimensioning of access networks.

Web traffic load is usually measured in units of HTTP requests per time period, with HTTP requests per second being the favoured unit [Wessels 01 p. 192, Luotonen 98 p. 296]. Benchmarking systems, such as Web Polygraph [Polyteam 04] used by the Measurement Factory [Rousskov 01], use performance figures of HTTP requests per second to compare Web proxy caches. Aggregate user traffic measured in HTTP requests per second is what is of interest for the model developed here. The goal is a relatively simple analytical model that is accurate enough to be useful in estimating expected peak HTTP request rate and the validation of artificial Web traffic workloads.

There have been a number of contributions in the area of modelling HTTP request traffic including [Abrahamsson 00, Arlitt 95, Arlitt 99, Barford 98a, Buchholz 02,

Busari 01, Cao 01, Catledge 95, Choi 99, Cleveland 00a, Cohen 99, Crovella 97, Deng 96, ETSI 98, Feldmann 98a, Feldmann 98b, Gribble 97b, Hlavacs 99, Kant 99, Mah 97, Molina 00, Morris 00, Reyes-Lecuona 99, Rousskov 01, Sedayao 94, Smith 01, Wessels 99]. In those contributions which have proposed models for HTTP request arrivals the majority have examined the traffic generated by single users [Abrahamsson 00, Arlitt 95, Barford 98a, Choi 99, Deng 96, ETSI 98, Hlavacs 99, Mah 97, Reyes-Lecuona 99, Rousskov 01]. A minority of contributions have examined models for aggregate HTTP traffic [Cao 01, Cleveland 00a, Kant 99, Molina 00, Wessels 99]. The problems that generally apply with existing models are that they are either too complex to be practical to apply to aggregate traffic [Barford 98a, Choi 99, Deng 96, ETSI 98, Mah 97, Reyes-Lecuona 99], model a higher level abstract and not HTTP request arrivals [Abrahamsson 00, Arlitt 95, Molina 00, Reyes-Lecuona 99], are based on a Poisson assumption and don't reflect actual traffic [Wessels 99], or are purely empirical based entirely on data points comprising a cumulative distribution function (CDF) [Abrahamsson 00, Hlavacs 99, Mah 97].

The contribution by [Cleveland 00a], updated by [Cao 01], details a relatively simple model for aggregate HTTP request traffic in a similar manner to the work presented here. There are two main differences. First, they are modelling HTTP request inter-arrivals while this work looks at the count of HTTP arrivals per second. The second difference is one of approach; they examine HTTP traffic traces over short periods of time in which they consider the number of traffic sources as constant. The work presented here looks at longer samples of traffic in which the underlying aggregate HTTP request rate is considered constant but the number of traffic sources may fluctuate. The fact that the number of traffic sources can vary is ultimately reflected in the choice of model. The model developed here has the advantage that it is directly applicable to generating estimates of peak HTTP arrival rate.

A challenge for a model of HTTP request traffic is to be invariant. The approach taken here has been to source traffic traces from different user populations over different time periods. Three potential sources of change in Web traffic are identified in Chapter 2; change in HTTP itself, change in the number of embedded objects per page and change in the number of non-human initiated HTTP requests. The models and methods developed here are considered invariant to the first two sources of

change. The traces were collected over a time period in which these changes were taking place and the derived model and methods are shown to hold to all of them. The third area of change is more problematic. The assumption is that the traces used in this work predominantly record human initiated action and the resulting models are most likely dependent on the random nature of human behaviour. Obvious non-human initiated requests were removed as outliers from one trace (Appendix C) before further analysis was performed. Similar removal of these types of requests as outliers has been reported by others [Choi 99, Mah 97]. If the ratio of non-human initiated requests rises significantly in future Web traffic then new models or re-validation of the work presented here may be required.

The results presented in this dissertation have been derived from an examination of Web traffic traces collected over an eight period between 1994 and 2002. The contributions concerning aggregate HTTP traffic are based on the analysis of four large independent traces of Web traffic. The derived models and heuristics are shown to apply to all four aggregate traces. One of the aggregate traces is from a large corporation while the other three are from university user populations. One of the traces sourced from a university consists largely of the traffic generated by users accessing the Web over dial-up and similar bandwidth connections while the other three traces host user populations on a high speed local area network (LAN). Two traces were collected in the USA while the other two were collected in Australia. Two traces were collected in 1996 by researchers at Digital Equipment Corporation [Kroeger 99] and University of California, Berkeley [Gribble 97b]. Both of these traces are publicly available from their respective sources for examination by other researchers. Two traces were sourced between 1999 and 2002 by this candidate from University of NSW (UNSW) in Australia.

A fifth trace used in this dissertation was obtained by this candidate from the Web traffic generated by a post-graduate laboratory at the University of Wollongong (UOW) in Australia. This trace details the traffic generated by a small number of users over a three year period between 1994 and 1997. The partitioned office environment of the laboratory allowed for matching each computer source to a specific user. The unusually long duration of the trace allowed for examination of HTTP request rate on a per user basis. Analysis of this trace provided useful insight into possible models of aggregate user traffic.

The model proposed for the marginal distribution of aggregate HTTP per second request rate is based on the Pólya-Aeppli probability distribution [Appendix F.4]. The model is shown to be a good fit to observed Web traffic. The model was proposed after observing the distributions of both the number of active Web users per second and the number of HTTP requests each user makes. These are matched by the Poisson and geometric distributions respectively. The Pólya-Aeppli distribution is a combination of the two.

Three applications of the HTTP request rate model are examined; estimation of peak HTTP request rate, use of the model in the development of two rules of thumb concerning HTTP request rate and for the sanity checking of models of HTTP request arrival proposed by others.

It is shown that the Pólya-Aeppli model allows for accurate estimation of peak HTTP request rate from known mean and variance. Less accurate, but still quite good, estimates can be obtained from knowledge of just the mean HTTP request rate. For example, for the traffic samples extracted from all four traces of aggregate traffic, the 95% quantile of peak per second HTTP request rate can be estimated to an average of ±8% just from a known mean rate.

The fact that use of the Pólya-Aeppli probability distribution provides such good estimates of peak HTTP request rate allows for the proposal of two new rules of thumb concerning HTTP request rate. The first follows from a result shown in Chapter 5 that there is a linear relationship between the mean and the standard deviation of per second HTTP request rate. It is a simple analytical expression for the estimation of the expected standard deviation of per second HTTP request rate from a known mean. The second is a simple analytical expression for the estimation of the expected peak per second HTTP request rate at the 95% quantile. That is, the peak HTTP request rate that is expected to be equalled or exceeded on average only once every 20 seconds. Comparison with samples of traffic from all four aggregate Web traffic traces shows the two rules of thumb provide fair approximations of the desired statistic. Both rules would benefit from testing against alternative Web traffic traces in the future which may lead to further refinement.

The third application is to exploit the Pólya-Aeppli result and the two proposed rules of thumb in the assessment of whether or not a model for HTTP request arrival

describes a HTTP request rate that resembles actual Web traffic. Four models of HTTP request rate proposed by others [Cao 01, Deng 96, Mah 97, Rousskov 01] are examined. Only one [Cao 01] is found to produce samples of simulated HTTP request rate traffic that sometimes resemble real traffic. The other three models either result in simulated traffic that exhibits less per second HTTP request rate variance than actual Web traffic [Deng 96, Rousskov 01] or more variance [Mah 97]. It is found that examining the marginal distribution of simulated per second HTTP request rate to see if it has a Pólya-Aeppli shape is not a very discriminatory test as all four models have this. The two proposed rules of thumb are more discriminatory as only the one model [Cao 01] partially meets these two criteria

1.1 Outline

Chapter 2 describes the Web and examines the work of others in characterising aspects of Web traffic. Previous contributions in modelling the HTTP request arrival process and estimating peak HTTP request rate are examined. The work and contribution of this thesis are placed in context.

Chapter 3 looks at the usefulness of the Poisson distribution in modelling HTTP request rate. The non-Poisson nature of TCP traffic is well known from a study by Vern Paxson et. al. [Paxson 94a]. The study included early traces of Web traffic and showed a non-Poisson nature at the TCP level. Chapter 3 demonstrates that the Poisson distribution is not a good model for describing the arrival of aggregate HTTP requests. In addition it is shown that there is no evidence that the HTTP request arrival process, as a count of HTTP request arrivals per second, may tend to Poisson with increasing aggregation. However the Poisson distribution does still have a part to play in modelling Web traffic. HTTP requests from a single user can be abstracted into sessions and it is shown that sessions do exhibit a Poisson arrival process. In addition it is also shown that the number of unique active users per second in an aggregate stream of HTTP requests also has a Poisson distribution.

Chapter 4 looks at the number of HTTP requests generated by single users. The chapter takes advantage of the long duration trace gathered at UOW to examine the distribution of HTTP requests generated per hour on a per user basis. It is found that

for an hour in which a user makes at least one HTTP request the request rate follows a geometric distribution. The geometric distribution is also found to approximate the number of HTTP requests generated per user on the minute and second time scale.

In Chapter 5 a new model for the marginal distribution of aggregate per second HTTP request rate is proposed. The model is based on the findings from previous chapters of an approximate Poisson number of active users combined with an approximate geometric distribution for the number of HTTP requests generated per active user. Combining these results a new model is proposed for the marginal distribution of HTTP request rate per second based on the Pólya-Aeppli distribution. The Pólya-Aeppli distribution is shown to be a good match to observed Web traffic. The match is confirmed using probability-probability (PP) plots and quantile-quantile (QQ) plots. It is shown that the Pólya-Aeppli model of the marginal distribution of HTTP request rate tends towards the normal distribution with increasing mean request rate but reaches an asymptotic limit at a mean request rate of around 1000 HTTP requests per second.

Chapter 6 looks at three applications of the proposed model for the marginal distribution of aggregate per second HTTP request rate. First, estimation of peak per second HTTP request rates. Second, the formulation of two new rules of thumb concerning per second HTTP request rate. Third, in the assessment of whether or not a model for HTTP request arrival proposed by others results in a HTTP request arrival rate that resembles actual Web traffic or not.

Chapter 7 concludes the dissertation and identifies areas for possible future work.

1.2 Contributions

The contributions claimed in this dissertation are grouped into two areas.

Contributions arising from the analysis of the trace of Web traffic collected from a postgraduate research laboratory at the UOW. The contributions are considered indicative rather than invariant over all the users on the Web due to the small trace size and specific nature of the user population.

1. The number of Web requests generated by a single user browsing the Web in an active hour (an hour where at least one request was generated) has a geometric distribution (Section 4.2) [Judge 99]

2. Web users have different underlying hourly mean request rates for active hours (Section 4.2)

3. The geometric distribution is an approximate match for the number of requests generated by a single user in an active minute and an active second in which at least one request was generated (Section 4.3)

Contributions from analysis of four independent traces of Web traffic collected between 1996 and 2001. Due to the independence of the traces and the diversity of collection methods, collection dates and user populations these contributions are considered more reliable and claimed as invariant properties of aggregate Web traffic. These contributions relate to aggregate HTTP request arrivals with constant mean rate.

4. The number of unique users generating HTTP requests in a second in an aggregate stream of HTTP traffic has a Poisson distribution (Section 5.1) [Judge 99]

5. The marginal distribution of the number of HTTP requests per second generated in an aggregate stream of Web traffic per second has a Pólya-Aeppli distribution (Section 5.1) [Judge 99]

6. A physical explanation for the Pólya-Aeppli shape of the marginal distribution of per second aggregate HTTP request rate is as a result of the geometric distributed number of HTTP requests per user and a Poisson distributed number of active users (Section 5.1) [Judge 99]

7. There is an approximate linear relationship between mean and standard deviation of aggregate per second HTTP request rate when mean HTTP request rate is over approximately 10 HTTP requests per second.

8. It is shown that the Pólya-Aeppli model of the marginal distribution of HTTP request rate per second tends towards the normal distribution with increasing

HTTP request rate but reaches as asymptotic limit at a mean HTTP request rate of around 1000 HTTP requests per second (Section 5.3)

9. The Pólya-Aeppli model of the marginal distribution of HTTP request rate results in good estimates of peak per second HTTP request rate from known mean and variance (Section 6.1) [Judge 99]

10. Two new rules of thumb concerning aggregate per second HTTP request rate which apply when mean HTTP request rate is over 10 HTTP requests per second:

    10i. An estimate $\tilde{s}$ of the standard deviation of aggregate per second HTTP request rate can be estimated from the mean rate $\bar{x}$ using the formula:

$$\tilde{s} = 0.186\bar{x} + 4.26 \quad \textbf{(Eqn 1.1)}$$

    10ii. An estimate $\tilde{p}$ of the peak aggregate per second HTTP request rate at the 95% quantile can be estimated from the mean rate using the formula:

$$\tilde{p} = 1.32\bar{x} + 7.33 \quad \textbf{(Eqn 1.2)}$$

1.3 Publications

Aspects of the work presented in this dissertation also appear in the following publications authored by the candidate:

1. John Judge, H.W. Peter Beadle and Joe Chicharo, “Modelling User Traffic in the WWW”, *Proc. Australian Telecommunication Networks & Applications Conference (ATNAC'95)*, Sydney, 1995, pp. 163-168 [Judge 95]

2. John Judge, H.W. Peter Beadle and Joe Chicharo, “Modeling World-Wide Web Request Traffic”, *Proc. IS&T/SPIE Multimedia Computing and Networking 1997*, San Jose, 1997, pp. 92-106 [Judge 97a]

3. John Judge, Joe Chicharo and H.W. Peter Beadle, “The Size of HTTP Response Packets and Calculation of WWW Traffic Volumes” *Proc. IEEE/IEE International Conference on Telecommunications 1997 (ICT'97)*, Vol. 1, 1997, pp. 257-262 [Judge 97b]

4. John Judge, H.W. Peter Beadle and Joe Chicharo, “Correlation of HTTP Response Packet Size and Estimating Confidence Intervals for Mean Packet Size and WWW Traffic Volume”, *Proc. APCC'97*, Sydney, 1997, pp. 382-386 [Judge 97c]

5. John Judge, H.W. Peter Beadle and Joe Chicharo, “Sampling HTTP Response Packets for Prediction of Web Traffic Volume Statistics”, *Proc. IEEE Globecom’98*, Sydney, 1998, pp. 2617-2622 [Judge 98]

6. John Judge, “Estimating Peak HTTP Request Rate for a Population of Web Users”, *Proc. 10th IEEE Workshop on Local and Metropolitan Area Networks LANMAN’99*, Sydney, 1999, pp. 108-111 [Judge 99]

7. John Judge, “A Model for the Marginal Distribution of Aggregate Per Second HTTP Request Rate”, *Selected papers from 10th IEEE Workshop on Local and Metropolitan Area Networks*, 2001, pp. 29-36 [Judge 01]

The following publication utilised some of the work presented in this dissertation:

1. Lorraine de Vere, John Judge, Gary Anido and H.W. Peter Beadle, “Internet Service Over ATM: The Effect on WWW Performance”, *Proc. 7th International Network Planning Symposium*, Sydney, 1996, pp. 227-232 [de Vere 96]

2. Related Work in Modelling HTTP Request Rate

This chapter provides an overview of the Web and examines previous contributions in the characterisation of HTTP request rate.

The basic architecture of the Web is relatively simple and has not varied much since the original proposal by Tim Berners-Lee et. al. [Berners-Lee 92a]. One aspect of the Web that was not described in the initial publications from Berners-Lee was the concept of Web proxy servers. These servers are positioned between Web clients and Web servers typically caching objects. Caches between Web clients and servers can provide bandwidth savings and faster response time. Publications examining the use of proxies started around 1994 [Glassman 94, Luotonen 94, Sedayao 94] and papers discussing the nature of aggregate Web traffic appeared soon after.

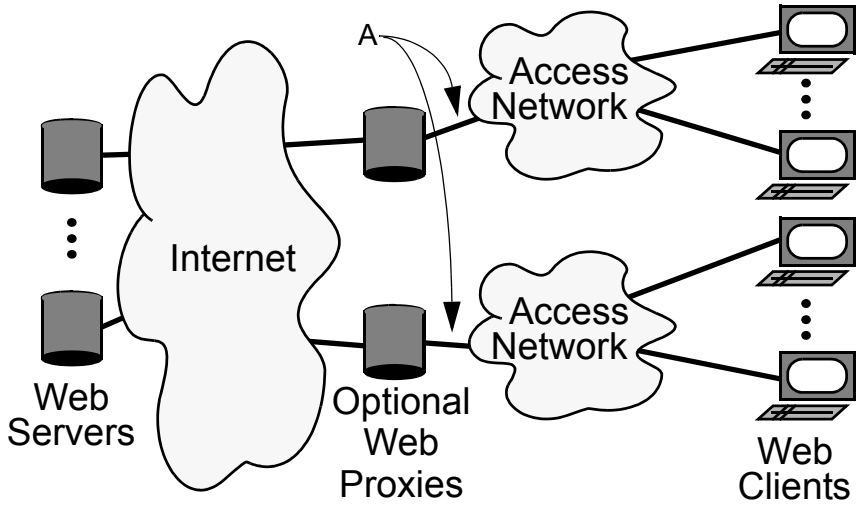


Figure 2.1 The General Topology of an Access Network

This dissertation is mainly interested in the aggregate traffic generated by a population of Web users. The term “access network” is used to describe a network connecting such a population to the Internet. Examples of access networks include the network operated by an ISP or the campus network of a university. The technology used to implement these networks can vary. For example, an ISP may have a large PSTN modem pool while a university may utilise a wide area ethernet network. The general topology is shown in Figure 2.1. The point of interest is where traffic from

multiple users comes together at points “A” on Figure 2.1. This traffic either directly enters the Internet or is relayed through a Web proxy server.

A challenge in HTTP traffic characterisation is to develop general models that are applicable to traffic generated by different user populations and at different points in time but specific enough to have some useful application. The approach taken here has been to develop models and methods using a diverse set of Web traffic traces collected over an 8 year period. The traces date from 1994 though to 2002 and come from corporate and university user populations. Two of the aggregate traffic traces are publicly available allowing for the analysis of the same data by other researchers.

Since 1992 there have been three general areas of change in the nature of Web traffic:

1. Change in HTTP from version 0.9 to 1.0 and then to 1.1
2. Changes resulting from an increase in the number of embedded objects in Web pages
3. Changes in network traffic from the introduction of software to automate the generation of HTTP requests. For example, traffic resulting from a periodically updating stock price figure shown on a web page.

The resulting models and methods are shown to hold despite changes in HTTP version and an increase in the number of embedded objects per page. The change in HTTP version from 1.0 to 1.1 occurred during the time period in which traces were collected and the number of embedded objects in Web pages also increased during this time frame. The third area of change is more problematic. The traces used in this work are assumed to be predominantly generated by the actions of human users. The model developed includes the Poisson distribution and is most likely dependent on the random behaviour of human beings. Obvious automated HTTP requests were filtered out of the raw trace data as outliers before further analysis as has been previously done by others [Choi 99, Mah 97]. If the ratio of non-human initiated HTTP requests rises in future Web traffic then new models and methods may be required.

The focus of the work presented in this thesis is the development of a new analytical model for the marginal distribution of aggregate HTTP request rate. The goal is a relatively simple model that is accurate enough to be useful in estimating peak HTTP request rate and in the sanity checking of artificially generated HTTP traffic.

There have been a number of contributions in the area of modelling HTTP request traffic [Abrahamsson 00, Arlitt 95, Arlitt 99, Barford 98a, Buchholz 02, Busari 01, Cao 01, Catledge 95, Choi 99, Cleveland 00a, Cohen 99, Crovella 97, Deng 96, ETSI 98, Feldmann 98a, Feldmann 98b, Gribble 97b, Hlavacs 99, Kant 99, Mah 97, Molina 00, Morris 00, Reyes-Lecuona 99, Rousskov 01, Sedayao 94, Smith 01, Wessels 99]. In those contributions which have proposed models for HTTP request arrivals the majority have examined the traffic generated by single users [Abrahamsson 00, Arlitt 95, Barford 98a, Choi 99, Deng 96, ETSI 98, Hlavacs 99, Mah 97, Reyes-Lecuona 99, Rousskov 01]. A minority of contributions have examined models for aggregate HTTP traffic [Cao 01, Cleveland 00a, Kant 99, Molina 00, Wessels 99].

The problems that generally apply with existing single user and aggregate models are:

- Most single user models are too complex to be practically applied to aggregate traffic. This applies to models which include size of the requested object as a parameter in the request arrival model [Barford 98a, Choi 99, ETSI 98, Reyes-Lecuona 99] or are ON/OFF type models requiring numerical simulation to model aggregate traffic [Barford 98a, Choi 99, Deng 96, Mah 97].

- A higher level abstract such as user clicking on links [Abrahamsson 00] or “page” arrivals [Arlitt 95, Molina 00, Reyes-Lecuona 99] is modelled rather than HTTP request arrivals.

- Poisson based models are not accurate. An example is the “PolyMix-1” workload used in the first Measurement Factory Web proxy benchmarking activity [Wessels 99]. This model under-estimates the burstiness of HTTP request traffic.

- Purely empirical models, based entirely on data points comprising a CDF for each parameter of interest, [Abrahamsson 00, Hlavacs 99, Mah 97] do not meet the goal of an analytical model.

This work and recent work by [Cao 01, Cleveland 00a] share the goal of finding simple yet accurate statistical models for aggregate HTTP request traffic. The difference is one of approach. They model HTTP request inter-arrivals during short periods of time over which the number of traffic sources is assumed to be constant. The work presented here models HTTP request rate measured in counts of arrivals per-second over longer periods in which the mean request rate was assumed to be constant. The number of sources may vary during the measurement period and the model reflects this. The resulting model has the advantage that it is directly applicable to estimating peak HTTP arrival rate.

The first section of this chapter briefly reviews the development and architecture of the Web. A comparison is made with the hypothetical "Memex" machine described by Vannevar Bush [Bush 45]. Web terminology is reviewed and various terms are defined. The second section of this chapter looks at change in the Web and the potential impact this change has on the modelling of HTTP request rate. The third section looks at the contributions by others in the area of HTTP request rate modelling and estimation of peak HTTP request rate. The fourth section of the chapter outlines the selection and collection of Web traffic traces used in this dissertation and includes a description of the methods used to extract samples of Web traffic from the traces.

2.1 The World-Wide Web

2.1.1 Development and Overview of the World-Wide Web

The fundamental protocols and basic design of the Web were created at the European Laboratory for Particle Physics (CERN) and are described in the Berners-Lee paper entitled "World-Wide Web: The Information Universe" [Berners-Lee 92a]. Essentially the Web is a distributed information system with primitive objects consisting of hypertext pages with extensions for the display of other media. The Web includes the definition of a common addressing scheme, protocol and format negotiation. Software for browsing and serving information on the Web is relatively easy to write using the defined interface. There is no centralised control of the Web appli-

cation, anybody is free to add to the information space merely by running a Web server on a computer appropriately connected to the Internet.

The World-Wide Web project at CERN commenced in 1989 [Berners-Lee 92b]. Widely published papers describing the concept and architecture of the Web and inviting participation appeared in two journals in 1992 [Berners-Lee 92a, Berners-Lee 92b]. By 1995 the Web was the single largest source of application level traffic by byte and by packet on the Internet [Pitkow 95a]. This trend has continued. Measurements in August 1997 on the MCI commercial backbone showed Web traffic comprising up to 75 percent of all bytes and 70 percent of all packets transmitted on the Internet [Thompson 97]. Recent statistics are unavailable due to the difficulty of gathering data after the break up of the network core in the US [McCreary 00]. A measurement in 2000 at a major Internet traffic exchange point (NASA Ames Internet exchange in Mountain View) showed HTTP remaining as the largest single source of TCP application level traffic [McCreary 00]. Measurements at the University of North Carolina at Chapel Hill performed in 1999 and 2000 showed that for traffic entering their campus HTTP was the single largest consumer of bandwidth [Smith 01]. However this trace also showed the actual percentage of bytes consumed by HTTP dropped between 1999 and 2000 traces from 56% to 35%. The drop was attributed to the rising popularity of audio file (MP3) sharing software.

The Web has a client server architecture and uses HTTP to transfer information between Web clients and Web servers. HTTP is a stateless protocol in which each request/response pair is independent of other transactions. Objects are addressed using a uniform resource locator (URL) [Berners-Lee 94]. The type and format of objects can be flagged using a Multipurpose Internet Mail Extension (MIME) type [Borenstein 93] in the header of the response packet. By interpreting the MIME header Web clients can determine how to display the retrieved object to the user. Objects are usually displayed directly within a browser window using either code native to the browser or third party software modules called “plugins”. Alternatively objects are either displayed to the user using an external process, called a “helper” application, or saved directly to disk. Typically the information retrieved from Web servers is formatted as a hypertext page. The development of the Web included the initial definition of hypertext mark-up language (HTML) [Richardson 95] which

defines the page layout and allows URLs for embedded objects as well as external links.

Web clients usually include some compatibility with previously existing Internet applications. Access to information over FTP (file transfer protocol) [Stevens 94], Gopher [Anklesaria 93] and NNTP (network news transfer protocol) [Stevens 96] are included in the URL addressing scheme. Web client software can usually communicate (as a client) using these protocols as well as HTTP. Email was incorporated by defining a specific URL type for email addresses.

Coordination of the development of the Web moved from CERN to an international industry consortium called the World-Wide Web Consortium (W3C) in 1994. The W3C is jointly hosted by the Massachusetts Institute of Technology Laboratory for Computer Science (MIT/LCS) and the Institut National de Recherche en Informatique et en Automatique (INRIA)[1].

2.1.2 Historical Perspective

While development of the Web commenced in 1989, the general concept of a hypertext information space was not new. Software, such as Apple's HyperCard, existed in the 1980's for building hypertext databases and the origins of hypertext are frequently cited as going back to Vannever Bush in 1945 [Baird 90].

Bush outlined the concept of an imaginary device called a "Memex" constructed out of technology available or conceivable in the 1940's [Bush 45]. Built into a desk the Memex was an electromechanical machine to automate the storage and retrieval of information. It was operated by manipulation of buttons, levers and a keyboard. Microfilm was the information storage medium and pages were projected on to glass panels in the top of the desk. Extrapolating slightly from the microfilm capabilities of the day it was calculated that the Memex could conceivably store around 10^7 indexed pages of text and images. Vannever's key feature of the Memex was to allow for the indexing of information by association. The Memex user could create and follow arbitrary links between one piece of information to another. The term "trail" was used to describe the path a user might take following these links. Memex

1. Home page for W3C is `http://www.w3c.org/`

users could share raw information and also share trails by physical exchange of microfilm.

Around 50 years later the Web has extrapolated from Vannever's concept. Computer and communications technology have replaced physical microfilm and electromechanical mechanisms. Hand to hand transfer of physical media is no longer required. The partition of an information space into a separate archive per user has been replaced by one vast, and growing, information space. In 1999 the size of the information space accessible via the Web was estimated at approximately $3x10^8$ pages [OCLC 99], in December 2002 the Google Web search engine claimed to have indexed over $3x10^9$ Web pages [Google 02].

The distributed nature of the Web and the fact that it is one large shared information space presents new challenges for teletraffic research. The issue investigated here concerns characterisation of the arrival of HTTP request packets that result from the aggregated demands of multiple end users. An analogy can be made to early work modelling the expected number of telephone calls to a central switch, for example, estimating the blocking probability for subscriber lines obtaining a telephone trunk [Molina 1922].

2.1.3 Definitions and Terminology

The Web Characterisation Activity (WCA) of the W3C produced a draft document outlining terms used in the characterisation of Web traffic [W3C 99a]. This dissertation uses these terms where possible.

A "Web user" is defined as an individual accessing the Web using Web client software. The term "Web client" is used to refer to an application process accessing a resource on the Web. Resources are accessed by "Web requests" which are initiated by the client and may be satisfied by a local cache built into the client software. In this dissertation the term "HTTP request" is used in place of Web request to make it clear that the focus is only on requests that result in network traffic. A HTTP request is processed by a Web server, or Web proxy server, and results in a HTTP response being sent back to the client. Requests satisfied by local client caches do not result in any network traffic and are not relevant to this work.

The focus of this dissertation is on aggregate Web traffic generated by a population of Web users on an access network (at point A in Figure 2.1). An access network may make use of a Web proxy server which may also be a cache. In this dissertation the terms “Web proxy”, “Web proxy server”, “Web cache” and “cache” are interchangeable. There is an important distinction between the use of the term “Web proxy” (and variants) in this dissertation and the definition of the term “proxy’ in the WCA characterisation draft document [W3C 99a]. The WCA draft states that:

“*Clients using a proxy know the proxy is present and that it is an intermediary*”

In this dissertation the use of the term “Web proxy” does not imply that a client using the proxy knows of the existence of the proxy. This is to take into account the use of transparent Web proxy caches that process Web traffic without the knowledge of the Web client.

The terms “HTTP request” and “HTTP response” refer respectively to the HTTP packets generated by Web clients and Web servers. Unless otherwise stated the term is independent of the HTTP protocol version which may be one of HTTP/0.9, HTTP/1.0 or HTTP/1.1.

HTTP requests may be considered as grouped into “user sessions” which is defined by the W3A [W3C 99a] as:

“A delimited set of user clicks across one or more Web servers”

This dissertation uses the term “HTTP session” to refer to the period of time in which a Web client is actively generating HTTP requests. HTTP sessions are delimited in Web traffic traces by defining an inactivity timeout. A HTTP session arrival is defined as the arrival of the first HTTP request in a HTTP session.

2.2 The Changing Nature of Web

The Web is under a constant state of change but this change has been evolutionary, rather than revolutionary. Web pages certainly have a more sophisticated look and feel since this author first browsed the Web in 1994 but the basic mechanism for the delivery and display of pages has remained the same. HTTP has been enhanced for

better support of caching and persistent connections but it is still a client server protocol. A single HTTP request still results in a single HTTP response. Web pages are still largely written in HTML. A single HTTP request for a HTML page usually results in a cascade of subsequent requests for embedded images and other objects. These aspects of the Web have existed since the early 1990's.

A good characterisation of Web traffic will remain unchanged in spite of changes to the Web. In order to build a model that is invariant this dissertation uses traffic traces spanning an eight year period, which is approximately half the lifetime of the Web. The specific results used to model the marginal distribution of HTTP request rate are based on aggregate traffic traces collected between 1996 and 2002. Much of the apparent changing nature of the Web discussed in the remaining part of this section occurred during this period of time and the models presented in this dissertation are shown to apply to all the traffic traces.

One area of change not accounted for in the work presented in this thesis is that of non-human initiated requests. These are requests for objects that have been generated entirely automatically and are not due to browsing behaviour. Examples include Web pages that periodically download a particular set of stock prices, sporting updates or weather information. The distinction is that no human has been involved in determining the point of time at which the download commenced, other than perhaps by prior programming. The effect of non-human initiated requests are discussed in Section 2.2.3.

2.2.1 HTTP; The Application Level Protocol for the Web

There have been three versions of HTTP; HTTP/0.9, HTTP/1.0 and HTTP/1.1. This section describes some aspects of the HTTP protocol and the development of persistent HTTP.

HTTP/0.9 was developed as a replacement protocol for FTP [Berners-Lee 92b]. It is a simple stateless client server application level protocol using TCP/IP for transport. A Web client opens a TCP connection to the a Web server and then sends a HTTP request of the form "GET URL" in ASCII text[2] where the URL is the address of the text to be returned. The Web server then responds with a stream of bytes which was

2. Some characters require encoding as defined in RFC1738 [Berners-Lee 94]

either a HTML document or plain text with a HTML header. The end of the response packet is indicated with the Web server closing the TCP connection. Every HTTP request required a new TCP connection. TCP port 80 was allocated for Web servers to listen for incoming HTTP connection requests.

HTTP/0.9 was soon replaced by HTTP/1.0 which is described (along with HTTP/0.9) in RFC1945 [Berners-Lee 96]. RFC1945 was written some time after HTTP/1.0 was in common usage and described current practice. A good general overview of HTTP/1.0 is given in Chapters 13 and 14 of the book “TCP/IP Illustrated Volume 3” by Richard Stevens [Stevens 96].

HTTP/1.0 introduced a number of features to the protocol including headers for content type, format negotiation and some support for caching. HTTP/1.0 remained a simple request/response protocol. The Web client initiates the TCP connection, sends the HTTP request packet, the Web server responds with a HTTP response packet and closes the connection. HTTP/1.0 included a content length field in the HTTP response so Web client software could determine the end of the response packet without waiting for the closure of the TCP connection.

The use of a new TCP connection for each HTTP request is inefficient [Padmanabhan 94]. A busy Web server could quickly accumulate a large number of TCP connections in “TIME-WAIT” state and perhaps run out of room in the kernel table. It was also claimed that since requests to a Web server from a Web client usually arrived in bursts that there was a response time penalty imposed by the creation of multiple connections. The latency penalty was seen as a combination of propagation delay, due to cumulative effects of round trip time for connection establishment packets, and the effect of TCP *slow-start* whereby each new connection did not utilise all available bandwidth immediately. In [Padmanabhan 94] they proposed the use of “long lived” HTTP connections in which one TCP connection would transmit a number of HTTP request and response pairs. In [Mogul 95] the proposal was expanded and the name P-HTTP was coined for a proposed enhancement to HTTP/1.0 to include persistent connections.

A study of the performance of P-HTTP over TCP questioned the magnitude of the latency improvement [Heidemann 97]. The study found that for Web clients accessing the Web over PSTN modems, ISDN lines and slow internet connections the per-

formance increase using P-HTTP was only marginal. Much better improvement was obtained with low latency high bandwidth connections.

Pending implementation of a P-HTTP, the "*keep-alive*" extension was added to HTTP/1.0 [Luotonen 98 pp. 47-48]. By including the *keep-alive* option in the HTTP request header Web client software could indicate to a Web server that the connection should be held up for multiple transactions. Latency improvements were also made by Web clients opening several TCP connections to a Web server for parallel HTTP requests [Luotonen 98 p. 50].

The current specification of HTTP/1.1 is predominately defined by RFC2616 [Fielding 99] with an update for using transport layer security in RFC2817 [Khare 00]. HTTP/1.1 includes persistent TCP connections and reduces the need for multiple parallel connections by supporting pipelining of HTTP requests over a single TCP connection. HTTP/1.1 also has increased support for Web cache control. The IETF status of RFC2616 is that of a draft standard [IETF 03].

HTTP/1.1 does not change the transactional nature of HTTP. It is still a client server protocol with the client issuing one request per Web object. In practical terms HTTP/1.1 makes little difference at the TCP level. A study by Eduardo Casilari et. al. has shown that in terms of network traffic generated there is no significant difference between HTTP/1.0 and HTTP/1.1 [Casilari 01]. It was found that the number of TCP connections per Web page visited made by a Web browser did not differ significantly between HTTP/1.0 and HTTP/1.1 due to the widespread use of the "keep-alive" extension in HTTP/1.0. The size of the TCP connection in bytes also remained the same.

The HTTP/1.1 draft standard is dated June 1999 [Fielding 99]. Compliance to HTTP/1.1 was investigated over a 16 month period between June 1999 and September 2000 [Krishnamurthy 01]. Over this time interval the number of Web servers claiming compliance to HTTP/1.1 rose from 73% to 92.5% with the rest supporting HTTP/1.0. The aggregate HTTP traffic traces used in this dissertation straddle this transition period between HTTP/1.0 and HTTP/1.1.

2.2.2 HTML and Web Page Content

The area of change in the Web that has been particularly dynamic is the definition of HTML. The last version of HTML was 4.01 which succeeded a number of earlier versions [W3C 99b]. HTML has now been replaced by the Extensible Hypertext Markup Language (XHTML) [W3C 03] which is a re-working of HTML in XML. XHTML version 1.0 [W3C 02] is the current W3C recommendation with work under way for version 2.0 [W3C 03].

The addition of new tags and features to HTML has certainly added to the increasing sophistication of Web pages. If there is an impact on HTTP request rate it would be due to the number embedded objects in a HTML page. A variation in the number of embedded objects per page may affect the observed burstiness of HTTP request rate.

Casual observation of Web pages over the last 10 years would suggest that the number of embedded objects has increased and there have been a number of publications discussing the frequency of embedded objects. A study of Web pages by Tim Bray in 1995 found a median number of one embedded image per page [Bray 96]. A study of 98371 documents found by recursively collecting URLs listed by `www.yahoo.com` in 1996 found an average of four embedded images per page [Lee 97]. Mikhail Mikhailov et. al. looked at embedded objects (which included images and other embedded URLs) in 1999 and 2000 and found a median of 7 to 17 depending on the source of the trace [Mikhailov 00]. Mikhailov contrasted his results with Bray and concluded that the number of embedded objects was increasing. A study of TCP/IP traffic headers at the University of North Carolina at Chapel Hill also found an increase in the number of embedded objects per Web page between 1999 and 2000 [Smith 01].

The traces used in this dissertation cover a long enough time span that the developed models and methods are assumed to be invariant to the increasing number of embedded objects in a Web page.

2.2.3 Web Client Programmability and Non-human Initiated Requests

There have been two stages of introducing programmability at the Web client. The first was the introduction of “client-pull” and “server-push” [Musciano 97 pp. 425-

432]. The second was the introduction of client side executable scripts and code such as Javascript and Java.

Client-pull works with the Web client recognising and parsing a “refresh” meta tag in the header of a HTML document. The refresh tag includes a time in seconds indicating when the next URL should be requested by the browser. If the next page requested also has a refresh tag the process is repeated.

Server-push works by utilising the multi-part mixed media feature of MIME. Used in conjunction with a Web browser that interprets this feature correctly, an object sent from a Web server can be in several sections with time delays between each section. Extensions to MIME allow for one section to over-write the proceeding section and various effects can be created. The client issues just one HTTP request and the HTTP response is potentially large.

Client-pull and server-push both have the potential to consume a large amount of bandwidth. If either is configured for rapid page updating then there is the potential for excessive request rates or connections to a Web server. Server-push has the added disadvantage that the TCP connection between the Web client and the Web server is maintained for the duration of the “push” process.

Java and Javascript are examples of programmability that was introduced at the client side that operates in a more sophisticated fashion than the periodic updates of client-pull and server-push. Part of the use of programmability is to enhance the look and feel of HTML, such as adding pull down menus and data validation in forms.

An initial application of client side programmability was “push content” [CSTB 02 p. 109]. Push content could be built on top of client-pull or utilise a client side program such as a Javascript. The idea was to construct dynamic Web pages that perhaps resembled the Web equivalent of TV stations. Content would be pushed to the clients by the Web site, triggered by timing updates programmed into a previous Web page. As originally created the Web is an “active” environment with users browsing around an information space using a mouse and a Web client as a navigational interface. Push content was an attempt to make the Web a more passive information environment. In [Benda 97] the introduction of push content was seen as the

introduction of “Internet TV channels”. The demise of push-content can be found documented in magazines and newspapers of the time. An article entitled “Gross Net Product” in the magazine 21C (Issue 24) in 1997 stated the opinion that Web users found push-content annoying and unnecessary. In an article printed on 6 July 2000 entitled “Peer-to-peer Makes a Splash” in the Sydney Morning Herald newspaper push-content was viewed as a 1996 concept that had failed.

Despite the failure of the push content concept the Web still includes a component of automated HTTP requests from client side software. In addition HTTP requests may be issued by non-browser applications like “wget”[3] or applications using HTTP as a transport protocol for tunnelling through firewalls such as SOAP [Albrecht 04].

The percentage of automated HTTP requests in Web traffic traces is hard to determine and they are a complicating factor in the characterisation of Web traffic. Obvious non-human initiated HTTP requests have been noticed in traces of Web traffic by others [Abrahamsson 00, Choi 99, Mah 97]. They are either explicitly removed as outliers [Choi 99, Mah 97] or discovered in the data set after some analysis has been performed and used to explain an anomaly [Abrahamsson 00]. The approach taken in this dissertation is to exclude obvious non-human initiated HTTP requests from traces of Web traffic as outliers. For example, the trace from Digital examined in this dissertation contained a large number of requests that were removed before the trace was analysed [Appendix C].

It is assumed that the models and methods developed in this dissertation apply to Web traffic that occurs predominantly as a result of human initiated action. The models developed include components based on the Poisson probability distribution and are assumed to model the random behaviour of humans. If automated or robot HTTP requests significantly increase in proportion to human generated traffic then new models may have to be developed to accommodate the phenomena.

3. Unix command line tool that can retrieve files over the Web. Home page `http://www.gnu.org/software/wget/wget.html`

2.2.4 Streaming Media

Streaming media in the Web has two basic variants. The first is where a HTTP response is directed at a “plug-in” or other software and played out to the user of the application as it is received. The second is where an alternate network connection is set up for the delivery of the media. The latter is commonly used by applications such as the RealOne[4] player. In this case it is the address of the media to be played and not the media itself that is passed to the playout software. Using the address the playout software makes a separate, not necessarily a HTTP request, connection to a remote server and plays the media out to the user. The current non-real time nature of the Internet (with no quality of service guarantees) presents a problem to streaming applications of uncertain network latency and jitter.

In this dissertation the network traffic generated by non-HTTP requests is not considered. The traffic traces examined in this dissertation do contain HTTP requests for video, audio along with other media types. As this media is carried in HTTP response packets they are included in all analysis.

2.3 Modelling and Estimation of HTTP Request Rate

This thesis focuses on modelling the marginal distribution of aggregate HTTP request rate traffic generated by a population of Web users. Such traffic has a natural measurement point on a shared network link and may be relayed through a Web proxy server (point A on Figure 2.1). This section looks at contributions made by others in characterising HTTP request rate and their use in modelling the marginal distribution of aggregate request rate.

The concept of Web proxy servers appeared in the literature around 1994 with contributions such as [Glassman 94, Luotonen 94, Sedayao 94]. At the same time initial results were being published on the characterisation of Web user traffic. These were based on observations of the traffic arriving at Web proxy servers, such as [Catledge 95, Glassman 94, Judge 95, Richardson 95, Sedayao 94], or on traces generated by instrumentation of Web browsers themselves, such as [Cunha 95, Catledge 95], or Web traces derived from TCP packet traces such as [Arlitt 95,

4. RealOne is media player product from RealNetworks `http://www.real.com`.

Paxson 94a]. Following these initial contributions there have been many others concerning the characterisation of various aspects of Web traffic. A useful, but dated, survey of general Web traffic characterisation results published prior to 1998 is [Pitkow 98].

The early contributions looking at Web proxies [Glassman 94, Luotonen 94, Sedayao 94] contain little information on the characterisation of HTTP request rate. The contribution from Jeff Sedayao in 1994 [Sedayao 94] looked at the Web usage of his colleagues at Intel and the load placed on the corporate network. The results presented would now be considered far from typical. From a Web user population of 800 individuals he found an overall average per user request rate of 3.5 HTTP requests per day. The maximum observed request rate was 36000 HTTP requests per week which averaged to 6.4 HTTP requests per user per day. In contrast the average request rates recorded in the traces used here range from 55 to 365 HTTP requests per user per day. The contribution from Ari Luotonen et. al. described HTTP protocol aspects of building a Web proxy with no observations of actual traffic [Luotonen 94]. The contribution from Steven Glassman describes the "caching relay" (his term for a Web proxy) set up for use at Digital Corporation [Glassman 94]. He describes the design of the cache and provides some early traffic analysis. Some interesting details are the observation that Web page popularity looked like having a Zipf distribution, early cache hit rate figures of between 30% and 50%, an average returned file size of around 14K bytes and that the use of the cache improved the average response time for accessing a remote (non-Digital) URL from 6 to 9 seconds down to 1.5 seconds. Glassman found an average request rate of around 40 HTTP requests per user per day. This is much closer to the averages found in the traces examined in this dissertation. The rest of this section concentrates on contributions dealing specifically with characterisation of HTTP request rate.

There are a number of ways to categorise HTTP request rate characterisation contributions. Some outline a model for per-user HTTP request rate arrival [Abrahamsson 00, Arlitt 95, Barford 98a, Choi 99, Deng 96, ETSI 98, Hlavacs 99, Mah 97, Reyes-Lecuona 99, Rousskov 01] or a model for aggregate HTTP request rate arrival [Cleveland 00a, Kant 99, Molina 00, Wessels 99]. A number of contributions do not specify a model for HTTP request rate but do characterise related

aspects of HTTP traffic. These contributions can be grouped as; the diurnal nature of HTTP request rate [Arlitt 99, Cohen 99], models of HTTP traffic detailing request type but not arrival rate [Buchholz 02, Busari 01, Smith 01], the concept of HTTP sessions [Catledge 95, Feldmann 98a], the bursty nature of HTTP request rate [Crovella 97, Feldmann 98b, Gribble 97b] and the non-Poisson nature of HTTP request rate [Morris 00].

Among the contributions providing a model of HTTP request rate some provide empirical descriptions [Abrahamsson 00, Hlavacs 99, Mah 97] or analytical descriptions [Arlitt 95, Barford 98a, Choi 99, Cleveland 00a, Deng 96, ETSI 98, Molina 00, Reyes-Lecuona 99, Wessels 99]. Typically the empirical descriptions are in the form of data defining one or more CDFs of some aspect of the traffic. The analytical descriptions typically name one or more probability functions that match the observed data and perhaps provide numerical values for the parameters.

The empirical descriptions are often used in the generation of data points for simulation purposes (following in the footsteps of work like Tcplib [Cáceres 91, Danzig 91]). One problem with this type of characterisation is that it cannot be applied to traffic intensities different to those originally observed. For example, the original empirical model may define a CDF based on observations of aggregate traffic during a busy hour with a mean arrival rate of 100 HTTP requests per second. How does this CDF then apply to an hour with a mean arrival rate of 200 HTTP requests per second? Attempts to develop analytical descriptions of aggregate traffic have found that the parameters (and shape) of matching probability distributions can vary with mean request rates [Cleveland 00a, Judge 99]. This is not an aspect of the traffic captured by the empirical descriptions. The approach taken by [Abrahamsson 00, Hlavacs 99, Mah 97] is to create an empirical model of one “typical” user. Different aggregate traffic intensities can then be created by combing together the desired number of users.

If empirical models include data points that define one or more CDFs then the model is only useful to others if the data comprising each of the CDF curves is available. Bruce Mah has provided a URL where data for his model can be obtained[5], however the other two contributions [Abrahamsson 00, Hlavacs 99] have not.

The following sections look at single user and aggregate request rate in more detail.

2.3.1 Per-User HTTP Request Rate Models

Contributions on single user, or per-user, HTTP request rate were examined for assistance in modelling the marginal distribution of aggregate HTTP request rate. None of the contributions examined provided a simple analytical model of HTTP request rate. Some of the contributions do not model the generation of HTTP requests directly but a more abstract concept such as “click arrivals” (user clicking on a URL) [Abrahamsson 00] or “page arrivals” (top level HTML page or equivalent) [Reyes-Lecuona 99, Arlitt 95]. Of the other contributions the main problem is one of model complexity. Nearly all the models examined provide, in some form, enough information for a characterisation of a marginal distribution of aggregate HTTP request rate but were complicated with one or both of the following:

1. The models include file size of the requested HTTP object as a parameter
2. The models could only provide descriptions of aggregate traffic through numerical simulation on a computer

The main reason for this complexity is that the models are mainly directed at describing more aspects of HTTP traffic than is the focus of this dissertation.

The HTTP request arrival models that include file size of the requested object as a parameter [Barford 98a, Choi 99, ETSI 98, Reyes-Lecuona 99] complicate the arrival process considerably. Firstly, they require a separate modelling effort to describe the size of requested objects. Secondly, the arrival process includes the time taken to retrieve these requested objects from the Web server. This necessitates another model parameter describing available bandwidth so the time duration can be calculated. To describe a stream of requests generated by a single user in one of these models takes at least six [ETSI 98, Reyes-Lecuona 99] or eight [Barford 98a, Choi 99] separate parameters.

Another differentiating factor was how the per-user models could be combined to create an aggregate traffic stream. The key difference between the characterisations is the concept of a user browsing “session” and how this is modelled. The term “ses-

5. http://http.cs.berkeley.edu/~bmah/Software/HttpModel/

sion" refers to the period of time a user is actively browsing the Web. In examining traces of Web traffic browsing sessions can be delimited by selecting a suitably long period of idle time to indicate a session has concluded. Some models define a separate arrival process for user sessions [ETSI 98, Hlavacs 99, Reyes-Lecuona 99], others model the duration of a session and the period of time between sessions [Barford 98a, Choi 99, Mah 97] and the remaining models assume users are constantly active [Abrahamsson 00, Arlitt 95, Deng 96, Rousskov 01].

The models that explicitly define the duration of time between sessions [Barford 98a, Choi 99, Mah 97] or between other groupings of HTTP requests [Deng 96] are categorised as ON/OFF models. An ON period consists of a number of HTTP requests while an OFF period contains no network traffic. The ON period is the duration of time taken for the retrieval of a Web "page" [Barford 98a, Choi 99, Deng 96, Mah 97]. The abstraction of a "page" is used to cover an initial HTTP request plus subsequent HTTP requests that are assumed to be part of, or related to, the initial request such as those for embedded images. The OFF period between sessions and pages is delimited using some threshold time. Some of the models define two types of OFF periods, one between components of a page and the other between pages [Barford 98a]. Some model the silence period between pages and sessions with the same parameter [Choi 99, Mah 97] while the remainder models the time between pages only [Deng 96].

Complexity arises in the aggregation of the ON/OFF models to represent traffic from multiple users. This is done using numerical simulation. Aggregate traffic is generated by combination of the traffic generated by each user. None of the ON/OFF model contributions include analytical descriptions of aggregate traffic.

The models where user sessions have an arrival process of their own are [ETSI 98, Hlavacs 99, Reyes-Lecuona 99]. The consensus is for user sessions to have a Poisson arrival process [ETSI 98, Reyes-Lecuona 99] or at least exponential inter-arrival times [Hlavacs 99]. Aspects of the Poisson nature of the arrival process for user sessions have been previously observed by others [Feldmann 98a, Judge 95, Reyes-Lecuona 99] and are discussed in more detail in Section 3.2.

Some contributions do not model sessions and describe only the traffic generated by users actively browsing the Web [Abrahamsson 00, Arlitt 95, Rousskov 01]. Like

the other single user models aggregate traffic is generated by combining the traffic of a number of users. One of these is used by the Measurement Factory in their Web cache benchmarking efforts [Rousskov 01]. Their model is in a constant state of enhancement and is updated between benchmarking efforts. The model considered here is the one used for their last benchmarking effort, the "Fourth Cache-Off" called "PolyMix-4" [Rousskov 01]. The workload consists of a tunable number of robots (software processes acting as Web client) each generating HTTP requests at a rate of 0.4 HTTP requests per second. In an effort to provide as realistic a workload as possible PolyMix-4 has a considerable number of parameters. A detailed description of the workload is available from the Measurement Factory Web site[6]. The problem with the PolyMix-4 workload is that it is not validated against any traces of traffic to compare HTTP request arrivals. The question regarding this model is whether the HTTP request rate is too smooth as the constant number of users each with a constant request rate would seem to imply.

The per-user models for HTTP request traffic outlined by Bruce Mah [Mah 97], Shuang Deng [Deng 96] and the Measurement Factory [Rousskov 01] lend themselves for comparison to the characterisation work in this dissertation. These models either have Web sites where data defining the model [Mah 97] or sample traces [Rousskov 01] can be downloaded, or request arrival times can be generated relatively easily from the model itself [Deng 96]. This is done in Section 6.3.

The other models either include file size as a parameter [Barford 98a, Choi 99, ETSI 98, Reyes-Lecuona 99], do not model HTTP request rate [Abrahamsson 00, Arlitt 95, Reyes-Lecuona 99] or the model is empirical (data points for CDF functions) and the data is not provided [Hlavacs 99].

2.3.2 Aggregate HTTP Request Rate Models

The simplest of the published models is the workload, termed "PolyMix-1", used in the first Measurement Factory Web proxy benchmarking effort [Wessels 99]. This model, described in full on the Measurement Factory Web site[7], assumes a Poisson arrival process for aggregate HTTP request arrivals. The appeal of this model is the

6. URL is `http://www.web-polygraph.org/docs/workloads/polymix-4/`

7. URL is `http://polygraph.ircache.net/Workloads/PolyMix-1/`

ease in which synthetic HTTP request arrival events can be generated but, as shown in Chapter 3, the Poisson distribution does not provide a good model for aggregate Web traffic. PolyMix-1 has been superseded in subsequent benchmarking by the same team (PolyMix-4 discussed in Section 2.3.1).

A simple analytical distribution for describing HTTP request inter-arrival time is provided by [Kant 99];

> "*The interarrival time distribution appears pretty tame - it is reasonably approximated by a branching Erlang distribution with a coefficient of variation of about 0.5*"

Unfortunately the contribution does not provide any other details. No contribution is made concerning the marginal distribution of the number of HTTP request per second.

A more comprehensive modelling of aggregate traffic is attempted in [Molina 00] and describes the modelling of Web "page" accesses. A trace of Web traffic was obtained by logging TCP traffic on an access link and filtering for the source or destination port number of their Web proxy (8080). Using a silence period of one second as a delimiter, the TCP connections for each end client machine were grouped together with each group corresponding to a single Web page access. It was found that for 20 minute periods the Web pages accesses had nearly exponential inter-arrival times. No results were presented for individual HTTP request arrivals.

William Cleveland et. al. provides the most comprehensive contribution in the area of modelling aggregate HTTP request traffic [Cleveland 00a]. An updated, and slightly simplified, model is provided by Jin Cao et. al. [Cao 01]. Some associated work (which is a useful aid to understanding) can be found in [Cao 02a, Cleveland 00b]. Cao and Cleveland find that the Weibull distribution can be used to model the inter-arrival time of aggregate HTTP request rate packets over 15 minute [Cleveland 00a] and 5 minute [Cao 01] time periods. The shape and scale parameters of the Weibull distribution were different in each block examined. They found a mathematical relationship between the shape and scale parameters of the Weibull distribution and the packet arrival rate. This meant that the parameters of the Weibull distribution could be estimated from a given aggregate mean HTTP request

arrival rate. Significantly they found that the shape parameter tended towards one with increasing request rate, hence the inter-arrival process tended towards exponential. Further they found that the auto-correlation of the log of inter-arrival times decreased with increasing request rate so that the overall arrival process tends towards Poisson. At first glance this is in contrast to the results presented in this dissertation. However Cao and Cleveland explain that their result differs from those based on counting arrival events in a fixed time period and that such methods would fail to reveal the local Poisson limit. The reason given is that as request rate increases the number of arrival events in a corresponding time period increases as well, masking the local limit.

This dissertation looks only at counts of HTTP request arrivals in one second periods. The main point of difference between the work in this dissertation and that of Cleveland and Cao is the consideration of a stationary sample of data. The modelling of Cleveland and Cao is based on "superposition of source traffic point processes" [Cleveland 00a]. They consider each source of traffic as independent and that aggregate traffic is stationary when there are a constant number of sources. Hence their choice of small sampling periods to try and minimise changes in the number of traffic sources. The work in this dissertation differs in that traffic is considered to be stationary when there is a constant mean HTTP request rate. Sampling periods are longer because the number of traffic sources can change and this change is catered for by the choice of model. The one hour sampling period used in this dissertation has been used by others examining HTTP traffic [Choi 99, Crovella 97] and Vern Paxson et. al. in looking at other application traffic on the Internet [Paxson 94a].

The marginal distribution of aggregate HTTP request rate as generated by the model proposed by Cao and Cleveland [Cao 01] is compared to the analytical distribution found in this dissertation in Section 6.3.

2.3.3 Peak HTTP Request Rate

One use of a model of aggregate HTTP request rate is in the estimation of peak HTTP request rate. This can be applied to observed traffic, for example estimating peak request rates from observed mean and variance (see Section 6.1), or in sanity checking other models and artificial workloads (see Section 6.3). This section dis-

cusses contributions from others in the general area of peak HTTP request rate. There is an absence of reported work looking specifically at estimating peak HTTP request rate.

A number of general observations and rules of thumb have been proposed for aggregate HTTP request rate observed over varying time periods [Abrahamsson 00, Arlitt 99, Cleveland 00a, Cleveland 00b, Gribble 97b, Luotonen 98, Morris 00]. At longer time periods, say an hour and more, the diurnal nature of HTTP traffic dominates. If traffic peaks at a certain number of requests per hour at 12 noon on one day then it most likely peaks at the same hour on all days. For shorter sampling time periods the diurnal nature of the traffic is not dominant. At these time periods there have been a number of contributions discussing the "bursty" nature of HTTP traffic [Cao 01, Cleveland 00b, Crovella 96a, Gribble 97b, Judge 95, Morris 00] although some of these look at traffic in terms of bytes per unit time period rather than request rate [Crovella 96a, Morris 00].

The nature of Web traffic varies with the time scale of the observation. At the hourly time the smooth diurnal cycle has been observed by many, for example [Abrahamsson 00, Arlitt 99, Cleveland 00b, Gribble 97b, Morris 00]. Steven Gribble examined the Home IP trace (described in Appendix B) and found that he could match a polynomial curve to traffic sampled every minute over the course of the day to the average of 15 days worth of the trace [Gribble 97b]. It was suggested that the curve could be used to calculate approximate load throughout the day. Cleveland looked at a trace of aggregate HTTP traffic collected on a connection between Bell Labs and the Internet [Cleveland 00b]. It was found that variation due to the diurnal nature of traffic was the dominant factor for traffic samples at the time scale of tens of minutes and more. A rule of thumb for aggregate HTTP request measured on an hourly time scale is given by Luotonen [Luotonen 98 p. 296]. In a discussion on capacity planning he suggests the "3/4 rule", where 75% of requests should be expected in 25% of the time.

For sampling time scales of an hour or more, where the diurnal nature of HTTP traffic dominates, there is no reason to believe that traditional time series modelling, such as described in [Chatfield 96], could not be applied to calculating and predicting peak traffic. Such an approach has been shown to be successful for modelling

and forecasting of Web traffic arriving at a server [Bolot 96] and suggested for aggregate user HTTP traffic [Cleveland 00a].

At shorter time scales variation due to the diurnal cycle is less apparent. For example see Figure 2.2 in Section 2.4.1 which shows HTTP request rate observed at three different time scales in one of the traces used in this dissertation. At the shorter time scales, such as one sample per second, aggregate HTTP request rate appears bursty. The mathematical definition of the term "bursty" as it applies to Web traffic is still an open question. Early contributions suggested that aggregate HTTP traffic displayed some properties of self-similarity, in terms of bytes per second [Crovella 96a], and HTTP request rate [Judge 95]. A later publication has limited this to suggesting that Web traffic has elements of long range persistence at limited time scales above hundreds of milliseconds and less than tens of minutes [Cleveland 00b]. Less than this time scale protocol behaviour is dominant and greater than this time scale diurnal variation is dominant. Other recent publications have suggested that correlation in HTTP traffic may reduce under aggregation [Morris 00, Cao 01]. Robert Morris et. al. examined two traces of Web traffic, one from the Harvard University Internet link and the other from a Lucent Bell Lab Internet link [Morris 00]. It was shown that Web traffic, as measured in bytes per 0.1 second intervals, aggregates as smoothly as Poisson traffic even though it is "burstier" than purely Poisson traffic. The lack of correlation in the generation of Web traffic between users is attributed as the cause for the smoothing of traffic under aggregation. Cao also attributes lack of correlation between sources of traffic for a reduction in long range dependence in HTTP connection inter-arrival times in samples of aggregate HTTP request traffic with increasing average connection arrival rate [Cao 01].

The unit of measure in common use for Web cache benchmarking is HTTP requests per second [Luotonen 98 pg. 296]. This is the unit of choice for the Measurement Factory Web Cache benchmarking events [Wessels 99, Rousskov 01]. Occasionally the figure is expressed as HTTP requests per day, or hour. These figures are provided in the expectation that load occurs evenly and that mean request rate per second can be obtained by simply dividing by the appropriate number of seconds [Luotonen 98 pp. 294-297].

In benchmarking the load applied to the proxy under test should have a constant average request rate [Wessels 01 p. 199]. The point of interest in this dissertation for these types of artificial Web traffic load is whether or not the excursions from the average request rate to peak are the same as observed in an actual bursty stream of HTTP request arrivals. A marginal model for HTTP request rate can show whether or not the magnitude of the excursion is consistent with actual traffic. One figure is provided by Gribble from his Home-IP trace [Gribble 97b]. He found peak to mean ratios of 5:1 at the 10 second time scale. For comparison Figure 2.2(c) shows excursions of 2:1 from the mean at the one second time scale for one of the traces used in this dissertation. The estimation of peak request rates is examined in Section 6.1.

2.4 Collecting and Sampling Web Traffic Traces

A number of Web traffic traces were collected for analysis. The traces were either collected by the candidate or collected by others over an eight year period between 1994 and 2002. The traces used for the modelling of HTTP request rate were collected between August 1996 and January 2002. A summary of the data sets is shown in Table 2.1. The trace name marked with the "†" symbol was used in the investigation of single user request rate in Chapter 4. The trace names marked with a "‡" symbol are the ones used for developing, validating and examining the usefulness of the HTTP request rate model in Chapters 3, 5 and 6.

Table 2.1 Summary of Web Traffic Traces

Data Source	Start Date	Duration (days)	Traffic Sources	HTTP Requests (x 10^6)	Average HTTP Requests / Day (x 10^3)
UOW SNRC†	26 Oct. 1994	1071	19	0.54	0.5
Digital‡	29 Aug. 1996	25	17354	24.1	963
Berkeley‡	1 Nov. 1996	18	8377	9.2	514
UNSW 1‡	1 Jan. 1999	181	8322	549	3036
UNSW 2‡	17 Oct. 2001	85	9364	69	n/a[a]

a. Filtered trace was to sparse to calculate average requests per day.

The UOW SNRC trace was collected by this candidate between 1994 and 1997 by monitoring the Web traffic generated by a laboratory of postgraduate students using

an instrumented CERN proxy server at the Switched Networks Research Centre (SNRC). The trace collection process is described in Appendix A. The purpose of the trace was to record the Web browsing behaviour of a constant group of users over an extended period of time. The trace has been used as a starting point for characterisation of Web traffic (Chapter 4).

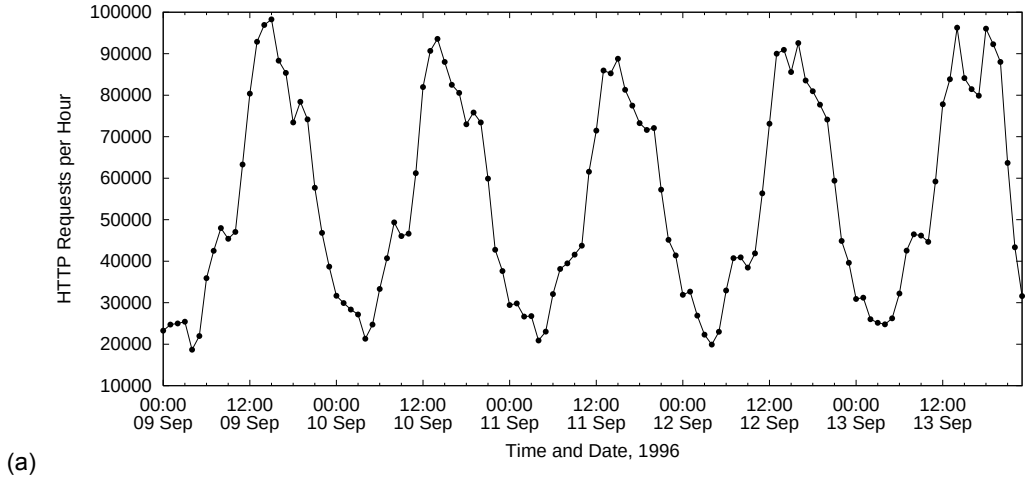
In 1997 Web traffic traces were released from Berkeley [Gribble 97a] and Digital [Kroeger 99]. These traces consist of relatively high volumes of traffic collected independently from each other using different collection methods and over different user populations. These traces and the identification of trace artefacts and outliers are discussed in Appendix B and Appendix C.

In 1999, and again in 2001, the Communications Unit at UNSW agreed to allow access by this candidate to the log files recorded by their campus Web cache. The later provided more recent traces of Web traffic than the two traces from Digital and Berkeley. The collection process, identification of outliers and trace artefacts is discussed in Appendix D and Appendix E. Unfortunately the second trace from 2001-2002 was severely corrupted by missing data in the log files and, although a couple of months of logs were collected, only 247 hours in the trace have data recorded for entire hour periods. The figures for the number of traffic sources and HTTP requests in Table 2.1 for the UNSW2 trace are for these 247 hours only.

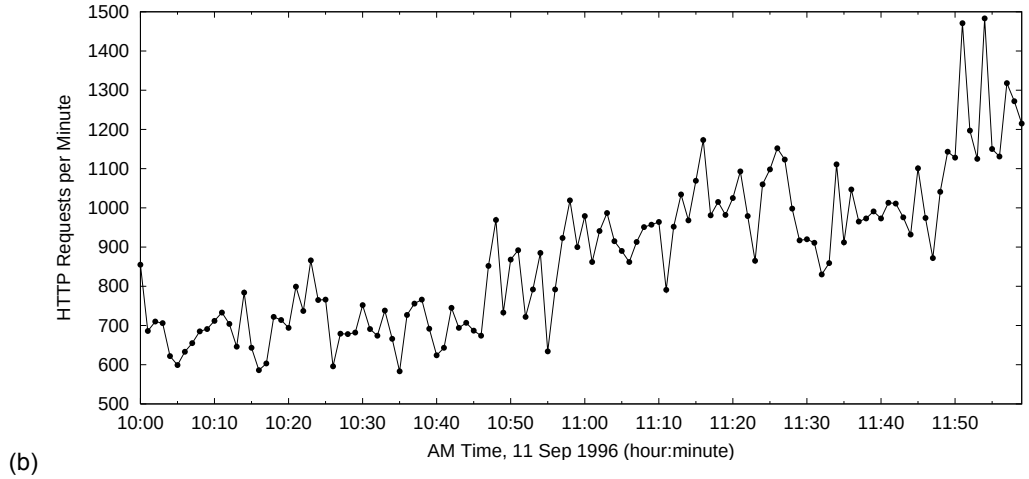
2.4.1 Selecting Samples of Web Traffic

The initial problem in modelling HTTP request rate from observations of aggregate Web traffic is sample selection. The sample duration needs to be long enough to obtain multiple observations of the statistic of interest yet short enough to be considered stationary.

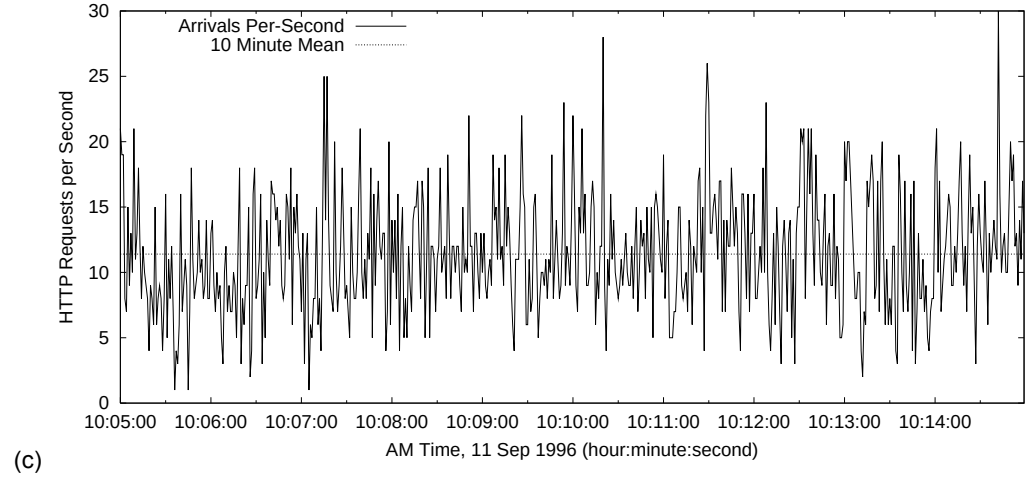
The variability of HTTP request rate at a number of time scales has been documented previously [Gribble 97b, Cleveland 00b] and discussed in Section 2.3.3. At the hour time scale observed aggregate Web traffic is relatively smooth with variation in HTTP request rate correlated with the time of day. At smaller time scales the same traffic is observed to be increasingly bursty. These same characteristics are apparent in the traces of aggregate Web traffic used here. For example, Figure 2.2 shows HTTP request rate at a number of time scales for a portion of the Digital



(a)



(b)



(c)

Figure 2.2 HTTP Request Rate for Web Traffic Observed in a Portion of the Digital Trace for the Time Scale of:
(a) Hour
(b) Minute
(c) Second

trace. At the hour time scale (Figure 2.2a) the predictability is evident with the HTTP request rate showing a clear diurnal cycle and relatively constant hourly rates from one day to the next. The variation in HTTP request rate is large with traffic at a minimum of 19k to 250k HTTP requests per-hour in the early morning and 90k to100k requests per-hour in the early afternoon. Figure 2.2b shows the diurnal cycle is still evident at the minute time scale as the mean HTTP request rate shows an increase over the two hours covered by the plot. Figure 2.2c shows the count of HTTP request arrivals each second for a ten minute period. For the ten minutes shown in the plot the traffic is bursty with significant excursions in HTTP request rate to either side of the ten minute mean. At this time scale it is hard to differentiate between a burst of HTTP requests, that may extend over a number of seconds, and change in mean HTTP request rate due time of the day.

Traffic burstiness at shorter time scales combined with the large diurnal variation in HTTP request rate at longer time sales presents a problem in selecting stationary samples of Web traffic. Capturing the bursty nature of HTTP request rate is a goal of the model. Capturing the variation in HTTP request rate due to the diurnal cycle is not a goal of the model. If the underlying HTTP request rate alters during the sample period due to the diurnal cycle then the sample is considered to not be stationary. The concern is to select samples of HTTP request arrivals that minimise the obvious nonstationarity of the diurnal cycle. Creation of stationary samples by removing a trend (a common procedure in time series analysis [Chatfield 96]) is not an option. The relationship between the number of active Web users and the request rate is of interest and this would be altered by trend removal in HTTP request rate. Instead a number of hours were sampled from each trace where the longer term mean HTTP request rate was approximately constant.

The methodology is to use these selected hours with relatively constant HTTP request rate to identify candidate probability distributions for statistics of interest. The validation and application of the derived model is tested on other hours randomly extracted from each of the traces in Chapter 6.

The Berkeley, Digital and UNSW1 trace were analysed to find groups of three adjacent hours where the number of HTTP request arrivals did not vary by more than a few percent from one hour to the next. The single hour from the middle of each

Table 2.2 Summary of the Hours Extracted from the Berkeley, Digital and UNSW Traces with Assumed Constant HTTP Request Rate

Trace	Hour	Date & Start Time	Number Requests	Number Users
Berkeley	B1	2 Nov. 11:00am 1996	24458	383
	B2	2 Nov. 3:00pm 1996	27747	388
	B3	2 Nov. 10:00pm 1996	32175	392
	B4	5 Nov. 4:00pm 1996	24801	409
	B5	11 Nov. 10:00am 1996	19979	392
	B6	11 Nov. 5:00pm 1996	24611	423
	B7	12 Nov. 11:00am 1996	20459	348
	B8	17 Nov. 11:00pm 1996	25616	355
Digital	D1	29 Aug. 2:00pm 1996	34904	991
	D2	31 Aug. 10:00pm 1996	9824	355
	D3	4 Sep. 2:00pm 1996	90715	1822
	D4	5 Sep. 3:00am 1996	26766	539
	D5	11 Sep. 7:00pm 1996	71627	1392
	D6	15 Sep. 1:00am 1996	19262	409
	D7	22 Sep. 5:00pm 1996	20893	452
UNSW1	U1	10 Feb 1:00pm 1999	221117	1241
	U2	10 Feb 2:00pm 1999	220907	1265
	U3	16 Mar. 10:00pm 1999	182321	814
	U4	22 Mar. 3:00pm 1999	291702	1670
	U5	16 Jun. 12:00pm 1999	288549	1695
	U6	29 Jun. 3:00pm 1999	312445	1579
UNSW2	N1	17 Nov. 9:00pm 2001	230151	831
	N2	24 Nov. 12:00pm 2001	325940	977
	N3	15 Dec. 3:00pm 2001	274681	829
	N4	27 Dec. 2:00pm 2001	206416	612
	N5	3 Jan. 12:00pm 2002	501928	1864
	N6	7 Jan. 8:00pm 2002	192300	755

these three hour series was extracted from the trace under the assumption that the underlying mean HTTP request rate over this sampled hour was approximately constant. Table 2.2 summarises each of the hours sampled from each trace. For the Berkeley and Digital trace the filter found sequences of hours where the hourly HTTP request rate varied by less than 2.5% from hour to hour. For the longer UNSW1 trace this was tightened to 1%. These percentage values are a compromise between

minimising the hourly variation in request rate and providing a number of sample hours from each trace.

A series of three adjacent logged hours was relatively rare in the sparse UNSW2 trace. In this trace hours of relatively constant mean request rate were found using linear regression. The approximate percentage rate of change in request rate per second over each hour was found using Equation 2.1 where i is the y-axis intercept and s is the slope.

$$\%\ \text{rate of change} = \frac{3600s}{i + 1800s} \times 100 \qquad \textbf{(Eqn 2.1)}$$

Six hours were selected from the trace which had a change in HTTP request rate over the hour of 0.7% or less.

2.5 Conclusion

This chapter has examined work related to the proposed marginal distribution model of aggregate HTTP request rate and applications to which the model may be used. The first area examined was the Web itself and the sources of change in network traffic generated by the Web. The second area examined was the contributed work by others in characterisation of Web traffic.

Since widespread introduction in 1992 the nature of the Web has changed including changes in network traffic. Three areas of change that may impact on models of Web traffic are:

- Migration in use of HTTP from version 1.0 to 1.1
- The change in the number of embedded objects in a typical Web page
- Change in the ratio of non-human initiated HTTP requests

The first two sources of change are taken into account by using traffic traces collected over periods of time in which these changes have taken place. If the developed models and methods are shown to hold over all the traces then they can be considered as independent from this change. The third area of change is not taken into account. Differentiating between human and non-human initiated requests is

difficult. Obvious non-human initiated requests have been removed from trace files before other analysis has taken place. This has also be done by others [Choi 99, Mah 97]. The model developed is assumed to reflect Web traffic predominantly generated by the actions of humans.

There are a large number of contributions discussing various aspects of HTTP request arrivals. None of the contributions examined provide a reliable or useful model of the marginal distribution of aggregate HTTP request rate. Most of the models are of the traffic generated by a single Web client [Abrahamsson 00, Arlitt 95, Barford 98a, Choi 99, Deng 96, ETSI 98, Hlavacs 99, Mah 97, Reyes-Lecuona 99]. A smaller number of contributions have examined models for aggregate HTTP traffic [Cleveland 00a, Kant 99, Molina 00, Wessels 99].

The problem with existing models is largely one of complexity. The single user models do not lend themselves to tractable descriptions of aggregate traffic.

The contributions that closest resemble the model desired here are those of Cao and Cleveland [Cao 01, Cleveland 00a]. The difference is that they have modelled HTTP request inter-arrival and have looked at sample periods during which it was assumed there were a constant number of traffic sources. In contrast the desired model here is of counts of HTTP request arrivals per second which can be directly applied to the task of estimating peak HTTP request rate. Sampling periods are longer and carefully selected to have a constant HTTP request rate rather than a constant number of traffic sources.

The proposed model of the marginal distribution of aggregate HTTP request traffic has a number of potential applications. The applications explored in this work concern estimation of peak HTTP request rate, use in developing new rules of thumb concerning HTTP request rate and in the assessment of HTTP request arrival models proposed by others.

The rate of HTTP requests generated by a population of Web users have a significant diurnal cycle. At longer sampling periods it looks likely that traditional time series modelling, such as [Chatfield 96], could be successfully applied to estimating request rate. Time series modelling has already been shown to work with HTTP request rate and traffic volume on a Web server [Bolot 96].

At shorter sampling periods Web traffic has been observed to be bursty with contributions discussing observed self similarity [Crovella 97] and long range dependence [Cao 01, Cleveland 00a]. Long range dependence in Web traffic may reduce with aggregation as individual sources of traffic appear to be independent [Cao 01, Morris 00]. The independent nature of sources has been attributed to the "smoothing" of Web traffic under aggregation [Morris 00] and to reducing long range dependence in HTTP request inter-arrival times and HTTP response packet size [Cao 01].

3. Web Traffic and the Poisson Distribution

The analytical simplicity and widespread use of the Poisson probability distribution in modelling telecommunications traffic make it an obvious candidate to consider when modelling Web traffic. Unfortunately HTTP request arrival is not Poisson. This was first suggested in 1994 when it was shown that the Poisson distribution did not describe TCP connection arrivals for Web traffic [Paxson 94a]. At that time Web network traffic consisted of HTTP version 0.9 or 1.0 and a TCP connection was a reasonable approximation for a HTTP request arrival. The ongoing consensus is that the Poisson distribution is not a good model for HTTP request arrival. Further the only model to use the Poisson distribution discussed in Chapter 2, PolyMix-1 [Wessels 99], has now been superseded with the non-Poisson based PolyMix-4 model [Rousskov 01]. However the Poisson distribution does apply to other aspects of Web traffic; as a limiting distribution for HTTP request arrivals [Cao 01, Cleveland 00a] and a model for HTTP session arrivals [ETSI 98, Judge 95, Reyes-Lecuona 99]. In addition it is shown here that the number of active Web clients per second has a Poisson marginal distribution.

The work by Cao and Cleveland shows that HTTP request inter-arrival times approach both an exponential marginal distribution and independence with increasing aggregation [Cao 01, Cleveland 00a]. This indicates that HTTP request arrivals approach Poisson as a limiting distribution as aggregation increases. However, when measured as counts of HTTP request arrivals per second no such trend is found here (Section 3.1). Cleveland explains that the correlation structure of counts of HTTP request arrivals prevents the Poisson limit showing up in counts of arrivals [Cleveland 00a]. In similar work looking at IP packets Cao suggests that IP packet counts should tend towards a normal marginal distribution with increasing aggregation [Cao 02b]. The tendency of the marginal distribution of HTTP request rate towards normal is examined in Section 5.3.

The Poisson distribution is already used to model HTTP session arrivals [ETSI 98, Judge 95, Reyes-Lecuona 99]. The Poisson distribution is also a good model for TELNET and FTP session arrivals [Paxson 94a]. There is some published work

matching aspects HTTP session arrivals to the Poisson distribution [Feldmann 98a, Hlavacs 99, Judge 95, Reyes-Lecuona 99]. It is confirmed here that the Poisson distribution is a good model for HTTP session arrivals.

The fact that HTTP session starts are Poisson suggests that the marginal distribution of sessions in progress may also be Poisson. This comes from a well known result in queuing theory that the number of customers in service in a queue with infinite servers, finite service time variance and Poisson arrivals has a Poisson marginal distribution irrespective of the actual service time distribution [Kleinrock 75]. However the number of sessions in progress may not be very informative about which source is actually generating traffic at an instant in time. A more useful result would be for the number of traffic sources actually generating one or more HTTP requests during a given second. Looking at each of the four traffic traces it is found that this has a Poisson marginal distribution. This result is used in the model developed for the marginal distribution of HTTP request rate in Chapter 5.

This first section of this chapter looks at the non-Poisson nature of aggregate HTTP request arrival. The second section looks at the Poisson nature of HTTP sessions arrivals. The third section shows that the number of active Web clients per second has a Poisson marginal distribution.

3.1 The Non-Poisson Nature of Aggregate HTTP Request Rate

Paxson tested TCP connection events against a Poisson arrival process assumption for a variety of TCP application traffic streams [Paxson 94a, Paxson 94b]. A number of traffic traces were divided up into both one hour and ten minute time periods and each was tested for Poisson arrivals. The traces used in their work included small amounts of early (circa 1994) Web traffic. At the TCP packet level they found that, for this small amount of Web traffic, TCP connection arrival events were correlated and the inter-arrival times did not have exponential marginal distribution.

A similar method was used to examine the four traces of aggregate HTTP traffic. The entire trace, for each of the four aggregate traffic traces, was divided into one

hour intervals and the Lexis ratio was calculated for HTTP request arrivals per second. The Lexis ratio is the variance divided by the mean [Law 91 pp. 358-360]. For the Poisson distribution, where the mean and variance are equal, the Lexis ratio is one. The division of each trace into one hour periods makes the assumption of a constant HTTP request arrival rate over the hour. To test this assumption 15 minute intervals were also examined.

Details of the number of time periods examined for each trace is shown in Table 3.1. Scattergrams of the Lexis ratio are shown in Figure 3.1. A line representing the expected value of the statistic should the marginal distribution be Poisson is also shown in each plot. A small number of time periods (less than 2% of those calculated) had Lexis ratio figures higher than those shown in Figure 3.1. On the plots the vertical axis range was constrained to show the bulk of the observed statistics and the difference from Poisson rather than all the statistics.

Table 3.1 Details of the Time Periods Examined from Each Trace

Trace	Period Length (min.)	Number Time Periods	Mean Number Traffic Sources	Mean Number of Requests per Traffic Source
Berkeley	60	422	314	72
Digital	60	592	833	47
UNSW1	60	3581	730	175
UNSW2	60	247	979	291
Berkeley	15	1690	157	35
Digital	15	2369	349	29
UNSW1	15	14324	423	76
UNSW2	15	988	545	130

All four traces show clusters of points above that which would be expected for a Poisson marginal distribution. No trace shows a trend towards a possible Poisson marginal distribution with increasing aggregation of traffic sources. In none of the traffic traces does there appear to be evidence that the arrival process may be a simple Poisson process. All the traces show a variance in HTTP request rate greater than the mean. The difference in the Lexis ratio between the traces may be due to the greater number of requests made by each traffic source by the later traces. Table 3.1 shows that on average the sources in the UNSW2 trace generate many more HTTP requests per source than traffic sources in the earlier traces.

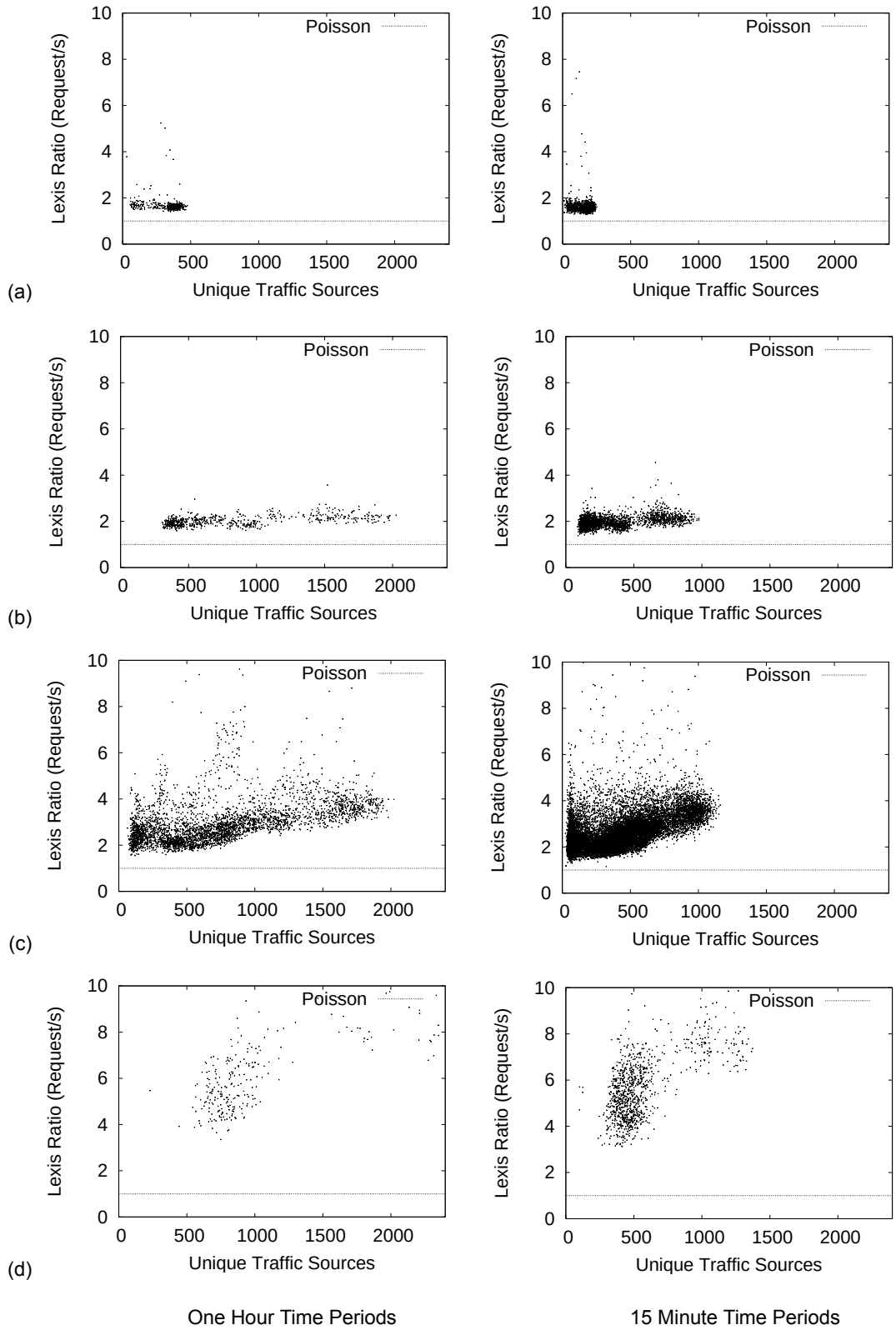


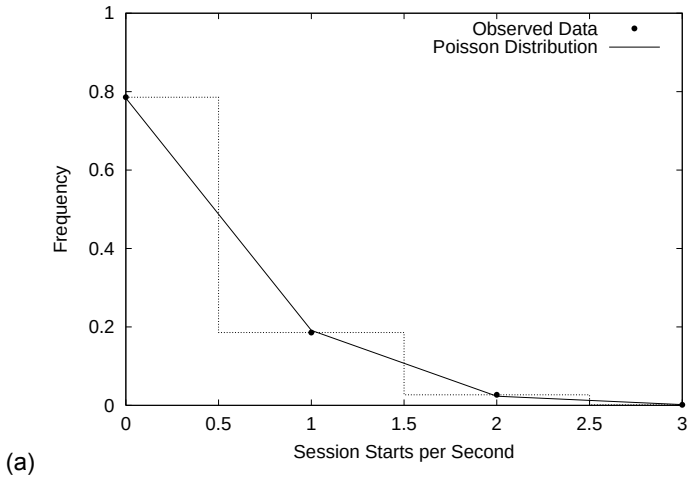
Figure 3.1 Observed Lexis Ratio for HTTP Request Arrivals Versus Unique Traffic Sources: (a) Berkeley Trace, (b) Digital Trace, (c) UNSW1 Trace, (d) UNSW2 Trace

3.2 The Poisson Nature of User Browsing Sessions

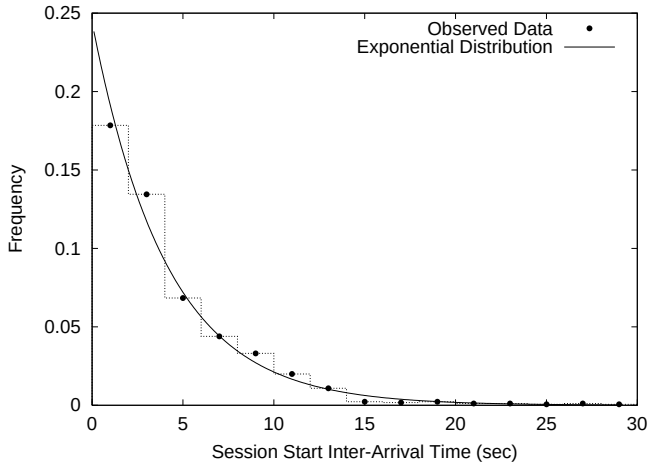
Web users engage in sessions of browsing activity. This behaviour is easily observed in practice and has been reported previously in literature [Catledge 95, Judge 95, Reyes-Lecuona 99]. Without direct observation of a user's browsing activity determining the start and end of a particular browsing session is difficult. Even instrumentation of Web browsing software may not provide a definitive record of a user session. In the trace collected by Lara Catledge et. al. it was found that users will leave Web browsers open and idle for periods of time returning for a number of browsing sessions with the one instance of the browser [Catledge 95]. For analysis user sessions are identified by selection of a suitable idle time between subsequent HTTP requests from the one source. Catledge choose to delimit session boundaries with 25.5 minutes of user inactivity with the browser. Reyes-Lecuona used a 30 minute gap between subsequent HTTP requests from the same Web client to delimit a session [Reyes-Lecuona 99]. Other sessions delimiters used include a 100 minute silence period [Deng 96], 15 minutes [Abrahamsson 00] and 30 minutes [Hlavacs 99].

Three of the aggregate traffic traces were filtered for HTTP sessions using a 20 minute delimiter. The UNSW2 trace was excluded as HTTP sessions could not be reliably identified in this sparse trace. The session starts during the sample hours (the hours listed in Table 2.2) are clearly Poisson. For example, Figure 3.2 shows a Poisson marginal distribution, exponential inter-arrival times and negligible auto-correlation for HTTP session arrivals in the hour *D1* in the Digital trace. Figure 3.2(c) is a correlogram showing the first 20 lags of the autocorrelation for the number of HTTP session arrivals per-second for hour *D1*. The correlogram includes lines showing the 95% confidence lines at $\pm 1.96\sqrt{n}$, where n is the size of the data set under examination. The 95% confidence lines are indications of the correlation of the data. For 20 lags the correlogram is assumed to show negligible auto-correlation even if one lag has magnitude greater than the 95% value [Chatfield 96].

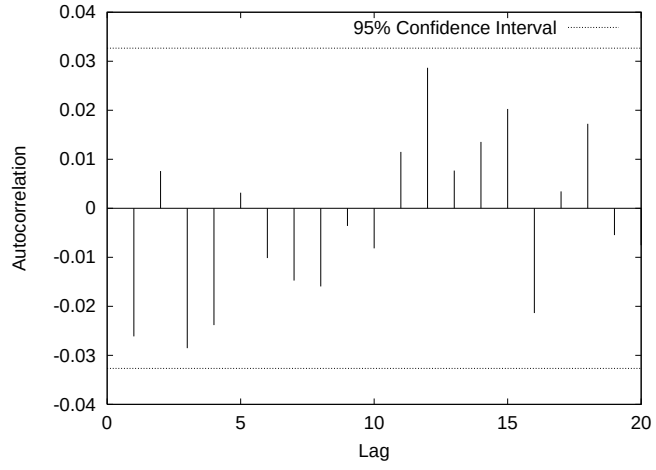
With the exception of the correlogram the other sampled hours show similar Poisson session arrival properties to the hour *D1* in Figure 3.2. Plots for the marginal distribution, correlogram and inter-arrival time of session arrivals for all the sampled hours are shown in Appendix G. The correlograms of six of the 21 sampled



(a)



(b)

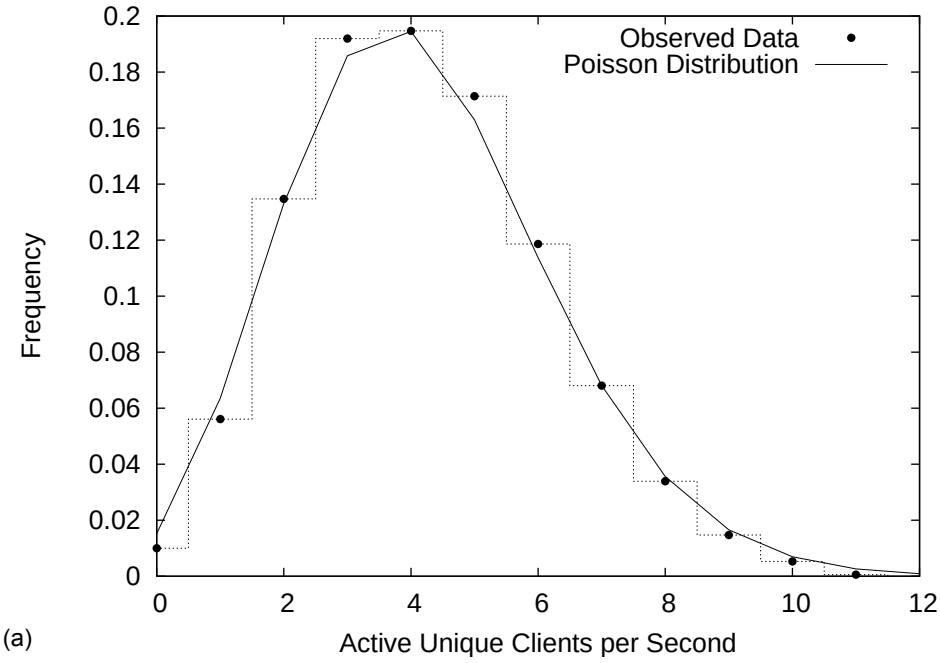


(c)

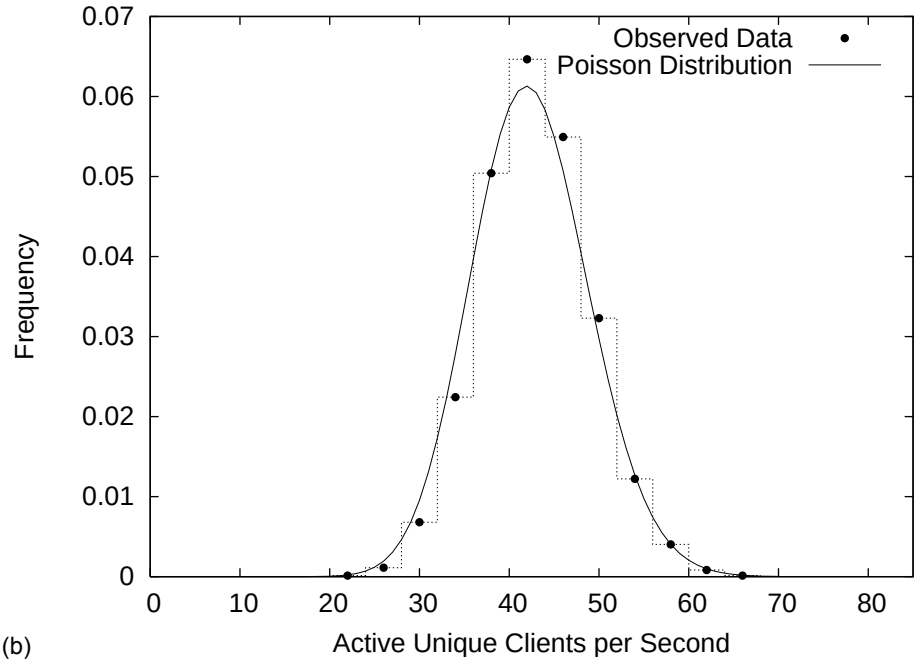
Figure 3.2 Three Properties Showing the Poisson Nature of Session Arrivals for Hour *D1* (2:00pm 22 August 1996 in the Digital Trace):
(a) Poisson marginal distribution of session arrivals per second
(b) Exponential marginal distribution of session inter-arrival times
(c) Negligible autocorrelation in session arrivals per second

hours have more than one lag with magnitude greater than the 95% confidence interval. However five of these hours (*B1, B5, D3, D5, U3*) have these lags randomly placed and the number of lags is small (two for *D3* and *D5*, three for *B1* and *U3* and four for *B5*). The graphs can still be interpreted as showing negligible autocorrelation in session arrival times. The other correlogram exception is the hour *D2* from the Digital trace. Hour *D2* shows a Poisson marginal distribution and exponential inter-arrival times for session arrivals but shows clear positive autocorrelation in the number of session arrivals per second. Closer examination of the session arrivals during the hour found that the correlation is entirely due to a large number of arrivals, over 5% of the entire arrivals for the hour, in a single ten second period. This burst of arrivals was unrepresentative of the remainder of the traffic in that hour and for all the other hours sampled from the three traces. Once this ten second period was removed the session arrivals show negligible autocorrelation.

Aspects of the Poisson nature of HTTP session arrivals have been previously reported [Feldmann 98a, Judge 95, Hlavacs 99, Reyes-Lecuona 99]. Anja Feldmann et al. looked at the modem call log for an ISP and equated a user modem connection to the ISP as a Web session [Feldmann 98a]. It was found that the modem connection arrivals were independent with an inter-arrival time exhibiting an “exponential shaped” tail. Reyes-Lecuona looked at traces of Web traffic gathered from the University of Málaga in 1998 and from Alcatel Alsthom in Madrid in 1997 and reported that an assumption of Poisson arrivals for session start time was valid [Reyes-Lecuona 99]. No supporting figures were shown. Hlavacs looked at a trace of Web traffic gathered from a proxy server at the University of Vienna in 1999. He found a good fit for the exponential distribution with session start inter-arrival times [Hlavacs 99]. A contribution from this author examined traces of aggregate Web traffic collected in 1994 and 1995 and showed a Poisson marginal distribution for HTTP session arrivals in a number of Web traffic traces [Judge 95]. The figures in Appendix G show comprehensively the Poisson aspect of HTTP session starts over a more diverse set of traffic traces than has been previously reported.



(a)



(b)

Figure 3.3 Histogram of the Number of Unique Active Web Clients Observed Each Second Compared to the Poisson Distribution for:
(a) Hour *D7* (5:00pm 22 September 1996 in Digital Trace)
(c) Hour *U6* (3:00pm 29 June 1999 in UNSW Trace)

3.3 Poisson Distributed Active Users

As stated in the introduction to this chapter queuing theory suggests that the number of HTTP sessions in progress may have a Poisson marginal distribution. However, HTTP sessions in progress does not easily translate into a model of HTTP request rate as each session contains periods of silence. An alternative is to consider the number of unique Web clients generating HTTP requests. The question is whether this would have a Poisson marginal distribution.

The number of unique active Web clients, those clients generating one or more HTTP requests, were obtained for each second from each of the sample hours from all four aggregate Web traffic traces. It turns out that the number of active unique Web clients does have a Poisson marginal distribution. For example, Figure 3.3 compares the marginal distribution of the number of active Web clients per second for the hour *D7* (with a low number of active users per second) and the hour *U6* (with a higher number of active users per second) with the Poisson distribution. A similar good fit was found for all the other sample hours listed in Table 2.2, the plots for all the sample hours are shown in Appendix H.

3.4 Conclusion

This chapter looked at the applicability of the Poisson distribution for modelling some aspects of Web traffic. Measured as counts of HTTP request arrivals per second that aggregate HTTP request arrivals do not have a Poisson arrival process. In addition there is no evidence of a trend towards Poisson for increasing levels of aggregation.

In contrast to HTTP request rate it has been shown that the arrival process of HTTP sessions is Poisson. The Poisson result for HTTP session arrival implies a Poisson marginal distribution for sessions in progress. Taking this a step further, each of the sampled hours from the four traces of aggregate traffic was examined looking at the distribution of the number of sources actually generating traffic (one or more HTTP requests) per second. It is found that this has a Poisson marginal distribution. In terms of developing a marginal distribution for HTTP request traffic this is an inter-

esting result. The next couple of chapters build on this result in the development of a proposed marginal distribution for HTTP request rate.

4. Single User HTTP Request Rate

In this chapter the HTTP request rate of a single person browsing the Web is described. It is found that the number of HTTP requests issued by someone, in an hour in which they generate at least one request, has a geometric distribution. The number of HTTP requests generated in a minute and a second is also reasonably approximated by a geometric distribution. This result in based on the analysis of a three year trace collected by the candidate at a postgraduate lab at UOW.

The purpose of the work in this chapter is to suggest a model that may be appropriate for the request rate of individual Web users. The idea is to then combine this result with the Poisson marginal distribution of active users result from the previous chapter to produce a model for aggregate Web traffic.

The analysis of single user HTTP request rate requires a trace in which the traffic generated by a particular individual can be identified. Usually traces of Web traffic identify the traffic source only by IP address which indicates little about the individual generating traffic. The source IP address may itself be a proxy server in use by a number of other client machines.

A small trace which did identify unique users was collected by Lara Catledge et. al. using instrumented Web browsers in 1994 [Catledge 95]. This trace is available on request and is comprised of 167 users over a three week period. Unfortunately, the traffic of individual users was not collected for a long enough duration to enable modelling HTTP request rate. A trace suitable for this purpose was collected by this candidate by logging the Web traffic generated by SNRC at UOW. Details of the trace collection process can be found in Appendix A.

The SNRC trace is a log of the Web traffic generated by the browsing behaviour of a small user population in a university postgraduate laboratory collected over a three year period. The long duration of the trace over a relatively constant user population makes it possible to extract data suitable for examining per user HTTP request rate statistics.

The rest of this chapter looks at the SNRC trace and the number of HTTP requests generated by single users over one hour, one minute and one second time periods. A

statistical goodness-of-fit (GOF) test is used to compare a number of probability distributions with the distributions of HTTP requests per hour. The GOF test indicates that the geometric distribution is a suitable model. Histograms of the number of HTTP requests per user over smaller time periods, one minute and one second, are compared to the probability mass function (PMF) of the geometric distribution. The plots indicate that the geometric distribution is also a good model for the number of user HTTP requests in these time periods also.

4.1 The SNRC Trace

The SNRC trace records the Web traffic generated by 19 users over a three year period from October 1994 to October 1997. The collection details are given in Appendix A. If the number of HTTP requests generated by each user is counted over every hour of the trace then for the majority of traced hours each user generated no traffic at all. The ratio of hours with traffic versus those without varies from user to user with influences from many factors outside the scope of this study on Web traffic. For example, some users attended the lab part time, other users used the Web heavily for recreational use outside work hours and others used it only sporadically. Attempting to model the likelihood of whether or not an individual user was accessing the Web in a given time period is not practical over such a small user population. Results on the randomness of the number of users browsing the Web at a given time was discussed in Chapter 3. A more practical use of the trace is to suggest models for the number of HTTP requests generated by individual users during the hours in which they were using the Web.

An “active hour” is defined as an hour in which a user generates at least one HTTP request. Two time periods were extracted from the SNRC trace, one at the start and the other at the end of the trace. Both time periods are of just long enough duration that most users have 100 or more active hours recorded. The reason for examining two time periods is that the underlying request rate of each user increased over the duration of three year trace. The two periods were selected as a compromise between minimising the effect of the non-stationarity of rising mean request rate and obtaining samples large enough for statistical analysis. The first period was from 26 October 1994 until 16 May 1995, the second period was from 30 June 1997

until 1 October 1997. The second period was shorter because not only did HTTP requests per hour increase between 1994 and 1997 but the users accessed the Web more often as well. Details of the observed traffic in these periods is given in Table 4.1. Two users *Alice* and *Mike* generated traffic in both periods. For the rest of this analysis their trace names are prepended with *"_94"* and *"_97"* to indicate the relevant sample period.

Table 4.1 The Sixteen Users with More Than 100 Active Hours in Each Sampled Period

Anonymous User Name	Time Period	Number of Active Hours	Number HTTP Requests per Hour	
			Mean	Std. Dev.
Alice_94	1994/5	133	44.9	56.5
Bobby	1994/5	117	14.8	14.4
Jan	1994/5	230	36.6	38.5
Marsha	1994/5	126	17.4	23.1
Mike_94	1994/5	130	14.9	13.8
Peter	1994/5	107	20.2	21.1
Sam	1994/5	216	15.7	18.2
Alice_97	1997	111	72.4	108.6
Siegfried	1997	124	66.1	72
Cindy	1997	106	40.6	41.8
Agent99	1997	126	41.7	56.5
Greg	1997	172	86.3	102.3
Chief	1997	228	104.8	130.7
Larabee	1997	159	96.5	81.6
Mike_97	1997	152	33.8	58.4
Agent13	1997	300	183.4	156.7

4.2 HTTP Requests Per Active Hour

Table 4.1 shows the rise in the mean number of HTTP requests generated by users in an active hour over the two year interval between the extracted trace periods. The mean request rate across the seven users for an active hour in the 1994/5 trace period was 23.5 HTTP requests, over the nine users in the 1997 trace period it was 80.6 HTTP requests. The two users, *Alice* and *Mike*, who had traffic recorded in both trace periods both approximately doubled the number of HTTP requests generated in an active hour between 1994/5 and 1997.

Table 4.1 also shows why the traffic for all the users could not simply be combined to create an idealised "average user" before fitting probability distributions. The mean and standard deviation for the number of HTTP requests generated by each user in an active hour varies significantly between users. For example *Marsha* in the 1994/95 trace period generated on average only 15.7 HTTP requests per active hour while *Agent13* in the 1997 trace period generated on average 183.4 HTTP requests per active hour.

A number of probability distributions were compared to the hourly HTTP request rate of each user during active hours. A good fit was found with the geometric distribution which is considered the discrete analog of the exponential distribution [Johnson 92 p. 201]. The probability mass function (PMF) used in this dissertation for the geometric distribution is given in Equation 4.1 [Larsen 86]. In other publications this form is sometimes called the shifted geometric distribution [Johnson 92].

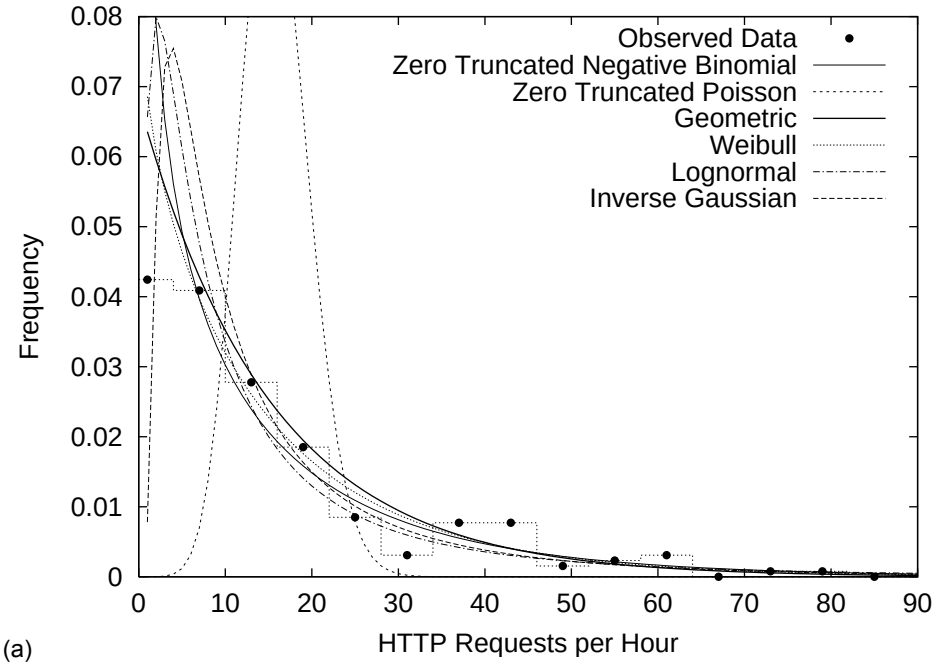
$$Pr[X = x] = p(1-p)^{x-1}, \qquad 0 < p < 1 \qquad \text{for } x = 1, 2, 3, \ldots \qquad \textbf{(Eqn 4.1)}$$

A graph comparing the histogram of the number of HTTP requests per active hour for users *Sam* and *Agent13* against a number of probability distributions is shown in Figure 4.1. The corresponding graphs for all the other users listed in Table 4.1 are shown in Appendix I.1. The number of requests per active hour was compared to the geometric distribution using the Chi-Square GOF test [Law 91 pp. 382-387]. The results are shown in Table 4.2.

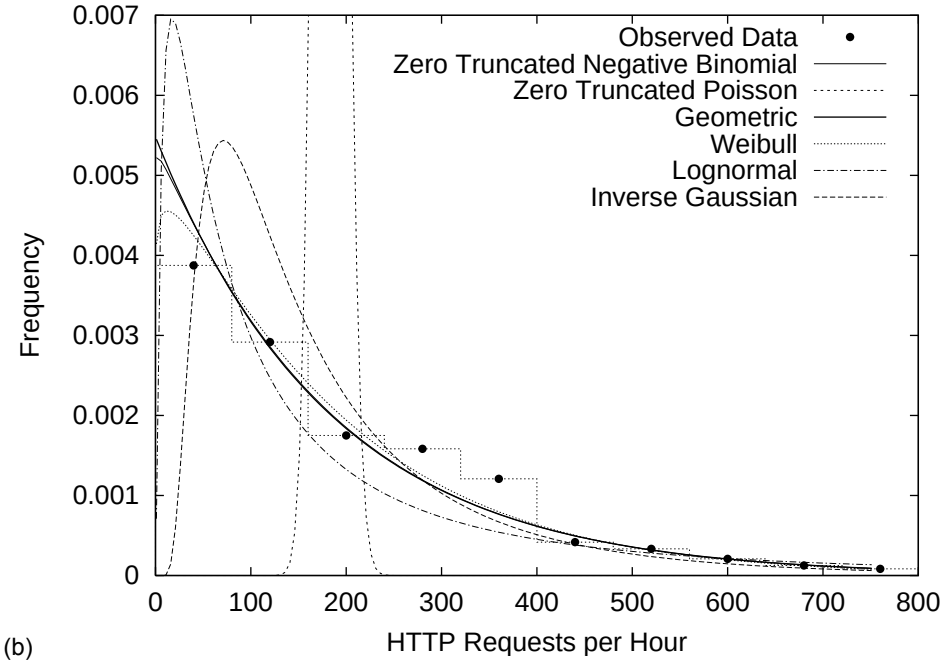
Table 4.2 also shows the GOF test results comparing the observed data against a number of other probability distributions:

- The continuous inverse Gaussian distribution [Johnson 94 Ch. 15] was tested because it has been used previously to model the number of successive HTTP requests from a single user to a Web site [Huberman 98].

- The geometric distribution is a special case of the negative binomial distribution [Johnson 92 Ch. 5]. Since the definition of an active hour is an hour in which at least one HTTP request is issued the observed user data was compared to the zero truncated negative binomial distribution (Appendix F.1). The shifted negative binomial distribution (Appendix F.2) was also considered but the zero trun-



(a)



(b)

Figure 4.1 HTTP Request Rate per Active Hour Compared to a Number of Probability Distributions for the Users;
(a) *Sam*
(b) *Agent13*

cated version proved better fit at this time scale. Prior to the wide adoption of the Web the shifted negative binomial distribution had been used to model the total path length (number of hypertext links followed) for users of hypertext applications [Qiu 94].

- The continuous Weibull distribution [Johnson 94 Ch. 21] is a distribution that includes the exponential distribution as a special case when the shape parameter equals one.

- The continuous Gamma distribution [Johnson 94 Ch. 17] also includes the exponential distribution as a special case.

The last three distributions; the zero truncated negative binomial, Weibull and gamma distributions, were tested to see if a more general family of distributions was a better fit than the Geometric distribution.

Users *Alice_97* and *Greg* both had a number of outliers removed before performing the GOF test. User *Alice_97* had an abnormally large number of HTTP requests in one single hour period which was not representative of the rest of the collected data. User *Greg* had walked away from their browser while it was accessing a Web page that periodically reloaded nearly every 30 seconds. A seven hour period of exactly 118 HTTP requests per hour was removed before performing the goodness of fit test.

Nine out of the sixteen user traces pass the Chi-Square GOF test at a 5% level of significance with the geometric distribution. This is a good fit for the geometric distribution. An extra four users pass the test against the more general truncated negative binomial distribution and Weibull distributions. Nine users passed the test against the Gamma distribution. Just one user trace passed the test with the inverse Gaussian distribution. Not shown in Table 4.2 were results for GOF tests against the lognormal distribution [Johnson 94 Ch. 14] and zero truncated Poisson distribution (Appendix F.3). Neither of these distributions passed against any of the users traces.

In the Chi-Square GOF test the observed data is divided into histogram bins and compared with the expected number of observations given by the proposed statistical distribution. For each of the seven users that failed the Chi-Square GOF test with the geometric distribution the main area of discrepancy was that the geometric dis-

Table 4.2 Results of the Chi-Square Goodness-of-Fit Test for the Number of HTTP Requests Generated by Individual Users in the SNRC Trace

User	χ^2 Goodness of Fit Test				
	Geometric	Zero Truncated Negative Binomial	Inverse Gaussian	Weibull	Gamma
Alice_94	Fail	Pass	Fail	Pass	Pass
Bobby	Pass	Pass	Fail	Pass	Pass
Jan	Pass	Pass	Fail	Pass	Pass
Marsha	Pass	Pass	Fail	Pass	Fail
Mike94	Pass	Pass	Fail	Fail	Fail
Peter	Fail	Pass	Fail	Pass	Pass
Sam	Fail	Pass	Fail	Pass	Pass
Alice_97	Fail	Pass	Fail	Pass	Pass
Siegfried	Pass	Fail	Fail	Pass	Pass
Cindy	Pass	Pass	Fail	Pass	Pass
Agent99	Fail	Fail	Fail	Pass	Pass
Greg	Pass	Pass	Fail	Fail	Fail
Chief	Pass	Pass	Fail	Pass	Fail
Larabee	Fail	Pass	Fail	Fail	Fail
Mike97	Fail	Fail	Pass	Pass	Fail
Agent13	Pass	Pass	Fail	Pass	Fail
Goodness of Fit Test Results (out of 16 tests)	9 Pass	13 Pass	1 Pass	13 Pass	9 Pass

tribution under estimated the number of hours in which only a small number of HTTP requests occurred. This is not evident in plots comparing observed request rate to the geometric PMF (Figure 4.1 for example) but was noted in the output of the routine used to perform the GOF test. The extra parameter in the zero truncated negative binomial and the Weibull distributions allowed for some extra weight towards the lower request rate end of the distribution. In contrast the inverse Gaussian distribution under estimated the frequency of hours with low request rates even more than the geometric distribution.

The GOF test results in Table 4.2 show the zero truncated negative binomial, the Weibull and the gamma distributions are also a good match to the observed data. All these distributions include the geometric (or exponential) distributions as a special case. The test against these distributions shows that a more general distribution only

matches the observed data marginally better than the one parameter geometric distribution.

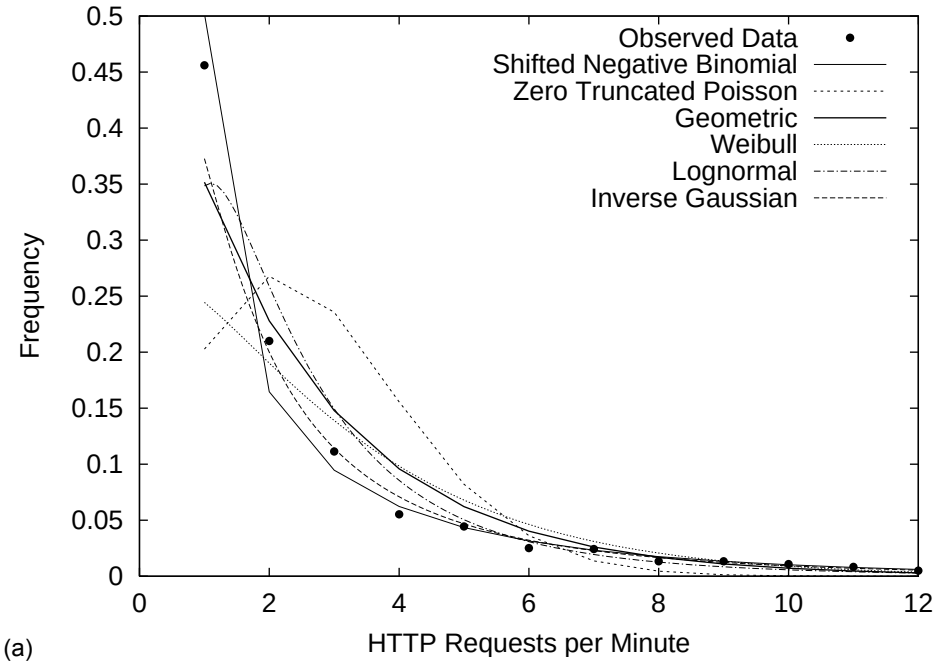
The single parameter geometric distribution was chosen as a reasonable model for the single user hourly request rate. It had the advantage that it was a discrete distribution and is simpler than the zero truncated negative binomial.

4.3 HTTP Requests Per Active Minute and Second

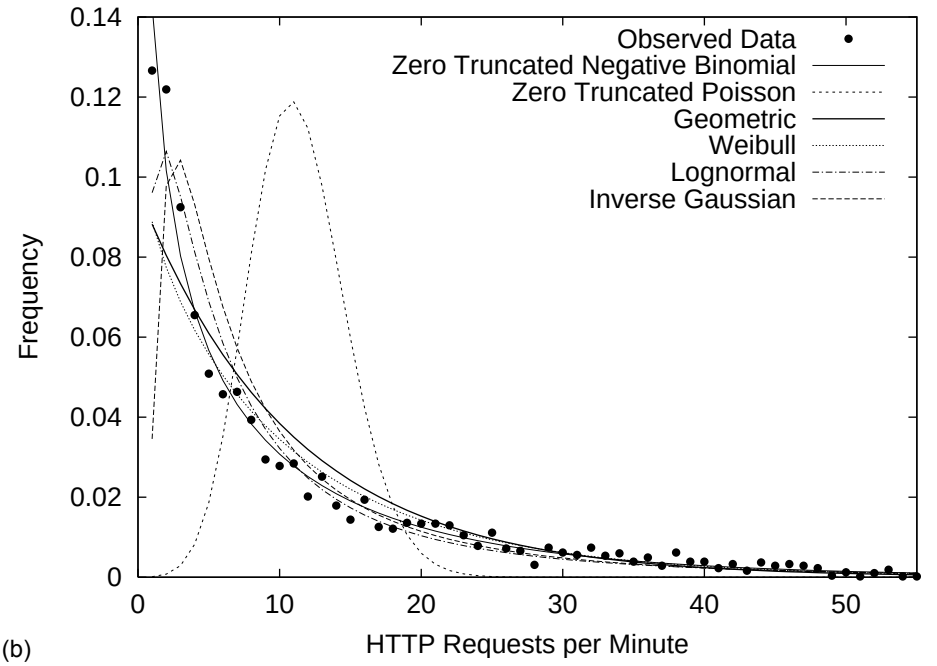
A property of the geometric distribution is that it is infinitely divisible [Johnson 92 pg. 202] and so it may also be a model for request rates during shorter time periods. The terms “active minute” and “active second” are defined similar to that of an active hour, that is, at least one HTTP request observed in the time interval. Using the same user trace data as listed in Table 4.1 the histograms of requests per active minute and active second were compared to the zero truncated negative binomial, zero truncated Poisson, geometric, Weibull, lognormal and inverse Gaussian distributions. The graphs for users *Sam* and *Agent13* are shown in Figures 4.2 and 4.3. The corresponding graphs for all the users are shown in Appendix I.2 and Appendix I.3 respectively. For time periods with low mean HTTP request rate the zero truncated negative binomial distribution has been replaced by the shifted negative binomial distribution. This is because the algorithm used to generate maximum likelihood (ML) estimators [Wyshak 74] failed to converge on sensible parameter estimates and/or the shifted version of the distribution provided a better fit to the data. For some time periods the algorithm used to find the ML estimators for the Weibull distribution [Law 91 p. 334] failed to find sensible values and the distribution was not plotted.

The continuous Weibull, lognormal and inverse Gaussian distributions were plotted to see if they provided a markedly better fit to the observed data than any of the discrete distributions. They did not and, given the discrete nature of the data, they have not been considered as potential models.

Figure 4.2 shows that the geometric distribution retains a good fit to HTTP request rate for users in an active minute. At this time scale the zero truncated Poisson distribution is a bad fit at higher request rates, for example user *Agent13* in Figure 4.2.

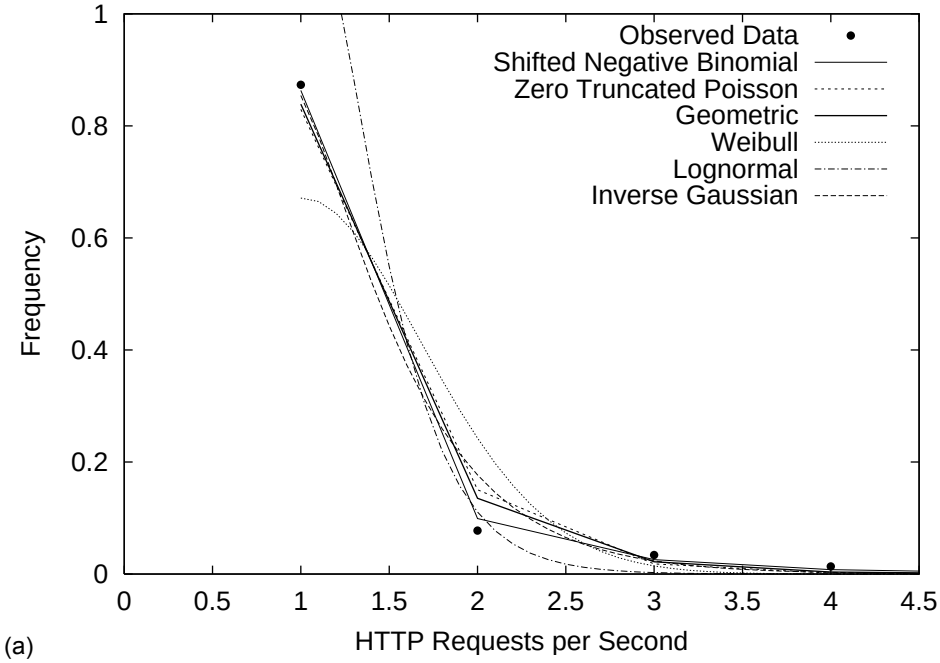


(a)

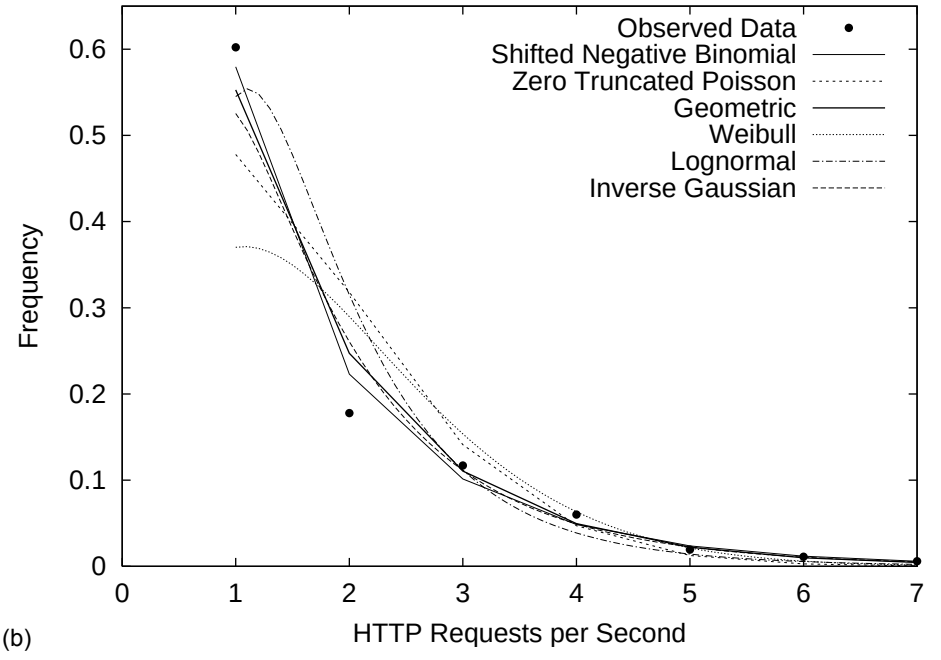


(b)

Figure 4.2 HTTP Request Rate per Active Minute Compared to a Number of Probability Distributions for the Users;
(a) *Sam*
(b) *Agent13*



(a)



(b)

Figure 4.3 HTTP Request Rate per Second Compared to a Number of Probability Distributions for the Users;
(a) *Sam*
(b) *Agent13*

Graphically there is not much difference between the geometric and the shifted negative binomial although the negative binomial had a tendency to under estimate the left hand side of the distribution for some of the users (for example, *Alice_94*, *Jan*, *Bobby*, *Marsha*, *Mike_94* and *Peter* in Figure G.2 (Part 1).

Figure 4.3 shows the geometric distribution also remains a good fit to user HTTP request rate at the per-second time scale. For some of the users there is not much difference between the discrete distributions and they overlap each other on the plot (for example, *Alice_94* in Figure G.3 (Part 1). At higher request rates the shifted negative binomial was not a good fit (for example *Alice_97* in Figure G.3 (Part 1). At this time scale there is not much of a difference between the geometric and zero truncated Poisson distributions although the zero truncated Poisson did under estimate the left hand side of the distribution at higher request rates (for example, *Chief*, *Greg* and *Larabee* in Figure G.3 (Part 1)

From the plots it is seen that the geometric distribution provides a good model for HTTP request rate at the one minute and one second time periods.

4.4 Conclusion

In conclusion the geometric distribution is a good model of per user HTTP request rate for time intervals where a user makes at least one HTTP request. At the hour time scale the geometric distribution passes a Chi-Square GOF test with over 50% of the samples of user traffic extracted from the SNRC trace and it remains a good fit to the observed data at the one minute and one second time scale. These results suggest:

1. That, due to the memoryless property of the geometric distribution, in a given hour the likelihood of another HTTP request being generated by a user is independent of how many have been generated so far.

2. That on a user level HTTP request generation may perhaps be approximated by a sequence of independent Bernoulli trials (as the geometric distribution is used to measure the number of trials before an event occurs). In this case the

stopping event may be either that the user decides to discontinue Web browsing or the end of the sample period.

3. That distributions of sums of geometrically distributed random variables may be appropriate for modelling the approximate aggregate number of HTTP requests for a population of users in the same time interval.

This last suggestion, using the geometric distribution result as a component in a model of the marginal distribution of HTTP request rate, is explored in Chapter 5.

5. Marginal Distribution of HTTP Request Rate

This chapter establishes a new model describing the marginal distribution of HTTP request arrivals per second in aggregate streams of Web traffic generated by a population of users on an access network. The model is based on the Pólya-Aeppli probability distribution [Johnson 92 pp. 378-382]. Formulas and other details concerning the Pólya-Aeppli distribution itself are given in Appendix F.4.

The choice of the Pólya-Aeppli distribution is based on the results from previous chapters. Chapter 3 shows that the number of unique Web users issuing one or more HTTP requests in a second has a Poisson distribution. Chapter 4 shows that the geometric distribution is an approximate model for the number of HTTP requests generated by an individual user in a second where they make at least one request. The Pólya-Aeppli distribution is a Poisson stopped sum of the geometric distribution [Johnson 92 pp. 378-382] and is essentially a combination of the two results.

In this chapter the Pólya-Aeppli distribution is compared to the observed marginal distribution of HTTP request rate in the sample hours listed in Table 2.2. Plots of the PMF of the Pólya-Aeppli distribution compared to histograms of observed request rate show a good match. The quality of the match is confirmed using PP plots and QQ plots. The body of the distribution is an excellent match to the observed request rate data, however there is some divergence between the model and the UNSW2 trace at the tail of the distribution (at quantiles of 0.95 and above).

The Pólya-Aeppli distribution is significant due to a good match with a diverse set of the traffic samples. These samples have hourly mean HTTP request rates ranging from 5.5 to 139 requests per second. They are from a range of time periods extending from 1996 to 2002, from both corporate and university sources and from US and non-US sources. The Pólya-Aeppli distribution is also less complex than other previously published characterisations (detailed in Section 2.3).

This chapter also compares the Pólya-Aeppli model to the Poisson and normal distribution alternatives for the marginal distribution of HTTP request rate. A Poisson model is considered as it is a default arrival process used in telecommunications

traffic modelling. The normal distribution is considered as it is a common distribution used in the absence of a known model. The Poisson distribution is a poor fit to observed traffic. The normal distribution is found to be a better match to aggregate HTTP request traffic than the Poisson distribution but not as good as the Pólya-Aeppli distribution.

The MOM parameter estimation procedure is used for fitting the Pólya-Aeppli distribution to observed per second HTTP request rate and requires the sample mean and standard deviation to be known. The standard deviation of request rate is not commonly reported as most request rates are usually reported in terms of the mean only. A fitting procedure based solely on the mean would therefore be useful. Analysis of the four traces of aggregate Web traffic shows an approximate linear relationship between the mean and standard deviation. Hence an estimate of the standard deviation from the mean can be derived.

A potential application of the Pólya-Aeppli model is to estimate peak per second HTTP request rate (Section 6.1). However the graphs in this chapter indicate that the Pólya-Aeppli distribution and the normal distribution may be converging at higher mean request rates. If that were so then the normal distribution would be more useful (and well known) for estimation of peak HTTP request rate in the region where the two distributions converged. Comparison of the two distributions over a wide range of hypothetical mean HTTP request rates shows that they do not converge. An asymptotic limit is reached at approximately a mean request rate of 1000 HTTP requests per second and higher. The normal distribution is not an adequate alternative.

The first section of this chapter compares the marginal distribution of HTTP request rate from the sample hours listed in Table 2.2 with the Poisson, normal and Pólya-Aeppli probability distributions. The fit of the Pólya-Aeppli distribution is examined using PP and QQ plots.

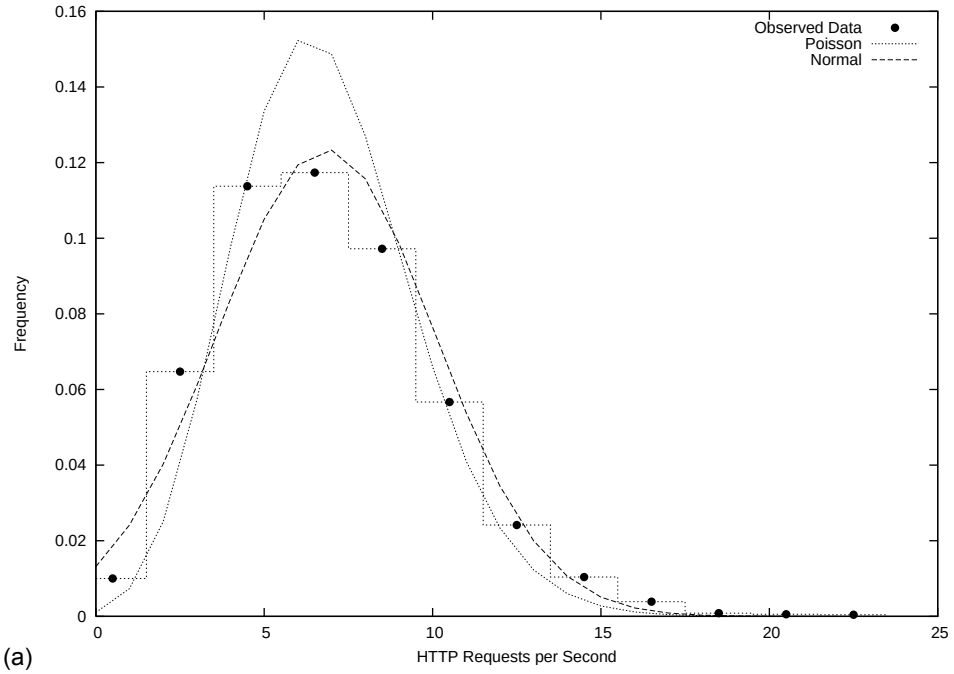
The second section looks at an alternative estimation technique for the parameters of the Pólya-Aeppli distribution using just the mean request rate. The third section looks at the rate of convergence of the marginal distribution of HTTP request rate to the normal.

5.1 Models for the Marginal Distribution of HTTP Request Rate

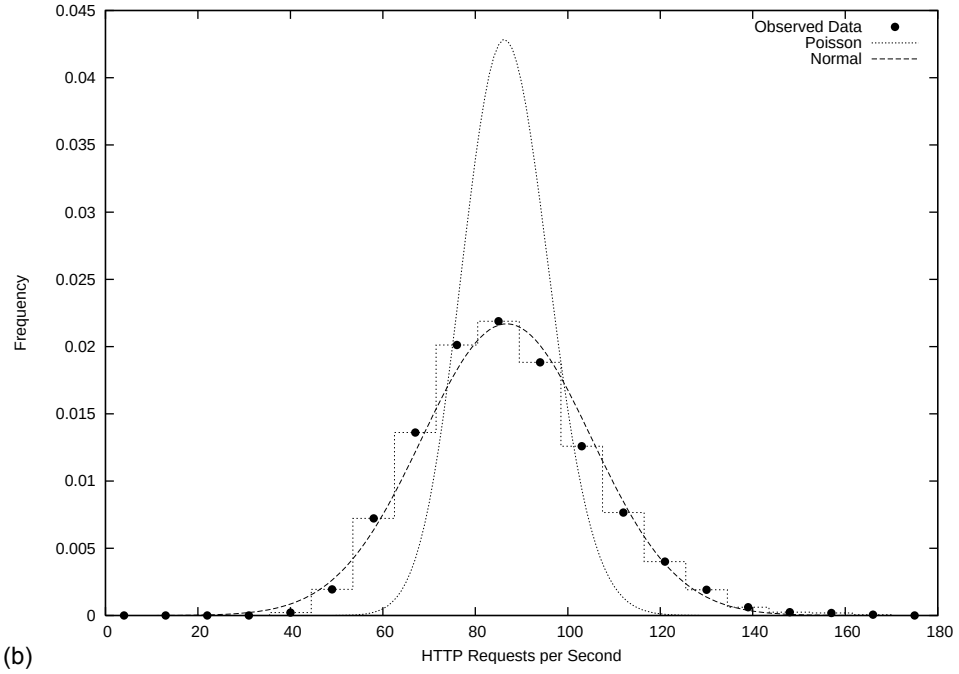
Section 3.1 has already shown that the arrival process of HTTP requests in an aggregate Web traffic stream is not Poisson. The variance of the request rate is too high in relation to the mean. Figure 5.1 shows the observed marginal distribution of HTTP request rate for the sample hours *B6* and *U6* against the Poisson and normal distributions. Similar graphs were obtained for the other hours listed in Table 2.2 and they are shown in Appendix J.1. Neither the Poisson or normal distributions provide a good fit to the marginal distribution of per second HTTP request rate, although the marginal distribution appears to converge towards the normal distribution at higher mean request rates. For example in Figure 5.1, the normal distribution is closer to the marginal distribution of request rate in hour *U6* which has a higher mean HTTP request arrival rate than *B6*. In general, the mass of the normal distribution is slightly to the right of the observed distribution and the right hand tail is lighter than the observed distribution.

In Chapter 4, analysis of the SNRC trace showed that the number of HTTP requests generated per hour by an individual user has a geometric distribution. Chapter 4 also showed that the geometric distribution approximated the number of requests made by users at shorter time scales. In Chapter 3 it was shown that the number of active Web clients per second has a Poisson marginal distribution. The Pólya-Aeppli distribution [Appendix F.4] can be viewed as a combination model of the geometrically distributed HTTP request rate and the Poisson distributed Web clients. The Pólya-Aeppli distribution is a “Poisson stopped sum” of the geometric distribution. The distribution is commonly used for counts of items that occur in clusters. If the number of items per cluster has a geometric distribution and the number of clusters has a Poisson distribution then the total count of items has a Pólya-Aeppli distribution.

The Pólya-Aeppli distribution was compared to the marginal distribution of HTTP request rate for the sample hours listed in Table 2.2. Figure 5.2 shows the comparison for the sample hours *B6* and *U6* along with the Poisson and normal distributions. Each probability distribution was matched to the data using the MOM (estimator equations for the Pólya-Aeppli distribution are detailed in Appendix F.4).

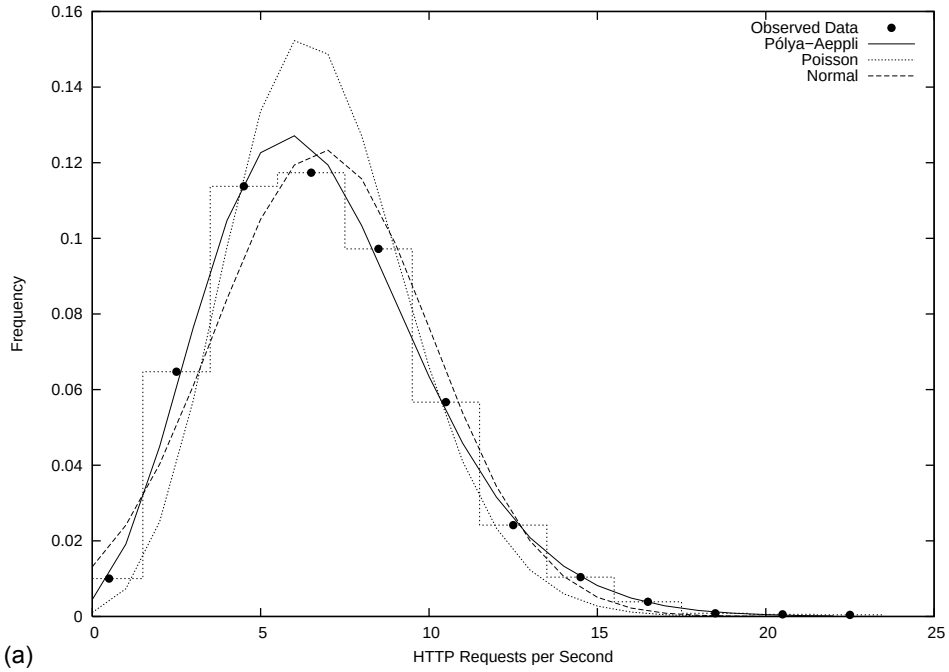


(a)

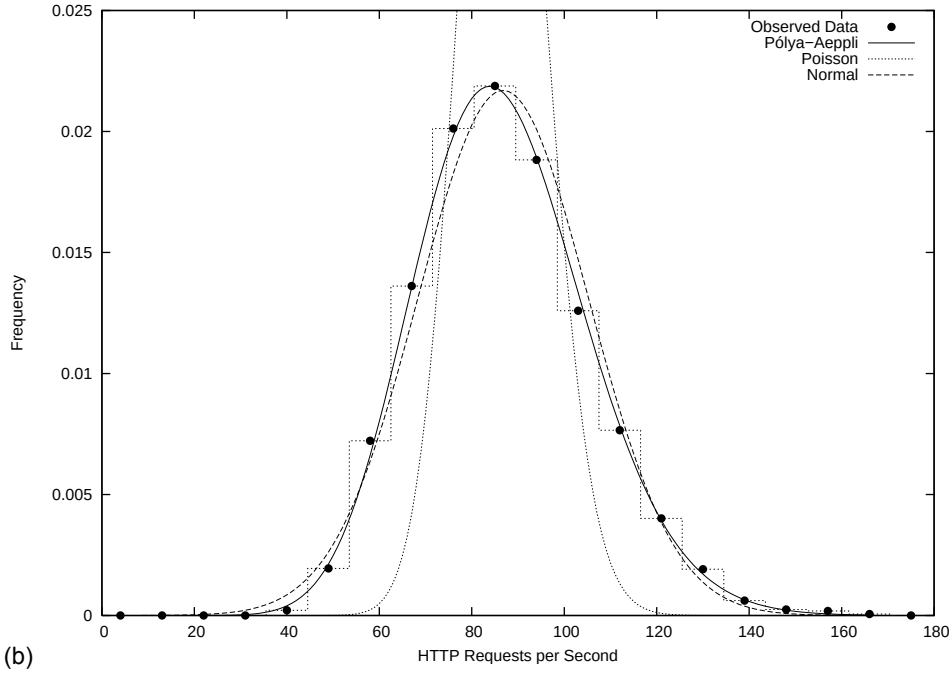


(b)

Figure 5.1 Marginal Distribution of HTTP Request Rate Compared to the Poisson and Normal Distributions for the Sample Hours;
(a) *B6* (5:00pm 11 November 1996 in Berkeley Trace)
(b) *U6* (3:00pm 29 June 1999 in UNSW1 Trace)



(a)



(b)

Figure 5.2 Marginal Distribution of HTTP Request Rate Compared to the Poisson, Normal and Pólya-Aeppli Distributions for the Sample Hours;
(a) *B6* (5:00pm 11 November 1996 in Berkeley Trace)
(b) *U6* (3:00pm 29 June 1999 in UNSW1 Trace)

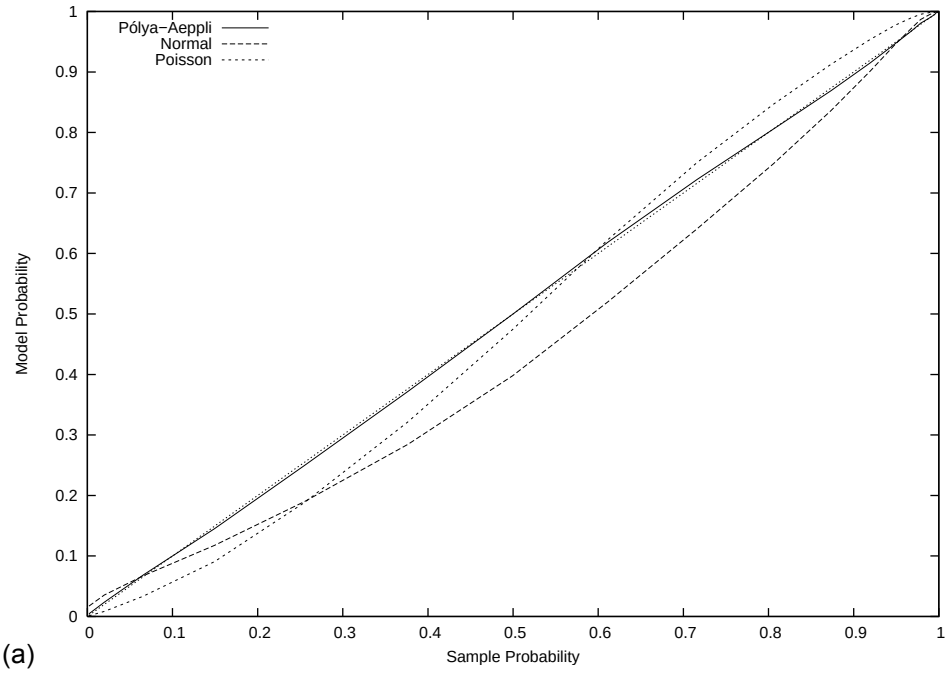
Figure 5.2 shows a good match between the observed marginal distribution of HTTP requests per second and the Pólya-Aeppli distribution. Similarly all the sample hours showed equally good fits and plots are shown in Appendix J.1. Parameter estimation using the MOM produced the best fit. Other parameter estimation methods [Johnson 92 p. 382] which are functions of observed counts of zero and one HTTP request per second produced poor fits.

The goodness of the fit of a proposed probability distribution to data is difficult to determine simply by viewing a PMF plot. Better graphical methods are the PP plot and the QQ plot [Law 91 pp. 372-380]. The PP plot compares the observed (empirical) distribution function with the proposed distribution function over probabilities between zero and one. The plot highlights differences between the proposed distribution and observed data for the middle of the distribution. A good fit is indicated on the plot by a straight line with a slope of 1.

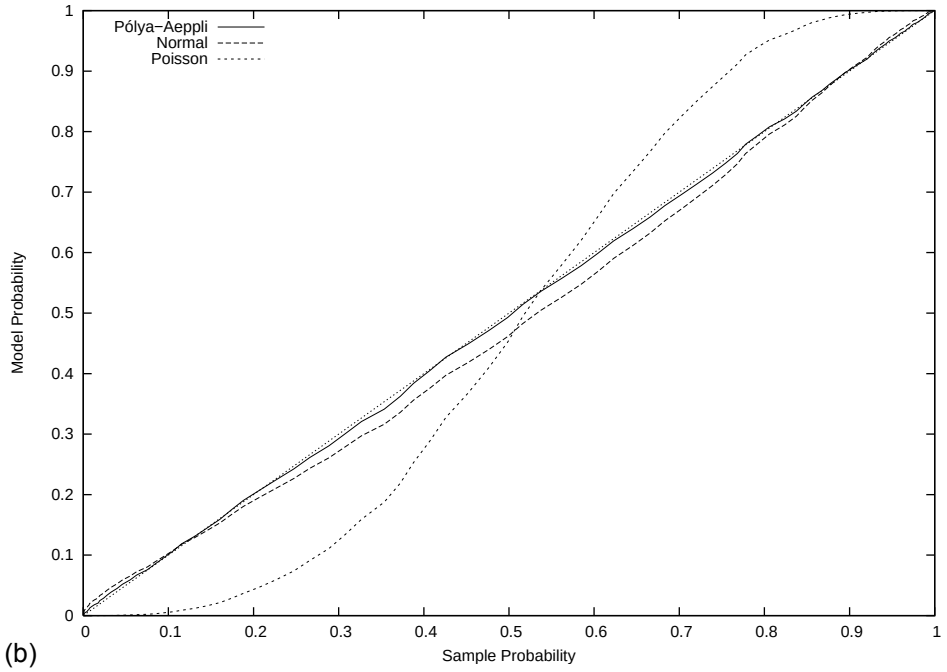
Figure 5.3 shows the PP plot for the sample hours *B6* and *U6* with the Pólya-Aeppli, Poisson and normal distributions, the plots for all the other sample hours are shown in Appendix J.2. The plot includes a dotted sloping line to show where a perfect fit would be plotted. The PP plots for all the sample hours show a good match to the Pólya-Aeppli distribution. The PP plots also show that the normal and Poisson distributions are not a good fit.

The QQ plot highlights differences in the tail [Law 91 pp. 372-380]. The QQ plots for the sample hours *B6* and *U6* with the Pólya-Aeppli, normal and Poisson distributions are shown in Figure 5.4. Again a good fit is indicated by a straight line (shown) on the plot. The plot also includes horizontal lines indicating the 0.01, 0.05, 0.95 and 0.99 quantiles. The QQ plots for all the sample hours are shown in Appendix J.3. On some of the plots one or both of the lower two quantiles are at zero and the lines are not shown. The QQ plots show discrepancy between the Pólya-Aeppli distribution and the observed request rate at higher quantiles for some of the hours in the UNSW2 trace but overall still indicate a good fit.

Good fit between the marginal distribution of HTTP request rate and the Pólya-Aeppli probability distribution is significant as it applies over a wide range of traffic samples. The distribution closely matches traffic samples collected between 1996 and 2002 with a wide range of mean HTTP request rates from 5.5 requests per sec-

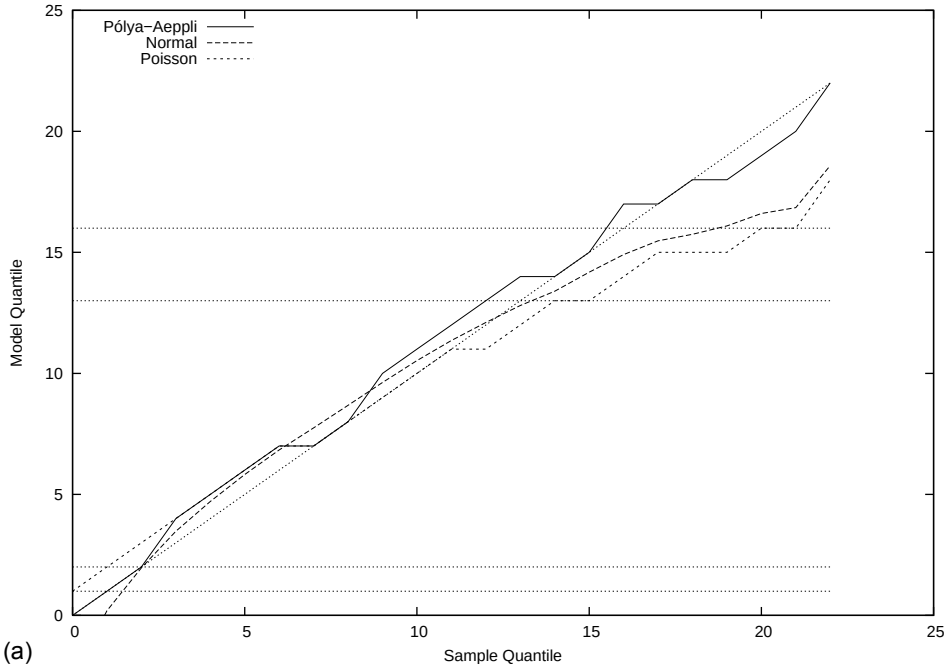


(a)

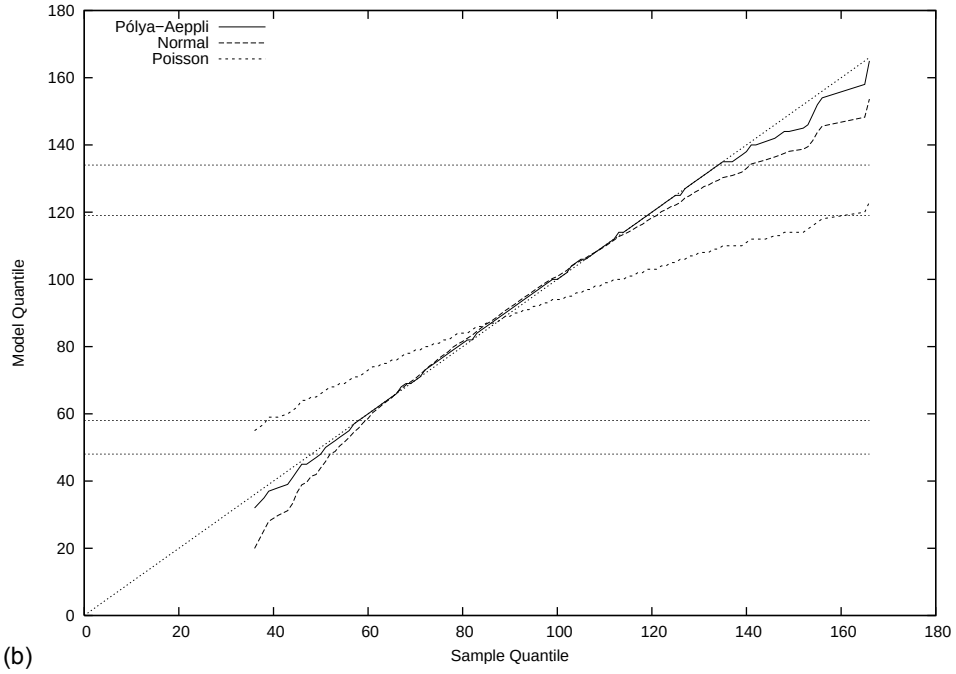


(b)

Figure 5.3 PP plots Showing the Fit of the Pólya-Aeppli, Normal and Poisson Distributions to HTTP Request Rate from the Sample Hours:
(a) *B6* (5:00pm 11 November 1996 in Berkeley Trace)
(b) *U6* (3:00pm 29 June 1999 in UNSW1 Trace)



(a)



(b)

Figure 5.4 QQ plots Showing the Fit of the Pólya-Aeppli, Normal and Poisson Distributions to HTTP Request Rate from the Sample Hours:
(a) *B6* (5:00pm 11 November 1996 in Berkeley Trace)
(b) *U6* (3:00pm 29 June 1999 in UNSW1 Trace)

ond (hour *B5*) to 139 requests per second (hour *N5*) however the fit diverges for some sample hours at higher quantiles. The poorest fit occurred with the UNSW2 trace at higher quantiles. Examples of the poorest fit are the hours *N2*, *N4* and *N6* on which the QQ plot divergence between observed request rate and the model upwards from the 0.95 quantile (these are all plotted in Appendix J.3).

5.2 Parameter Estimation from the Mean Request Rate

In previous sections the parameters of the Pólya-Aeppli distribution have been estimated using the MOM in which parameter estimates are calculated as functions of the sample mean and variance (equations in Appendix F.4). Unfortunately HTTP request rate variance is not usually reported. Since only the mean HTTP request rate is usually reported a method to estimate parameters of the Pólya-Aeppli distribution as a function of the sample mean only is required. Such a method would allow for estimation of peak per second HTTP request rates from a known mean rate (in Section 6.1), comparison with other Web traffic models (in Section 6.3) and assist in analysis of the convergence between the normal and Pólya-Aeppli distributions at higher request rates (later in this chapter).

Fortunately, for traffic with a mean HTTP request rate of around 10 HTTP requests per second and higher, an approximate linear relationship between the mean and standard deviation of HTTP request rate exists. Figure 5.5 shows scattergrams for the mean versus the standard deviation of observed per second HTTP request rate for each hour in each of the four traces of aggregate Web traffic. For the Digital, UNSW1 and UNSW2 traces a line has been fitted above a mean request rate of 10 requests per second. The fitted lines have slopes and intercept values listed in Table 5.1. The mean value of each is also shown in the table and using these an approximate relationship between the sample mean $\bar{x}$ and the estimate of the sample deviation $\tilde{s}$ is given by Equation 5.1.

$$\tilde{s} = 0.186\bar{x} + 4.26 \qquad \textbf{(Eqn 5.1)}$$

Comparison of the estimate of standard deviation using Equation 5.1 to the actual sample standard deviation of HTTP request rate was made for the 14 sample hours listed in Table 2.2 with a mean request rate of over 10 HTTP requests per second. At

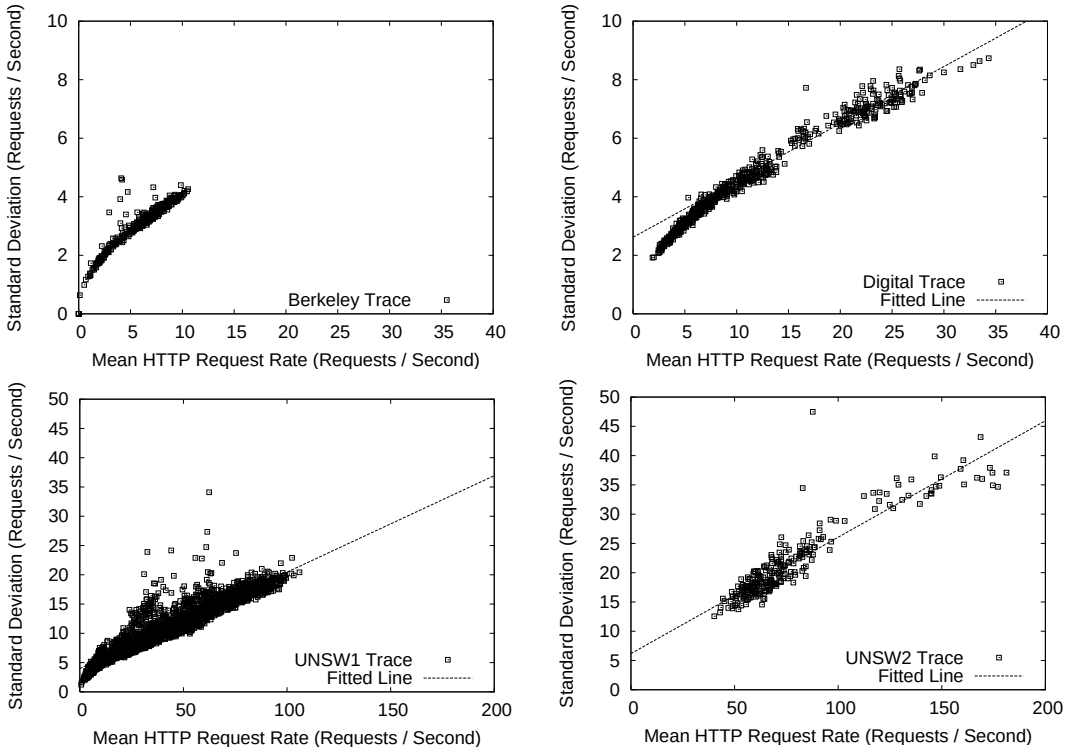


Figure 5.5 Scattergrams of Mean versus Standard Deviation of HTTP Request Rate for Each Hour in the Four Aggregate Traffic Traces

Table 5.1 Slope and Intercept of Fitted Lines in Figure 5.5

Trace	Slope	Y Axis Intercept
Digital	0.194	2.62
UNSW1	0.165	3.97
UNSW2	0.199	6.18
Mean	0.186	4.26

most the estimate of the standard deviation was 27% higher than that observed (hour *D5)*, or 21% lower than that observed (hour *N4)*. On average the estimate of standard deviation was only 3.3% lower than that observed.

5.3 Convergence to the Normal Distribution

The plots shown in Section 5.1 suggest that the normal distribution may converge to with the fitted Pólya-Aeppli model with increasing request rate. The possible convergence is indicated as an increasingly closer fit between the normal distribution and the Pólya-Aeppli distribution with increasing mean request rate on the PP and

QQ plots shown in Section J.2 and J.3 respectively. This section examines the convergence over a larger range of mean request rates. The issue of convergence is important because if the two distributions can be shown to converge then the normal distribution and associated statistical methods could be utilised in the estimation of various parameters of interest such as peak HTTP request rate.

In order to extrapolate the marginal distribution of per second HTTP request rate at higher mean HTTP request rates than observed in the four aggregate Web traces the relationship between sample mean and standard deviation given in Equation 5.1 is utilised. The range of HTTP request rates examined covers the range of Web cache performance figures published in the most recent Web proxy benchmark effort at the Measurement Factory in December 2001 [Rousskov 01].

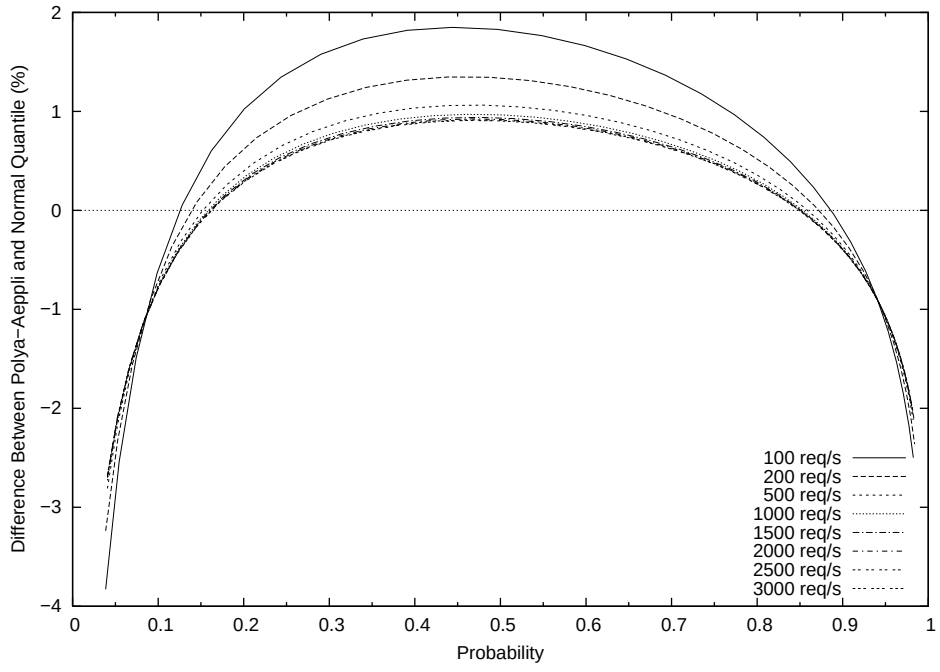


Figure 5.6 Percentage Difference in Quantiles of the Pólya-Aeppli and Normal Distributions for a Range of Mean HTTP Request Rates

Figure 5.6 shows the percentage difference between quantiles of the normal distribution and the Pólya-Aeppli distribution at probabilities between 0.05 and 0.95 for a range of HTTP request rates. For example, at a mean request rate of 100 HTTP requests per second the 0.7 quantile of the normal distribution differs from the Pólya-Aeppli distribution by approximately 1.4%.

Figure 5.6 shows the difference between the normal and Pólya-Aeppli distributions decreasing with increasing mean HTTP request rate with this difference most pronounced at the tails. However, the figure also shows that this difference appears to be approaching an asymptotic limit as the lines representing higher request rates on the plot overlap.

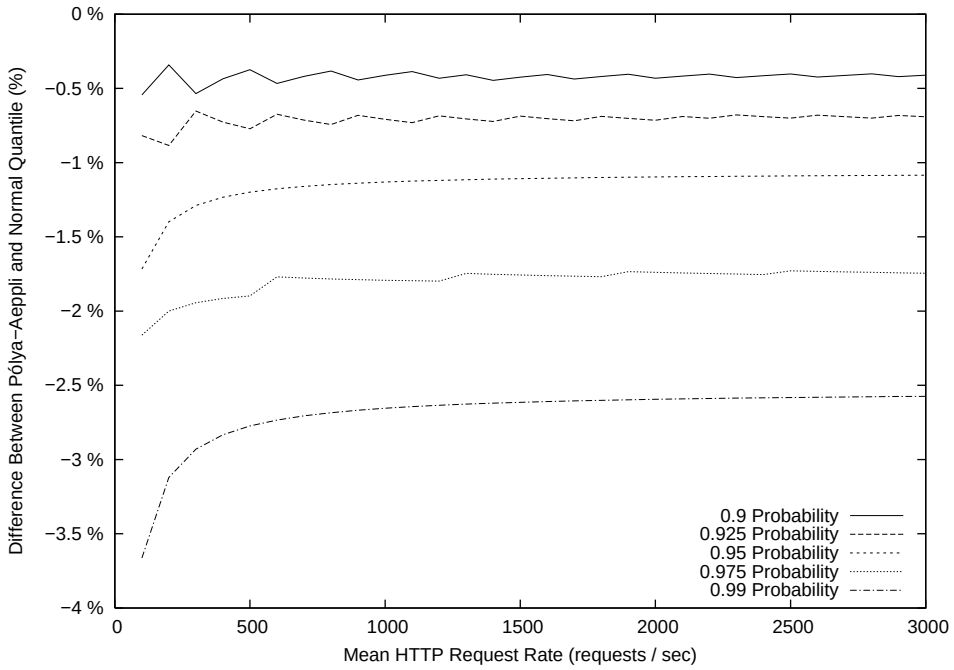


Figure 5.7 The Amount by Which the Normal Distribution Under Estimates the Pólya-Aeppli distribution with Increasing Mean Request Rate

Figure 5.7 shows the percentage difference between quantiles of the normal and Pólya-Aeppli distributions at mean HTTP request rates between 100 and 3000 requests per second for a range of upper probabilities. The figure shows an asymptotic limit at around 1000 HTTP requests per second above which the difference between the normal and Pólya-Aeppli distribution is approximately constant. The jagged lines on the graph are due to the discrete nature of the Pólya-Aeppli distribution. The result shows that the normal distribution under estimates the upper quantiles of the Pólya-Aeppli distribution and does not converge. For quantiles between 0.9 and 0.975 and the normal distribution under estimates the Pólya-Aeppli by up to 2%.

5.4 Conclusion

It is concluded that the Pólya-Aeppli distribution is a good model for the marginal distribution of aggregate HTTP request rate measured in counts of request arrivals per second. The result holds for a diverse set of samples of Web traffic extracted from four independent traces of Web traffic collected between 1996 and 2002. The Pólya-Aeppli distribution result diverges slightly above the 0.95 quantile for some samples of traffic in the UNSW2 trace.

The finding of a Pólya-Aeppli marginal distribution suggests per user request rates may be geometrically distributed. Although the result in Chapter 4 relied on analysis of a trace from a small group of postgraduate research students in a single university laboratory, a good match with the Pólya-Aeppli distribution suggests that the geometric distribution result may hold for other users.

Analysis of the four traces of aggregate traffic shows an approximate linear relationship between the mean and standard deviation of HTTP request rate during hours with a mean request rate greater than 10 HTTP requests per second. In the absence of known sample standard deviation the formula shown in Equation 5.1 is suggested as a method to estimate sample standard deviation from the sample mean.

The Pólya-Aeppli model for the marginal distribution of HTTP request rate was compared to the Poisson and normal distribution models. The Poisson model was found to fit poorly with observed traffic and the normal distribution has a poor fit for sample hours with low HTTP request rate but improves for hours with higher request rates. The convergence between the Pólya-Aeppli distribution and the normal distribution was examined over a range of per second HTTP request rates using the linear relationship between mean and standard deviation of HTTP request rate found in the traces of Web traffic. It was found that for hours with mean request rates over 1000 HTTP request per second the difference between the two distributions approaches an asymptotic limit. The normal distribution cannot be used as a substitute model for the marginal distribution of HTTP request rate at high mean HTTP request rates. At mean rates of 1000 HTTP requests per second and higher the normal distribution under estimates upper quantiles between 0.9 and 0.975 by up to 2%.

6. Application of the Pólya-Aeppli Model

This chapter discusses three possible applications of the Pólya-Aeppli model of HTTP request rate. The first application is estimating peak HTTP request rates, the second is the formulation of two rules of thumb concerning HTTP request rate, and the third is sanity checking models of HTTP request arrival such as artificial Web proxy workloads.

In Chapter 5 the Pólya-Aeppli distribution was shown to be a good model for the marginal distribution of per second HTTP request rate. In this chapter the Pólya-Aeppli result is applied to the estimation of peak HTTP request rate. It is shown to accurately estimate peak HTTP request rate for a large number of samples of busy hour traffic from each of the four traces of aggregate Web traffic. These traffic samples are independent from those used to develop the model. The most accurate estimates are obtained when the mean and standard deviation of HTTP request rate are already known. Exploiting the linear relationship between request rate mean and standard deviation expressed in Equation 5.1 allows estimation of peak HTTP request rate from the mean request rate alone. In this case the estimates of peak request rate are still quite good, for example an average ±8% of HTTP requests per second at the peak 95% quantile. A more accurate estimate of an average HTTP request rate (within 2%) is obtained if Equation 5.1 is tailored to the particular traffic trace in question. If the standard deviation is known then the estimate is even better at an average of ±1%.

As discussed in Section 2.3.3 there are few published results concerning peak HTTP request rates. The ability of the Pólya-Aeppli distribution to provide good estimates of peak request rate in combination with the linear relationship between mean and standard deviation of request rate suggest two new rules of thumb. First is the relationship expressed in Equation 5.1 applied as a method for the estimation of expected standard deviation (and hence variance) of per second HTTP request rate from a given mean rate. Second is a similar equation for the estimation of peak per second HTTP request rate from a given mean. The peak chosen is the 95% quantile, that is, the expected per second HTTP request rate that should be equalled or

exceeded on average once every 20 seconds. The two proposed rules of thumb are compared to the samples of busy hour traffic from the aggregate HTTP traffic traces. It is found that both rules provide approximate estimates (as is expected from a rule of thumb) and results are provided for how much each estimate deviates from the observed traffic.

A further application of the Pólya-Aeppli model is in the sanity checking of models of HTTP request arrival. The idea is to examine whether or not the per second HTTP request rate suggested by the model appears to be representative of actual Web traffic. In conjunction with the two proposed rules of thumb three aspects of HTTP request arrival can be examined:

1. The marginal distribution of per second HTTP request rate - does it have the expected Pólya-Aeppli shape?

2. The standard deviation of per second HTTP request rate - is it similar to the expected value?

3. The peak per second HTTP request rate - is the 95% quantile of request rate similar to the expected value?

Four models of HTTP request arrival were previously identified in Chapter 2 for comparison. Only one model [Cao 01] is found to generate some hours of Web traffic that appear to be realistic according to these three criteria. The other models [Deng 96, Mah 97, Rousskov 01] produce streams of aggregate HTTP request rate that are either not variant enough [Deng 96, Rousskov 01] or too variant [Mah 97]. It is found that the test for a Pólya-Aeppli shaped marginal distribution of per second HTTP request rate does not discriminate between different HTTP request arrival models. All four of the models produce an aggregate per second HTTP request rate stream with a marginal distribution that is a good match to the Pólya-Aeppli distribution. Testing for expected standard deviation and peak request rate using the proposed rules of thumb is more discriminatory with only one model [Cao 01] partially meeting these two other criteria.

6.1 Estimation of Peak HTTP Request Rate

The previous chapter showed that the Pólya-Aeppli distribution is a good model for the marginal distribution of per second HTTP request. This section applies the model to estimation of peak request rate. The statistic of interest is the likelihood and magnitude of excursions of per second request rate from the mean during periods of time where the underlying mean is otherwise constant. The peak rate is estimated from the Pólya-Aeppli distribution by inversion at the probability of interest.

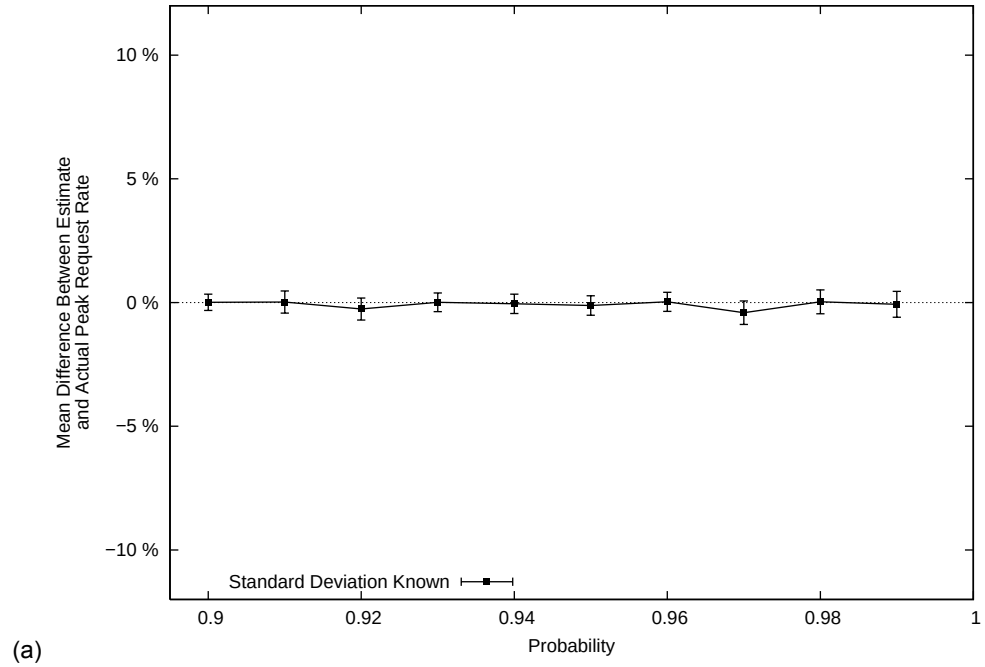
The 100 hours with the highest mean request rate (the busy hours) were extracted from each of the four traces of aggregate Web traffic. This process was independent of the original sampling process that was used in Chapter 2 to select samples of Web traffic to develop the Pólya-Aeppli model. In each of the sampled busy hours, peak HTTP request rates were compared to estimates obtained using the Pólya-Aeppli distribution. Parameters of the Pólya-Aeppli distribution were calculated using the MOM from the mean and standard deviation of per second HTTP request rate of the sample in question. Three different estimates were calculated using three different values for the sample standard deviation:

1. The measured standard deviation of the HTTP request rate of the traffic sample

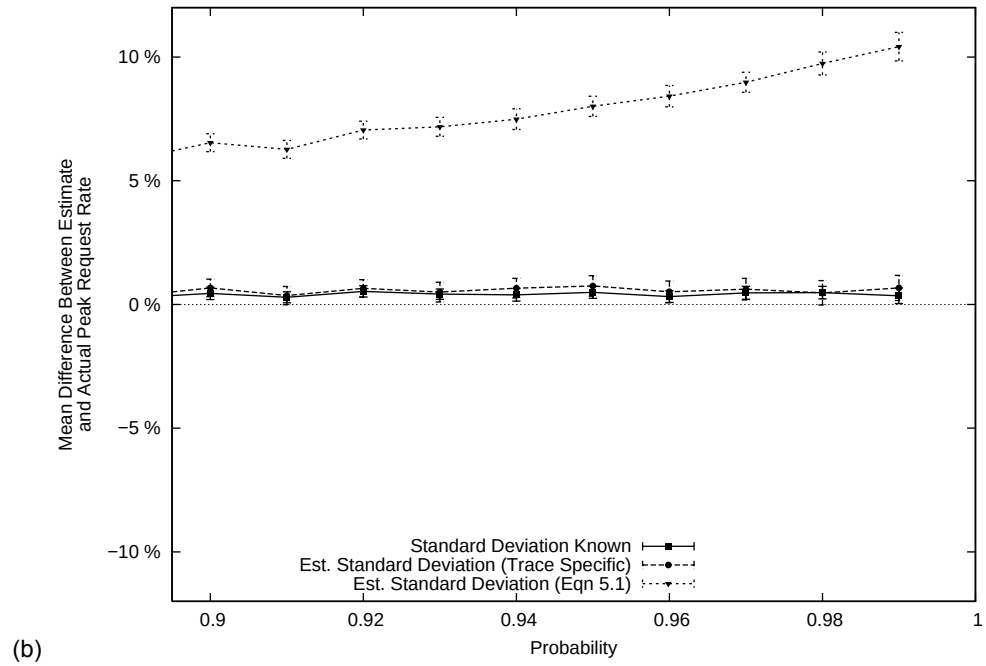
2. Standard deviation estimated from the mean HTTP request rate of the traffic sample using Equation 5.1

3. Standard deviation estimated from the mean HTTP request rate of the traffic sample using Equation 5.1 but substituting values listed in Table 5.1 specific to that traffic trace

Peak HTTP request rate estimates for hours from the Berkeley trace were obtained using the first method only as the mean request rate of the sample hours is too low for Equation 5.1 to apply. The difference between the estimate and observed peak HTTP request rate were expressed as a percentage. The mean percentage difference between the estimate and the observed peak request rate over the 100 hundred hours in each trace was calculated. Figures 6.1 and 6.2 show the 95% confidence intervals for this mean percentage difference at a number of probabilities for each of the four traces of aggregate Web traffic.

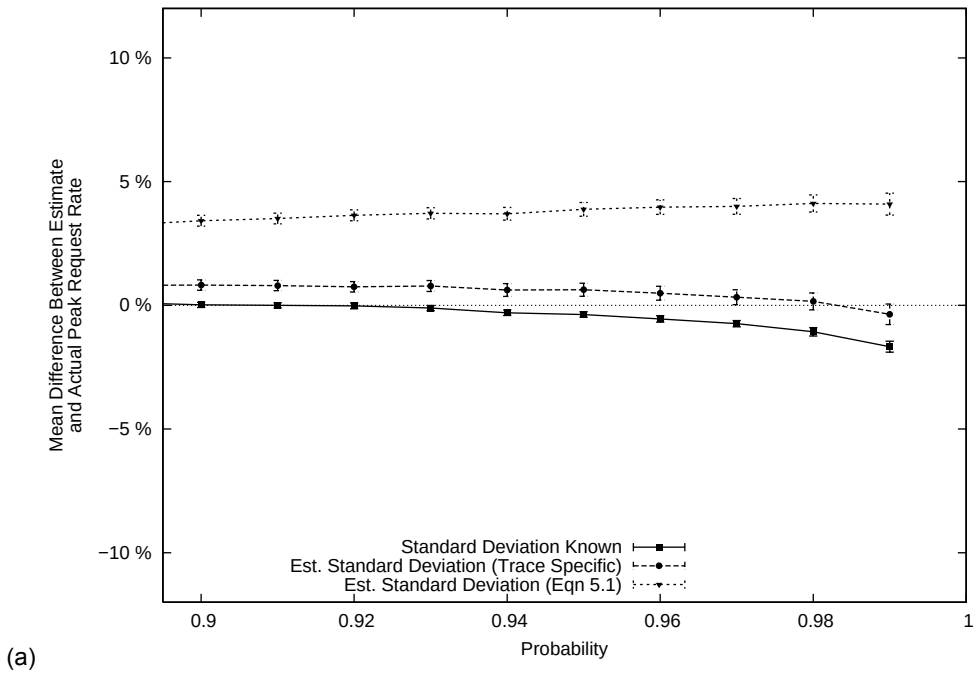


(a)

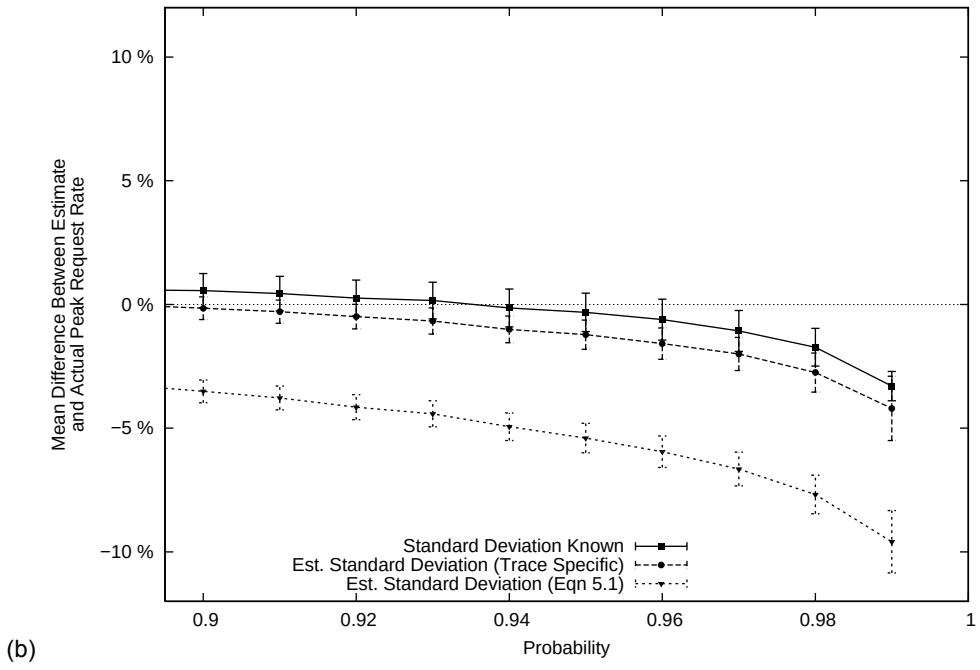


(b)

Figure 6.1 Mean Difference Between Estimated and Actual Peak HTTP Request Rate for 100 Busy Hours Samples from:
(a) Berkeley Trace
(b) Digital Trace

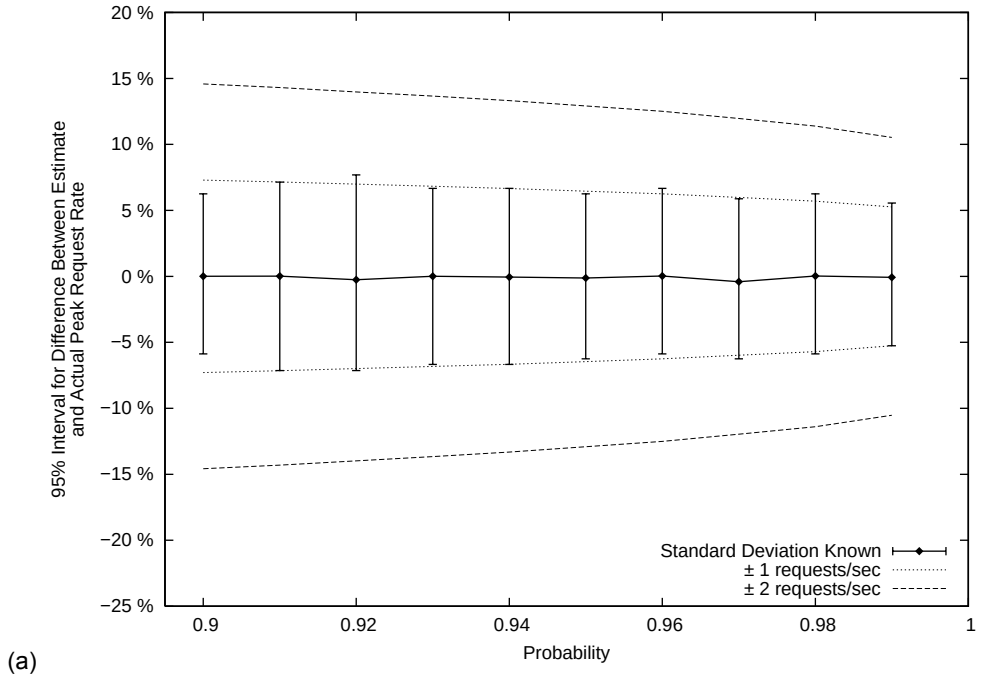


(a)

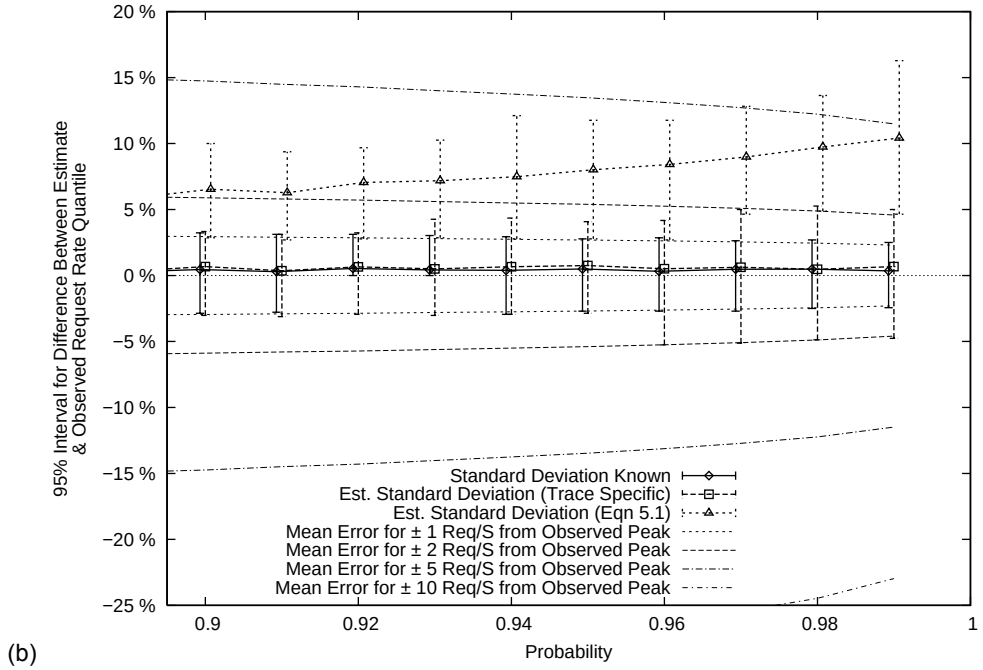


(b)

Figure 6.2 Mean Difference Between Estimated and Actual Peak HTTP Request Rate for 100 Busy Hours Samples from:
(a) UNSW1 Trace
(b) UNSW2 Trace

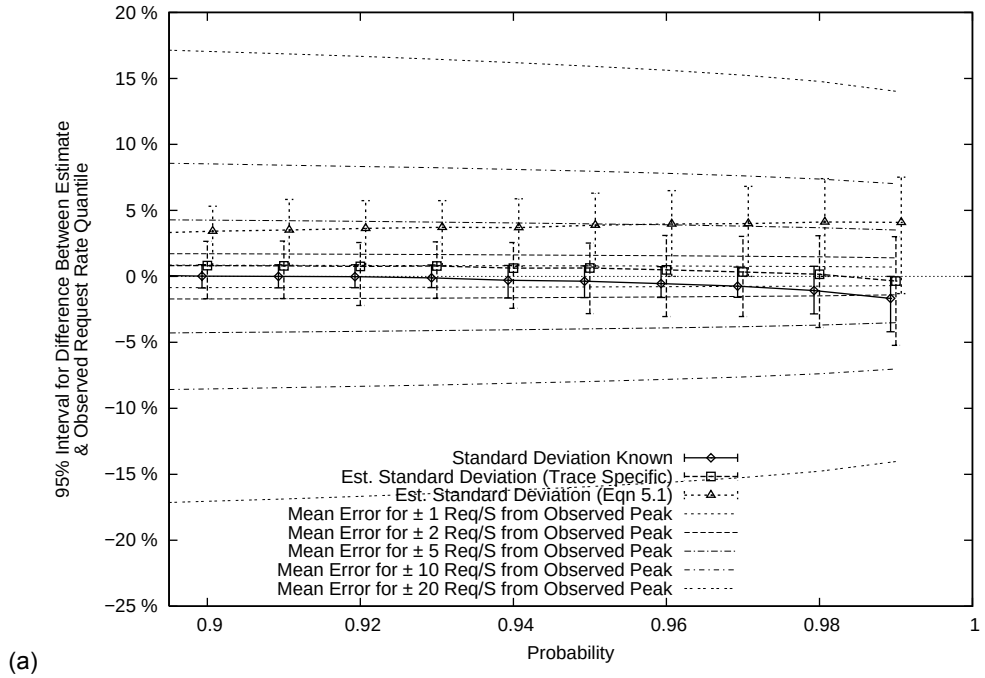


(a)

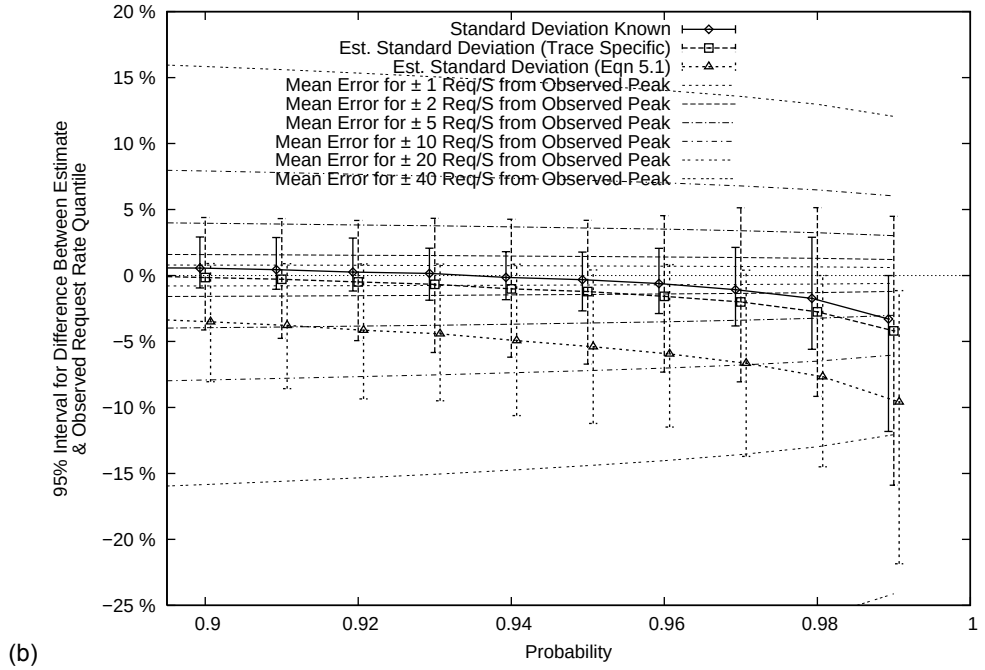


(b)

Figure 6.3 95% Interval for the Difference Between Estimated and Actual Peak HTTP Request Rate for 100 Busy Hours Samples from:
(a) Berkeley Trace
(b) Digital Trace



(a)



(b)

Figure 6.4 95% Interval for the Difference Between Estimated and Actual Peak HTTP Request Rate for 100 Busy Hours Samples from:
(a) UNSW1 Trace
(b) UNSW2 Trace

Figures 6.1 and 6.2 show that the average estimate provided by the Pólya-Aeppli distribution is close to the observed peak per second HTTP request rate. The best estimate of peak request rate is obtained when the Pólya-Aeppli distribution is matched directly to the actual mean and standard deviation of the data sample. For three of the traces the estimate of peak request rate obtained this way is very close to the actual value. There is some divergence for the UNSW2 trace at higher quantiles. In the UNSW2 trace the peak HTTP request rate at the 0.99 probability is under estimated by approximately 3%.

The least accurate estimation method is where the parameters of the Pólya Aeppli distribution are obtained using just the mean of the data sample and applying Equation 5.1 for an estimate of the standard deviation. With this method the average error at the 95% peak HTTP request rate is ±8% rising to approximately ±10% at the 99% peak rate.

Tailoring Equation 5.1 to be specific to the trace in question using the values provided in Table 5.1 results in more accurate average estimates. With this method the peak HTTP request rate estimate was within ±2% for all the traces with the exception of the higher quantiles of the UNSW2 trace which were under estimated by up to 5%.

Figures 6.1 and 6.2 show that the average estimate of peak HTTP request rate provided by the Pólya-Aeppli distribution is good. However they do not provide information on the spread, or coverage, of the estimates. Figures 6.3 and 6.4 show confidence intervals that indicate the range of 95% of the estimates of peak HTTP request rate at each probability of interest. These probabilities are 0.9 through to 0.99 in 0.01 increments. On the plots with multiple confidence interval estimates at each probability the confidence intervals are plotted slightly to the left and right to improve readability of the graph. Figures 6.3 and 6.4 also include dashed lines showing the integral number of HTTP requests per second by which the peak HTTP request rate is over, or under, estimated by.

Figures 6.3 and 6.4 show that when mean and standard deviation are known that 95% of the estimates of peak request rate are approximately within ±1 HTTP request per second from the actual value for the Berkeley and Digital traces. For the UNSW1 and UNSW2 traces with higher mean request rates the estimate of peak

request rate is within ±5 HTTP requests per second except at very high quantiles. The other two estimation methods show a wider range of peak HTTP request rate estimates at each probability of interest. The result is still good. For example based on just knowledge of the mean HTTP request rate estimates of the peak rate at the 0.95 probability mostly fall within ±12% of the actual value.

The use of the Pólya-Aeppli model to estimate peak HTTP request rate provides good results. The most accurate result is obtained when knowledge of both the mean and standard deviation of the HTTP request rate are known. An approximate peak request rate estimate can be obtained with knowledge of just the mean request rate using Equation 5.1 to estimate the sample standard deviation. This later approximation can be "tuned" if there is knowledge of the long term relationship between mean and standard deviation for a particular source of Web traffic.

6.2 Two New Rules of Thumb

The results from Section 6.1 and Equation 5.1 can be utilised to propose two new rules of thumb for HTTP request rate. The first is for the estimation of the standard deviation (and hence variance) of per second HTTP request rate, the second is for the estimation of the 1 in 20 peak per second HTTP request rate.

In Chapter 5 a linear relationship was observed between the mean and standard deviation of HTTP request rate for mean request rates above approximately 10 requests per second. This relationship was expressed in Equation 5.1 and is proposed here as being a new rule of thumb. The rule provides an estimate of the expected standard deviation of per second request rate given the mean request rate. The estimate was compared against the standard deviation measured for the same 400 busy hours from the aggregate traffic traces used in the previous section. The linear relationship is known to not hold for the hours from the Berkeley trace because the mean HTTP request rates was too low. For the other three traces 95% of the hours had a standard deviation of per second HTTP request rate between -21% and 32% of the estimate. Figure 6.5 shows the comparison between the observed and estimate of standard deviation.

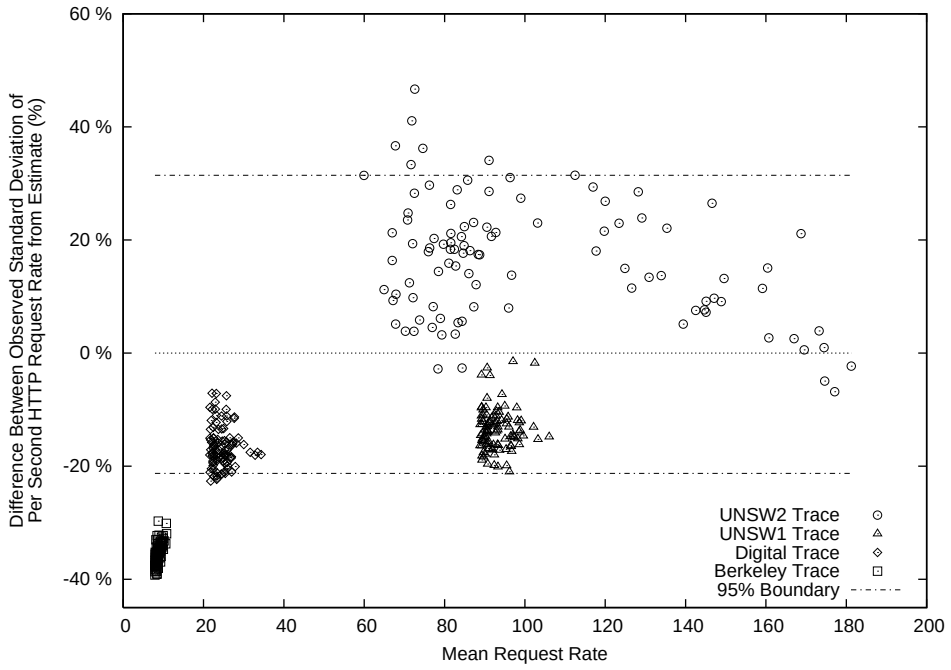


Figure 6.5 Rule of Thumb for the Estimation of the Expected Standard Deviation of Per-Second Request Rate Compared to the Busy Hours

An example of the use of this first rule of thumb is; if we had an hour of Web traffic in which the only information known is that the mean per second HTTP request rate is 100 HTTP requests per second it would be expected that the standard deviation of request rate would be approximately 23 HTTP requests per second. An alternative estimate is that the standard deviation would be expected to fall between approximately 18 and 30 HTTP requests per second.

A second rule of thumb is now suggested for estimating peak HTTP request rate. The 95% quantile of the Pólya-Aeppli distribution provides a 1 in 20 peak per second HTTP request rate estimate. In the plots shown in Figures 6.1 and 6.2 this estimate is shown to be accurate on average to around ±8%. The inverse of the Pólya-Aeppli distribution was calculated at the 95% probability using parameters for the distribution for a number of mean HTTP request rates $\bar{x}$ using an estimate of the standard deviation $\tilde{s}$ obtained by applying Equation 5.1. This inverse, expressed in Equation 6.1, was solved to find j over a range of mean request rates.

$$P(X < \bar{x} + j\tilde{s}) = 0.95 \qquad \textbf{(Eqn 6.1)}$$

It was found that j varies with mean request rate but asymptotically approaches a value of approximately 1.72 for a mean request rate of over 100 HTTP requests per second. Substituting Equation 5.1 in for $\tilde{s}$ in Equation 6.1 and using the value of 1.72 for j gives an estimate for the 95% quantile of per second HTTP request rate which is shown in Equation 6.2.

$$95\% \text{ Request Rate Quantile } = 1.32\bar{x} + 7.33 \quad \textbf{(Eqn 6.2)}$$

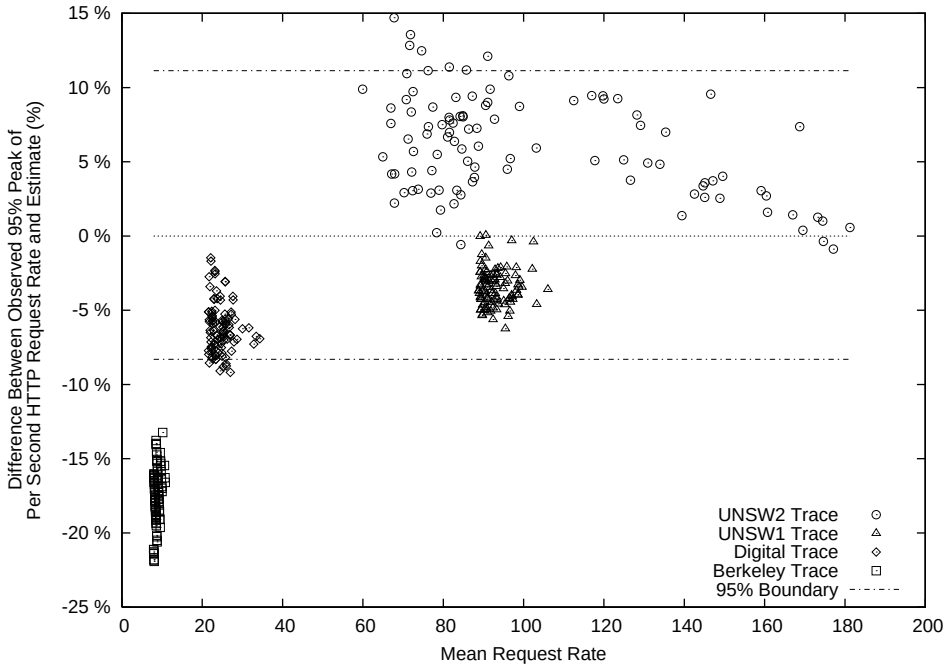


Figure 6.6 Rule of Thumb for the Estimation of the Expected Peak Per-Second Request Rate Compared to the Busy Hours

The linear relationship between the mean and standard deviation of per second request rate does not hold for mean request rates less than approximately 10 HTTP requests per second. Hence the estimate of peak request rate does not hold in this region either. At mean request rates of between 10 and 100 HTTP requests per second the estimate is expected to be slightly lower than the actual quantile as the multiplier j is approximately 1.8. In practice the difference is small. At the worst case of a mean request rate of 10 HTTP requests per second Equation 6.2 under estimates peak HTTP request rate quantile by just 2.4%.

To test the proposed second rule of thumb the 95% quantile of peak HTTP request rate for the top 400 busy hours from the aggregate Web traffic traces was compared

to the estimate. Figure 6.6 shows the comparison. Excluding the Berkeley trace 95% of the busy hours had an observed peak rate that differed from the estimate by -8.3 to 11.4%.

An example of the use of this second rule of thumb is; if we had an hour of Web traffic in which the only information known was that the mean request rate is 100 HTTP requests per second then the 1 in 20 peak request rate would be expected to be approximately 139 HTTP requests per second. An alternative estimate is that the peak HTTP request rate would be expected to fall between approximately 128 and 155 HTTP requests per second.

6.3 Comparison to Marginal Distributions from Synthetic Workloads and Models

The third application of the Pólya-Aeppli model of HTTP request rate is in the assessment of HTTP request arrival models proposed by others. Three factors can be examined to see if a proposed HTTP arrival model appears to provide for a HTTP request rate that resembles actual Web traffic:

1. Does the marginal distribution of per second HTTP request rate have a Pólya-Aeppli shape?

2. Is the standard deviation of per second HTTP request rate similar to the expected value provided by Equation 5.1?

3. Is the peak per second HTTP request rate (the 95% quantile) similar to the expected value provided by Equation 6.2?

Together these criteria can be regarded as a “sanity check” for a model of HTTP request rate.

A number of models of HTTP request arrivals were discussed in Section 2.3. Four of these models were identified as suitable for comparison with the results in this dissertation. These models were:

1. Fractional Sum-Difference Model by Cao and Cleveland [Cao 01]

2. PolyMix-4 Model by the Measurement Factory [Rousskov 01]

3. Empirical Single User Model by Mah [Mah 97]

4. Single User ON/OFF Model by Deng [Deng 96]

A sequence of HTTP request arrival times simulating a number of hours of aggregate Web traffic was obtained for each of these four models. For each of the simulated traffic streams the marginal distribution of per second HTTP request rate was compared to the Pólya-Aeppli distribution. The standard deviation and 95% quantile of peak request rate were also compared to the rules of thumb proposed earlier in this chapter.

Using PP and QQ plots it was found that all four of the models generated HTTP request arrivals with a marginal distribution of per second request rate which had a good fit with the Pólya-Aeppli distribution. The parameters for the Pólya-Aeppli distribution were matched to the mean and standard deviation of the simulated traffic using the MOM. However, the fact that the marginal distribution of per second HTTP request rate has a Pólya-Aeppli shape does not indicate if the variance of the simulated traffic is realistic. The Pólya-Aeppli has a bell like shape which can be thin or fat. In Section 6.2 two rules of thumb were proposed for expected standard deviation and peak of the per second request rate. When compared to these rules of thumb the simulated traffic was either significantly less or significantly more variant than the would be expected of actual Web traffic. Only the model from Cao produced some samples of Web traffic that fell inside the expected range. A summary of the comparison of the simulated HTTP request traffic is shown in Table 6.1. The

Table 6.1 Summary of Comparison with Four HTTP Request Arrival Models

Model	Cao[a]	Deng	Mah	PolyMix-4
Match to Pólya-Aeppli	good	good	good	good
Request Rate (req / sec)	35.4	34.9	35.1	100.22
Number of Simulated Users or Sources	N/A	12195	15557	250
Hourly HTTP Request Rate per User (req / hour)	N/A	10.3	8.1	1443
Expected Std. Dev. Request Rate (req / sec) [Expected Range (req / sec)]	10.8 [8.5 - 14.2]	10.8 [8.5 - 14.1]	10.8 [8.5 - 14.2]	22.9 [18 - 30.1]
Std. Dev. Request Rate (req / sec) [% Diff. from Expected]	14.7 [+36%]	7.5 [-30%]	16.5 [+53%]	13.7 [-40%]
Expected 95% Peak Request Rate (req / sec) [Expected Range (req / sec)]	54.0 [49.5 - 60.2]	53.4 [49 - 59.5]	53.6 [49.2 - 59.7]	139.6 [128 - 155.5]
95% Peak Request Rate [% Diff. from Expected]	62.5 [+16%]	48 [-10%]	65 [+21%]	123.4 [-12%]

a. Mean of 1000 independent hours of artificially generated traffic

table shows both the point estimate of the expected standard deviation and peak request rate and the range in which both these values is expected to lie. The Deng and Polymix-4 models produced Web traffic that had a lower standard deviation of per-second HTTP request rate and lower peak rate than would be expected of actual Web traffic. The model from Mah had a higher standard deviation of per second HTTP request rate and higher peak rate than would be expected of actual Web traffic. The model proposed by Cao produced a range of HTTP request rate traffic some of which fell inside the expected range for standard deviation and peak HTTP request rate.

Testing for a Pólya-Aeppli shaped marginal distribution of per second HTTP request rate does not discriminate between different models. All four of the proposed models had a Pólya-Aeppli shaped marginal distribution. A more discriminatory test is comparison of the simulated HTTP request traffic stream against the proposed rules of thumb. Even given the wide expected range for the standard deviation and peak per second HTTP request rate from Section 6.2 only one of the models [Cao 01] appears to generate some samples of traffic that appear to have an overall realistic per second HTTP request rate. The assessment of whether or not the per second HTTP request rate generated by the traffic model appears realistic does not of course include any assessment of the auto-correlation of HTTP request rate which is outside the parameters of this discussion.

Another interesting aspect of the per user models of HTTP request rate is the number of simulated users (sources) of Web traffic must be aggregated to produce the desired mean HTTP request rate. The models by Deng [Deng 96] and Mah [Mah 97], which are both proposed as per-user models of HTTP request rate, produce quite low average per user HTTP request rates of 10.3 and 8.1 HTTP requests per hour respectively. The PolyMix-4 model has a quite high average request rate of 1443 HTTP requests per hour per source of traffic. However the PolyMix-4 model does not claim to model per-user HTTP traffic and is designed explicitly for aggregation to produce a traffic stream with the desired mean HTTP request rate. In contrast the aggregate Web traffic examined in previous chapters ranges from an average of 47 HTTP requests per hour per source in the Digital trace and an average of 291 HTTP requests per hour per source in the UNSW2 trace (figures detailed previously in Table 3.1).

The following sections briefly describe the simulation of HTTP request rate from each of the four models, the comparison with the Pólya-Aeppli distribution and the rules of thumb. For a discussion of the models themselves please refer to Section 2.3 and appropriate references.

6.3.1 Fractional Sum-Difference Model by Cao

Cao has proposed a model for HTTP request inter-arrival times based on a Weibull marginal distribution with long range dependence (LRD) described by a "fractional sum-difference" (FSD) model [Cao 01]. Cao found that the shape and scale of the matched Weibull distribution varied with mean request rate and that HTTP request inter-arrival times tended towards independence and exponential with rising mean request rate. Cao used FSD to model the auto-correlation between HTTP request inter-arrival times and the Weibull distribution to model the marginal distribution. Both parts of the model are a function of the mean HTTP request rate ρ.

The simulation of HTTP inter-arrival times follows a number of steps outlined in [Cao 01]. The first step is to simulate the LRD time series s_i specified by the fractional autoregressive integrated moving average (ARIMA) model shown in Equation 6.3 (reproduced from [Cao 01]). In Equation 6.3, B is the backward shift operator, d is a value $0 \le d < 0.5$ and ε is Gaussian white noise with zero mean and variance specified by Cao.

$$(1 - B)^d s_i = \varepsilon_i + \varepsilon_{i-1} \quad \textbf{(Eqn 6.3)}$$

For the data used here the s_i were generated using the fast fourier transform method described in [Stoev 04] adjusted so that the variance of the Gaussian white noise was that specified by Cao. The other parameters used in the remainder of the process of simulating the HTTP request inter-arrival times are shown in Table 6.2 These

Table 6.2 Parameters Used in the Simulation of FSD Inter-arrival model by Cao

Parameter	Value
d	0.45
ρ	33.87
$\theta(\rho)$	0.84
$\lambda(\rho)$	0.655

parameters were specified in [Cao 01] with the exception of the scale parameter for the marginal Weibull distribution, $\theta(\rho)$. This was calculated using the relationship between the mean μ, shape $\lambda(\rho)$ and scale $\theta(\rho)$ parameters for the Weibull distribution from [Law 91 p. 333] shown in Equation 6.4 where $\rho = 1/\mu$.

$$\mu = \frac{\lambda(\rho)}{\theta(\rho)}\Gamma\left(\frac{1}{\lambda(\rho)}\right) \quad \textbf{(Eqn 6.4)}$$

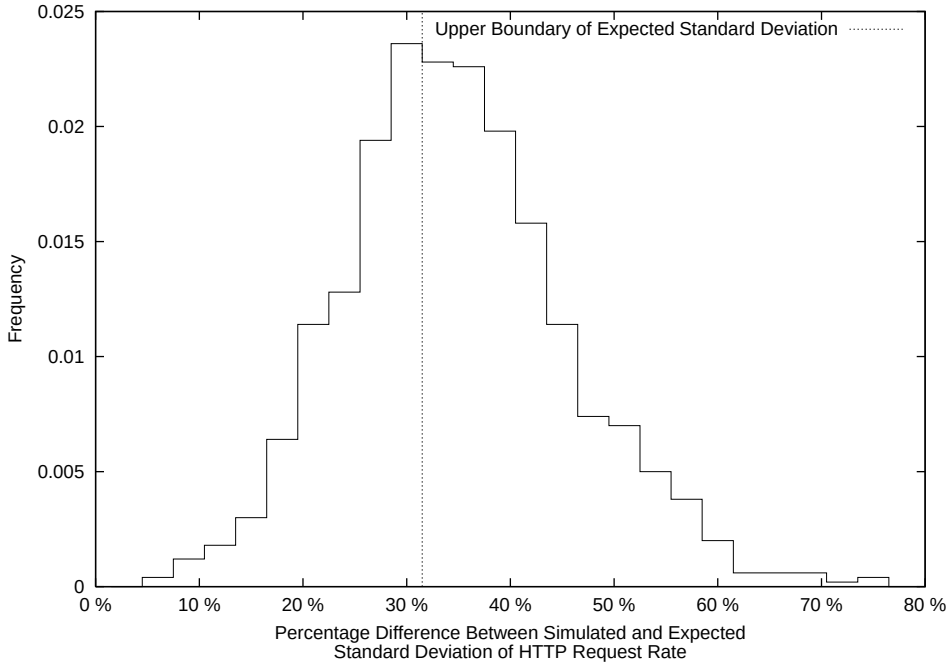


Figure 6.7 Histogram of the Difference in Standard Deviation of HTTP Request Rate in Simulated Traffic from the FSD Model Compared to Estimate of Expected Standard Deviation Using Equation 5.1

The parameter *d* of 0.45 indicates a strong long range dependence and the hours of simulated traffic varied significantly between simulation runs. To enable comparison with the Pólya-Aeppli based model 1000 independent simulated hours of traffic were generated. The target request rate, parameter ρ, was 33.87 HTTP requests per second. The mean request rate for the hours of simulated traffic varied from 21.5 to 58.9 HTTP requests per second. For each of the thousand hours the standard devia-

tion and peak per second HTTP request rate were compared with the expected values generated from the rules of thumb.

A histogram plot of the difference between the standard deviation of simulated traffic and the expected standard deviation is shown in Figure 6.7. The mean difference was that the standard deviation of the simulated request rate was 36% above the expected HTTP request rate. In Section 6.2 the 95% range was between -21% and 32% of the expected value. The 32% upper boundary is plotted on Figure 6.7, approximately 40% of the simulated hours of traffic were within this boundary. However, of these hours only 151, approximately 15% of the total simulated hours, had a peak HTTP request rate that fell inside the -8.3% to 11.4% range for peak request rate.

The Pólya-Aeppli distribution was matched to the marginal distribution of per second HTTP request rate with the MOM. Figures 6.8 and 6.9 show PP and QQ plots respectively for nine of the simulated hours compared to the Pólya-Aeppli distribution. The 0.01, 0.05, 0.95 and 0.99 quantiles are marked on the QQ plots with horizontal lines. The marginal distribution of the simulated request rate is close to the Pólya-Aeppli distribution. On the PP plot there is some evidence of a small difference in shape as the plot is not quite a straight line, on the QQ plot the simulated traffic diverges from the Pólya-Aeppli just before the 0.99 quantile.

Overall the proposed HTTP request rate model proposed by Cao appears to produce some traffic samples that appear to have a realistic overall per second HTTP request rate. However the simulated traffic varies considerably between runs, both in terms of the mean and variance of HTTP request rate. It is suspected that this is due to the high degree of long range auto correlation in the model due to the choice of a value of 0.45 for the parameter d.

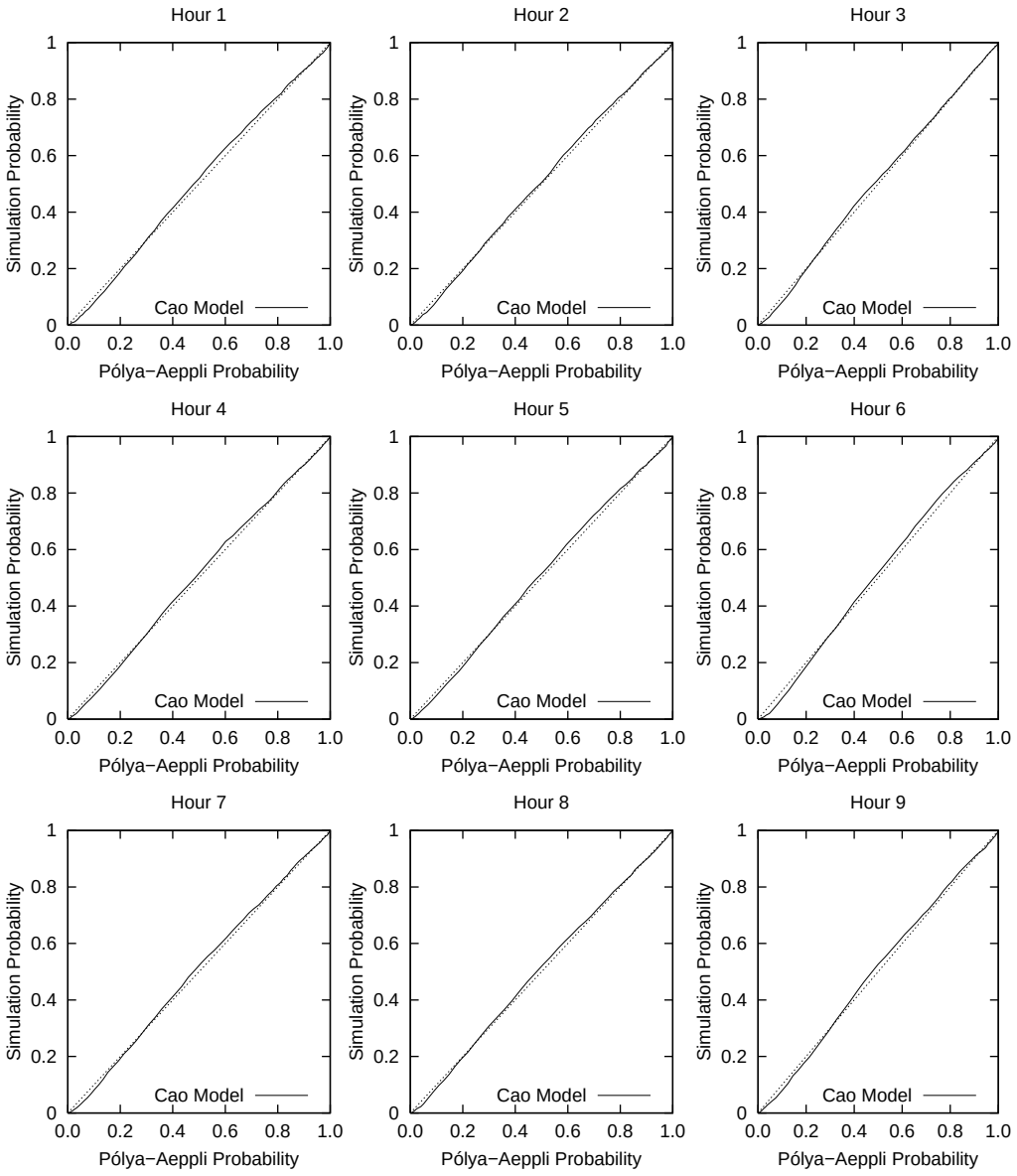


Figure 6.8 PP Plot Comparison of the Marginal Distribution of HTTP Request for Simulated Traffic from the Cao Model with the Pólya-Aeppli Distribution

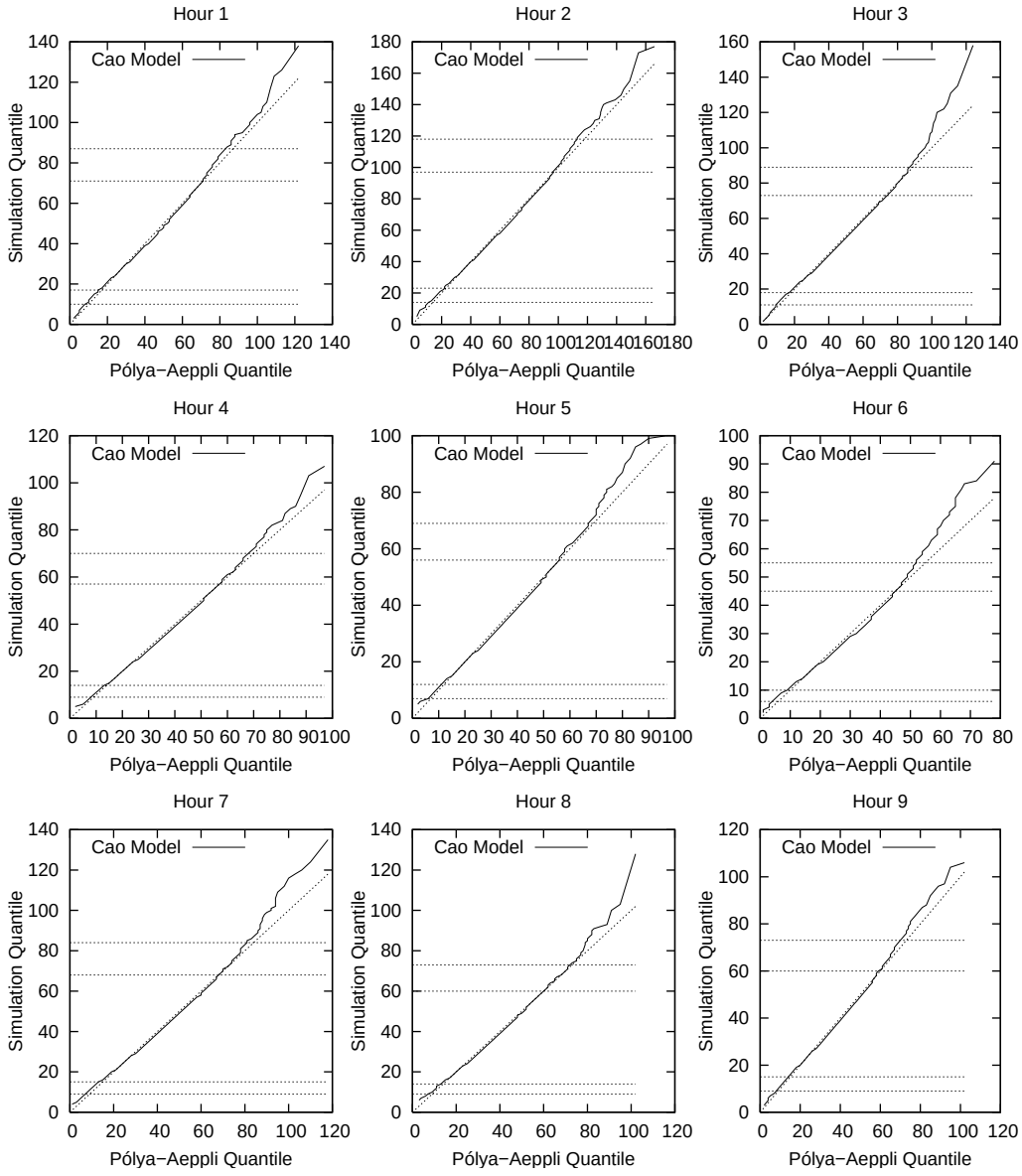


Figure 6.9 QQ Plot Comparison of the Marginal Distribution of HTTP Request for Simulated Traffic from the Cao Model with the Pólya-Aeppli Distribution

6.3.2 PolyMix-4 Model by the Measurement Factory

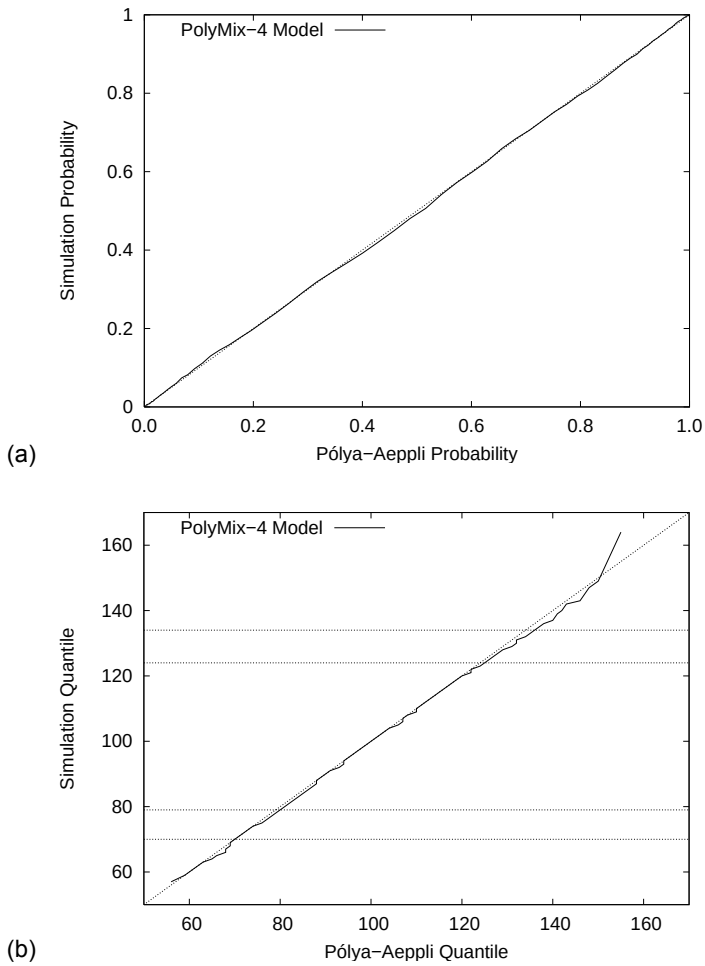


Figure 6.10 Comparison of the Marginal Distribution of HTTP Request for Simulated Traffic from the PolyMix-4 Model with the Pólya-Aeppli Distribution
(a) PP Plot
(b) QQ Plot

The Measurement Factory has an artificial Web traffic workload called “PolyMix-4” which is part of the PolyGraph software used to benchmark Web proxy servers. The workload was used in the last Measurement Factory benchmarking effort, the “Fourth Cache-Off”, results of which can be found at [Rousskov 01]. The workload has a considerable number of parameters and a detailed description is available from the Measurement Factory Web site[1]. Trace files for the Fourth Cache-Off are available from the Measurement Factory[2] but unfortunately the statistics are averaged out

1. URL is `http://www.web-polygraph.org/docs/workloads/polymix-4/`

2. URL is `http://www.measurement-factory.com/results/public/cache-off/N04/logs/`

for five second periods and were not suitable for comparison with the results here. An alternative trace file with statistics averaged over one second periods was obtained [NICTA 04].

The per second HTTP request rate was obtained from the trace file for the period of time in which proxy server was under full load. The mean request rate and other details were shown previously in Table 6.1. Unlike the Cao model the simulated Web traffic was quite constant between different hours and the hour examined was representative of other hours of simulated traffic. The per second HTTP request rate had a marginal distribution with a Pólya-Aeppli shape, PP and QQ plots are shown in Figure 6.10.

Despite the Pólya-Aeppli shaped marginal distribution the simulated aggregate HTTP request rate had too low a standard deviation and peak HTTP request rate to be considered realistic. The standard deviation of the simulated HTTP request rate was 40% below the expected value and the 95% peak request rate was approximately 16 HTTP requests per second (12%) below the expected value.

6.3.3 Empirical Web Traffic Model by Mah

The Mah model of Web traffic is a ON/OFF model using empirical CDF data [Mah 97]. The data is available for download along with some C++ source code for generation of random samples from each CDF used in the model. The difficulties encountered when using the model to simulate an aggregate HTTP request traffic stream were selecting the number of users to aggregate and simulating a HTTP request stream with a constant mean rate. Trial and error was used to determine the number of users to aggregate to obtain an approximate mean HTTP request rate of 35 HTTP requests per second. The OFF periods used in the model have a long tail which resulted in the simulation having a large initial transient.

The simulated HTTP request rate varied more than expected. Both the standard deviation and peak HTTP request rate were higher than the expected values (shown previously in Table 6.1). The marginal distribution of per second HTTP request rate did have an approximate Pólya-Aeppli shape. The PP and QQ plots are shown in Figure 6.11. There is some divergence on the QQ plot just before the 0.99 quantile

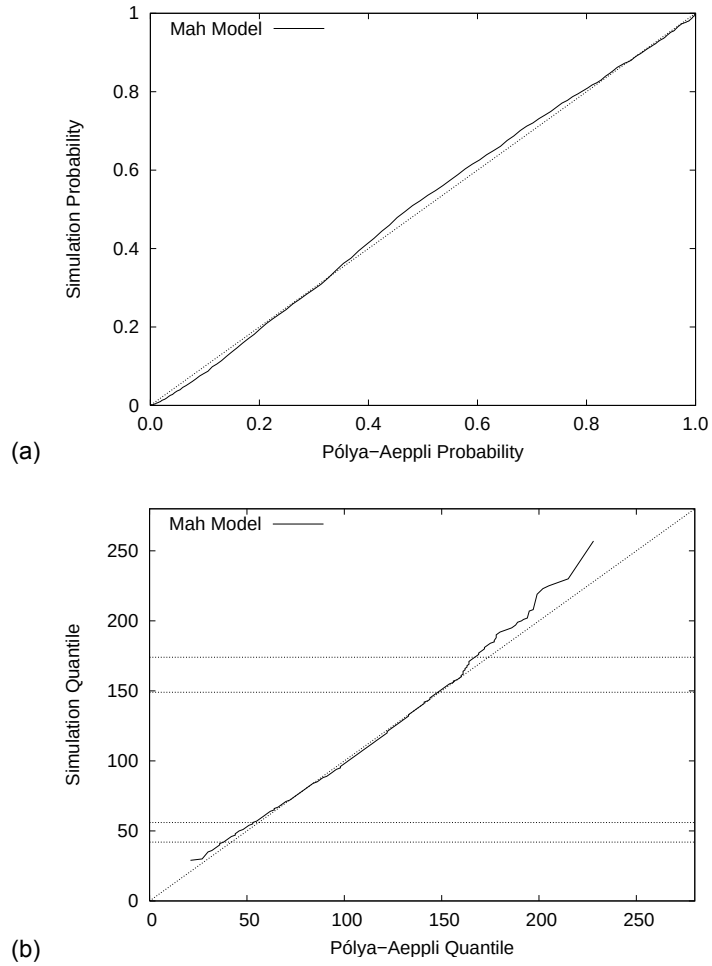


Figure 6.11 Comparison of the Marginal Distribution of HTTP Request for Simulated Traffic from the Mah Model with the Pólya-Aeppli Distribution
(a) PP Plot
(b) QQ Plot

and the PP plot has a slight S shape. Overall, due to the high request rate variance this model does not appear realistic.

6.3.4 Single User ON/OFF Model by Deng

The Web traffic model from Deng is also an ON/OFF model [Deng 96]. The same problems were encountered in simulating HTTP request arrivals using this model as the Mah ON/OFF model. Trial and error was used to find the number of users to aggregate together to obtain an approximate mean HTTP request rate of 35 HTTP requests per second. Like the Deng model the simulation had a large initial transient.

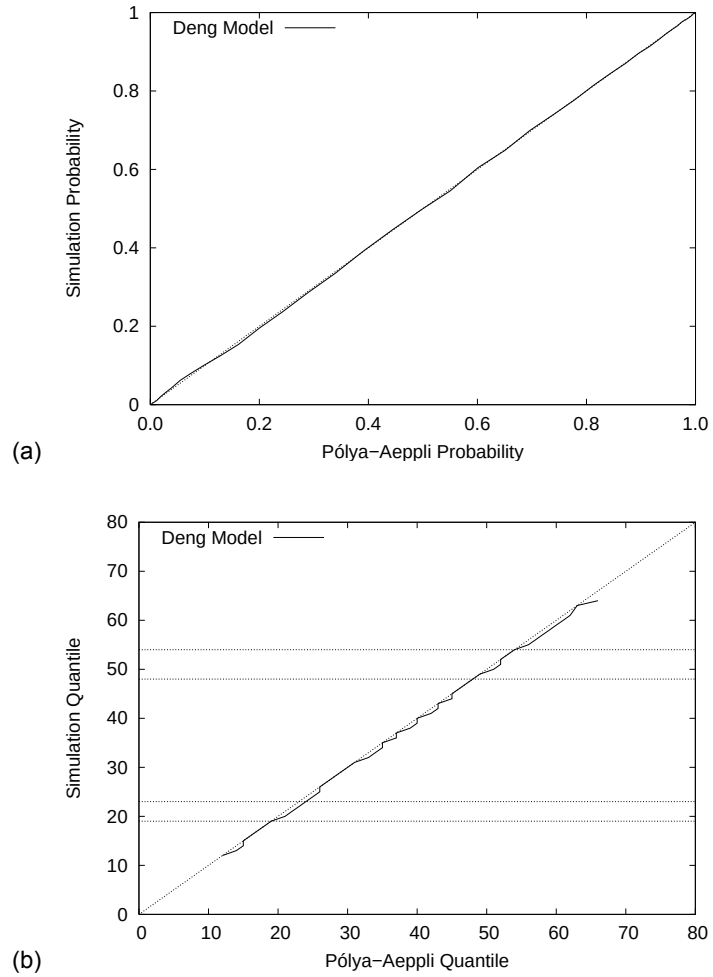


Figure 6.12 Comparison of the Marginal Distribution of HTTP Request for Simulated Traffic from the Deng Model with the Pólya-Aeppli Distribution
(a) PP Plot
(b) QQ Plot

The simulated HTTP request rate had a lower variance than expected. The standard deviation and peak per second HTTP request rate were both below expected values (shown previously in Table 6.1). The marginal distribution of per second request rate did have a Pólya-Aeppli shape. PP and QQ plots are shown in Figure 6.12. The match with the Pólya-Aeppli was very close, the closest of the four models examined. However, overall the simulated traffic does not appear realistic due to the lower than expected standard deviation and peak request rates.

6.4 Conclusion

Three possible applications of the Pólya-Aeppli based model were examined in this chapter; estimation of peak HTTP request rate, use in proposing two new rules of thumb for HTTP request rate and sanity checking simulated HTTP traffic generated by models proposed by others.

It was found that the Pólya-Aeppli distribution can provide excellent estimates of peak HTTP request rate. The most accurate estimates are obtained when the sample mean and standard deviation are known. If the standard deviation is not known then a less accurate estimate of peak request rate is possible using an estimate of the standard deviation obtained from the mean request rate exploiting a linear relationship between the two. Equation 5.1 can be used in the absence of any other information concerning the linear relationship. A better alternative is to exploit previous knowledge from the traffic under measurement.

Two new rules of thumb were proposed for estimating the standard deviation and peak of per second HTTP request rate given a value for the mean. The rules were compared to 100 busy hours from each of the aggregate traffic traces and shown to approximately hold. The comparison with the busy hours suggested some 95% confidence intervals of -21 to 32% for the estimate of expected standard deviation of request rate and -8.3 to 11.4% for the estimate of expected peak request rate.

The Pólya-Aeppli result for the marginal distribution of per second HTTP request rate and the rules of thumb were compared with HTTP traffic simulated using four different models of HTTP request arrival. As a discriminatory tool the Pólya-Aeppli distribution by itself is not effective as it matched the shape of the per second marginal distribution of HTTP request rate for all four of the models tested. The problem is that simple matching of the distribution to the simulated data does not reveal whether the request rate variance is realistic. The bell shape of the Pólya-Aeppli distribution can shrink or grow to cover a wide range. The rules of thumb are more discriminatory, only one of the models [Cao 01] produced some hours of Web traffic with both a standard deviation and peak per second HTTP request rate that matched the rules of thumb. Approximately 15% of the hours of simulated Web traffic from the Cao model had a standard deviation of HTTP request rate and a peak HTTP request rate that fell inside the estimated confidence intervals for the rules of thumb.

The other models either resulted in simulated traffic that had too high a request rate variance [Mah 97], with a higher than expected standard deviation and peak per second HTTP request rate, or too low a request rate variance [Deng 96, Rousskov 01], with lower than expected standard deviation and peak HTTP request rates.

7. Conclusion and Future Work

7.1 Conclusion

This dissertation has presented the Pólya-Aeppli probability distribution as a new model for the marginal distribution of HTTP request rate. The model applies to the aggregate traffic generated by a population of Web users accessing the Web over the Internet. The model is shown to provide excellent estimates of peak HTTP request rate and prompted the proposal of two new rules of thumb.

The choice of the Pólya-Aeppli distribution was motivated by observations of the distribution of HTTP requests made by single users and the distribution of the number of active Web users. It is shown that the number of HTTP requests made individually by users browsing the Web has a geometric distribution. It is also shown that the number of users browsing the Web in a given second has a Poisson distribution. The Pólya-Aeppli is a Poisson stopped sum of the geometric distribution and therefore an appropriate combination of the two observations.

The Pólya-Aeppli result is less complex than previously reported models of HTTP request rate. In general, previous results are based on single user models of HTTP request rate and their use to describe aggregate traffic is difficult. There are fewer existing models of aggregate HTTP request rate. One comparable model [Cao 01] to that proposed here describes at HTTP request inter-arrival time and not HTTP request rate.

The presented results are based on analysis of traces of Web traffic collected over an eight year period between 1994 and 2002. The results concerning aggregate Web traffic were based on four large independent traces of Web traffic collected between 1996 and 2002. The Pólya-Aeppli model is shown to be a good match with samples of traffic from all four traces.

The proposed rules of thumb allow for the estimation of the standard deviation and peak per second HTTP request rate from a given mean rate. These rules are shown to provide approximate estimates of both. Future comparison with new traces may lead to refinement of both rules.

The Pólya-Aeppli distribution and the two rules of thumb are used to compare four models of HTTP request arrival proposed by others [Cao 01, Deng 96, Mah 97, Rousskov 01] to actual Web traffic. While the Pólya-Aeppli distribution is an excellent model of actual Web traffic it is found to be not a good discriminatory tool for checking artificial Web traffic generated by models. The two rules of thumb provide better evaluation criteria of artificial Web traffic. Only one model assessed produced some samples of Web traffic that appeared realistic [Cao 01].

7.2 Future Work

7.2.1 Non-Poisson Nature of HTTP Traffic

In Section 3.1 the scattergrams for the Lexis ratio versus the number of traffic sources suggest that HTTP request traffic trends toward increasing variance with aggregation. An area of future work is to compare this against the result from Morris [Morris 00] who found Web traffic smoothing under aggregation.

7.2.2 Single User HTTP Request Rate

An analytical GOF test was used on hour samples of user traffic and not the shorter duration samples. An area of future work would be to perform GOF testing on the one minute and one second samples with consideration of the appropriate amount of sub-sampling that may be required.

The SNRC trace used in Chapter 4 is now approximately 7 years old. An area of future work would be to collect and analyse a similar long term, per user trace. This could be used to validate if the number of HTTP requests generated per user per time period has a geometric distribution like the SNRC trace. Related issues that could be examined are; the effect of client side caching and long term variation in HTTP request rates.

7.2.3 The Pólya-Aeppli Model

The issue of stationarity in the arrival of HTTP requests could be examined further. It has been stated that HTTP request inter-arrival times are inherently non-stationary due to the fact that the number of users (traffic sources) varies over time

[Cleveland 00a]. Cleveland and Cao therefore choose to examine short time intervals, 15 minute [Cleveland 00a] and 5 minute [Cao 01], and then looked for variation in model parameters as a function of HTTP request arrival rate. However, in the graphs shown in Appendix G, even the Berkeley trace (the trace with the lowest mean request rate and lowest number of unique traffic sources) inter-arrival times for user sessions were on average less than 30 seconds. So even over a 5 minute trace it would be reasonable to expect that the number of users/sources is not constant. In choosing trace samples to examine Cleveland and Cao explicitly selected samples in which the variation in arrival rate due to diurnal cycle was minimised. Likewise the hours selected from the traces examined in this dissertation were selected explicitly due to the fact that diurnal variation was negligible. But the difference is that results presented in this dissertation have been built on the observation that the number of active Web users can fluctuate. An area of future work would be to examine if this fluctuation in users/sources was around a constant mean (as is implied by the constant request rate). Another area of future work would be to look in detail at the issue of stationarity and LRD of HTTP request rate. If HTTP request rate is non-stationary what does this imply for previously presented results?

Another possible area of future work is to examine the robustness of the Pólya-Aeppli model. Questions that could be explored are whether or not the distribution retains a good fit to smaller sampling periods and the sensitivity of distribution parameters to sample period.

7.2.4 The Proposed Rules of Thumb for Estimation of Standard Deviation and Peak HTTP Request Rate

The two proposed rules of thumb would benefit from further comparison with other samples of HTTP request rate traffic. There is scope for the parameters of both Equation 5.1 and Equation 6.2 to be “tuned” with the addition of data from other traces. Section 6.2 also provided figures for a 95% range around the point estimates provided by the rules of thumb. With repeated use on a large number of data sets the usefulness of the expected range could be assessed or a more general level of comfort for the accuracy of the rules of thumb may evolve.

The plots shown in Figures 6.5 and 6.6 are interesting in that the data from the Berkeley trace is clustered towards the lower left. The linear relationship between mean

and standard deviation of aggregate per second HTTP request rate was observed in the scattergram plots shown in Figure 5.5 for mean HTTP request rates approximately above 10 HTTP requests per second. However, it is not obvious where the relationship actually starts. Some of the data for the Berkeley trace approaches a mean request rate of 10 HTTP requests per second (5 out of 100 samples over 10 HTTP requests per second with the highest being 10.7). However, estimated standard deviation and peak HTTP request rates from the rules of thumb are still higher than the observed values from the Berkeley trace. An area of future work is to examine the point at which the linear relationship holds more closely. There is some evidence on Figures 6.5 and 6.6 for a rise in convergence between observed and estimated standard deviation and peak HTTP request rate with rising mean HTTP request rate for the Berkeley trace data. This may show that the linear relationship between mean and standard deviation may only apply at slightly higher values, say 20 HTTP requests per second. Looking for evidence of the linear relationship and applying the rules of thumb in other traces would also be useful.

7.2.5 Sanity Checking HTTP Request Models

The model proposed by [Cao 01] produces some samples of HTTP request rate traffic that have a HTTP request rate marginal distribution that resembles actual Web traffic. However, the majority of simulated hours have too high a variance in HTTP request rate. The standard deviation and peak HTTP request rate are both higher than expected. An area of future work would be to examine the model proposed by Cao more closely and, staying within his proposed framework, see if the model parameters could be adjusted to produce a more realistic looking HTTP request rate. The obvious parameters to examine first are those concerned with the LRD.

References

Abrahamsson 00 — Henrik Abrahamsson and Bengt Ahlgren, “Using Empirical Distributions to Characterize Web Client Traffic and to Generate Synthetic Traffic”, *Proc. Globecom 2000*, San Francisco, 2000, pp. 428-433

Albrecht 04 — Conan C. Albrecht, “How Clean is the Future of SOAP?”, *Communications of the ACM*, Vol. 47, No. 2, 2004, pp. 66-68

Anklesaria 93 — F. Anklesaria, M. McCahill, P. Lindner, D. Johnson, D. Torrey and B. Alberti, “The Internet Gopher Protocol: A Distributed Document Search and Retrieval Protocol”, RFC1436, University of Minnesota, 1993

Arlitt 95 — Martin F. Arlitt and Carey L. Williamson, “A Synthetic Workload Model for Internet Mosaic Traffic”, *Proc. Summer Computer Simulation Conference*, Ottawa, 1995, pp. 852-857

Arlitt 99 — Martin Arlitt, Rich Friedrich and Tai Jin, “Workload Characterization of a Web Proxy in a Cable Modem Environment”, Technical Report HPL-1999-48, HP Laboratories Palo Alto, 1999

Baird 90 — P.M. Baird, “Hypertext and Hypermedia for the Information Scientist”, *Proc. IEE Colloquium on Hypertext*, London, 1990, pp. 5/1-5/5

Barford 98a — Paul Barford and Mark Crovella, “Generating Representative Web Workloads for Network and Server Performance Evaluation”, *Proc. ACM SIGMETRICS*, 1998, pp. 151-160

Barford 98b — Paul Barford, Azer Bestavros, Adam Bradley and Mark Crovella, “Changes in Web Client Access Patterns; Characteristics and Caching Implications”, Technical Report BUCS-TR-1998-023, Boston University Computer Science Department, 1998

Benda 97 — Miro Benda, “The Right User Interface”, *IEEE Internet Computing*, Vol. 1, No. 2, 1997, pp 68-70

Berners-Lee 92a — Tim Berners-Lee, Robert Cailliau, Jean-Francois Groff and Bernd Pollermann, “World-Wide Web: The Information Universe”, *Electronic Networking: Research Applications and Policy*, Vol. 1, No. 2, 1992

Berners-Lee 92b — T. J. Berners-Lee, R. Cailliau and J. F. Groff “The world-wide web”, *Computer Networks and ISDN Systems*, Vol. 25, 1992, pp. 454-459

Berners-Lee 94 — Tim Berners-Lee, L. Masinter and M. McCahill, “Uniform Resource Locators (URL)”, RFC1738, CERN, Xerox PARC and University of Minnesota, 1994

Berners-Lee 96 — Tim Berners-Lee, R. Fielding and H. Frystyk, “Hypertext Transfer Protocol -- HTTP/1.0”, RFC1945, MIT/LCS and UC Irvine, 1996

Bolot 96 — Jean-Chrysostome Bolot and Philipp Hoschka, “Performance Engineering of the World-Wide Web: Application to Dimensioning and Cache Design”, *Computer Networks and ISDN Systems*, Vol. 28, No 7-11, 1996, pp. 1397-1405

Borenstein 93 — N. Borenstein and N. Freed, “MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies”, RFC1521, Bellcore, Innosoft, 1993

Bray 96 — Tim Bray, “Measuring the Web”, *Computer Networks and ISDN Systems*, Vol. 28, Issues 7-11, 1996, pp. 993-1005

Buchholz 02 — Sven Buchholz, Steffen Jaensch and Alexander Schill, “Flexible Web Traffic Modeling for New Application Domains”, Proc. IASTED International Conference on Applied Modelling and Simulation (AMS 2002), Cambridge MA, 2002

Busari 01 — Mudashiru Busari and Carey Williamson, “On the Sensitivity of Web Proxy Cache Performance to Workload Characteristics”, *Proc. INFOCOM 2001*, Anchorage, 2001, pp. 1225-1234

Bush 45 — Vannevar Bush, “As We May Think”, *The Atlantic Monthly*, July 1945

Cáceres 91 Ramón Cáceres, Peter B. Danzig, Sugih Jamin and Danny J. Mitzel, "Characteristics of Wide-Area TCP/IP Conversations", *ACM SIGCOMM Computer Communication Review*, Vol. 21, No. 4, 1991, pp. 101-112

Cao 01 Jin Cao, William S. Cleveland, Dong Lin and Don X. Sun, "On the Nonstationarity of Internet Traffic", *Proc ACM SIGMETRICS*, 2001, pp. 102-112

Cao 02a Jin Cao, William S. Cleveland, Dong Lin and Don X. Sun, "The Effect of Statistical Multiplexing on the Long-Range Dependence of Internet Packet Traffic", Bells Labs Technical Report, 2002

Cao 02b Jin Cao, William S. Cleveland, Dong Lin and Don X. Sun, "Internet Traffic Tends *Toward* Poisson and Independent as the Load Increases", *Nonlinear Estimation and Classification*, eds. C. Holmes, D. Denison, M. Hansen, B. Yu, and B. Mallick, Springer, New York, 2002

Casilari 01 Eduardo Casilari, Francisco J. González and Francisco Sandoval, "Modeling of HTTP Traffic", *IEEE Communications Letters*, Vol. 5, No. 6, June, 2001, pp. 272-274

Catledge 95 Lara D. Catledge and James E. Pitkow, "Characterizing Browsing Strategies in the World-Wide Web", *Computer Networks and ISDN Systems*, Vol. 27, 1995, pp. 1065-1073

Chatfield 96 Chris Chatfield, *The Analysis of Time Series: An Introduction*, Chapman & Hall, London, 1996

Choi 99 Hyoung-Kee Choi and John O. Limb, "A Behavioral Model of Web Traffic", *Proc. ICNP'99*, Toronto, 1999, pp. 327-333

Cisco 99a Cisco Systems Inc., "White Paper; Network Caching", Available at `http://www.cisco.com/warp/public/cc/pd/cxsr/500/tech/cds_wp.htm`, last viewed 27 May 2004

Cisco 99b — Cisco Systems Inc., “Using Cisco Cache Engine, Version 1.7: Chapter 4, Optimizing the Cache Engine”, Available at `http://www.cisco.com/en/US/products/sw/contsw/ps547/products_user_guide_chapter09186a0080087513.html`, last viewed 27 May 2004

Cleveland 00a — William S. Cleveland, Dong Lin and Don X. Sun, “IP Packet Generation: Statistical Models for TCP Start Times Based on Connection-Rate Superposition”, *Proc. ACM SIGMETRICS 2000*, Santa Clara, 200, pp. 166-177

Cleveland 00b — William S. Cleveland and Don X. Sun, “Internet Traffic Data”, *Journal American Statistical Association*, Vol. 95, 2000, pp. 979-985

Cohen 99 — Edith Cohen and Haim Kaplan, “Exploiting Regularities in Web Traffic Patterns for Cache Replacement”, *Proc. STOC’99*, Atlanta, 1999, pp. 109-117

Crovella 96a — Mark E. Crovella and Azer Bestavros, “Self-Similarity in World-Wide Web Traffic Evidence and Possible Causes”, *Proc. ACM SIGMETRICS’96*, Philadelphia, 1996

Crovella 97 — Mark E. Crovella and Azer Bestavros, “Self-Similarity in World-Wide Web Traffic: Evidence and Possible Causes”, *IEEE/ACM Transactions on Networking*, Vol. 5, No. 6, 1997, pp. 835-846

CSTB 02 — Computer Science and Telecommunications Board, National Research Council, *Broadband: Bringing Home the Bits*, National Academy Press, Washington, 2002

Cunha 95 — Carlos Cunha, Azer Bestavros and Mark E. Crovella, “Characteristics of WWW Client-based Traces”, Technical Report BU-CS-95-010, Boston University Computer Science Department, 1995

Danzig 91 — Peter B. Danzig and Sugih Jamin, “tcplib: A Library of TCP Internetwork Traffic Characteristics”, Technical Report USC-CS-91-495, Computer Science Department, University of Southern California, 1991

Deng 96 — Shuang Deng, “Empirical Model of WWW Document Arrivals at Access Link”, *Proc. ICC’96*, Dallas, 1996, pp. 1797-1802

ETSI 98 — “Selection Procedures for the Choice of Radio Transmission Technologies of the UMTS”, ETSI Technical Report TR 101 112 (UMTS 30.03 version 3.2.0), ETSI, 1998

Feldmann 98a — A. Feldmann, A. C. Gilbert, W. Willinger and T. G. Kurtz, “The Changing Nature of Network Traffic: Scaling Phenomena”, *Computer Communication Review*, April, 1998, pp. 5-29

Feldmann 98b — Anja Feldmann, Jennifer Rexford and Ramón Cáceres, “Efficient Policies for Carrying Web Traffic Over Flow-Switched Networks”, *IEEE/ACM Transactions on Networking*, Vol. 6, No. 6, 1998, pp. 673-685

Fielding 99 — R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach and T Berners-Lee, “Hypertext Transfer Protocol -- HTTP/1.1”, RFC2616, IETF, 1997

Glassman 94 — Steven Glassman, “A Caching Relay for the World Wide Web”, *Computer Networks and ISDN Systems*, Vol. 27, 1994, pp. 165-173

Google 02 — Google home page, Available at `http://www.google.com`

Gribble 97a — Steven D. Gribble, “UC Berkeley Home IP HTTP Traces”, 1997, Available at `http://ita.ee.lbl.gov/html/contrib/UCB.home-IP-HTTP.html`, last viewed 27 May 2004

Gribble 97b — Steven D. Gribble and Eric Brewer, “System Design Issues for Internet Middleware Services: Deductions from a Large Client Trace”, *Proc. 1997 Usenix Symposium on Internet Technologies and Systems*, Monterey (1997)

Heidemann 97 — John Heidemann, Katia Obraczka and Joe Touch, “Modeling the Performance of HTTP Over Several Transport Protocols”, *IEEE/ACM Transactions on Networking,* Vol. 5, No. 5, 1997, pp. 616-630

Hlavacs 99 Helmut Hlavacs and Gabriele Kotsis, “Modeling User Behavior; A Layered Approach”, *Proc. 7th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, 1999, Los Alamitos, pp. 218-225

Huberman 98 Bernardo A. Huberman, Peter L. T. Pirolli, James E. Pitkow and Rajan M. Lukose, “Strong Regularities in World Wide Web Surfing”, *Science*, Vol. 280, 1998, pp. 95-97

IETF 03 “Internet Official Protocol Standards”, J. Reynolds and S. Ginoza (eds), IETF RFC3600, 2003

Johnson 92 Norman L. Johnson, Samuel Kotz and Adrienne W. Kemp, *Univariate Discrete Distributions*, 2nd Ed., John Wiley & Sons, New York, 1992

Johnson 94 Norman L. Johnson, Samuel Kotz and N. Balakrishnan, *Continuous Univariate Distributions, Volume 1*, 2nd Ed., John Wiley & Sons, New York (1994)

Judge 95 John Judge, H. W. Peter Beadle and Joe Chicharo, “Modelling User Traffic in the WWW”, *Proc. Australian Telecommunications Networks & Applications Conference (ATNAC’95)*, Sydney, December 1995, pp. 163-168

Judge 97a John Judge, H.W. Peter Beadle and Joe Chicharo, “Modeling World-Wide Web Request Traffic”, *Proc. IS&T/SPIE Multimedia Computing and Networking 1997*, San Jose, 1997, pp. 92-106

Judge 97b John Judge, Joe Chicharo and H W. Peter Beadle, “The Size of HTTP Response Packets and Calculation of WWW Traffic Volumes” *Proc. IEEE/IEE International Conference on Telecommunications 1997 (ICT'97)*, Vol. 1, 1997, pp. 257-262

Judge 97c John Judge, H.W. Peter Beadle and Joe Chicharo, “Correlation of HTTP Response Packet Size and Estimating Confidence Intervals for Mean Packet Size and WWW Traffic Volume”, *Proc. APCC'97*, Sydney, 1997, pp. 382-386

Judge 98 John Judge, H.W. Peter Beadle and Joe Chicharo, “Sampling HTTP Response Packets for Prediction of Web Traffic Volume Statistics”, *Proc. IEEE Globecom’98*, Sydney, 1998, pp. 2617-2622

Judge 99 John Judge, “Estimating Peak HTTP Request Rate for a Population of Web Users”, *Proc. 10th IEEE Workshop on Local and Metropolitan Area Networks LANMAN’99*, Sydney, 1999, pp. 108-111

Judge 01 J. Judge, “A Model for the Marginal Distribution of Aggregate Per Second HTTP Request Rate”, *Selected papers from 10th IEEE Workshop on Local and Metropolitan Area Networks*, 2001, pp. 29-36

Kant 99 Krishna Kant and Youjip Won, “Server Capacity Planning for Web Traffic Workload”, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 5, No. 5, 1999, pp. 731-747

Khare 00 R. Khare and S. Lawrence, “Upgrading to TLS Within HTTP/1.1”, IETF RFC2817, 2000

Kleinrock 75 Leonard Kleinrock, *Queueing Systems, Volume 1: Theory*, John Wiley & Sons, 1975, New York

Krishnamurthy 01 Balachander Krishnamurthy and Martin Arlitt, “PRO-COW: Protocol Compliance on the Web---A Longitudinal Study”, *Proc. 2001 USENIX Symposium on Internet Technology and Systems*, San Francisco, 2001, pp. 109-122

Kroeger 99 Tom M. Kroeger, Jeffrey C. Mogul and Carlos Maltzahn, “Digital’s Web Proxy Traces”, `Available at ftp://ftp.digital.com/pub/DEC/traces/proxy/webtraces.html`, last viewed 27 May 2004

Larsen 86 Richard J. Larsen and Morris L. Marx, *An Introduction to Mathematical Statistics and its Applications*, 2nd Edition, Prentice Hall, London, 1986

Law 91 Averill M. Law and W. David Kelton, *Simulation Modeling and Analysis*, 2nd Edition, McGraw-Hill, New York, 1991

Lee 97 David C. Lee and Scott F. Midkiff, “A Sample Statistical Characterization of the World-Wide Web”, *Proc. IEEE Southeastcon*, Blacksburg VA, 1997, pp. 174-178

Lewis 97 Ted Lewis, “Wired Wired World”, *IEEE Internet Computing*, Vol. 1, No. 3, 1997, pp. 94-96

Luotonen 94 Ari Luotonen and Kevin Altis, “World-Wide Web Proxies”, *Computer Networks and ISDN Systems*, Vol. 27, 1994, pp. 147-154

Luotonen 96 Ari Luotonen, Henrik Frystyk Nielson and Tim Berners-Lee, “CERN httpd”, Available at `http://www.w3.org/Daemon/`, last viewed 27 May 2004

Luotonen 98 Ari Luotonen, *Web Proxy Servers*, Prentice Hall PTR, Upper Saddle River, New Jersey, 1998

McCreary 00 S. McCreary and k. claffy, “Trends in Wide Area IP Traffic Patterns - A View from Ames Internet Exchange”, *Proc. 13th ITC Specialist Seminar on Internet Traffic Measurement and Modelling*, Monterey, 2000.

Mah 97 Bruce A. Mah, “An Empirical Model of HTTP Network Traffic”, *Proc. INFOCOM’97*, Kobe, 1997

Maltzahn 97 Carlos Maltzahn and Kathy J. Richardson, “Comparing the Performance of CERN’s httpd and Squid”, *Proc. NLANR Web Cache Workshop*, Boulder, 1997

Mikhailov 00 Mikhail Mikhailov and Craig E. Wills, “Embedded Objects in Web Pages”, Technical Report WPI-CS-TR-00-05, Computer Science Department, Worcester Polytechnic Institute, 2000

Mogul 95 Jeffrey C. Mogul, “The Case for Persistent-Connection HTTP”, *Proc. SIGCOMM ‘95*, Cambridge, 1995, pp. 299-313

Molina 1922 Edward C. Molina, “The Theory of Probabilities Applied to Telephone Trunking Problems”, *The Bell System Technical Journal*, Vol. I, No. 2, 1922, pp. 69-81

Molina 00 — Maurizio Molina, Paolo Castelli and Gianluca Foddis, "Web Traffic Modeling Exploiting TCP Connections' Temporal Clustering through HTML-REDUCE", *IEEE Network*, Vol. 14, No. 3, 2000, pp. 46-55

Morris 00 — Robert Morris and Dong Lin, "Variance of Aggregated Web Traffic", *Proc. INFOCOM'2000*, Tel Aviv, 2000, pp. 360-366

Musciano 97 — Chuck Musciano and Bill Kennedy, *HTML The Definitive Guide*, 2nd Ed., O'Reilly & Associates, 1997

NICTA 04 — Unpublished PolyMix-4 Trace File obtained from Aidan Williams, Networks and Pervasive Computing Program, National ICT Australia, 18 March 2004

OCLC 99 — OCLC Online Computer Library Center, Inc., "June 1999 Web Statistics", 1999, Available at `http://web.archive.org/web/2000-1999re_/http://www.oclc.org/oclc/research/projects/webstats/statistics.htm`, last viewed 27 May 2004

Padmanabhan 94 — Venkata N. Padmanabhan and Jeffrey C. Mogul, "Improving HTTP Latency", *Proc. 2nd International WWW Conference '94: Mosaic and the Web*, Chicago, 1994, pp. 995-1005

Paxson 94a — Vern Paxson and Sally Floyd, "Wide-Area Traffic: The Failure of Poisson Modelling", *Proc. SIGCOMM'94*, London 1994

Paxson 94b — Vern Paxson and Sally Floyd, "Wide-Area Traffic: The Failure of Poisson Modelling (Extended Version)", technical report LBL-35238, Lawrence Berkeley Laboratory, 1994

Paxson 94c — Vern Paxson, "Empirically-Derived Analytic Models of Wide-Area TCP Connections", *IEEE/ACM Transactions on Networking*, Vol. 2, No. 4, 1994, pp 316-336

Pederson 90 — Shane P. Pederson and Mark E. Johnson, "Estimating Model Discrepancy", *Technometrics*, Vol. 32, No. 3, 1990, pp. 305-314

Pitkow 95a — James Pitkow, "GVU's NSFNET Backbone Statistics", 1995, Available at http://www.cc.gatech.edu/gvu/stats/NSF/merit.html, last viewed 27 May 2004

Pitkow 98 — James E. Pitkow, "Summary of WWW Characterizations", *Computer Networks and ISDN Systems*, Vol. 30, 1998, pp. 551-558

Polyteam 04 — "Web PolyGraph", 2004, Available at http://www.web-polygraph.org/, last viewed 5 February 2004

Qiu 94 — Liwen Qiu, "Frequency Distributions of Hypertext Path Patterns: A Pragmatic Approach", *Information Processing and Management*, Vol. 30, No. 1, 1994, pp. 131-140

Radford 99 — Personal communication with Adam Radford, Communications Unit, UNSW, 7 July 1999.

Richardson 95 — Kathy J. Richardson, "Request Load and Performance Characterization of a WWW-Proxy", Unpublished document from Compaq Network Systems Laboratory, 1995, Available at http://research.compaq.com/nsl/people/kjr/proxycache1.ps, last viewed 2 September 03

Reyes-Lecuona 99 — A. Reyes-Lecuona, E. Gonzalez-Parada, E. Casilari, J. C. Casasola and A. Diaz-Estrella, "A Page-Oriented WWW Traffic Model for Wireless Systems Simulations", *Proc. 16th International Teletraffic Congress*, Edinburgh, 1999

Rousskov 01 — A. Rousskov, M. Weaver, and D. Wessels, The Fourth Cache-off. Raw data and independent analysis at http://www.measurement-factory.com/results/, last viewed 27 May 2004

Sedayao 94 — Jeff Sedayao, "Mosaic Will Kill My Network! - Studying Network Traffic Patterns of Mosaic Use", *Proc. The Second International WWW Conference - Mosaic and the Web*, Chicago, 1994

Smith 01 F. Donelson Smith, Félix Hernández Campos, Kevin Jeffay and David Ott, “What TCP/IP Protocol Headers Can Tell Us About the Web”, *Proc. ACM SIGMETRICS*, Cambridge, 2001, pp. 245-256

Squid 97 “Squid Internet Object Cache”, 1997, Available at `http://www.squid-cache.org/`, last viewed 27 May 2004

Stevens 94 W. Richard Stevens, *TCP/IP Illustrated Volume 1, The Protocols*, Addison-Wesley, Reading, 1994

Stevens 96 W. Richard Stevens, *TCP/IP Illustrated Volume 3, TCP for Transactions, HTTP, NNTP, and the UNIX Domain Protocols*, Addison-Wesley, Reading, 1996

Stoev 04 Stilian Stoev and Murad S. Taqqu, “Simulation Methods for Linear Fractional Stable Motion and FARIMA Using the Fast Fourier Transform”, *Fractals [Complex Geometry, Patterns and Scaling in Nature and Society]*, Vol. 12, No. 1, 2004, pp. 95-121

Thompson 97 Kevin Thompson, Gregory J. Miller and Rick Wilder, “Wide-Area Internet Traffic Patterns and Characteristics”, *IEEE Network*, Vol. 11, No. 6, 1997, pp. 10-23

de Vere 96 Lorraine de Vere, John Judge, Gary Anido and H.W. Peter Beadle, “Internet Service Over ATM: The Effect on WWW Performance”, *Proc. 7th International Network Planning Symposium*, Sydney, 1996, pp. 227-232

W3C 99a “Web Characterization Terminology & Definitions Sheet”, W3C Working Draft 24-May-1999, Eds. Brian Lavoie and Henrik Frystyk Nielsen, 1999, Available at `http://www.w3.org/1999/05/WCA-terms/01`, `Last` viewed 2 February 2004.

W3C 99b “HTML 4.01 Specification”, W3C Recommendation 24 December 1999, Eds. Dave Raggett, Arnaud Le Hors and Ian Jacobs, 1999, Available at `http://www.w3.org/TR/html401/`, last viewed 27 May 2004

W3C 02 "XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition)", W3C Recommendation 26 January 2000, revised 1 August 2002, Available at `http://www.w3.org/TR/xhtml1/`, last viewed 27 May 2004

W3C 03 HyperText Markup Language Activity Statement, Available at `http://www.w3.org/MarkUp/Activity`, last viewed 27 May 2004

Wessels 99 Duane Wessels, Alex Rousskov and Glenn Chisholm, "The First IRCache Web Cache Bake-off", *Proc. Fourth Web Caching Workshop*, San Diego, 1999, Raw data and independent analysis available at `http://cacheoff.ircache.net/N01/`, last viewed 27 May 2004

Wessels 01 Duane Wessels, *Web Caching*, O'Reilly, Sebastopol, June 2001

Wyshak 74 G. Wyshak, "A Program for Estimating the Parameters of the Truncated Negative Binomial Distribution", Algorithm AS68, *Applied Statistics*, Vol. 23, No. 1, 1974, pp. 87-91

Appendix A. WWW Traffic Measurement at SNRC

This appendix describes the Web traffic trace collected from SNRC at UOW. In this dissertation this trace is referred to as the "SNRC Trace".

A standard CERN proxy server [Luotonen 96] was instrumented to monitor the use of the Web of a group of postgraduate students in SNRC at UOW. The laboratory is divided into cubicles with each student having their own desk and computer. The proxy was modified to log details of each HTTP transaction by host name and by keeping track of the assignment of cubicles in the laboratory the trace was divided into individual users.

Students were asked to participate in the project by configuring their Web clients to use the instrumented proxy. Participation was voluntary and some students chose not to. The students who did participate were promised that the details of their browsing behaviour would remain confidential. The trace recorded traffic against fictitious host names and the mapping between actual and fictitious names was not revealed. As a further measure, due to the small number of users in the lab and the ease of guessing which fictitious name mapped to which user, only general statistics derived from the trace were ever publically revealed. After it was apparent that their detailed browsing behaviour was not under scrutiny most of the students in the laboratory participated freely. This candidates Web traffic was recorded also (it was a useful method of checking that the proxy was working on a day to day basis) but that traffic was discarded from all analysis.

The CERN Web proxy server was instrumented to record the following information in a log file for each Web transaction:

- the name of the Web client machine mapped to a fictitious name via a lookup table
- the time in seconds and microseconds that the first byte of the request was received from the client

- the time in seconds and microseconds that the first byte of the response was sent back to the client
- the time in seconds and microseconds that the last byte of response was sent back to the client
- the size of the response in bytes (excluding size of HTTP headers)
- the request method (GET, PUT or POST)
- the full URL requested
- the HTTP protocol version

The instrumented CERN proxy did not act as a cache and did not noticeably affect the response time for users. For a period during 1995 and 1996 the proxy forwarded requests to the UOW campus proxy cache but after poor performance of the campus cache forwarding was discontinued.

The Web traffic logging continued for just under three years from 26 October 1994 until 1 October 1997. Over this period 642654 requests from 35 separate hosts were recorded. Because the purpose of the trace was to track the Web traffic for individual users the traffic from a number of host machines was discarded. This consisted of traffic generated from machines which were only used temporarily in the laboratory or were shared on an ad-hoc basis by a number of users. Also the traffic from the authors host machine and the traffic from a small number of machines outside the laboratory, such as Web robots, was discarded. Remaining in the trace were the details of 536585 requests from 19 users over a 1071 day period. A graph showing the activity of each user is shown in Figure A.1. Each ‘x’ in Figure A.1 signifies that the user generated at least one Web request on the given date. The users are identified by names chosen at random from the TV shows the “Brady Bunch”[1] and “Get Smart”[2].

1. Character names from the TV show the Brady Bunch, “Cindy”, “Jan”, “Marsha”, “Bobby”, “Peter”, “Greg”, “Mike”, “Carol”, “Alice”, “Sam” and “Tiger”

2. Character names from the TV show Get Smart, “Maxwell”, “Agent99”, “Chief”, “Hymie”, “Larabee”, “Agent13”, “Shtarker” and “Siegfried”

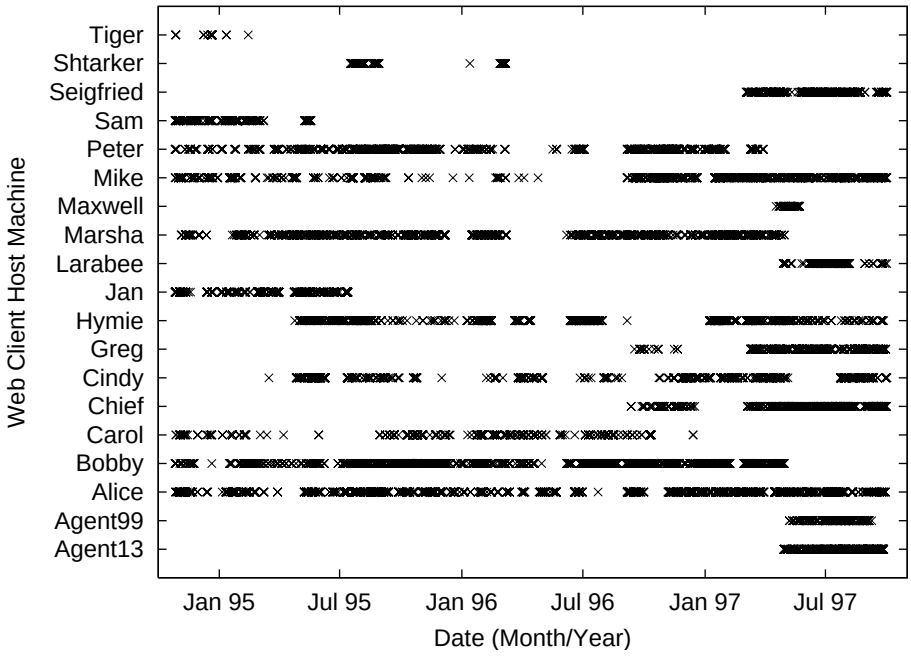


Figure A.1 Activity of Each User Over Three Years in the SNRC Trace

It was a continuing problem to ensure the Web client of each participating user in the laboratory kept pointed at the instrumented Web proxy. Periodically Web clients would not use the proxy. The three main reasons for failure to use the proxy for a period of time were:

1. losing the proxy server configuration after upgrading or changing Web client software

2. users turning off use of the Web proxy to regain use of the client after failure of the UOW campus Web proxy cache

3. frustrated users turning off use of the Web proxy to try to achieve better response time performance after poor performance of the UOW campus Web proxy cache

It was after repeated instances of 2 and 3 above the instrumented proxy discontinued use of the UOW campus Web proxy cache in early 1997.

Appendix B. WWW Traffic Measurement at Berkeley

This appendix describes the Web traffic trace sourced from Berkeley. In this dissertation this trace is referred to as the "Berkeley Trace".

B.1 Source and Initial Analysis

Steve Gribble utilised packet snooping software to trace the HTTP traffic on a portion of the campus network at the University of California Berkeley (UCB) [Gribble 97a]. The traffic monitored included the "Home IP" service which provides dial up IP connectivity to UCB network users away from the campus. Processing of the data by Gribble reconstructed HTTP request response transactions. HTTP traffic was identified by logging only traffic destined for TCP port 80. More information on the collection method and information on how to obtain the traffic traces is available over the Web [Gribble 97a]. The trace details 9.2 million requests over an 18 day period commencing 1 November 1996.

Initial analysis of the Berkeley Trace revealed an unusually high percentage of one byte objects (nearly 15% of the trace). After communication with Steve Gribble it was concluded that the high percentage of one byte objects was an artifact of the collection process and actually represented zero byte objects. In addition, it was found that the trace included a small number of Web objects of an impossibly large size. These objects were filtered out of the trace by checking the recorded time to transmit each object over 10^7 bytes in size and discarding the object from the data set if the sustained data rate was over 5×10^6 bits per second. Thirteen large objects were discarded. Both alterations to the trace were made before the rest of the analysis presented in this dissertation was performed.

The Berkeley trace used in this dissertation consists of 9244716 requests from 8377 Web client hosts over 18 days commencing 1 November 1995.

Appendix C. WWW Traffic Measurement at Digital

This appendix describes the Web traffic trace sourced from Digital (now Hewlett-Packard Company). In this dissertation this trace is referred to as the "Digital Trace".

C.1 Source and Initial Analysis

T. Kroeger *et. al.* [Kroeger 99] collected traces of Web traffic at Digital Equipment Corporation (DEC) using two instrumented squid [Squid 97] proxy servers. For the duration of the collection period the servers were not used as caches. Information on the collection of the traces and the trace data is available via anonymous FTP [Kroeger 99]. The traces detail 24.5 million requests over a 25 day period commencing 29 August 1996.

In the initial analysis of the Digital trace there was a clear outlier of traffic for a single 24 hour period. On closer examination it was found that one client had made an unusually large number of frequent requests to the same URL at a mean rate of nearly 7 requests per second. The requests were generated in an automated fashion that was clearly unrepresentative of the rest of the trace and not repeated during any other period. After communication with Jeff Mogul at Digital the requests were filtered out of the trace. They are identifiable by client ID# 3068 making request with URL ID# 5241814. In total 591429 requests starting at time stamp 842145989 were discarded.

The Digital trace used in this dissertation consists of 24067753 requests from 17354 Web client hosts over 25 days commencing 29 August 1996.

Appendix D. First Web Traffic Measurement at UNSW

This appendix describes the collection and filtering of the first Web traffic trace obtained from UNSW between 1 January and 30 June 1999. Throughout this dissertation this trace is referred to as the "UNSW1" trace.

D.1 Collection

UNSW is a Sydney based university with approximately 30 thousand students. The University generates a substantial amount of Internet traffic and uses a transparent Web cache to reduce the Web related component. The cache consists of a pair of model CE2050 Cisco cache engines running version 1.7.5 of the Cache Engine software. A Cisco router re-directs all HTTP requests generated on the campus network to the cache engines without the need for configuration of client software. A white paper explaining the Cisco Cache Engine and related cache-router protocol "WCCP" can be found at the Cisco Web site [Cisco 99a].

Each cache is configured to log details concerning user traffic to a log file on local disk. The UNSW Cache Engines are configured to start a new log file each hour. A periodic process running elsewhere in the UNSW network downloads the completed log files from each cache every hour automatically to more permanent disk storage using FTP. The log file is then deleted on the cache freeing up disk space for new log information. The time on each cache is synchronised with hourly calls to a common NTP server.

The log file format is documented in [Cisco 99b] and is compatible with the popular Squid log file format. Each HTTP transaction is logged with ten space separated fields of information detailing the transaction. Before UNSW would release the files a number of the fields were encrypted using a MD5 hash. The client IP number, the server IP number and the URL string were all encrypted. A "salt" only known to UNSW was added to each field before hashing to ensure reverse engineering of IP numbers was not possible. To minimise storage space requirements the hashes for the client IP numbers were stored in a database file and assigned a sequence

number. The first IP client in the log was “1”, the second “2” and so on. The database file ensured consistency over the separate log files. Each file was then copied over to the computer system at UOW where the following fields from each HTTP transaction were extracted:

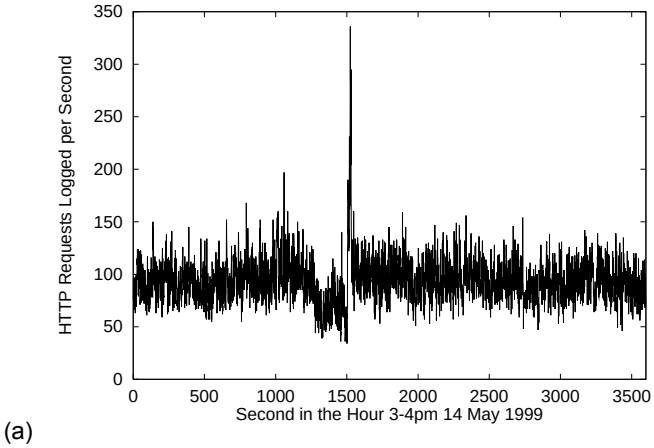
- time - timestamp in seconds and microseconds
- size - size of the returned object in bytes
- status - HTTP response status code
- client - an integer identifying each unique client IP number

The rest of the data was archived for future use.

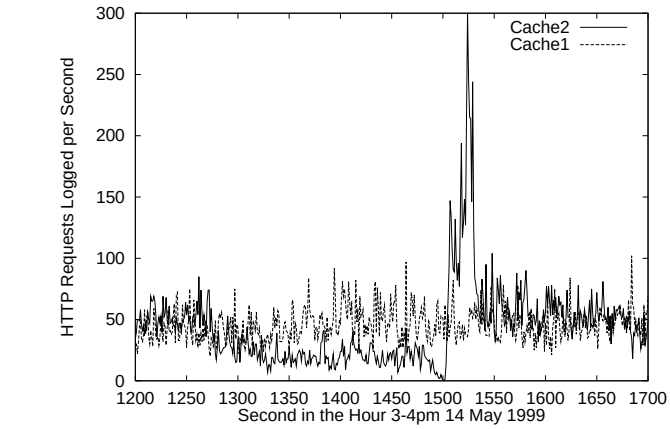
D.2 Trace Processing

Initial processing of the trace files indicated a number of problems with the trace collection method. UNSW were collecting the log files for there own internal use which did not include detailed statistical analysis. When the collection process failed it sometimes took a while before anybody noticed and corrected the situation. Consequently a number of the hourly log files had no traffic recorded at all. The other problem concerned introduced artefacts in the traces. These artefacts were introduced by the trace collection process rather than user browsing behaviour. Three separate artefacts were identified each due to a different problem with the caches or trace collection process; first, a cache engine “seizes” up for a period of time, second, a cache engine failing to log user requests for a period of time and last, log files being duplicated from one hour to the next.

Figure D.1 shows the hour of 3:00pm to 4:00pm on 14 May 1999 which is an example of an hour in which one of the cache engines (in this case Cache2) had problems. The problem manifests itself as one cache engine logging a reduced request rate for a period of a couple of minutes and then logging a higher request rate seemingly in an effort to “catch up”. Figure D.1(b) shows a period of 500 seconds surrounding the artefact with the HTTP request rate logged by each cache. It can be seen that Cache1 continues with a consistent request rate but Cache2 logs a reduced request



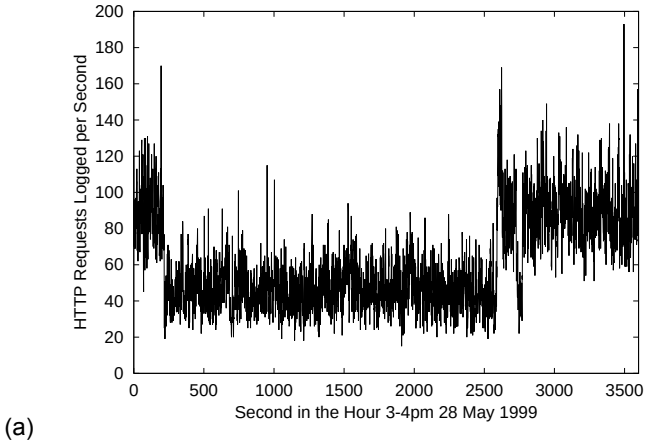
(a)



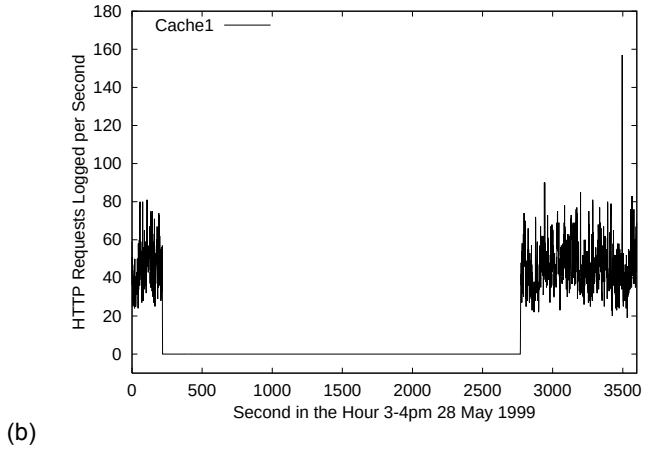
(b)

Figure D.1 Requests per Second Logged for 3-4pm 14 May 1999;
(a) Overall Traffic
(b) Traffic Logged by Each Cache Engine Over 500 Second Period Surrounding Artefact

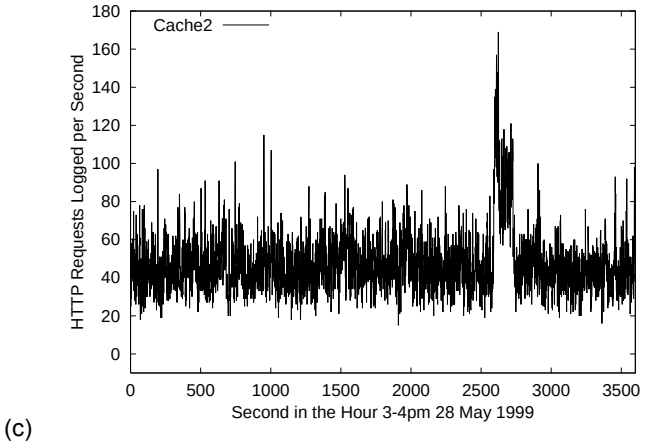
rate for about 250 seconds from 1250 seconds into the hour and then logs a short burst of requests at a much higher rate. The fact that Cache1 shows a steady HTTP request rate indicates that the artefact is due to the trace collection process rather than user browsing behaviour. After communication with UNSW [Radford 99] it was concluded that the problem was due to faulty Beta software being run on the cache engines between 5:00pm 6 May 1999 and 3:00pm 21 May 1999. The software had caused noticeable problems with the operation of the cache engines and the upgrade on the 21 May 1999 was specifically to move to a more stable version. In the hours examined outside this period this particular artefact was not evident. In contrast it appeared in nearly every hour examined during this time period.



(a)



(b)



(c)

Figure D.2 HTTP Requests per Second Logged for the Hour 3-4pm 28 May 1999;
(a) Overall Traffic
(b) Traffic Logged by Cache1
(c) Traffic Logged by Cache2

Figure D.2 shows an example of an hour where one of the cache engines failed to log HTTP transactions for a period of time. Even though the router re-directing

HTTP request traffic to the cache engines is supposed to perform load balancing this does not appear to be reflected properly in the logs. The failure of an engine appears to introduce transients into the recorded logs which do not appear indicative of user traffic behaviour.

The third problem was that a small number of log files were repeats of previous hours. For some reason the cache engine would fail to delete the old log file and create a new one. The automated process copying log files from the cache would simply copy the same file as the previous hour. This resulted in a double count for HTTP requests in one hour and a gap in the next with no HTTP requests recorded.

A filtering algorithm was devised to discard affected hours from the trace. Removing duplicate log files and discarding all the hours between 5:00pm 6 May and 3:00pm 21 May was not hard. The difficulty was identifying hours in which one or more of the cache engines had failed for a period of time. The generation of HTTP requests is a random process with more requests arriving during some seconds than others. The problem was in determining the number of seconds in a logged hour in which no HTTP requests were received that was a reasonable number versus that which indicated a problem with a cache engine or the trace collection process.

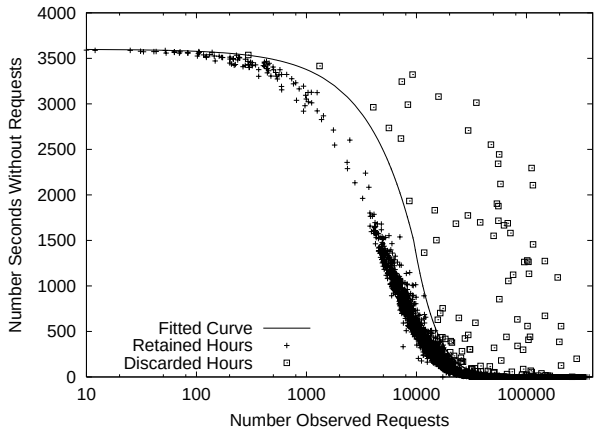
A temporary data set was created by discarding the hours in the affected period in May and discarding hours affected by duplicate log files. A scattergram plot was made of the remaining hours by plotting the number of HTTP requests recorded against the number of seconds in which HTTP requests were not logged. In total 7890 hours were plotted (3945 traced hours for each cache engine).

A curve was fitted to the outer right hand edge of the plot to delineate the bulk of traced hours from those in which it was suspected that a cache failure may have occurred. The equation of the fitted curve is given in Equation D.1 where x is the number of HTTP transactions logged in the hour an y is the maximum acceptable number of seconds in the hour with logged traffic. Figure D.3 shows the fitted curve and a scattergram of hours discarded

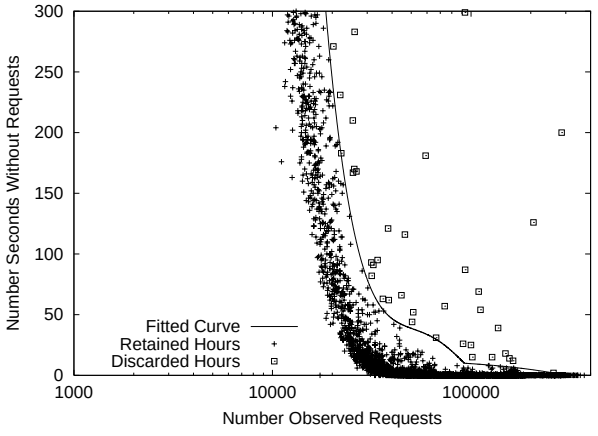
$$y = \begin{cases} -0.222859x + 3600, & 0 \le x \le 9423 \\ p_1x^4 + p_2x^3 + p_3x^2 + p_4x + p_5, & 9423 < x < 92503 \\ \dfrac{-1}{20749.3}x + 14.4581, & 92503 \le x < 300000 \\ 0, & 300000 \le x \end{cases} \quad \textbf{(Eqn D.1)}$$

where:

$$p_1 = 0.0137813, \quad p_3 = 10.2826, \quad p_5 = 212.541$$
$$p_2 = -0.614963, \quad p_4 = -76.3631,$$



(a)



(b)

Figure D.3 Curve Fitted to Scattergram Plot of HTTP Requests Versus Number of Seconds Without Observed Traffic
(a) All the Hours
(b) Close-up of the "Knee" of the Plot

Table D.1 gives a summary of the hours recorded in the trace. In total 762 hours were discarded from 4343 hours originally in the trace leaving 3581 hours for further analysis.

Table D.1 Summary of Initial Analysis of Logged Hours in UNSW 1 Trace

Hours in Original Trace Period	4343
Hours with no Data Recorded	271
Hours in Bad Period in May	367
Hours Affected by Duplicate Log Files	31
Suspect Hours with Too Many Seconds Without Traffic	93
Hours Left for Further Analysis	3581

Appendix E. Second Web Traffic Measurement at UNSW

This appendix describes the collection and filtering of the second Web traffic trace obtained from UNSW between 17 October 2001 and 10 January 2002. Throughout this dissertation this trace is referred to as the "UNSW2" trace.

E.1 Collection

The second trace sourced from UNSW was from the same cache engine farm as described in Appendix D. Since the first trace a third cache engine had been added and the operating system software upgraded a number of times (to a version that UNSW did not specify). The staff at UNSW collected log files from the cache engines on an hourly basis as described in Appendix D. The format of the log files was also the same.

The traces made available by UNSW consisted of just four fields:

- time - timestamp in seconds and microseconds
- size - size of the returned object in bytes
- status - HTTP response status code
- client - an integer identifying each unique client IP number

This was in contrast to the previous trace where all the log files (with some fields encrypted) had been made available.

The client ID was encrypted using a MD5 hash and a "salt" only known to UNSW to ensure reverse engineering of IP numbers was not possible. The field extraction and encrypting was done on UNSW computers and compressed files were loaded onto a notebook computer hard drive. UNSW processed data was collected from each of the three cache engines between 10:00pm 17 October 2001 and 11:00pm 10 January 2002.

E.2 Trace Processing

Initial processing of the trace files showed significant corruption of the trace files. UNSW did not regularly look at the traces from the cache engines and the process to collect log files was entirely automatic. In most hours at least one or more of the cache engines failed to log all requests. This resulted in trace artefacts similar to those seen in the previous trace file detailed in Appendix D and shown in Figure D.2. A more minor problem was a small number of repeated log files where the log file for one hour was duplicated for the next hour also.

The duplicate log files were removed from the data. Hours in which one of the cache engines had failed to record request rate data for some period of time were also removed. Out of the 2041 hours in the collection time period only 247 hours had data logged over the entire hour. In these 247 hours no other trace artefacts were apparent and the data looked similar to what had been collected just under three years previously from the same source.

Table E.1 gives a summary of the hours recorded in the trace. In total 1794 hours were discarded from 2041 hours originally in the trace leaving 247 hours for further analysis.

Table E.1 Summary of Initial Analysis of Logged Hours in UNSW 2Trace

Hours in Original Trace Period	2041
Hours with no Data Recorded	59
Hours Affected by Duplicate Log Files	12
Suspect Hours with Too Many Seconds Without Traffic	1735
Hours Left for Further Analysis	247

Appendix F. Description of Some Less Well Known Probability Distributions

Some of the probability distributions used in this dissertation are not widely known. This appendix details the definition and the method of parameter estimation for each of these distributions.

F.1 Zero Truncated Negative Binomial Distribution

The zero truncated negative binomial distribution is the negative binomial distribution without observation of zero values. The distribution is described in [Johnson 92 pp. 225-7]. The PMF of the distribution is given in Equation F.1 and has two parameters, k and Q, where $Q = 1 + P$. The CDF was generated numerically by summing values obtained from the PMF over the required range. Unless otherwise mentioned the zero truncated negative binomial distribution was fit to observed data using ML estimators which is the method recommended by [Johnson 92]. The ML estimators $\hat{k}$ and $\hat{Q}$ are obtained by solving Equation F.2 where f_x is the observed frequency of the observation x. An algorithm for doing so is given by [Wyshak 74].

$$Pr[X = x] = (1 - Q^{-k})\binom{k + x - 1}{k - 1}\left(\frac{P}{Q}\right)^x\left(1 - \frac{P}{Q}\right)^k, \qquad x = 1, 2, 3, \ldots \quad \textbf{(Eqn F.1)}$$

$$\begin{aligned} 0 &= \sum_{x \geq 1} f_x\left(\frac{1}{\hat{k}} + \frac{1}{\hat{k} + 1} + \ldots + \frac{1}{\hat{k} + x - 1} - \frac{\ln \hat{Q}}{1 - \hat{Q}^{-\hat{k}}}\right), \\ 0 &= \sum_{x \geq 1} f_x\left(\frac{x}{1 - \hat{Q}} - \frac{\hat{k}}{1 - \hat{Q}^{-\hat{k}}}\right), \end{aligned} \quad \textbf{(Eqn F.2)}$$

F.2 Shifted Negative Binomial Distribution

The shifted negative binomial distribution used in this dissertation is the negative binomial distribution shifted one value to the right so the minimum observation is

one rather than zero. The text of the dissertation sometimes refers to [Qiu 94] who specifies a PMF with parameters V and p. The PMF used in this dissertation is given in Equation F.3 [Johnson 92] with two parameters k and Q. A translation into the form used by [Qiu 94] is $p = 1/q$ and $V = k$. Unless otherwise stated the distribution was fit to observed data using the MOM with estimators for $\tilde{k}$ and $\tilde{Q}$ given by Equation F.4 where $\bar{x}$ is the sample mean and s^2 is the sample variance [Johnson 92].

$$Pr[X = x] = \binom{k+x-2}{k-1}\left(1-\frac{P}{Q}\right)^k\left(\frac{P}{Q}\right)^{x-1}, \qquad x = 1, 2, \ldots \qquad \textbf{(Eqn F.3)}$$

$$\tilde{Q} = \frac{s^2}{\bar{x}-1}$$
$$\tilde{k} = \frac{(\bar{x}-1)^2}{s^2-(\bar{x}-1)} \qquad \textbf{(Eqn F.4)}$$

F.3 Zero Truncated Poisson Distribution

The zero truncated Poisson distribution is the Poisson distribution where zero values are not included as observations. The distribution used in this dissertation and parameter estimation techniques are described in [Johnson 92 pp. 181-4]. The PMF is shown in Equation F.5 and has a single parameter, θ. Unless otherwise stated the distribution was fit to observed data using ML estimators obtained by numerically solving Equation F.6 for estimate $\hat{\theta}$ where $\bar{x}$ is the sample mean.

$$Pr[X = x] = \frac{\theta^x}{(e^\theta - 1)x!}, \qquad x = 1, 2, \ldots \qquad \textbf{(Eqn F.5)}$$

$$\bar{x} = \frac{\hat{\theta}}{1-e^{-\hat{\theta}}} \qquad \textbf{(Eqn F.6)}$$

F.4 The Pólya-Aeppli Distribution

The Pólya-Aeppli distribution and parameter estimation techniques used in this dissertation are detailed in [Johnson 92 pp. 378-382]. The distribution is a Poisson stopped sum of the Geometric distribution. The geometric distribution has PMF

shown in Equation 4.1 and is referred by Johnson et. al. as the shifted geometric distribution. The PMF of the Pólya-Aeppli distribution is shown in Equation F.7 and has two parameters, θ and p. Unless otherwise stated the distribution was fit to observed data using the MOM with estimators $\tilde{\theta}$ and $\tilde{p}$ given by Equation F.8 where $\bar{x}$ is the sample mean and s^2 is sample variance.

$$Pr[X = x] = \begin{cases} e^{-\theta} & x = 0 \\ e^{-\theta} p^x \sum_{j=1}^{x} \binom{x-1}{j-1} \frac{[\theta(1-p)]^j}{j!} & x = 1, 2, \ldots \end{cases} \quad \textbf{(Eqn F.7)}$$

$$\tilde{\theta} = \frac{2\bar{x}^2}{s^2 + \bar{x}}$$

$$\tilde{p} = \frac{s^2 - \bar{x}}{s^2 + \bar{x}} \quad \textbf{(Eqn F.8)}$$

Appendix G. Poisson HTTP Session Arrivals

In Section 3.2 it is shown that the arrival process of HTTP sessions in an aggregate stream of Web traffic is Poisson. Figure 3.2 shows three properties of the arrival process for the sample hour *D1*; Poisson distributed session arrivals per second, exponential session inter-arrival times and negligible autocorrelation in session arrivals. The following figures show these characteristics for each of the sample hours listed in Table 2.2 with the exception of the hours sampled from the UNSW2 trace. More discussion on the correlograms can be found in Section 3.2.

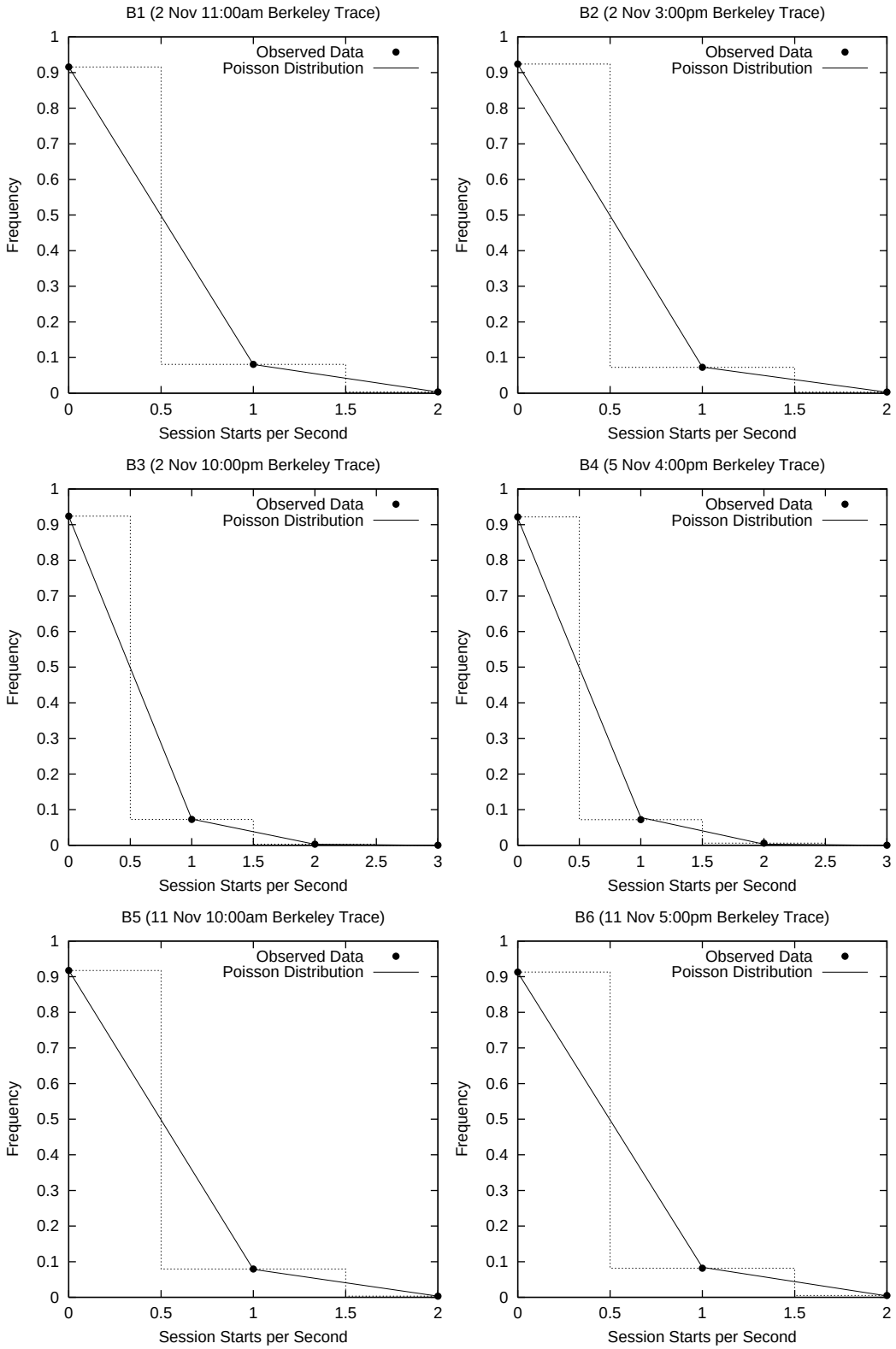


Figure G.1 (Part 1) Histogram of the Number of HTTP Session Arrivals per Second Compared to Poisson Distribution for Hours Listed in Table 2.2

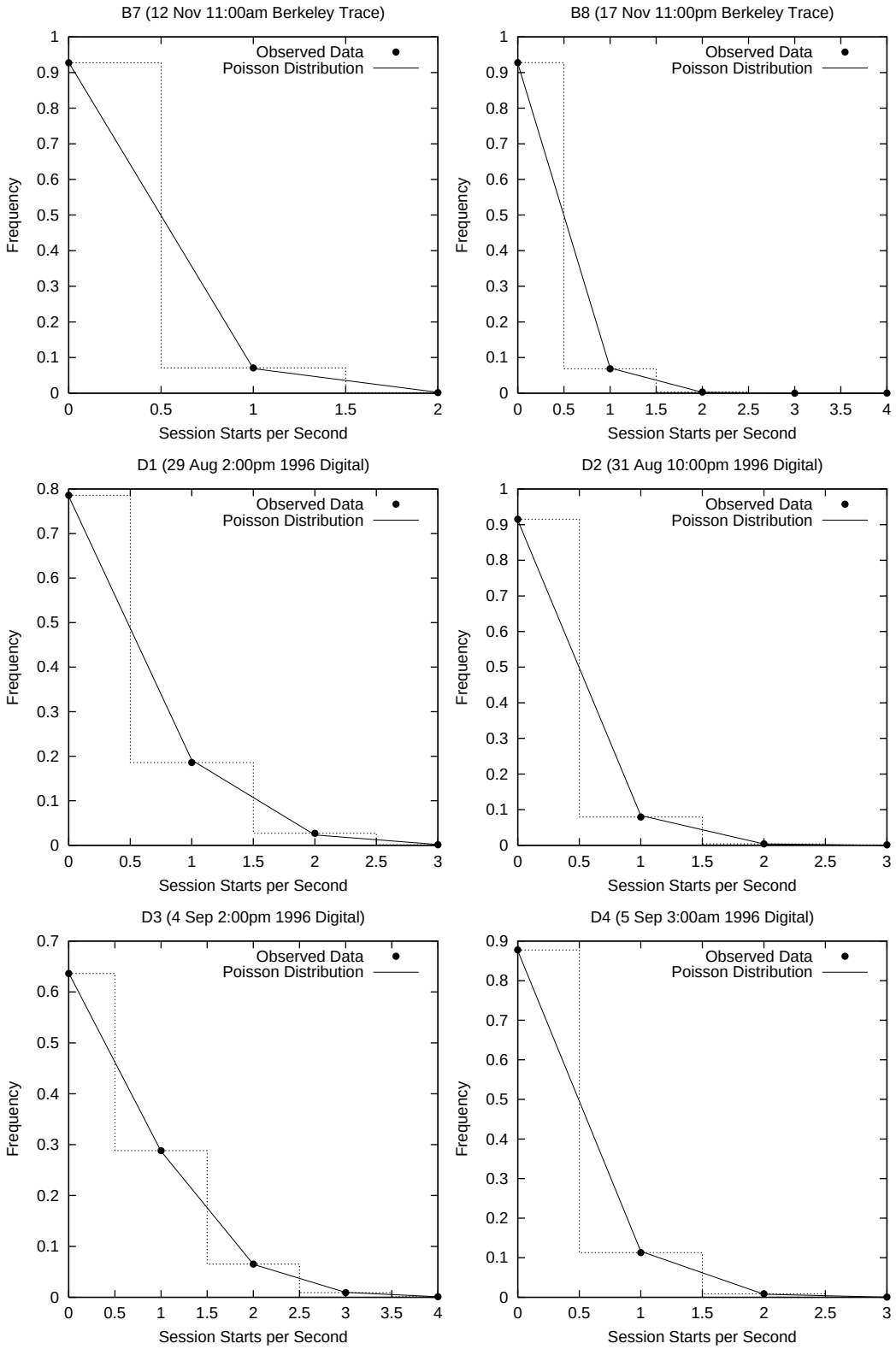


Figure G.1 (Part 2) Histogram of the Number of HTTP Session Arrivals per Second Compared to Poisson Distribution for Hours Listed in Table 2.2

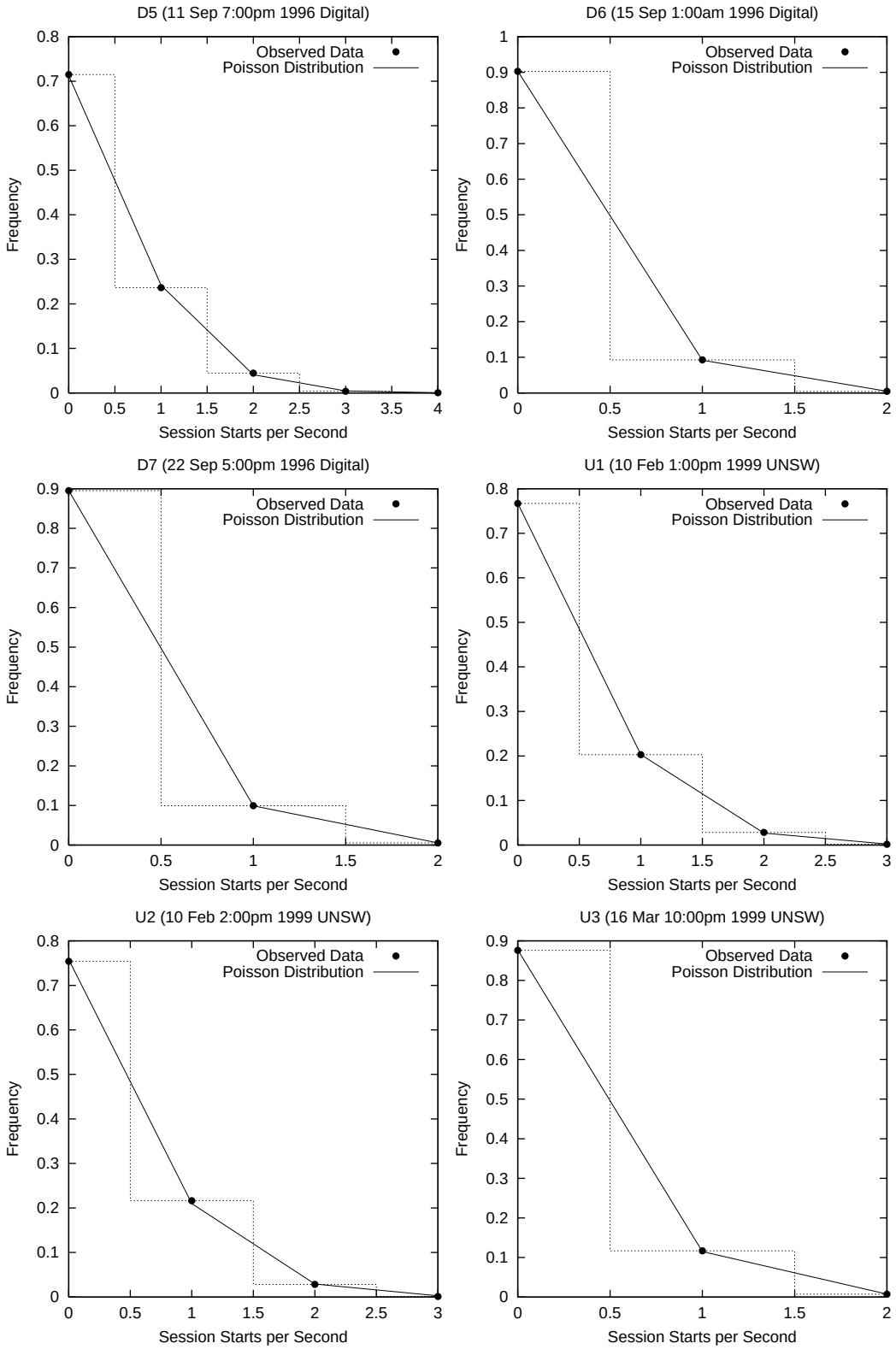


Figure G.1 (Part 3) Histogram of the Number of HTTP Session Arrivals per Second Compared to Poisson Distribution for Hours Listed in Table 2.2

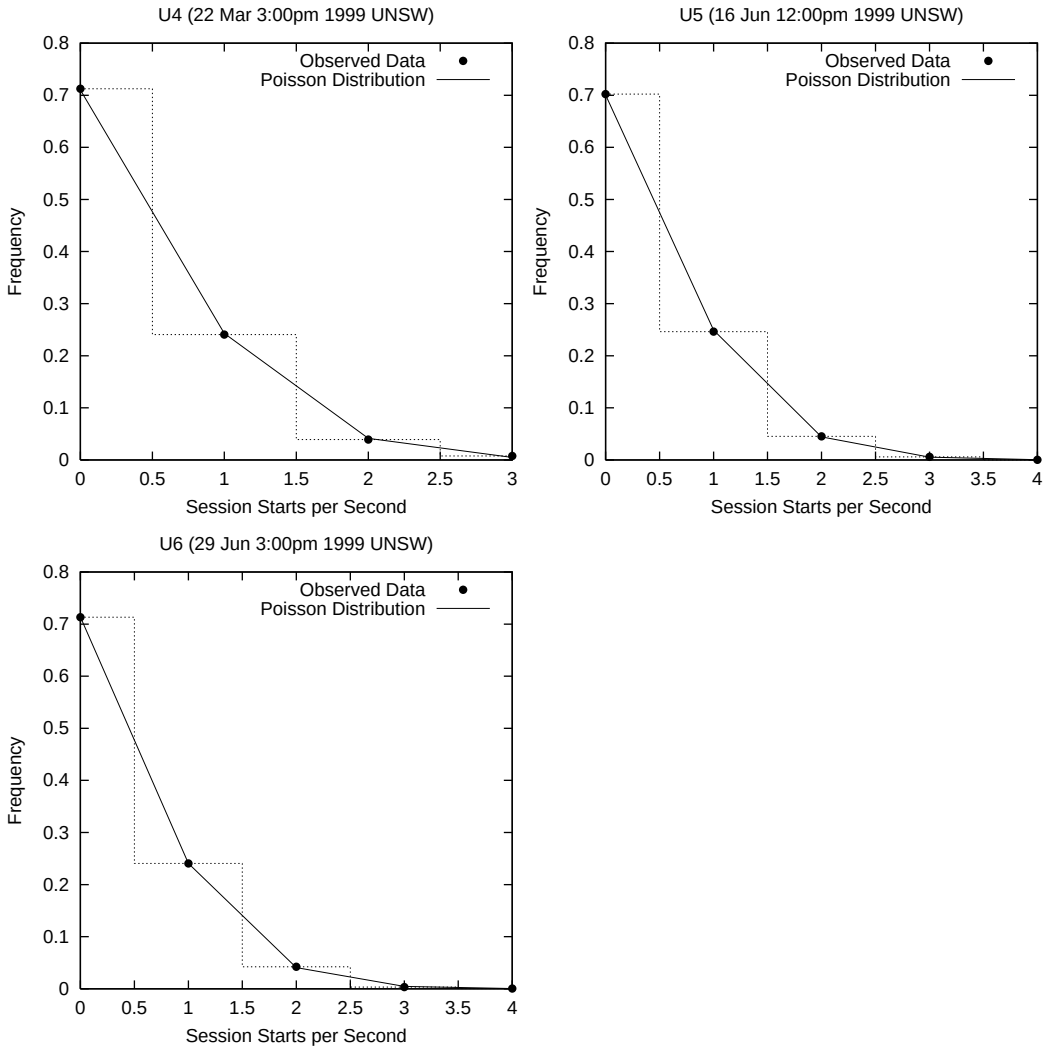


Figure G.1 (Part 4) Histogram of the Number of HTTP Session Arrivals per Second Compared to Poisson Distribution for Hours Listed in Table 2.2

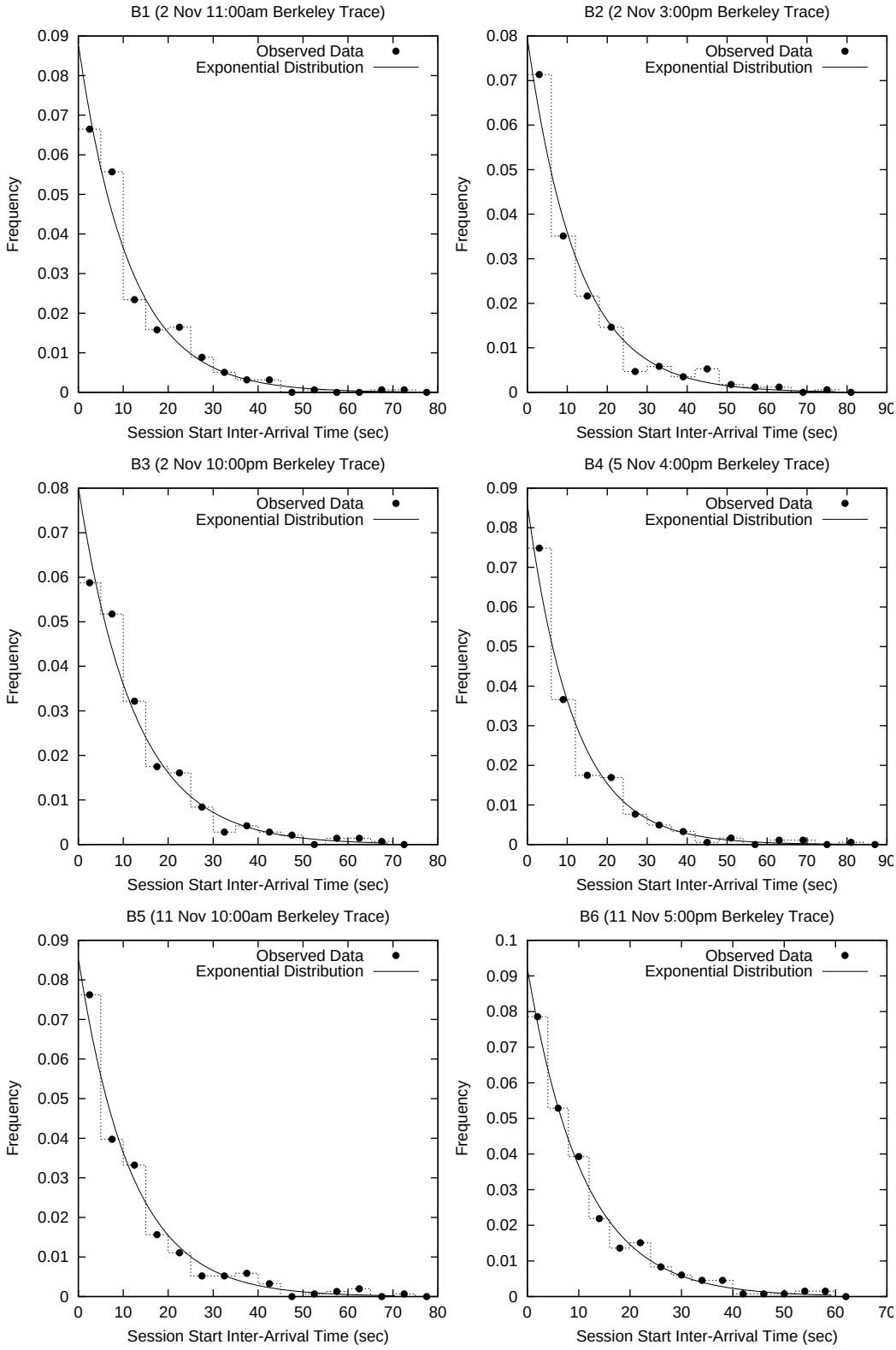


Figure G.2 (Part 1) Histogram of HTTP Session Interarrival Time Compared to Exponential Distribution for Hours Listed in Table 2.2

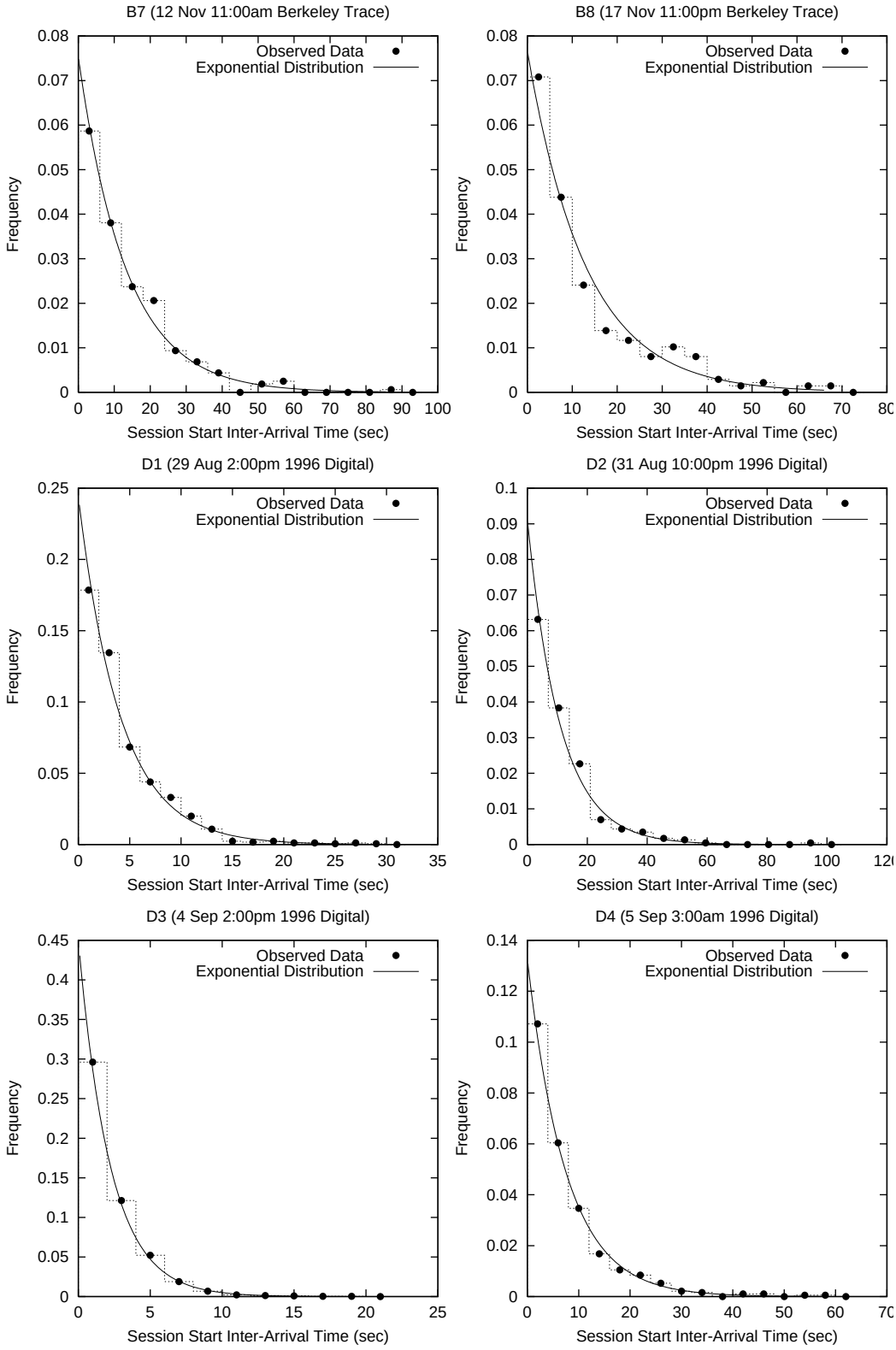


Figure G.2 (Part 2) Histogram of HTTP Session Interarrival Time Compared to Exponential Distribution for Hours Listed in Table 2.2

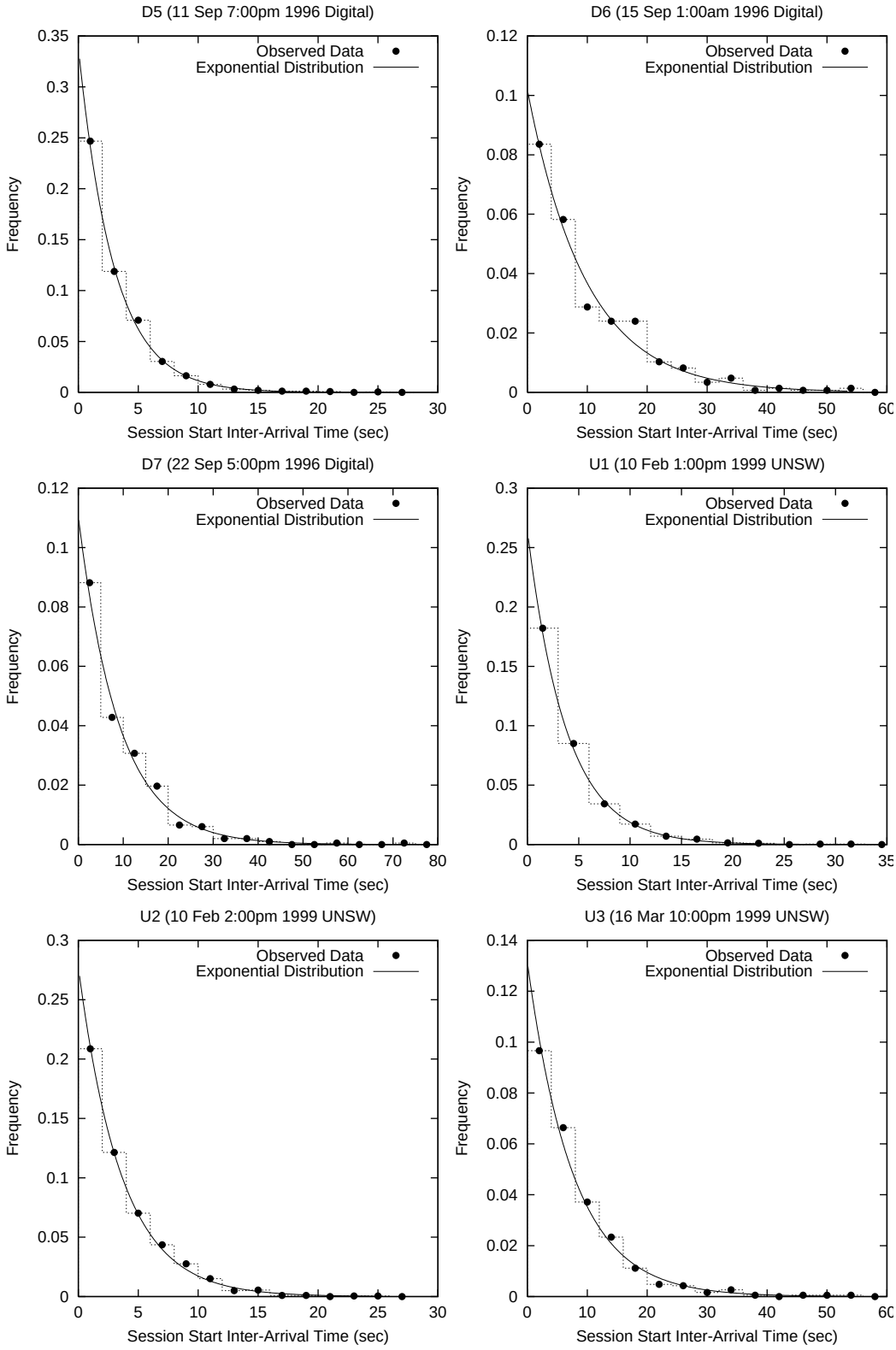


Figure G.2 (Part 3) Histogram of HTTP Session Interarrival Time Compared to Exponential Distribution for Hours Listed in Table 2.2

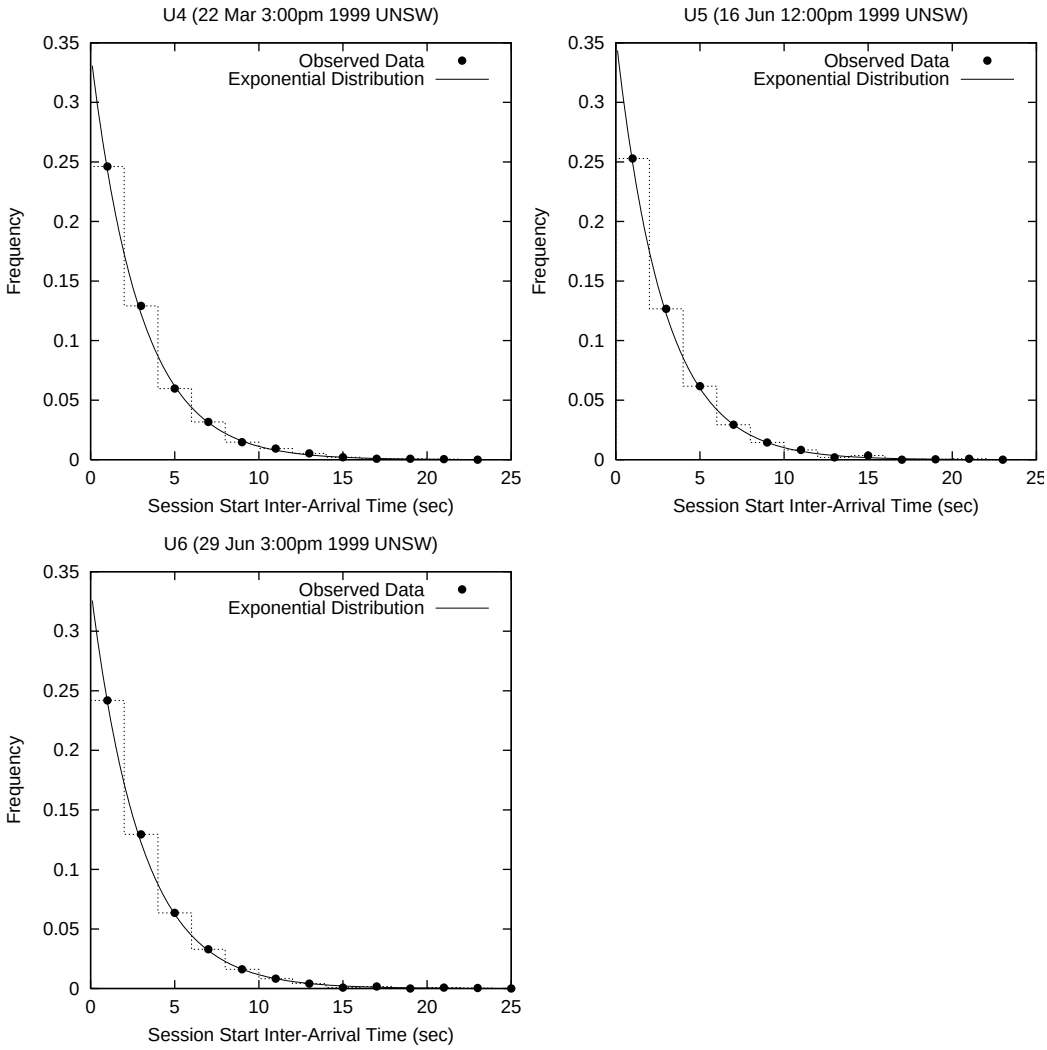


Figure G.2 (Part 4) Histogram of HTTP Session Interarrival Time Compared to Exponential Distribution for Hours Listed in Table 2.2

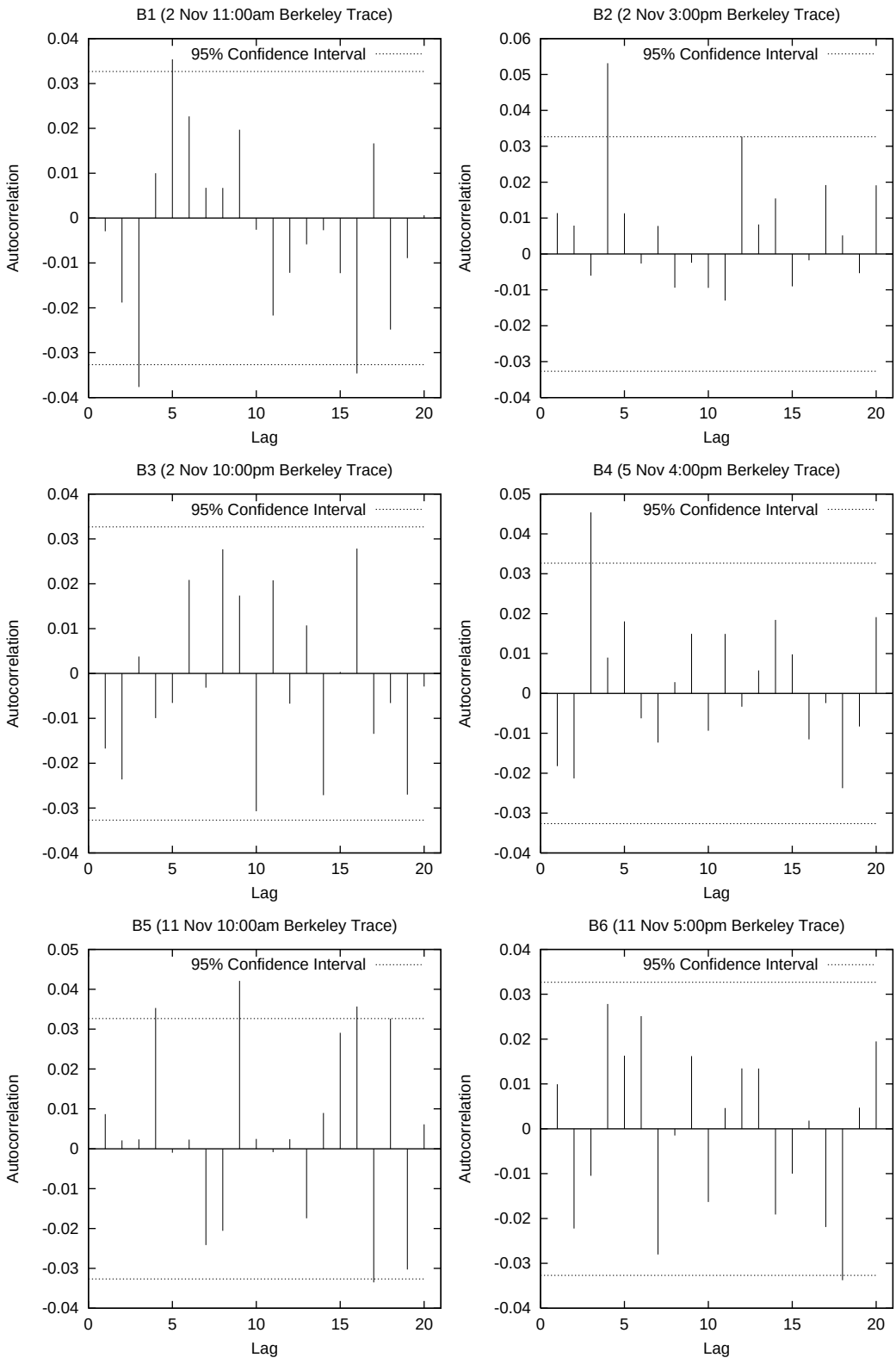


Figure G.3 (Part 1) Correlogram of Number of HTTP Session Arrivals per Second for Trace Hours Listed in Table 2.2

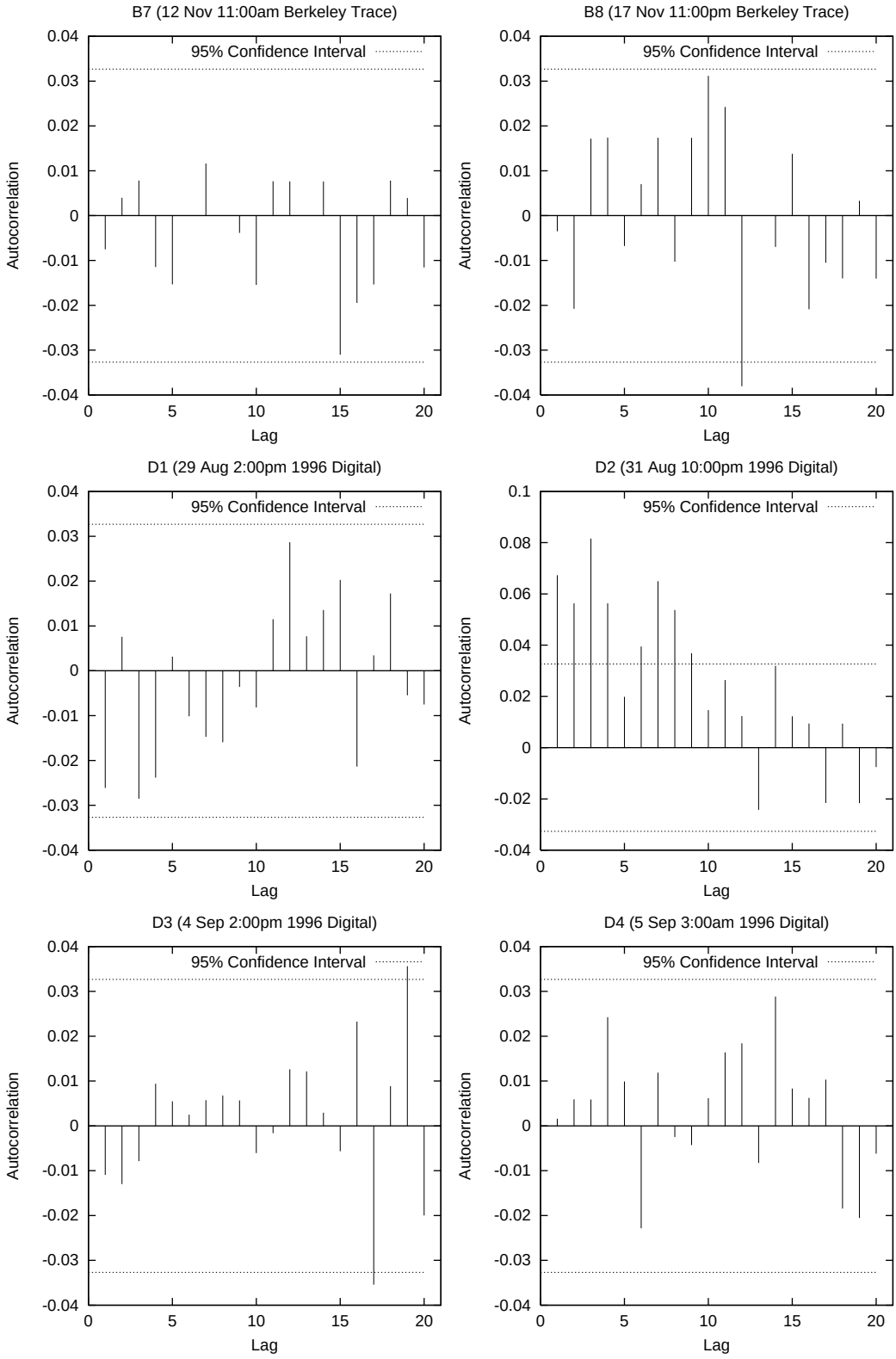


Figure G.3 (Part 2) Correlogram of Number of HTTP Session Arrivals per Second for Trace Hours Listed in Table 2.2

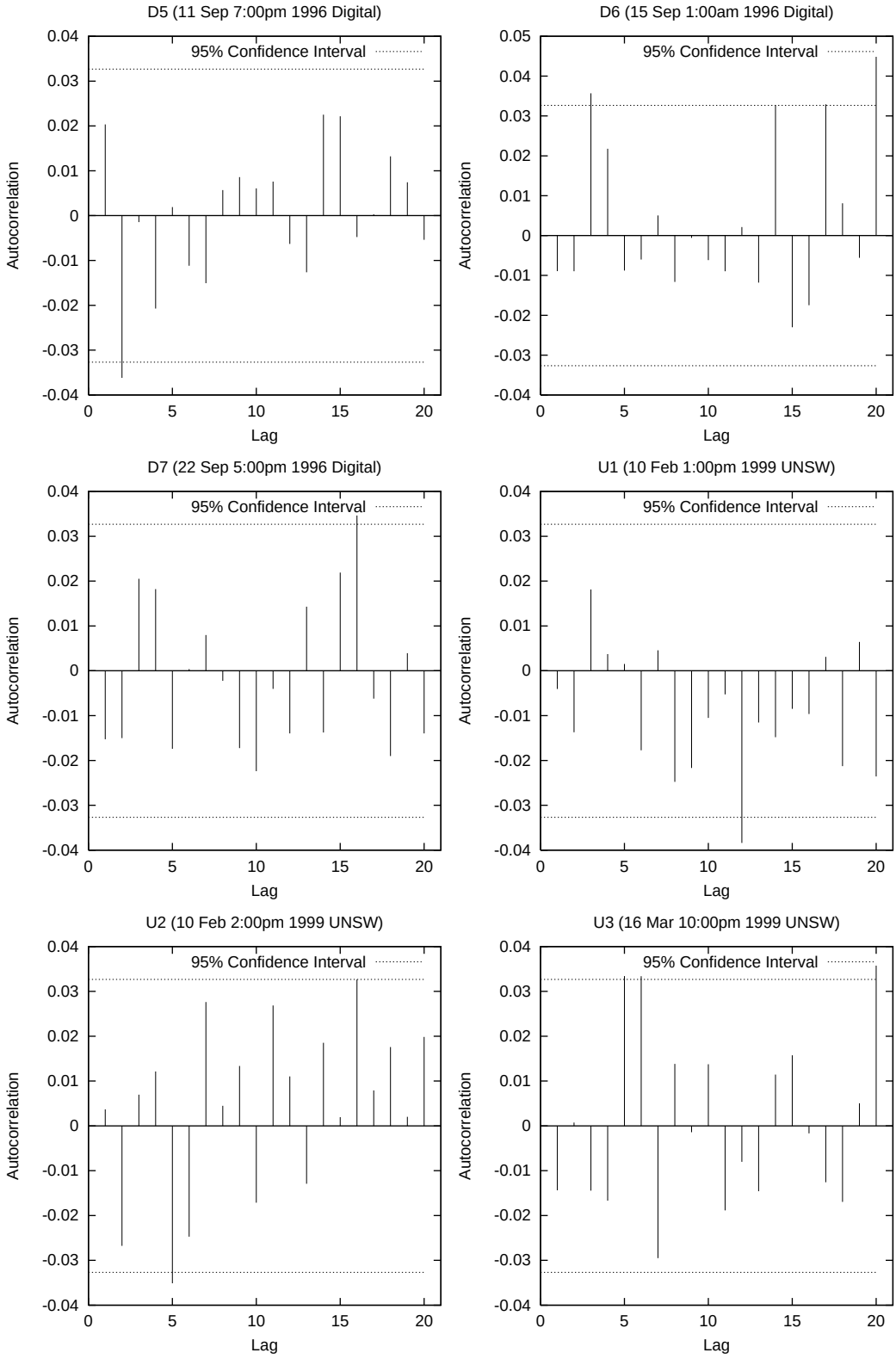


Figure G.3 (Part 3) Correlogram of Number of HTTP Session Arrivals per Second for Trace Hours Listed in Table 2.2

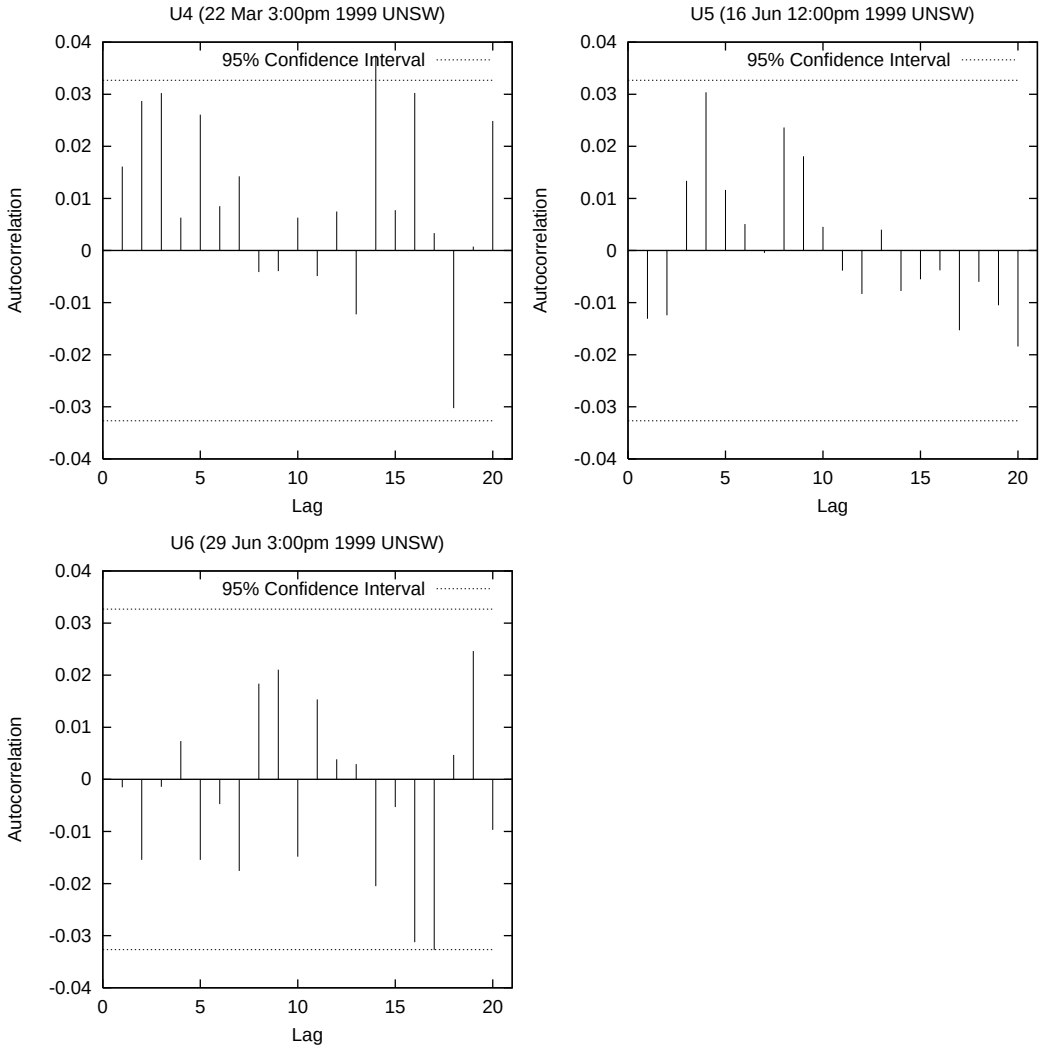


Figure G.3 (Part 4) Correlogram of Number of HTTP Session Arrivals per Second for Trace Hours Listed in Table 2.2

Appendix H. Poisson Distribution of Active Sources per Second

In Section 3.3 it is shown that the number of unique active sources of Web traffic per second has a Poisson distribution. Figure 3.3 shows the Poisson distribution against the observed data for the sample hours D7 and U6. The following figure shows the same comparison for all the hours listed in Table 2.2.

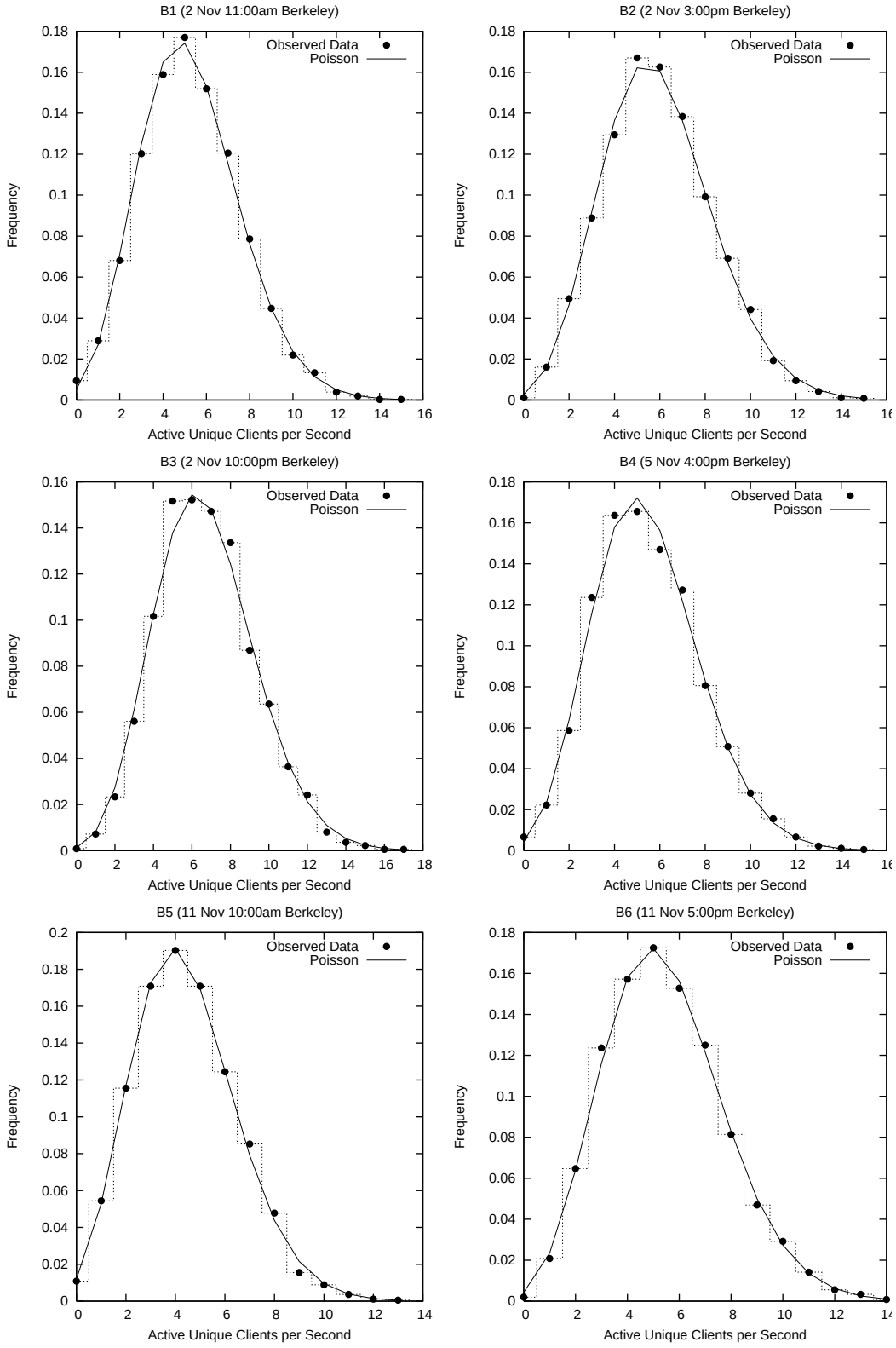


Figure H.1 (Part 1) Histogram of the Number of Active Web Clients Observed Each Second Compared to the Poisson Distribution

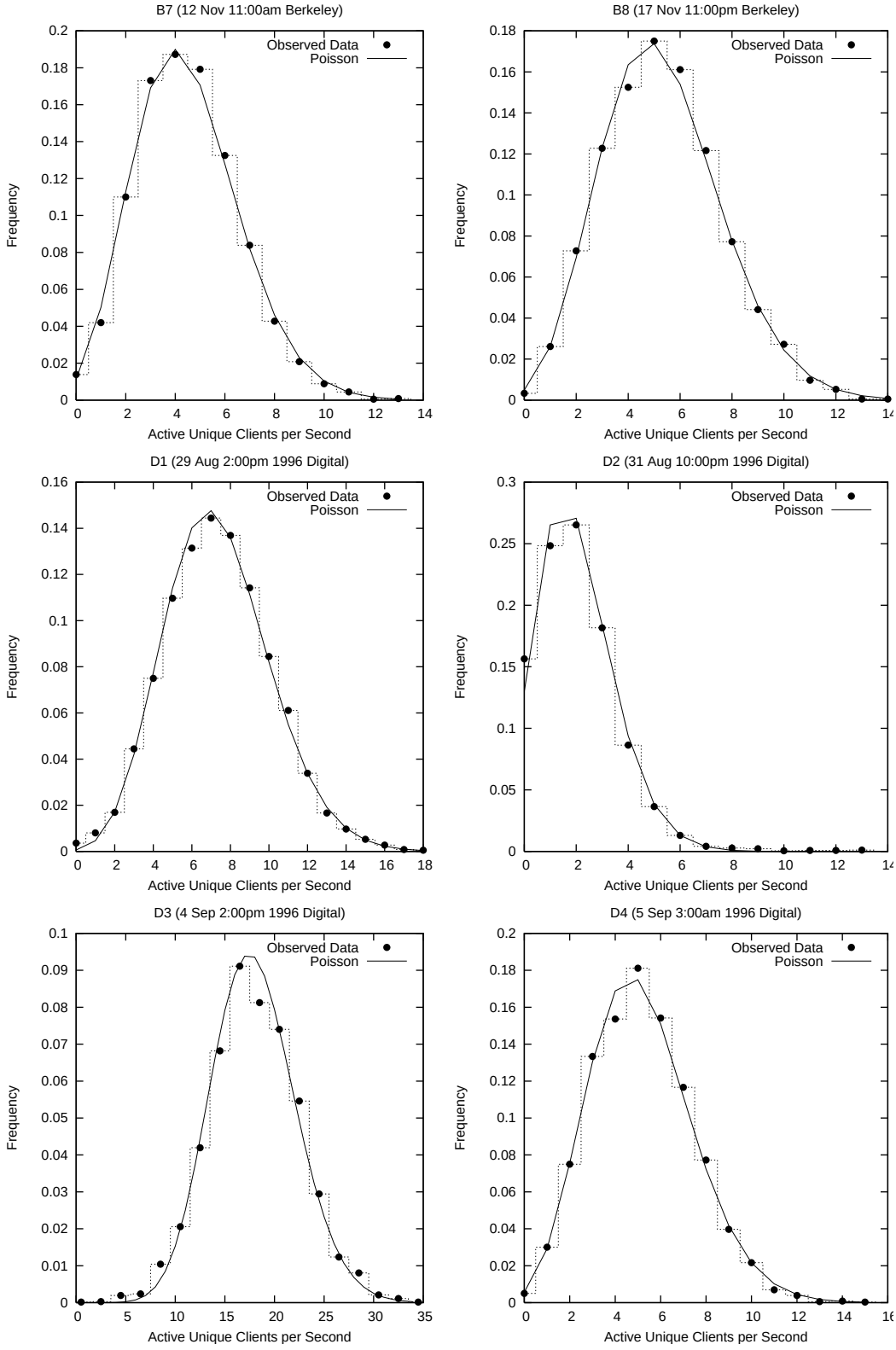


Figure H.1 (Part 2) Histogram of the Number of Active Web Clients Observed Each Second Compared to the Poisson Distribution

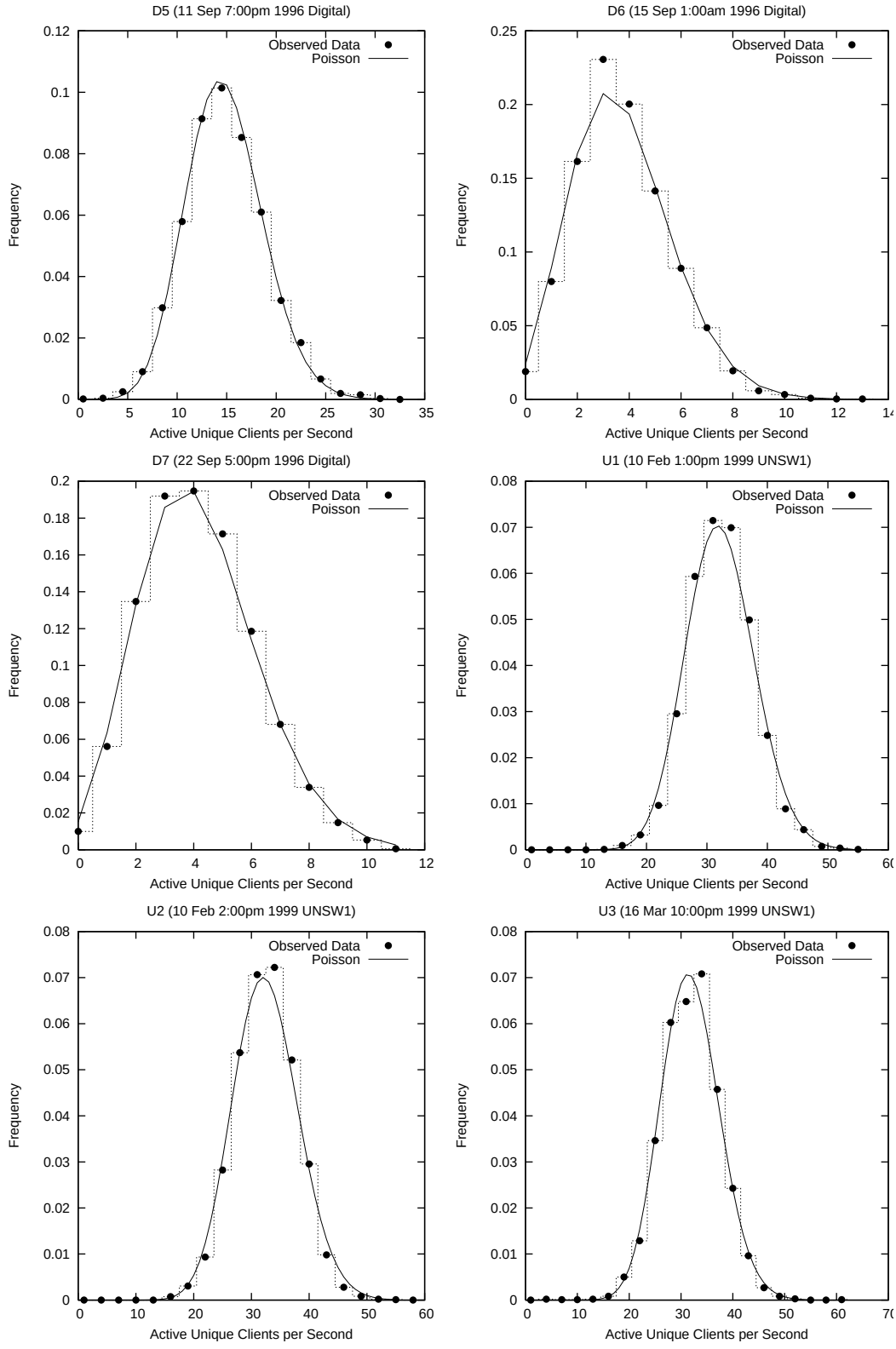


Figure H.1 (Part 3) Histogram of the Number of Active Web Clients Observed Each Second Compared to the Poisson Distribution

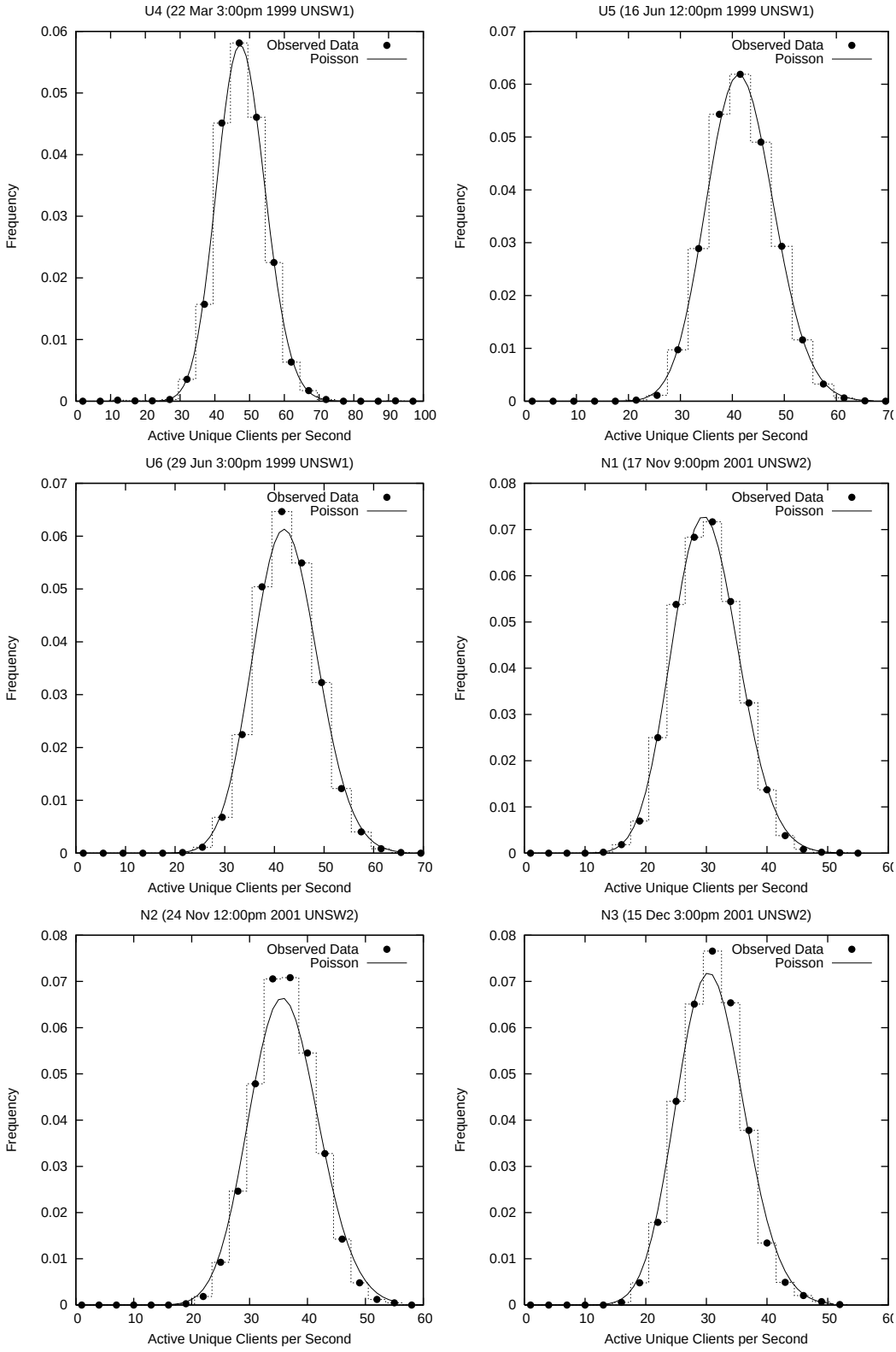


Figure H.1 (Part 4) Histogram of the Number of Active Web Clients Observed Each Second Compared to the Poisson Distribution

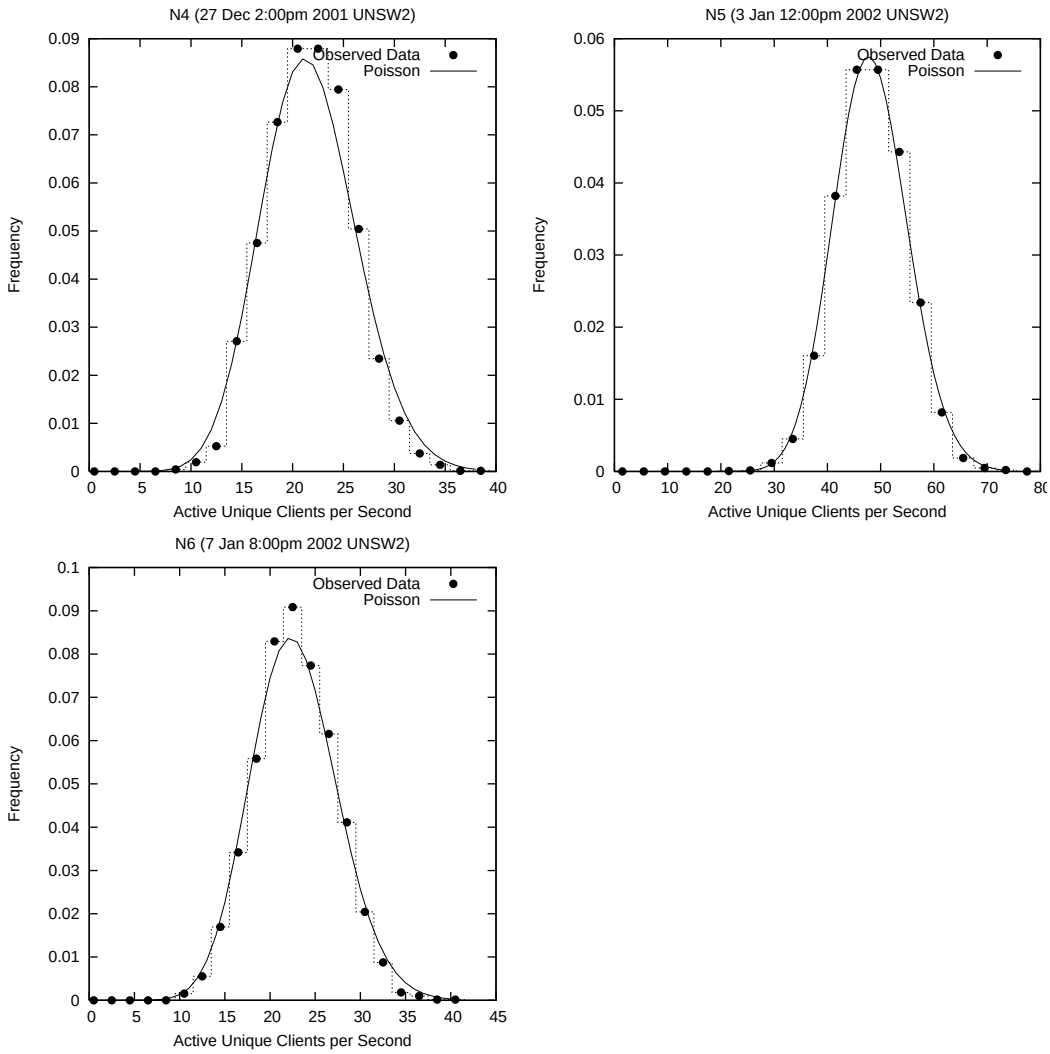


Figure H.1 (Part 5) Histogram of the Number of Active Web Clients Observed Each Second Compared to the Poisson Distribution

Appendix I. HTTP Request Rate for Single Users

In Chapter 4, a number of graphs are shown comparing the observed number of HTTP requests generated by single users against various probability distributions. This appendix contains the plots for all the users listed in Table 4.1. Examples from this appendix are shown in Chapter 4 in Figures 4.1, 4.2 and 4.3.

I.1 HTTP Request Rate per Active Hour

The Figure I.1 shows plots of the number of HTTP requests issued by each of the sixteen users in an active hour versus the zero truncated negative binomial, zero truncated Poisson, geometric, Weibull, lognormal and inverse Gaussian distributions.

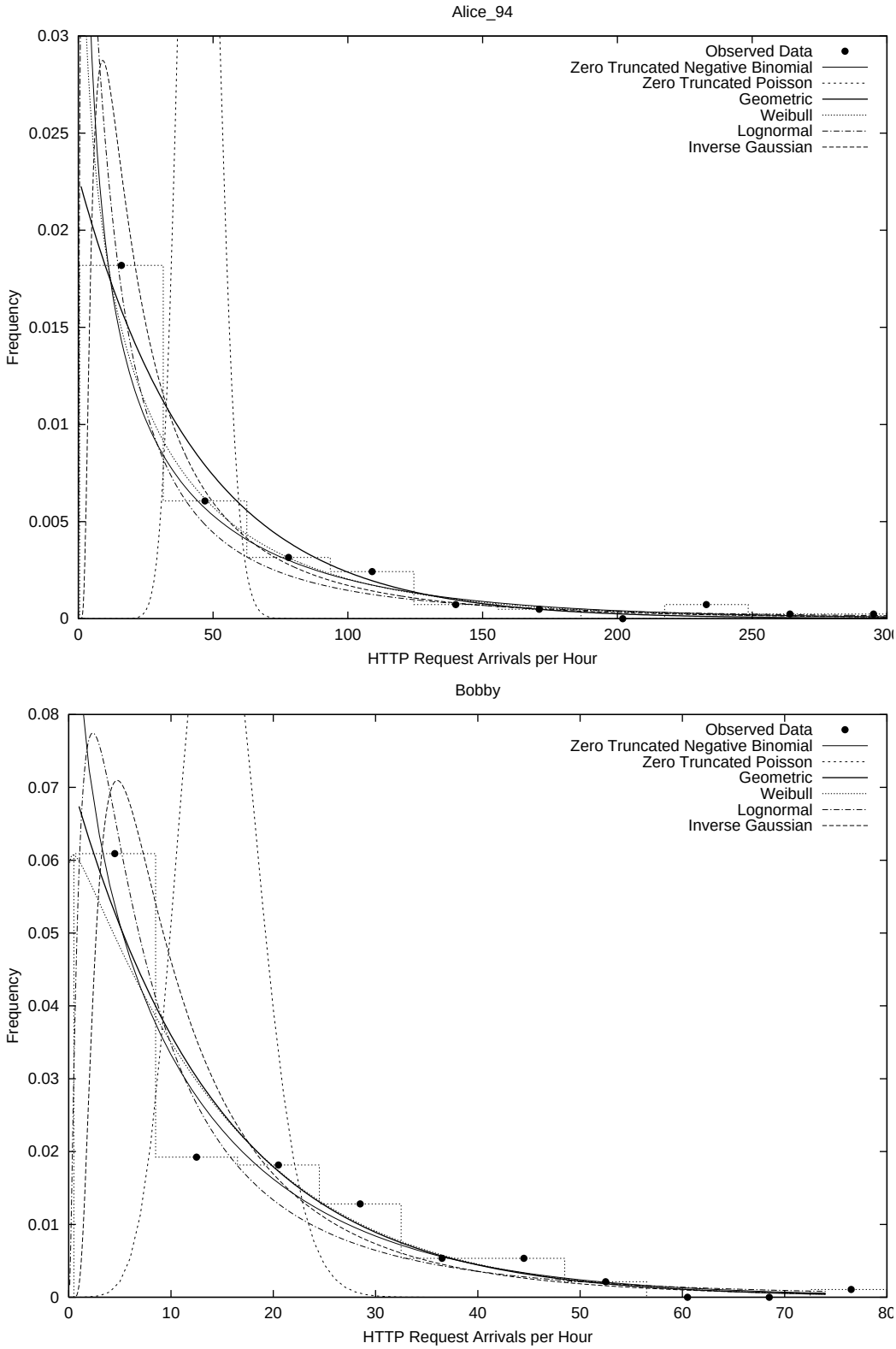


Figure I.1 **(Part 1)** HTTP Request Rate per Active Hour Compared to a Number of Probability Distributions

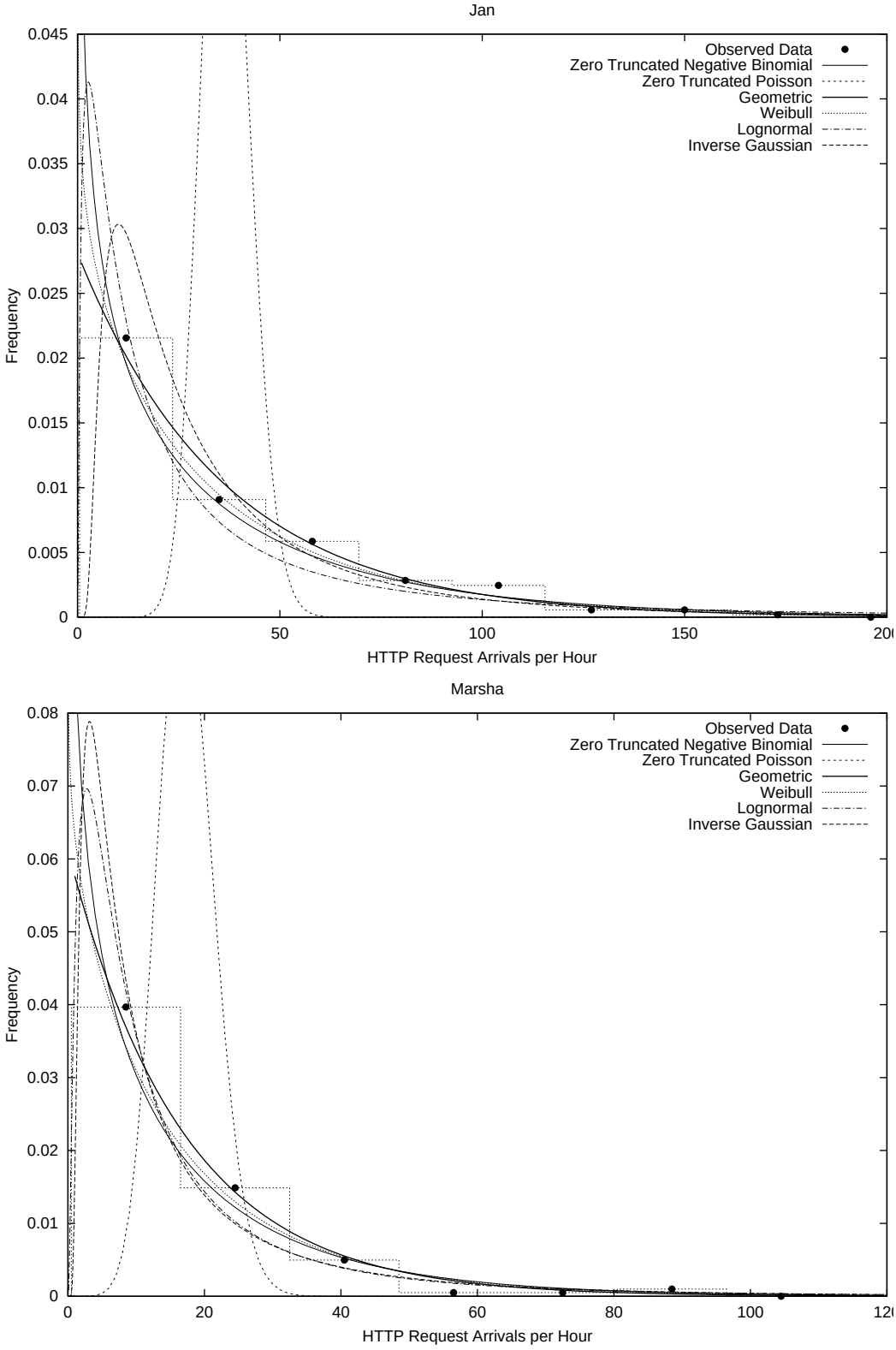


Figure I.1 (Part 2) HTTP Request Rate per Active Hour Compared to a Number of Probability Distributions

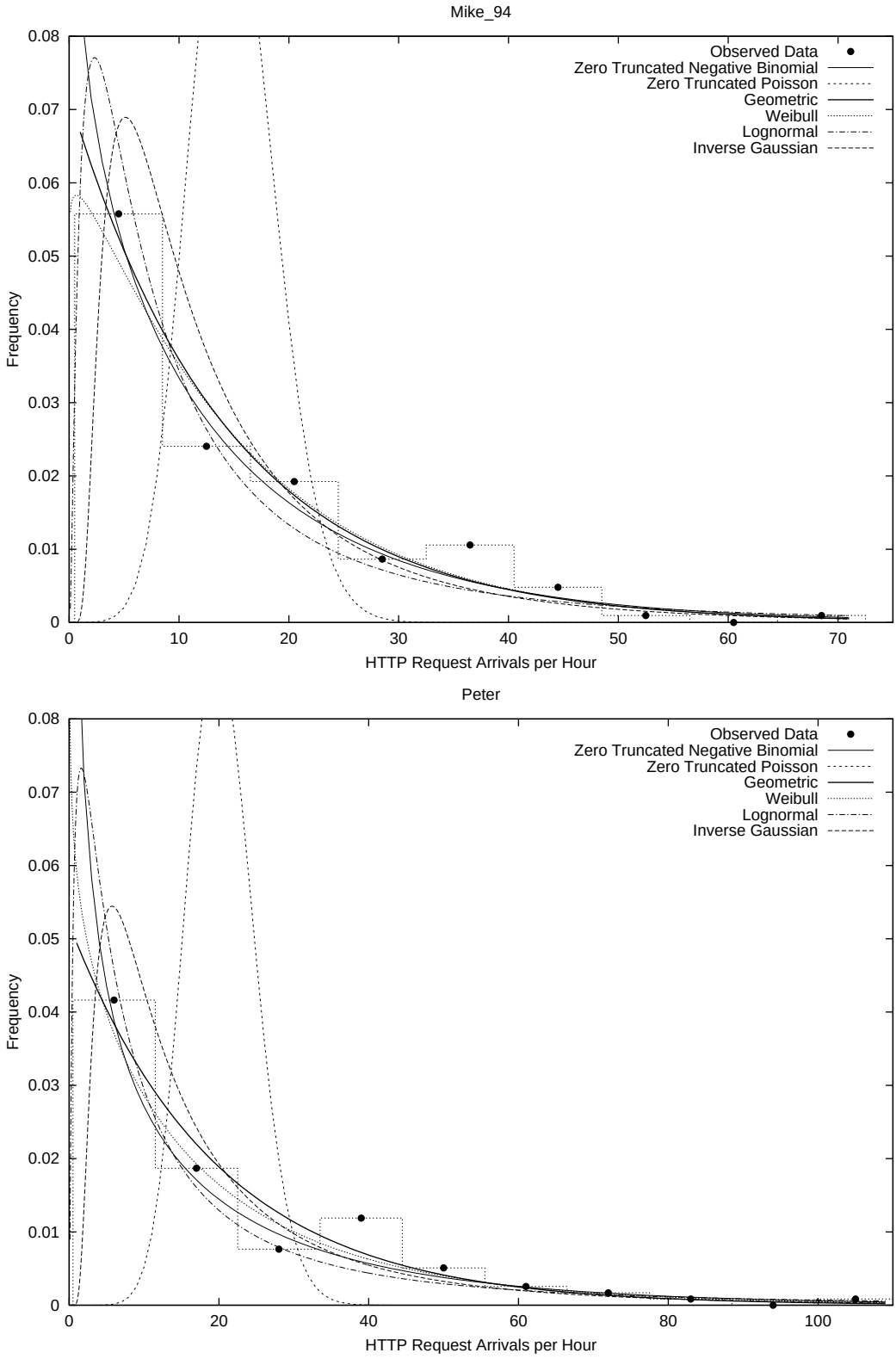


Figure I.1 **(Part 3)** HTTP Request Rate per Active Hour Compared to a Number of Probability Distributions

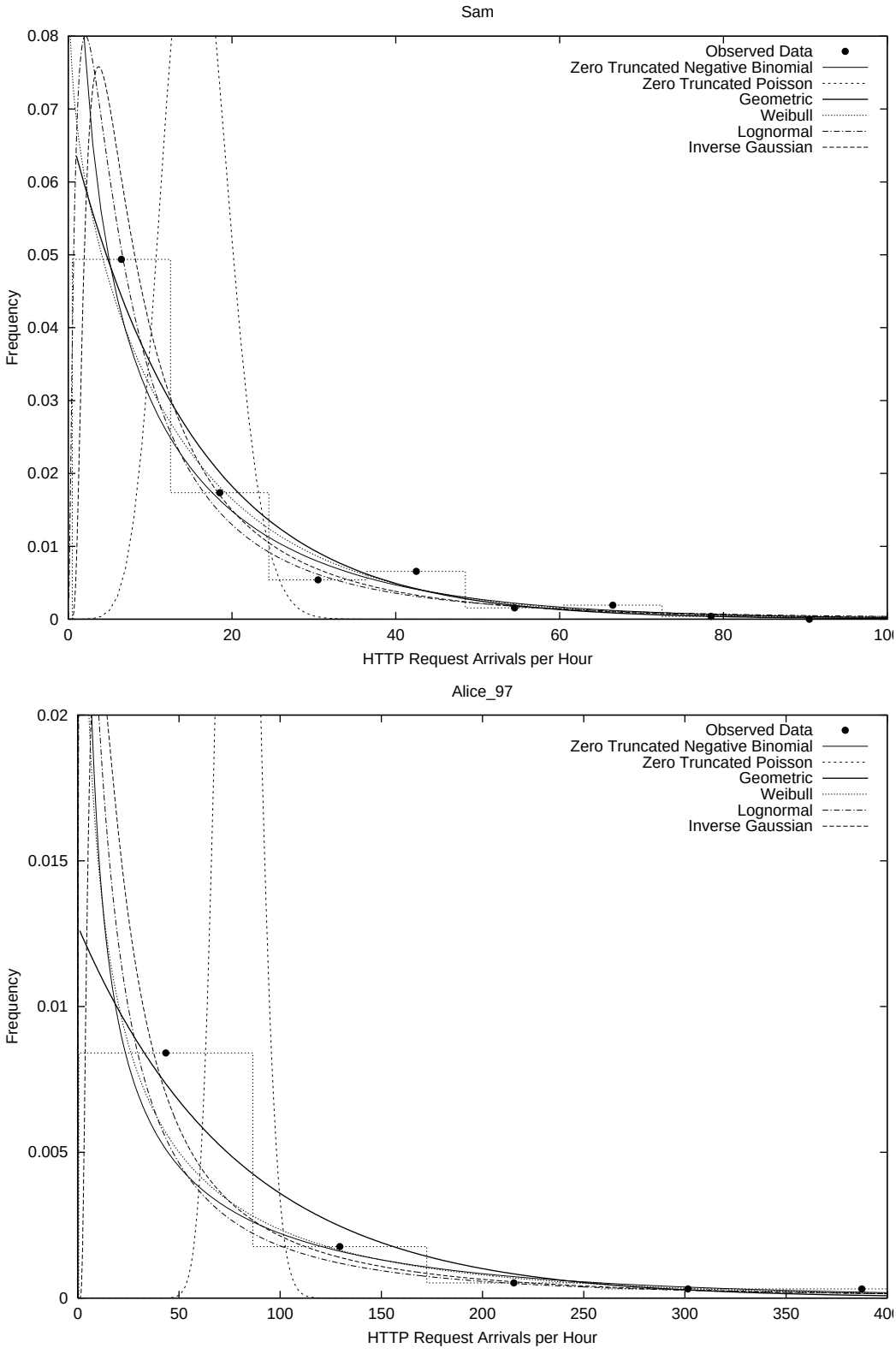


Figure I.1 (Part 4) HTTP Request Rate per Active Hour Compared to a Number of Probability Distributions

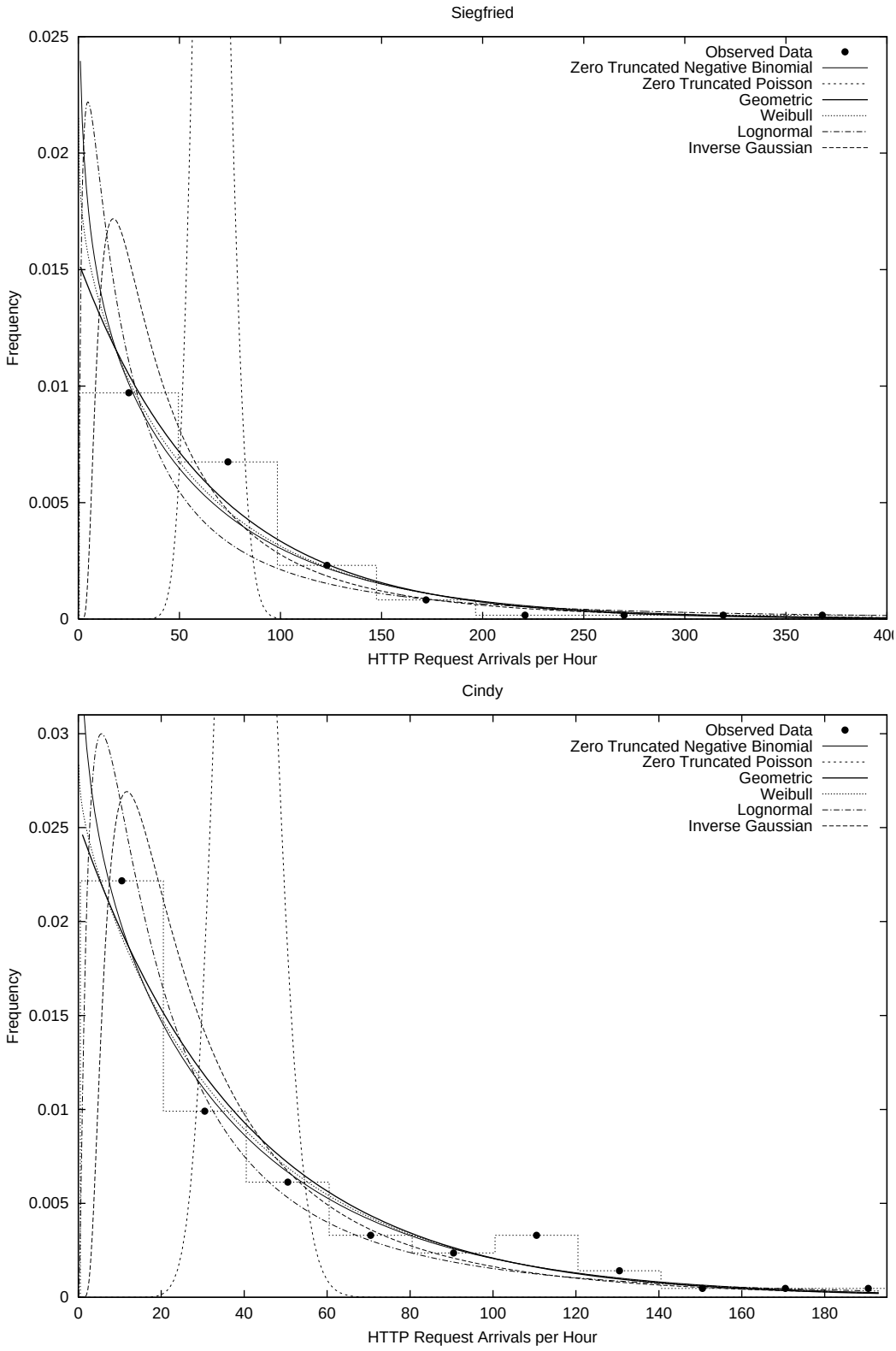


Figure I.1 (Part 5) HTTP Request Rate per Active Hour Compared to a Number of Probability Distributions

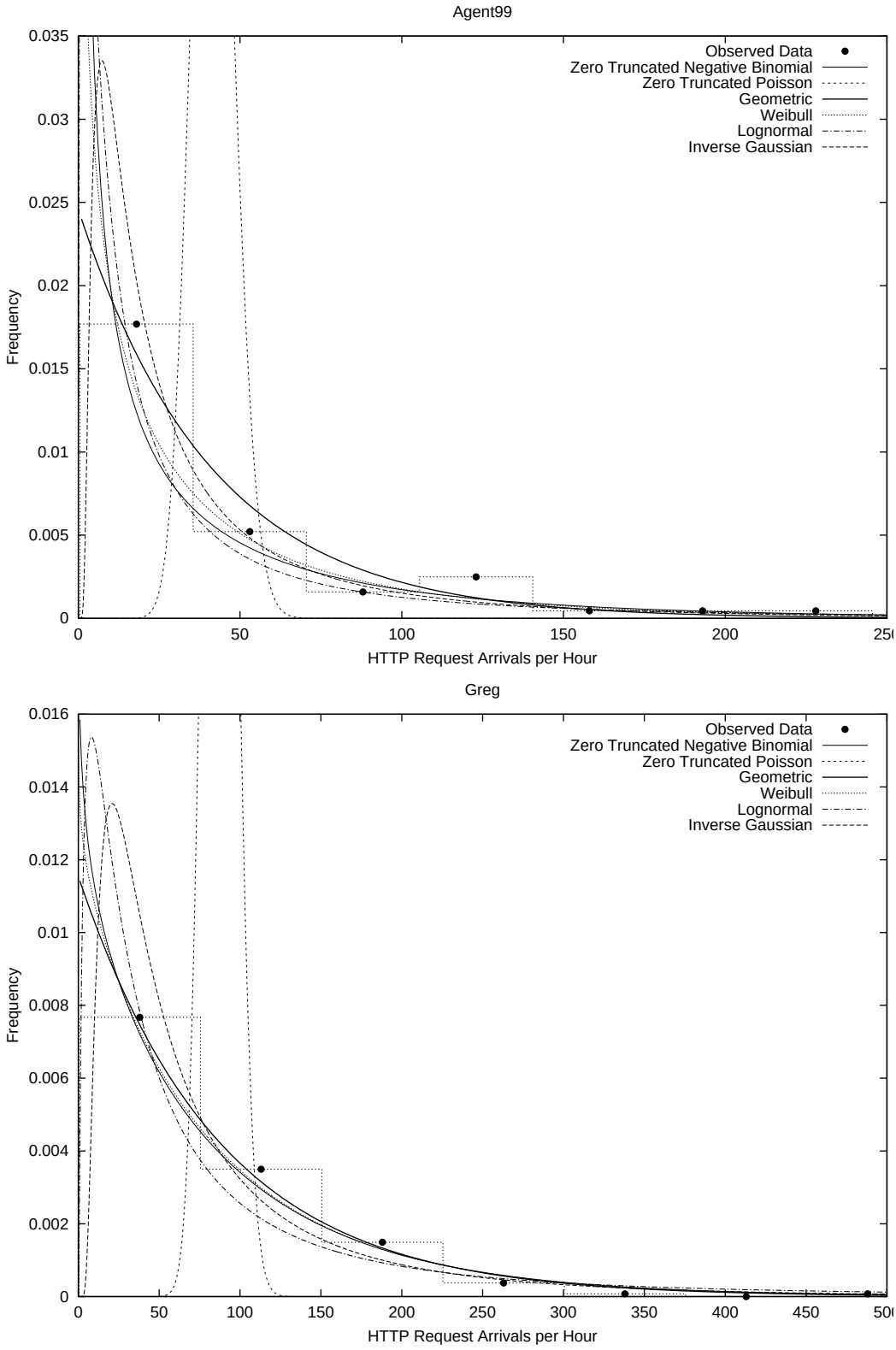


Figure I.1 (Part 6) HTTP Request Rate per Active Hour Compared to a Number of Probability Distributions

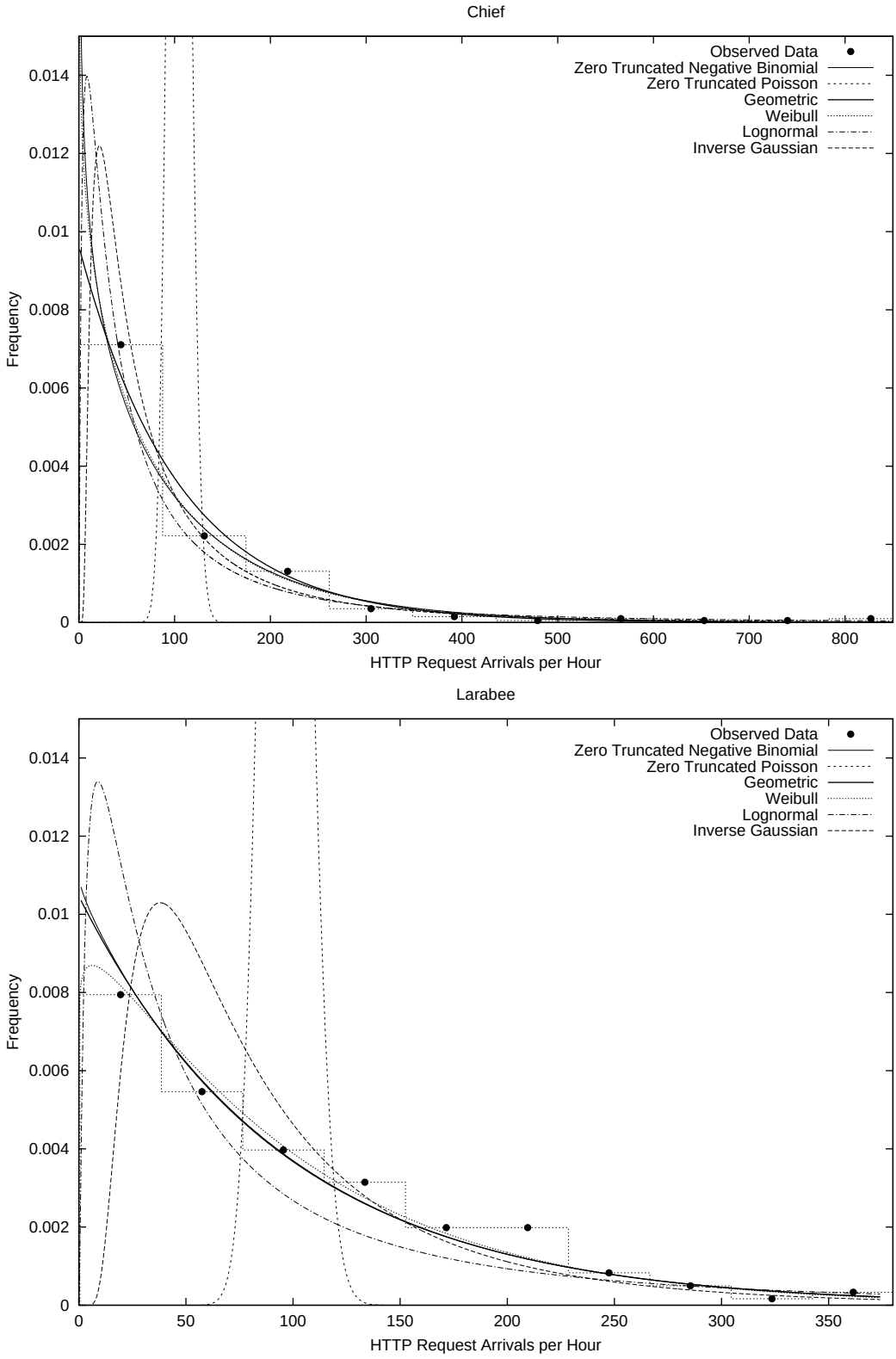


Figure I.1 (Part 7) HTTP Request Rate per Active Hour Compared to a Number of Probability Distributions

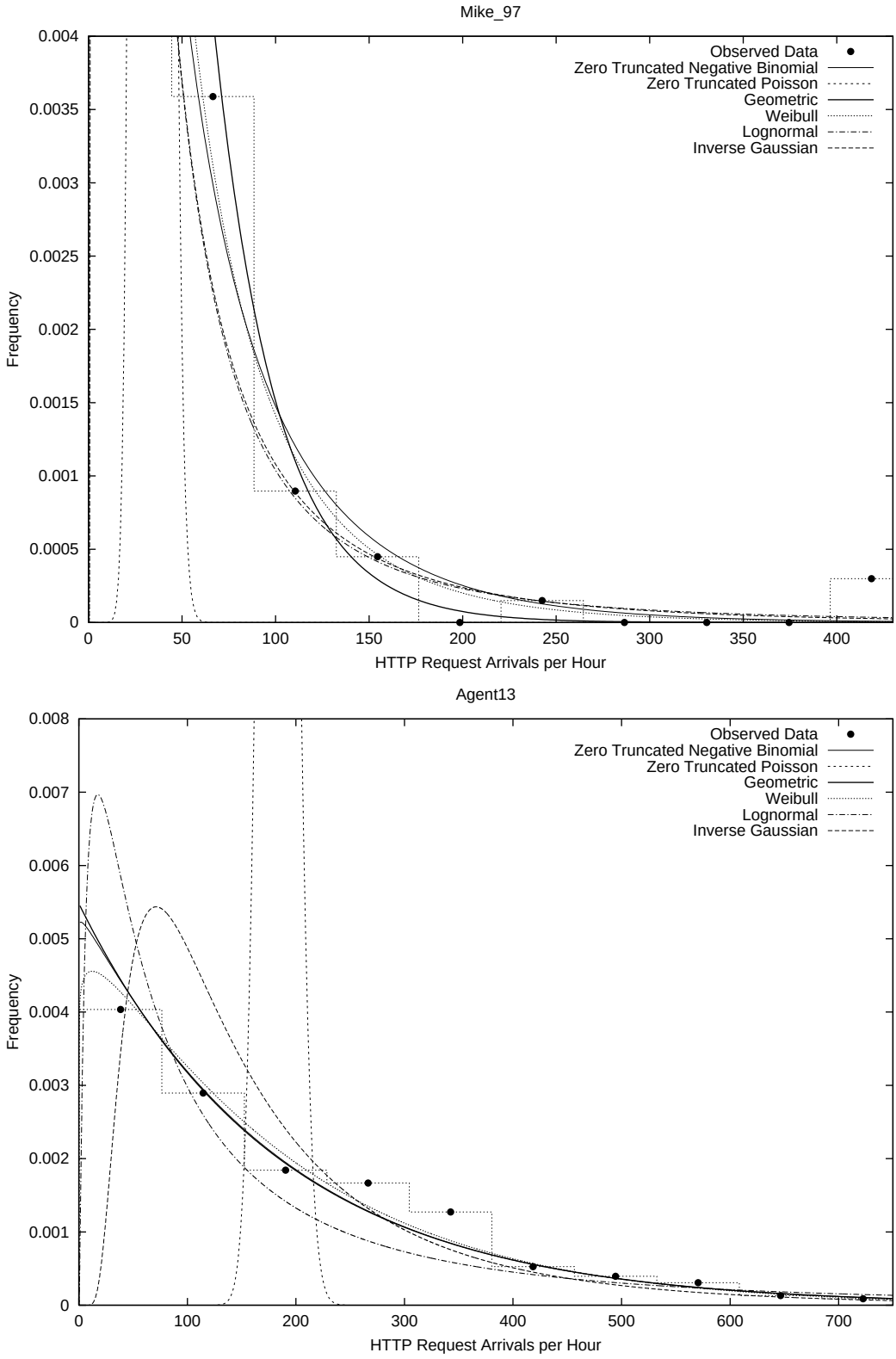


Figure I.1 (Part 8) HTTP Request Rate per Active Hour Compared to a Number of Probability Distributions

I.2 HTTP Request Rate per Active Minute

Figure I.2 shows plots of the number of HTTP requests issued by each of the sixteen users listed in Table 4.1 in an active minute versus the zero truncated negative binomial, zero truncated Poisson, geometric, Weibull, lognormal and inverse Gaussian distributions.

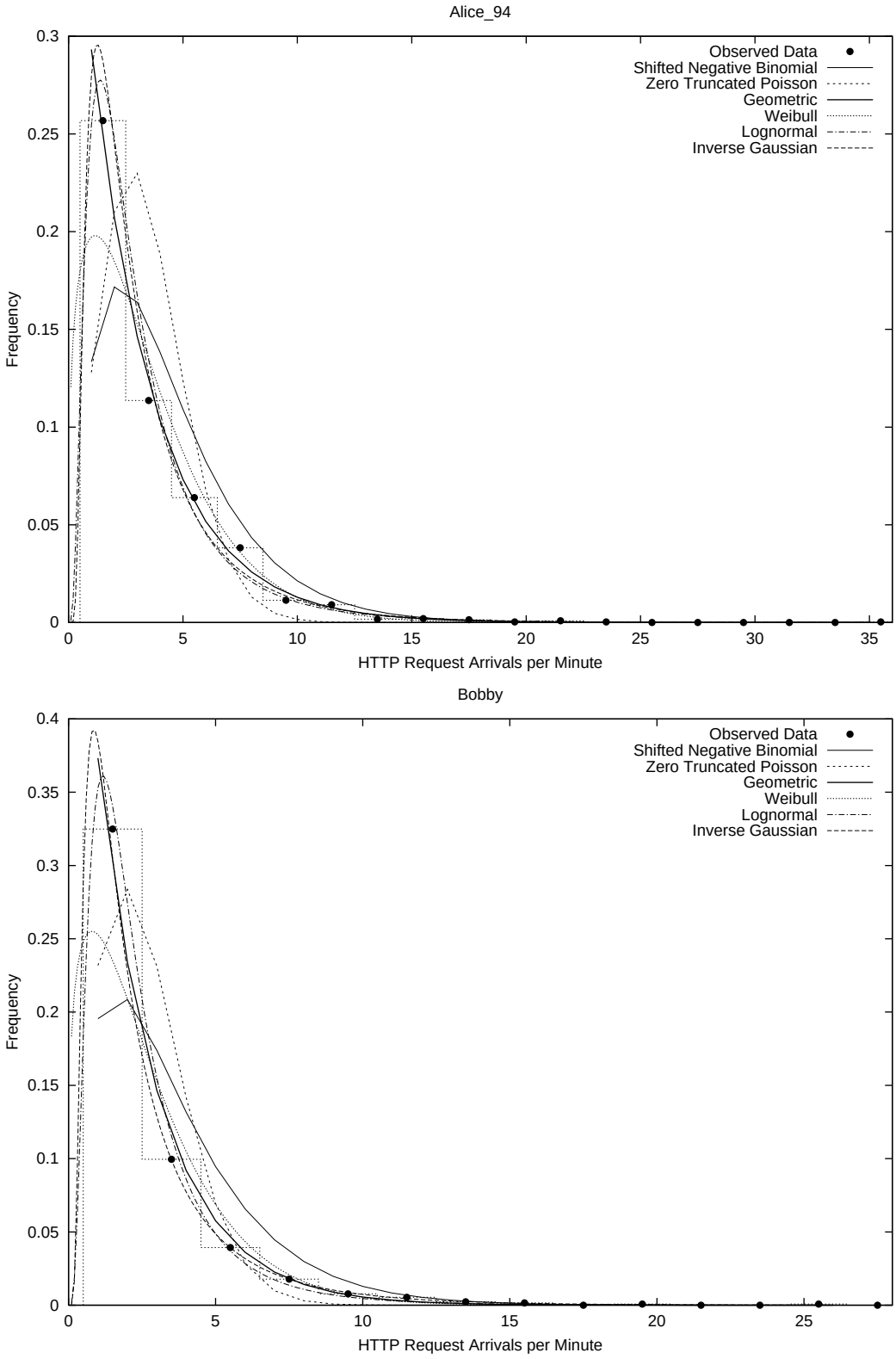


Figure I.2 **(Part 1)** HTTP Request Rate per Active Minute Compared to a Number of Probability Distributions

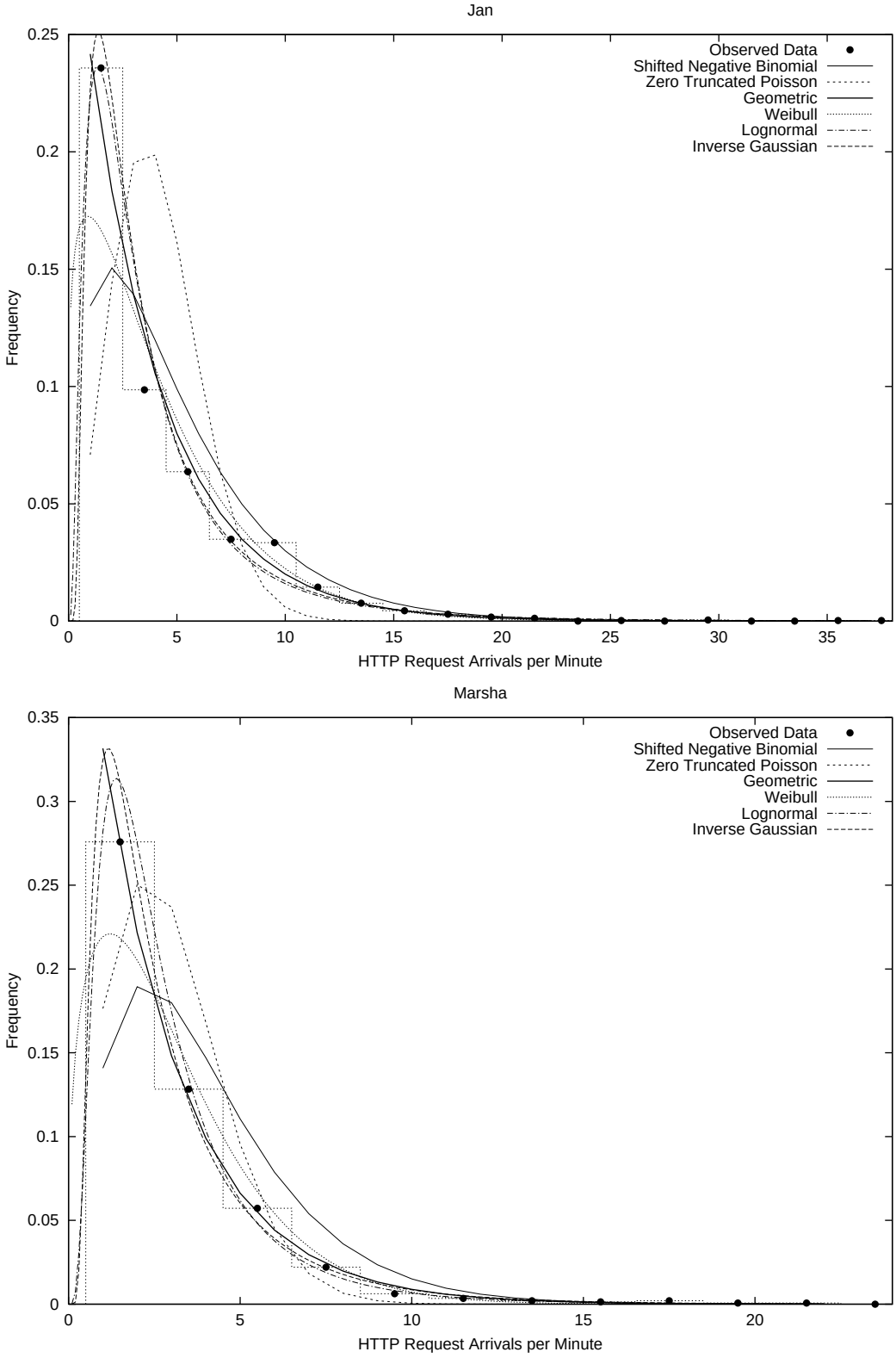


Figure I.2 **(Part 2)** HTTP Request Rate per Active Minute Compared to a Number of Probability Distributions

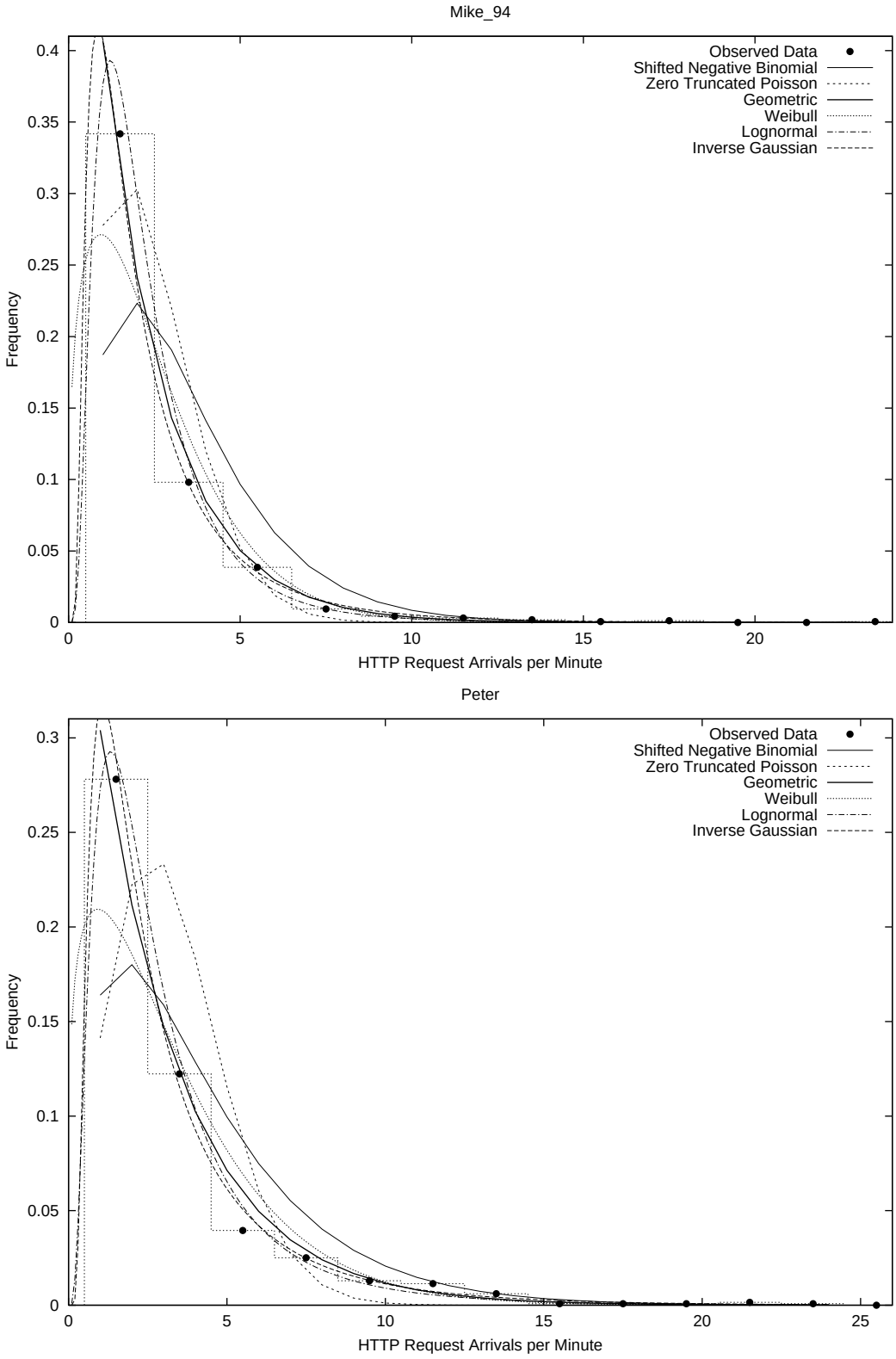


Figure I.2 **(Part 3)** HTTP Request Rate per Active Minute Compared to a Number of Probability Distributions

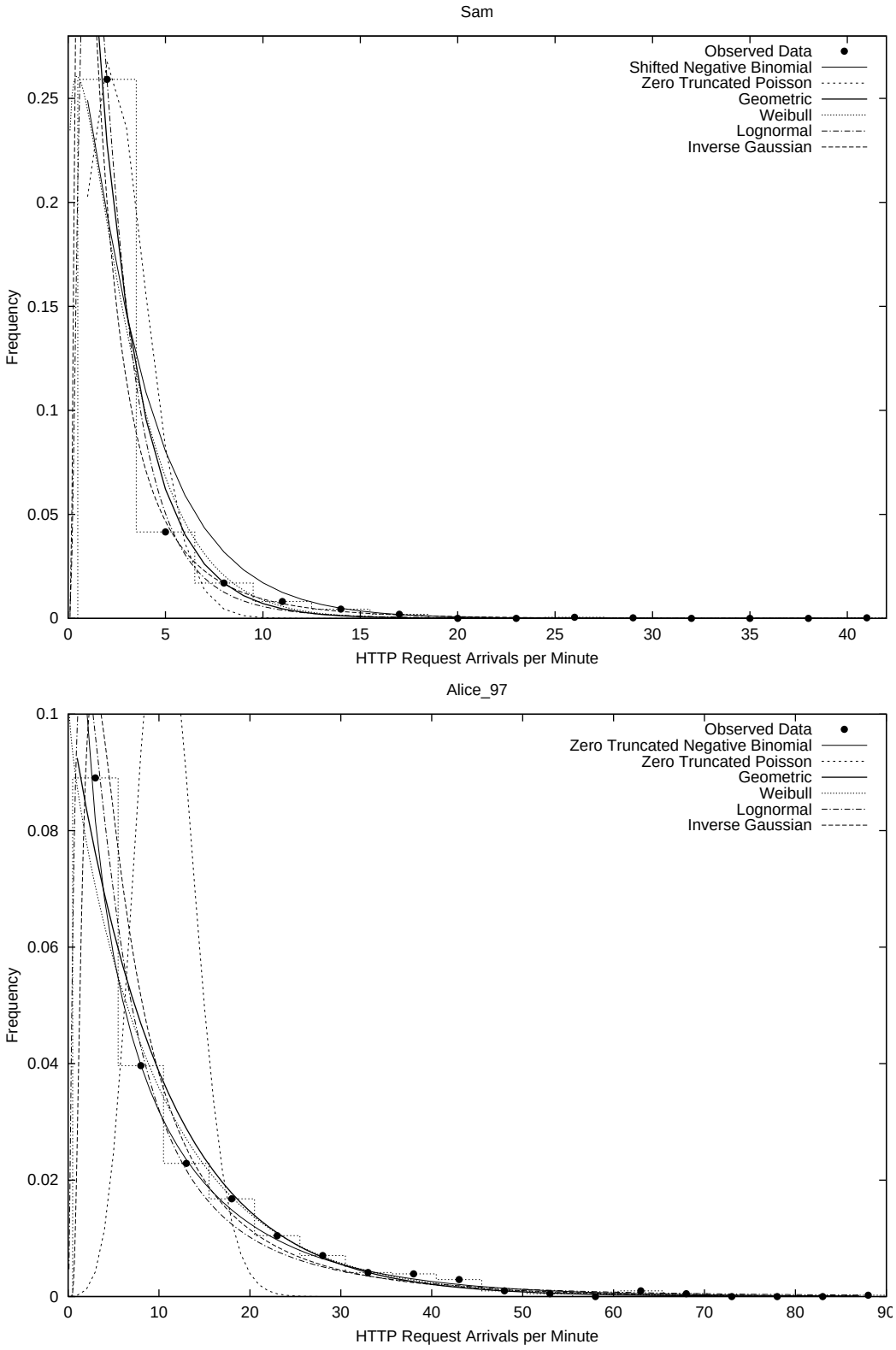


Figure I.2 **(Part 4)** HTTP Request Rate per Active Minute Compared to a Number of Probability Distributions

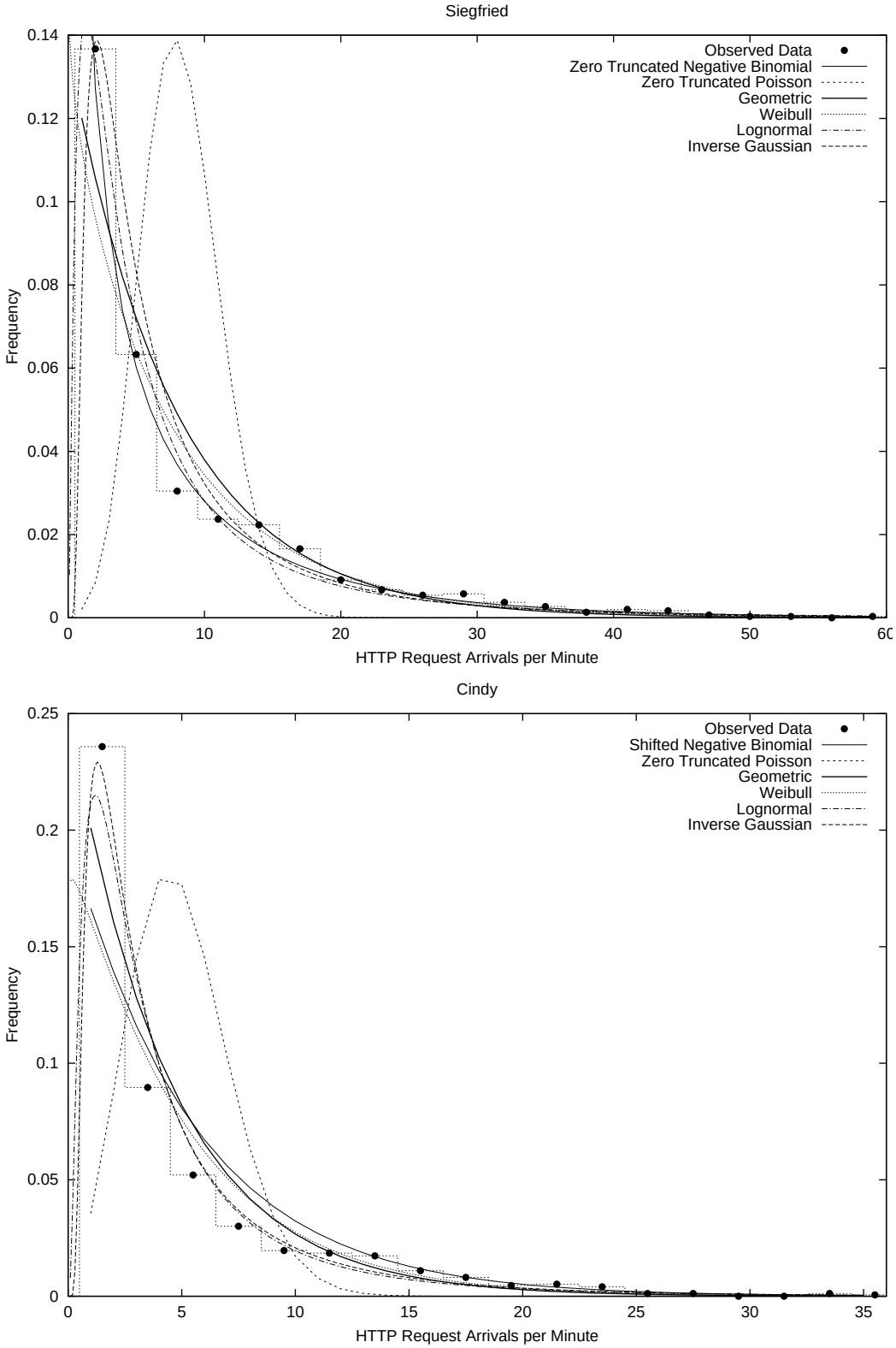


Figure I.2 **(Part 5)** HTTP Request Rate per Active Minute Compared to a Number of Probability Distributions

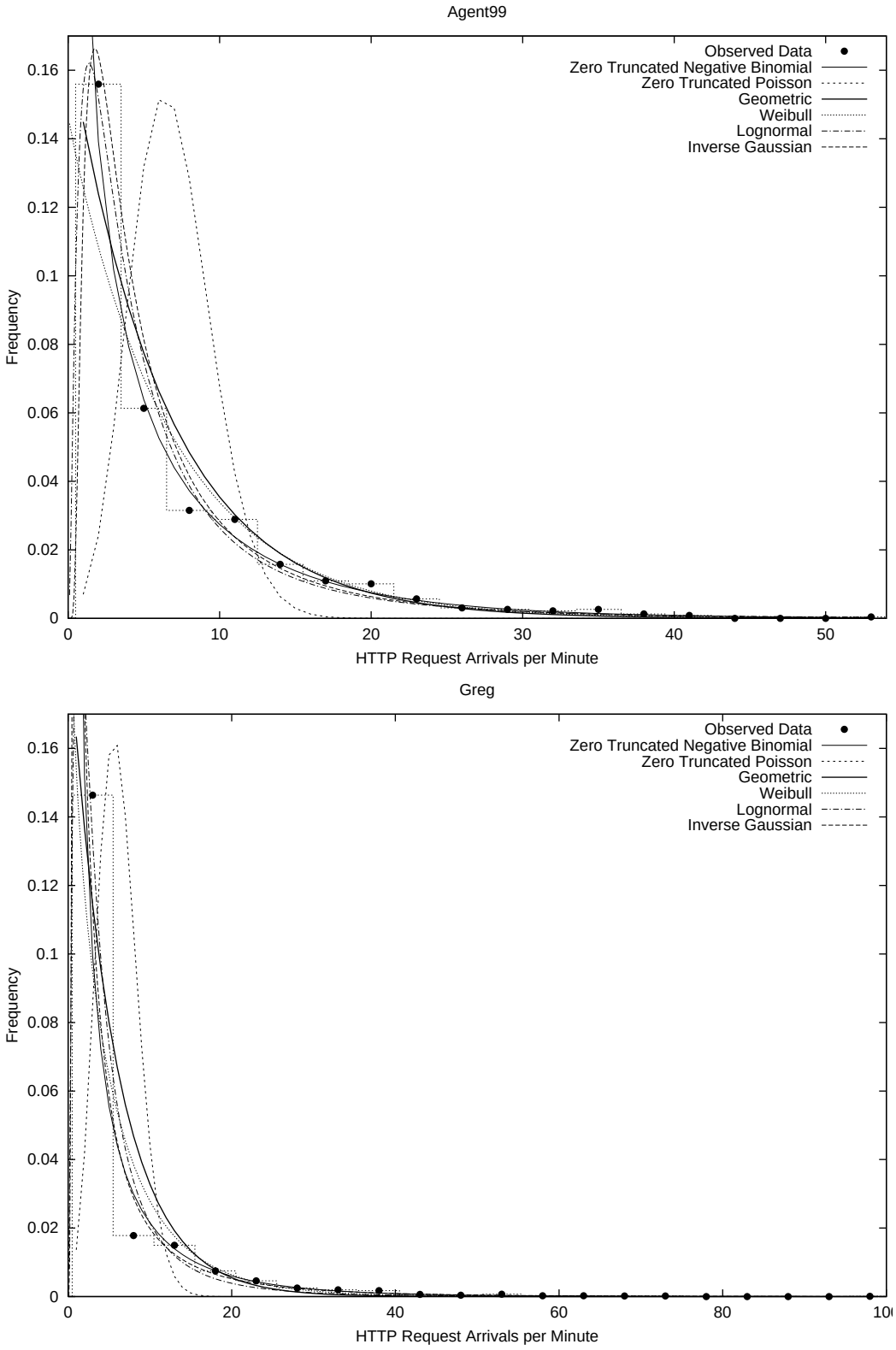


Figure I.2 **(Part 6)** HTTP Request Rate per Active Minute Compared to a Number of Probability Distributions

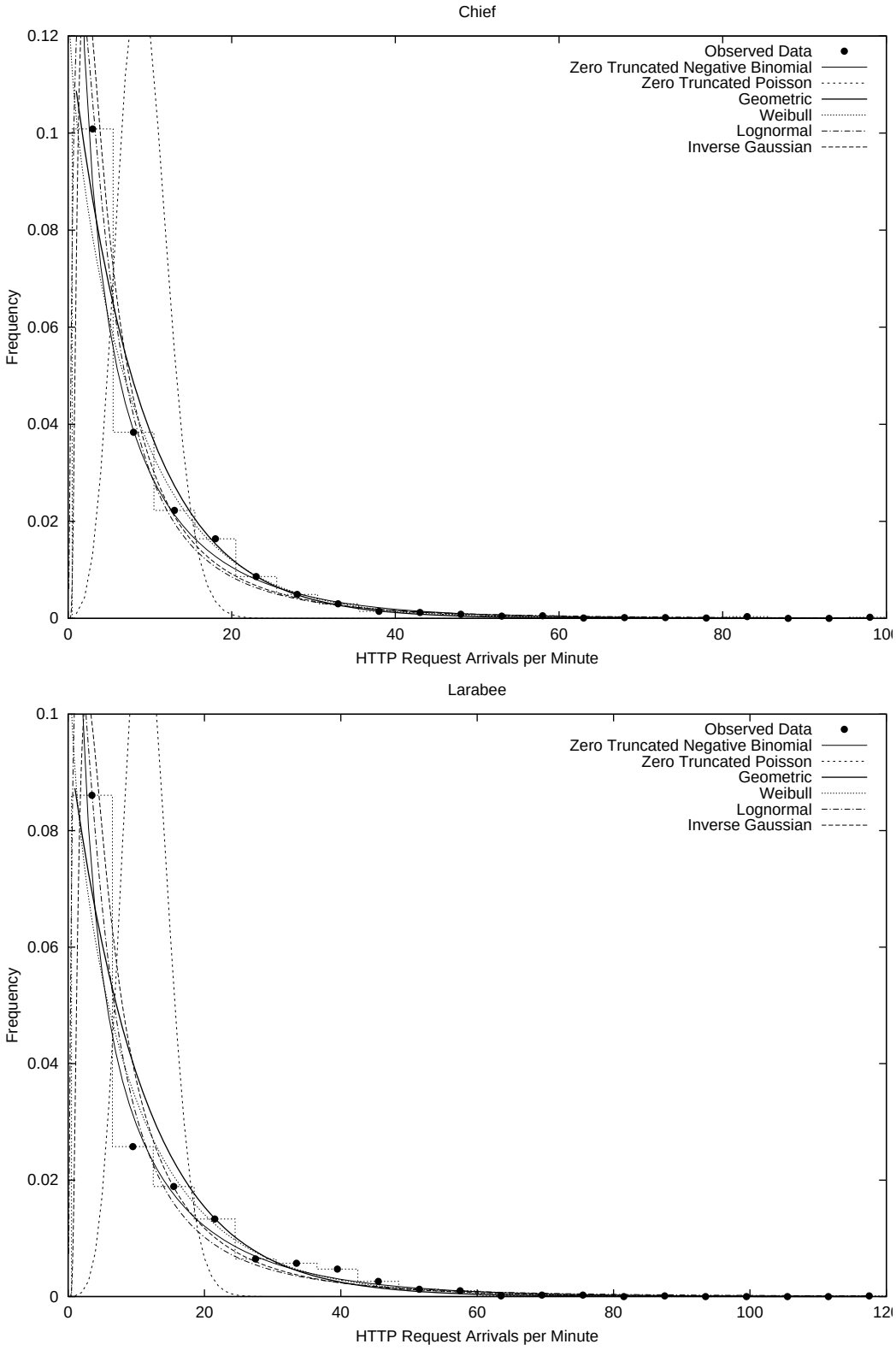


Figure I.2 **(Part 7)** HTTP Request Rate per Active Minute Compared to a Number of Probability Distributions

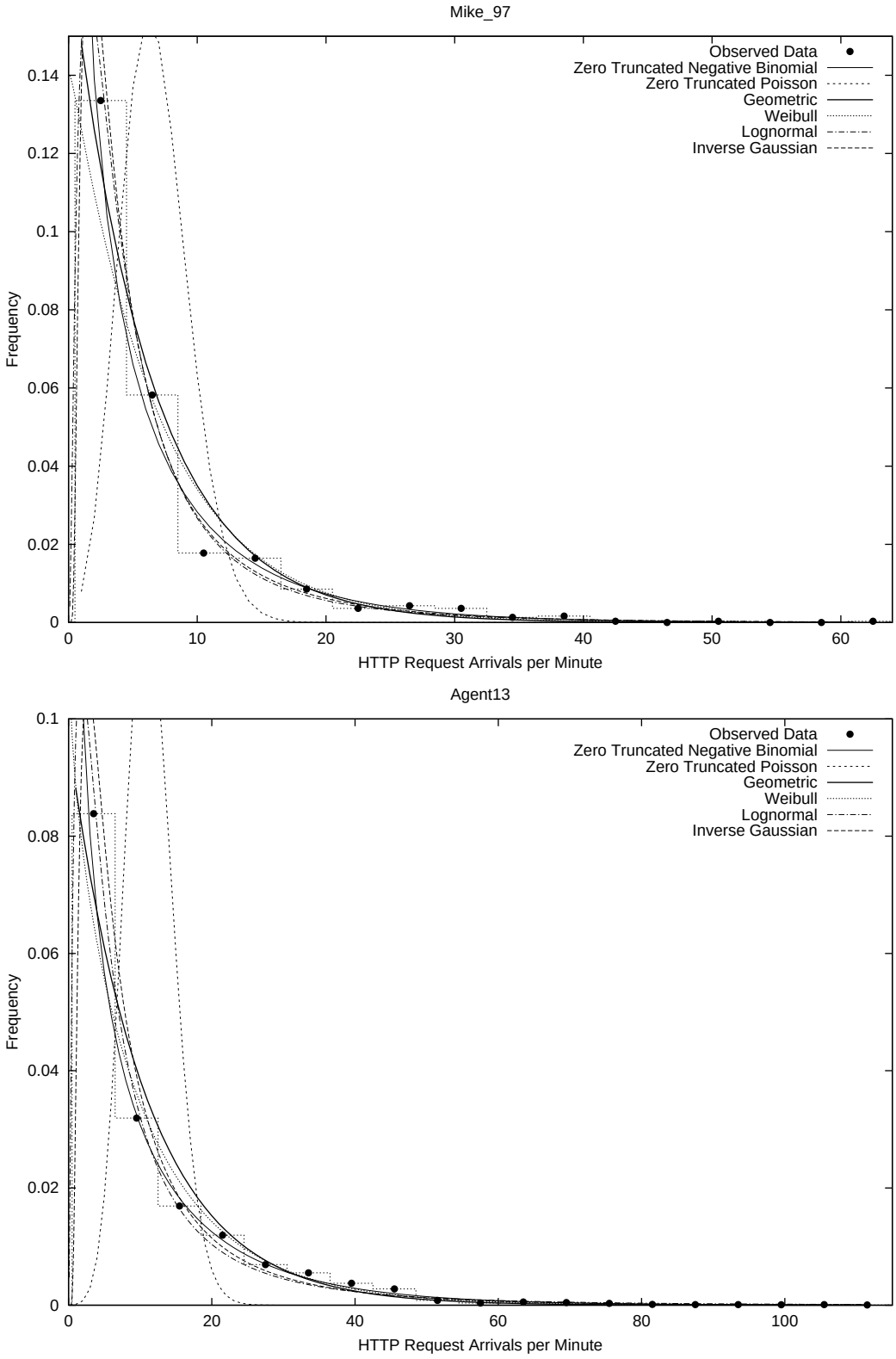


Figure I.2 **(Part 8)** HTTP Request Rate per Active Minute Compared to a Number of Probability Distributions

I.3 HTTP Request Rate per Active Second

Figure I.3 shows plots of the number of HTTP requests issued by each of the sixteen users listed in Table 4.1 in an active second versus the shifted negative binomial, zero truncated Poisson, geometric, Weibull, lognormal and inverse Gaussian distributions.

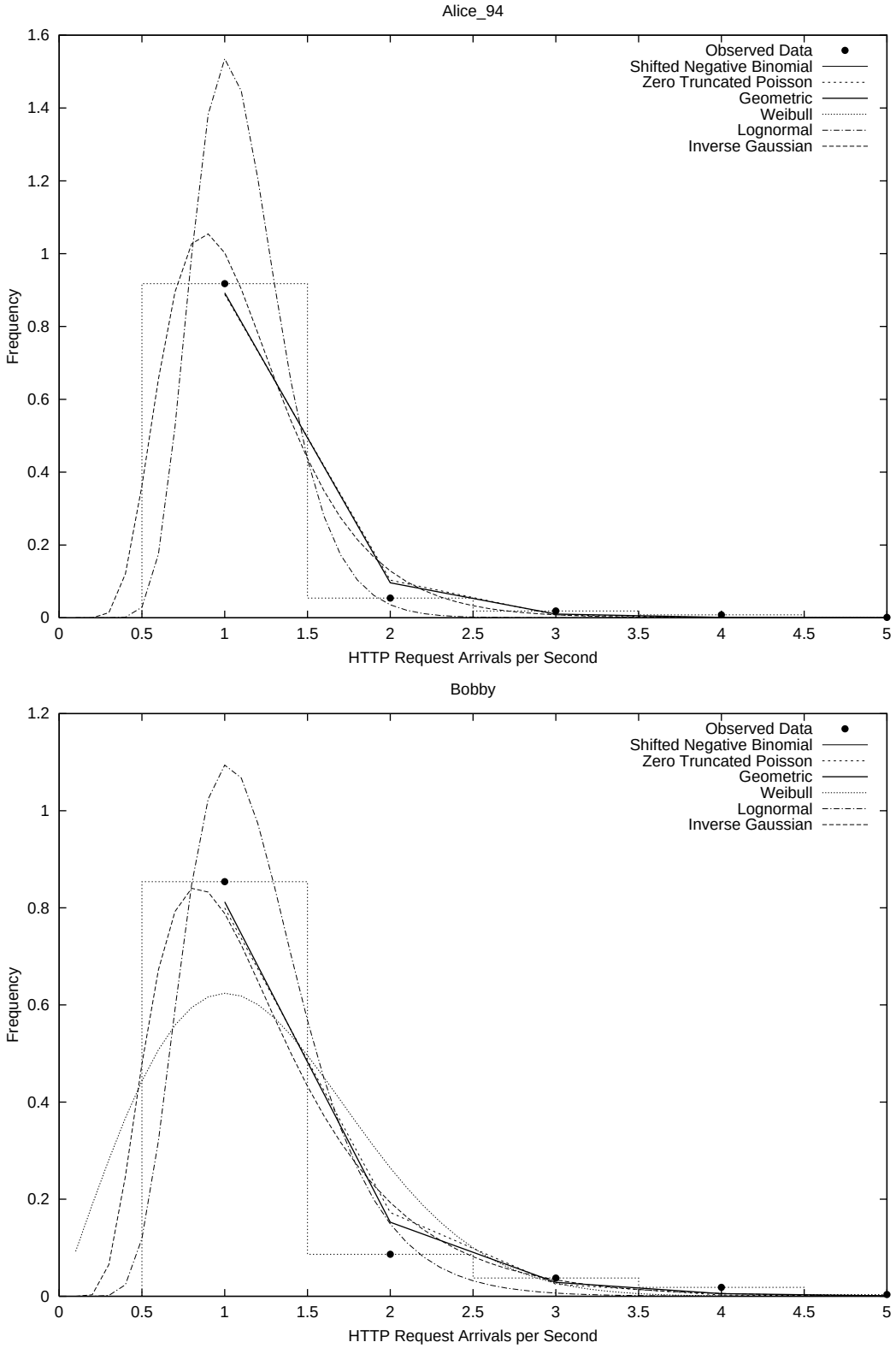


Figure I.3 **(Part 1)** HTTP Request Rate per Active Second Compared to a Number of Probability Distributions

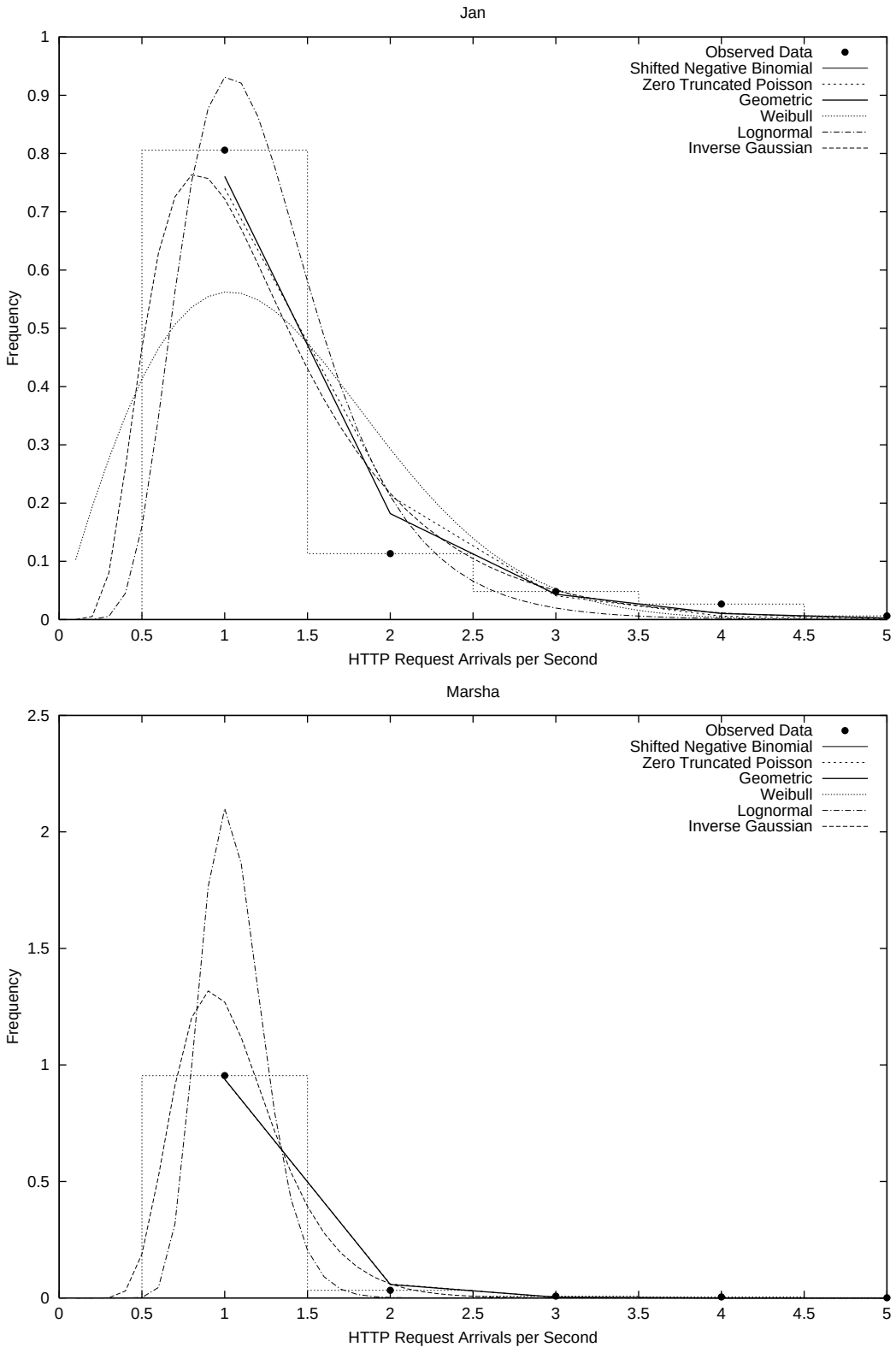


Figure I.3 (Part 2) HTTP Request Rate per Active Second Compared to a Number of Probability Distributions

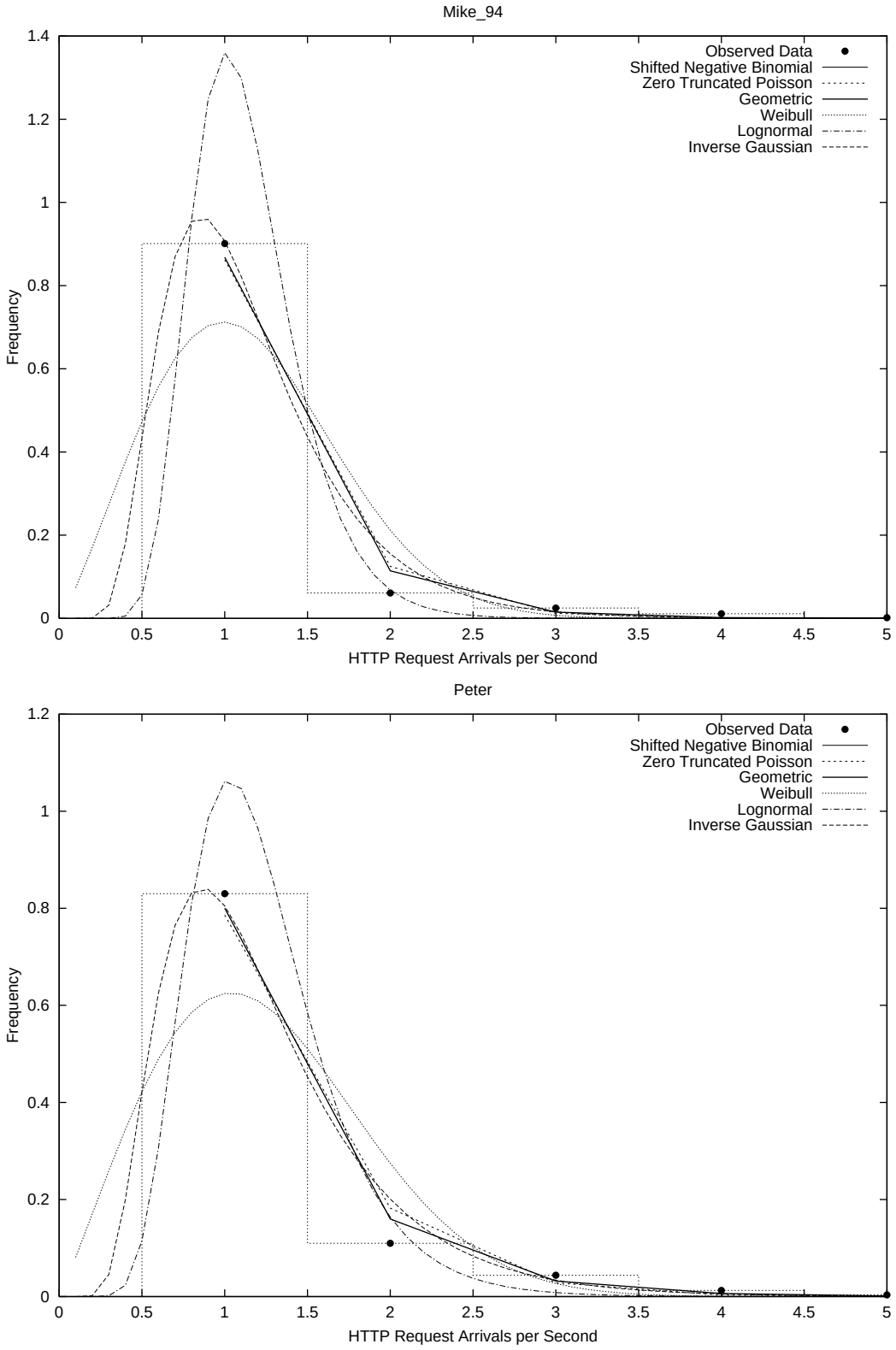


Figure I.3 **(Part 3)** HTTP Request Rate per Active Second Compared to a Number of Probability Distributions

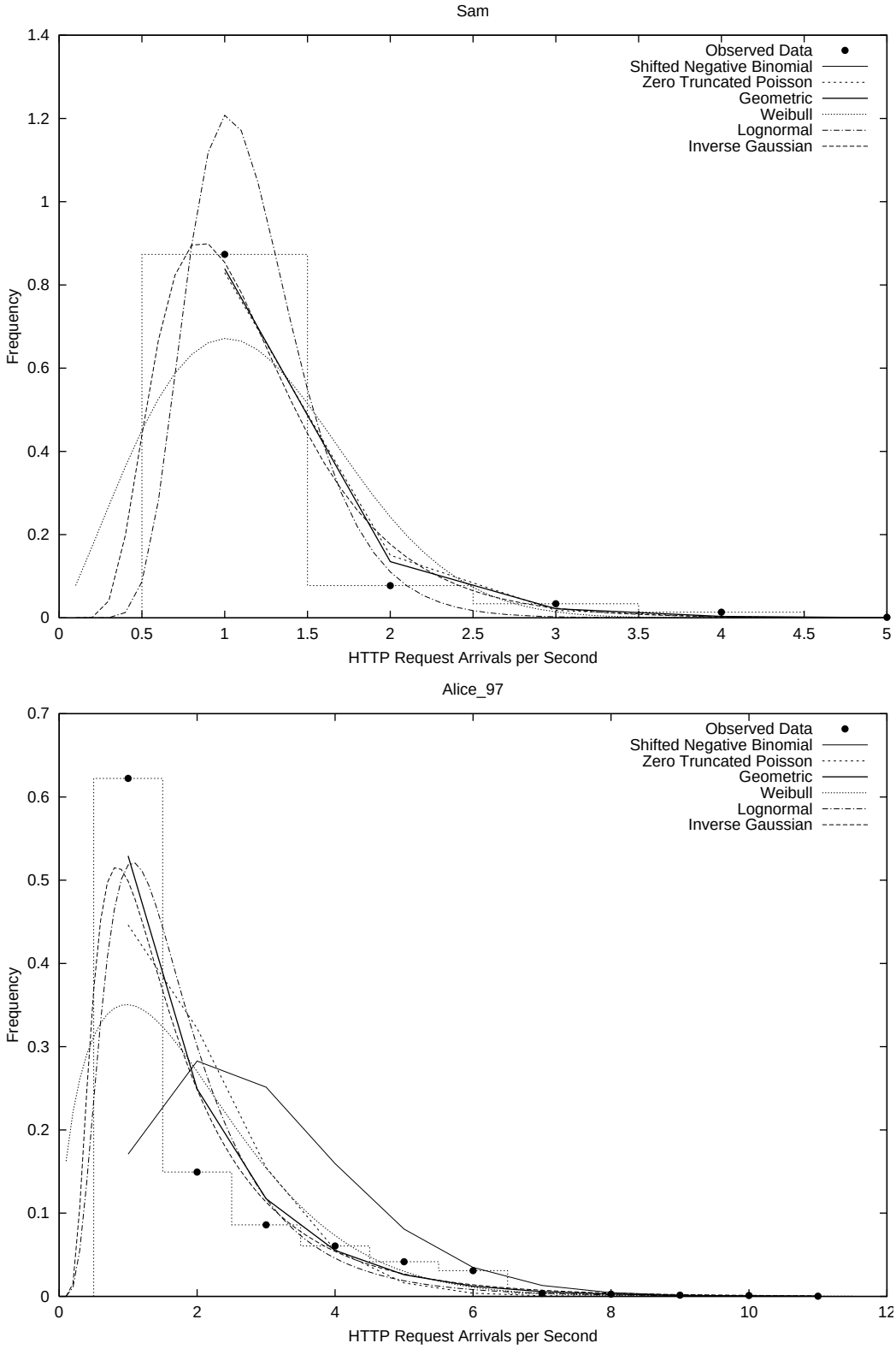


Figure I.3 (Part 4) HTTP Request Rate per Active Second Compared to a Number of Probability Distributions

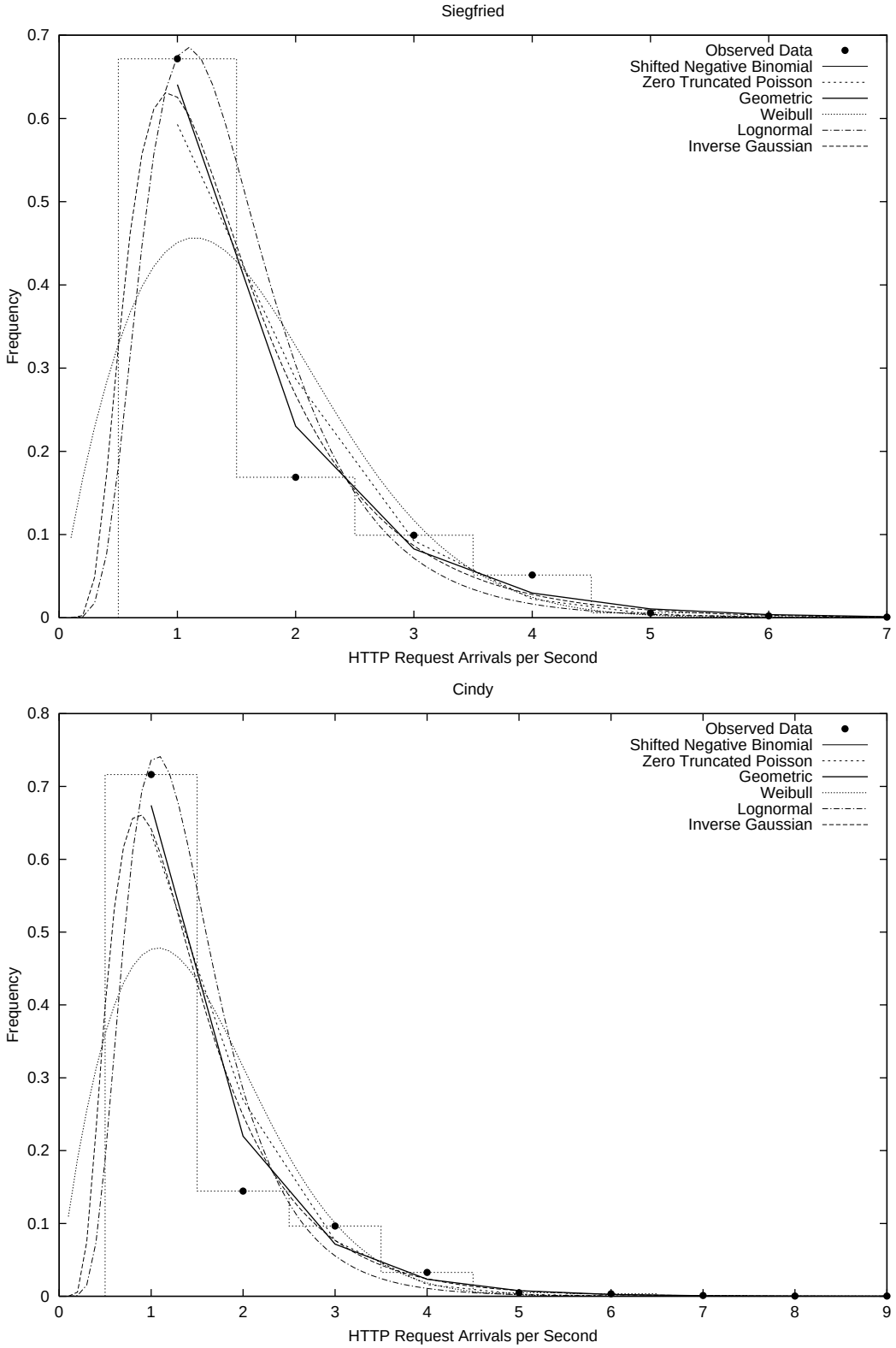


Figure I.3 (Part 5) HTTP Request Rate per Active Second Compared to a Number of Probability Distributions

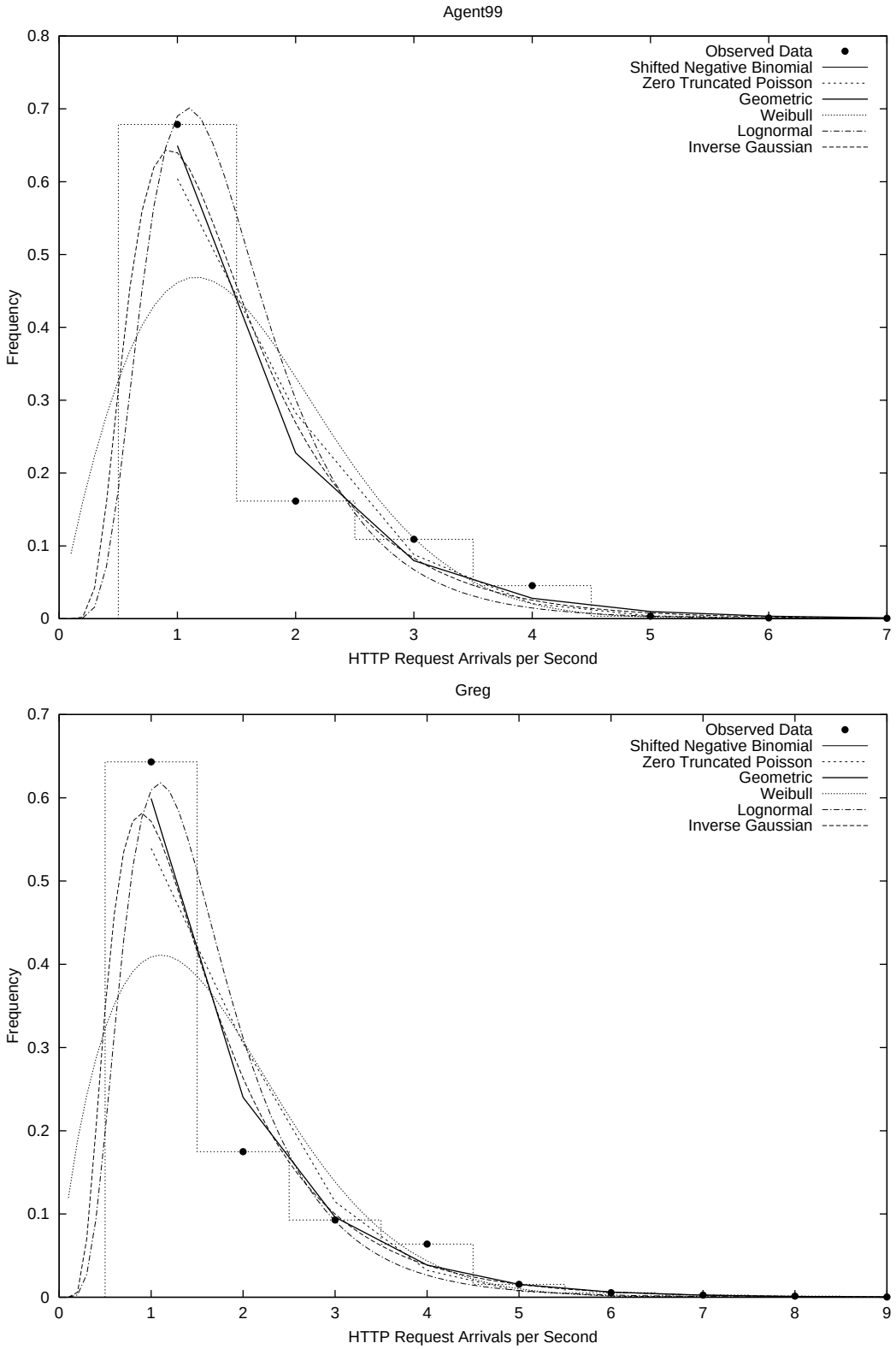


Figure I.3 **(Part 6)** HTTP Request Rate per Active Second Compared to a Number of Probability Distributions

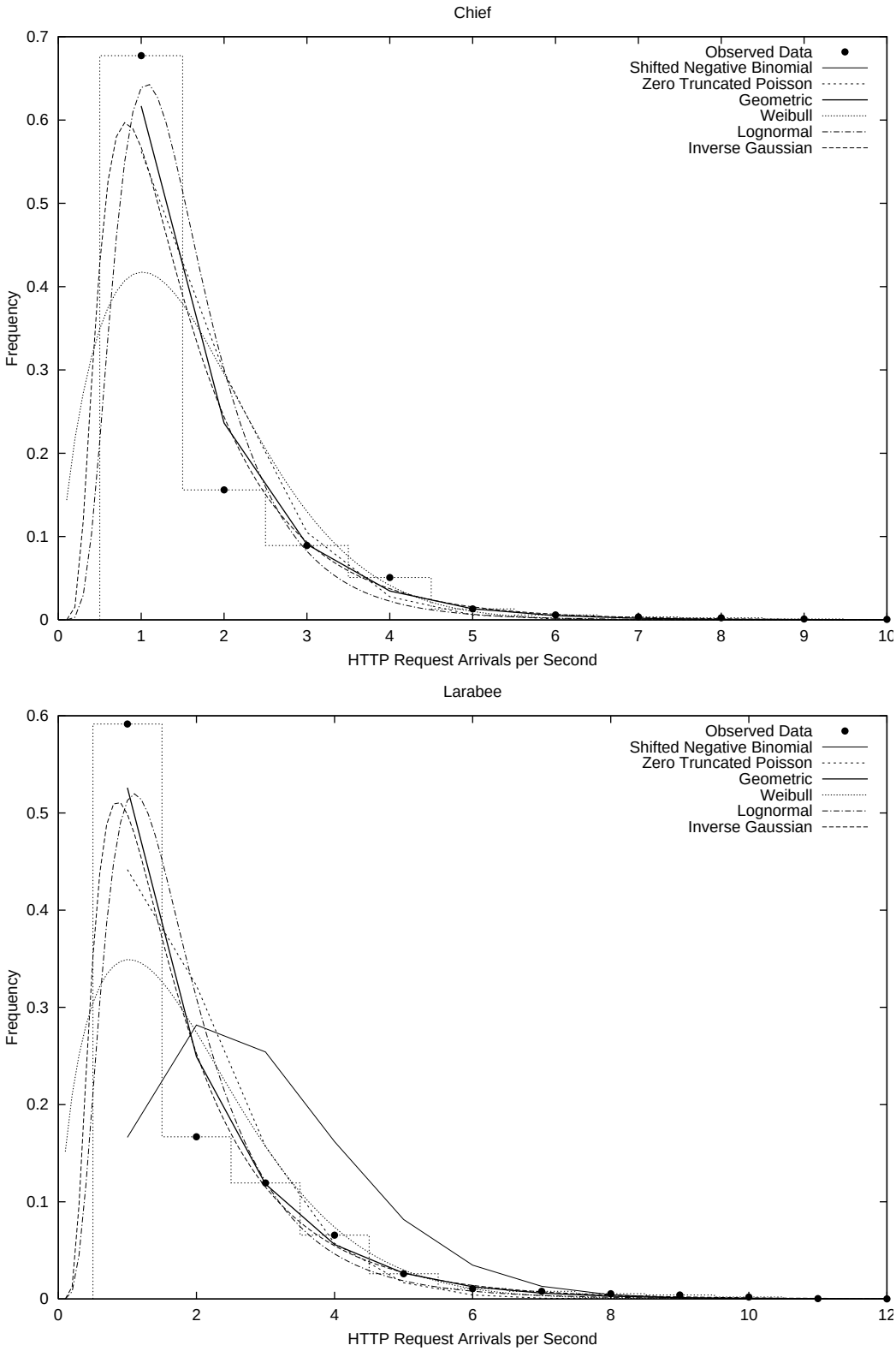


Figure I.3 **(Part 7)** HTTP Request Rate per Active Second Compared to a Number of Probability Distributions

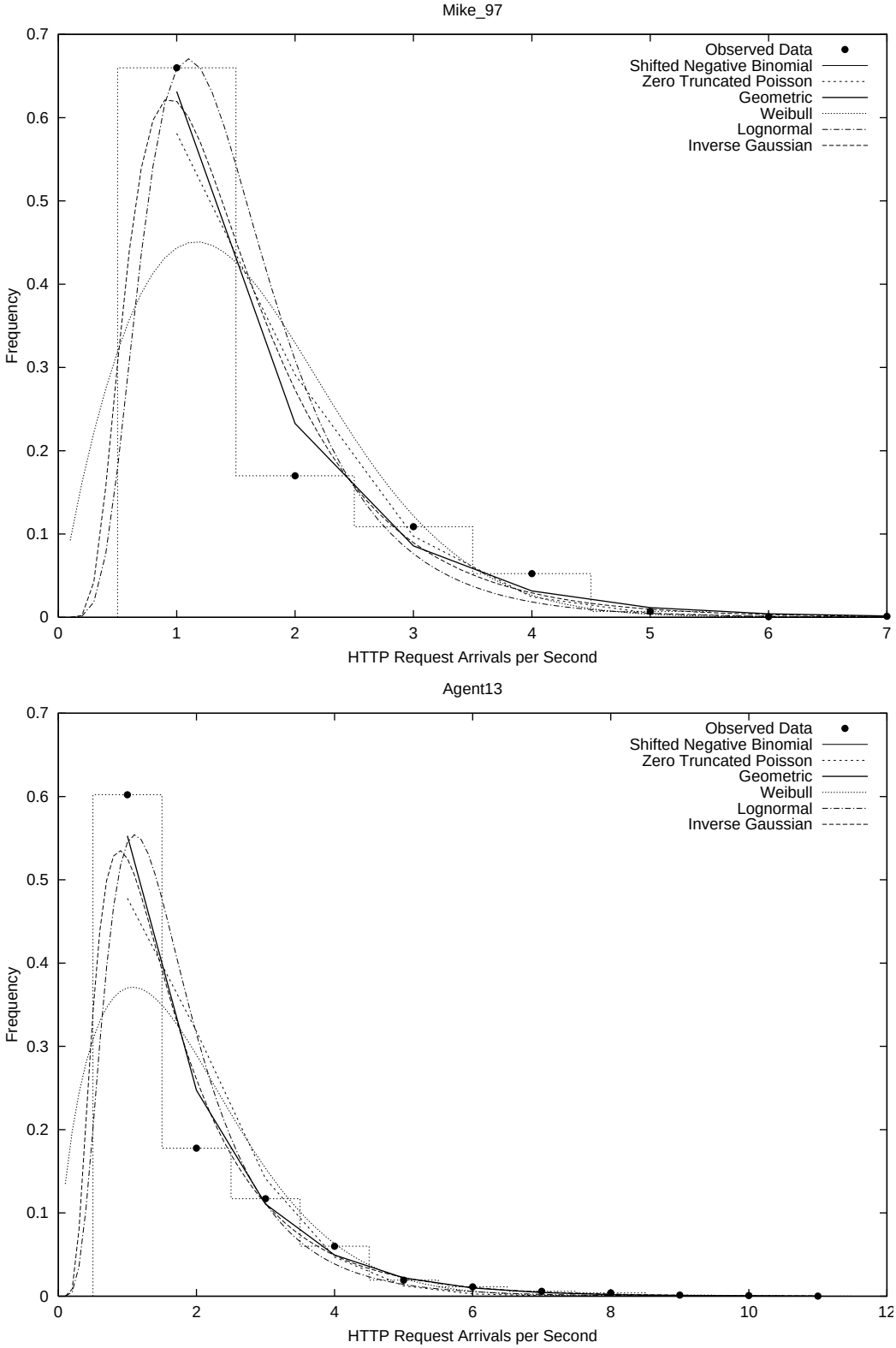


Figure I.3 **(Part 8)** HTTP Request Rate per Active Second Compared to a Number of Probability Distributions

Appendix J. Marginal Distribution of HTTP Request Arrivals Per Second

J.1 Marginal Distribution of Aggregate HTTP Request Rate Compared to the Pólya-Aeppli Distribution

Figures 5.1 and 5.2 compare the histogram of the observed HTTP request rate per second with two or more of the Pólya-Aeppli, normal and Poisson distributions for the sample hours *B6* and *U6*. The following figures show the comparison between the histograms of request rate and the PMF of the Pólya-Aeppli and Poisson distributions and the PDF of the normal distribution for each of the sample hours of Web traffic listed in Table 2.2.

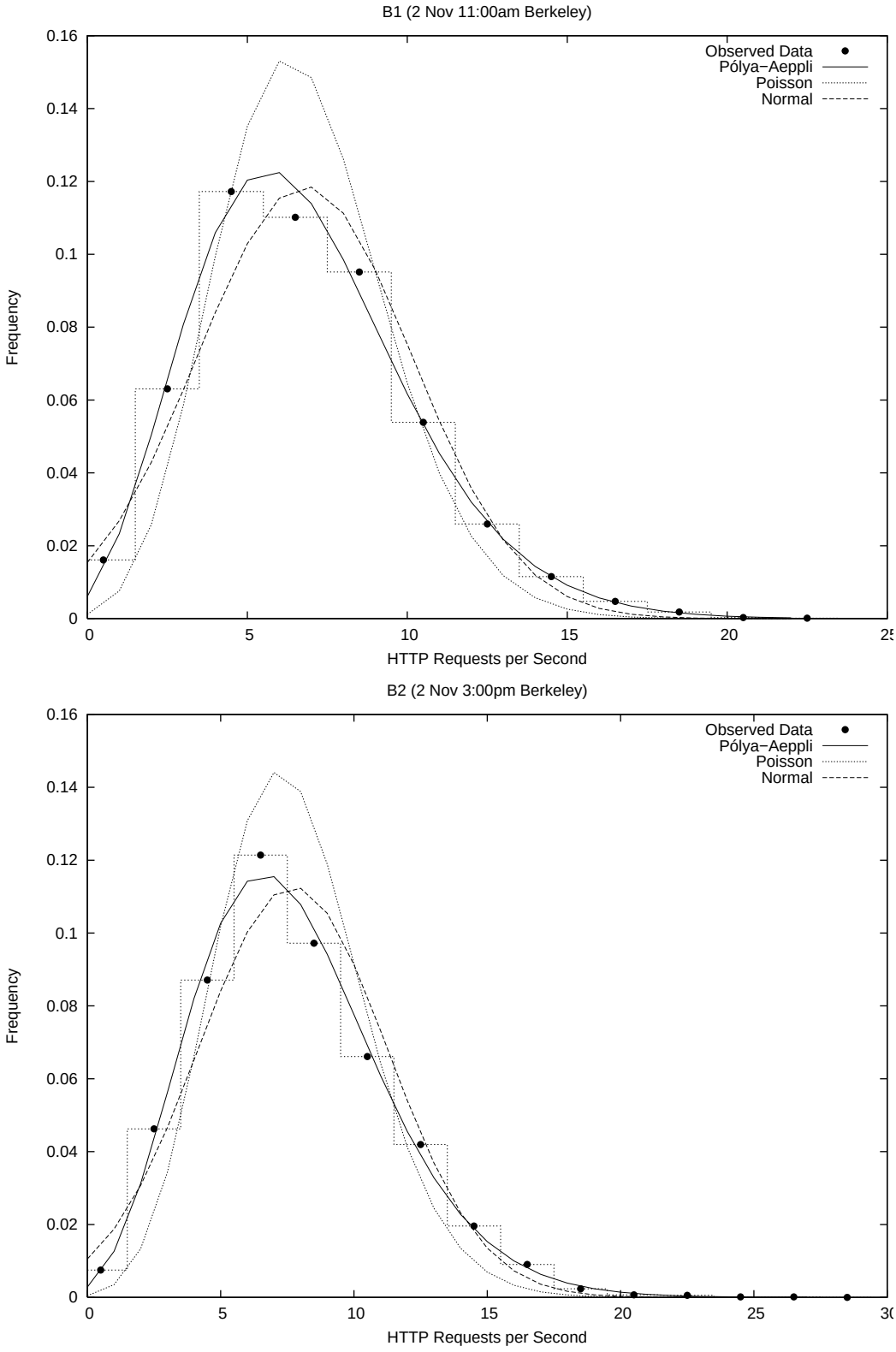


Figure J.1 (Part 1) Histogram of HTTP Request Arrival Per Second Compared to Pólya-Aeppli, Normal and Poisson Distributions

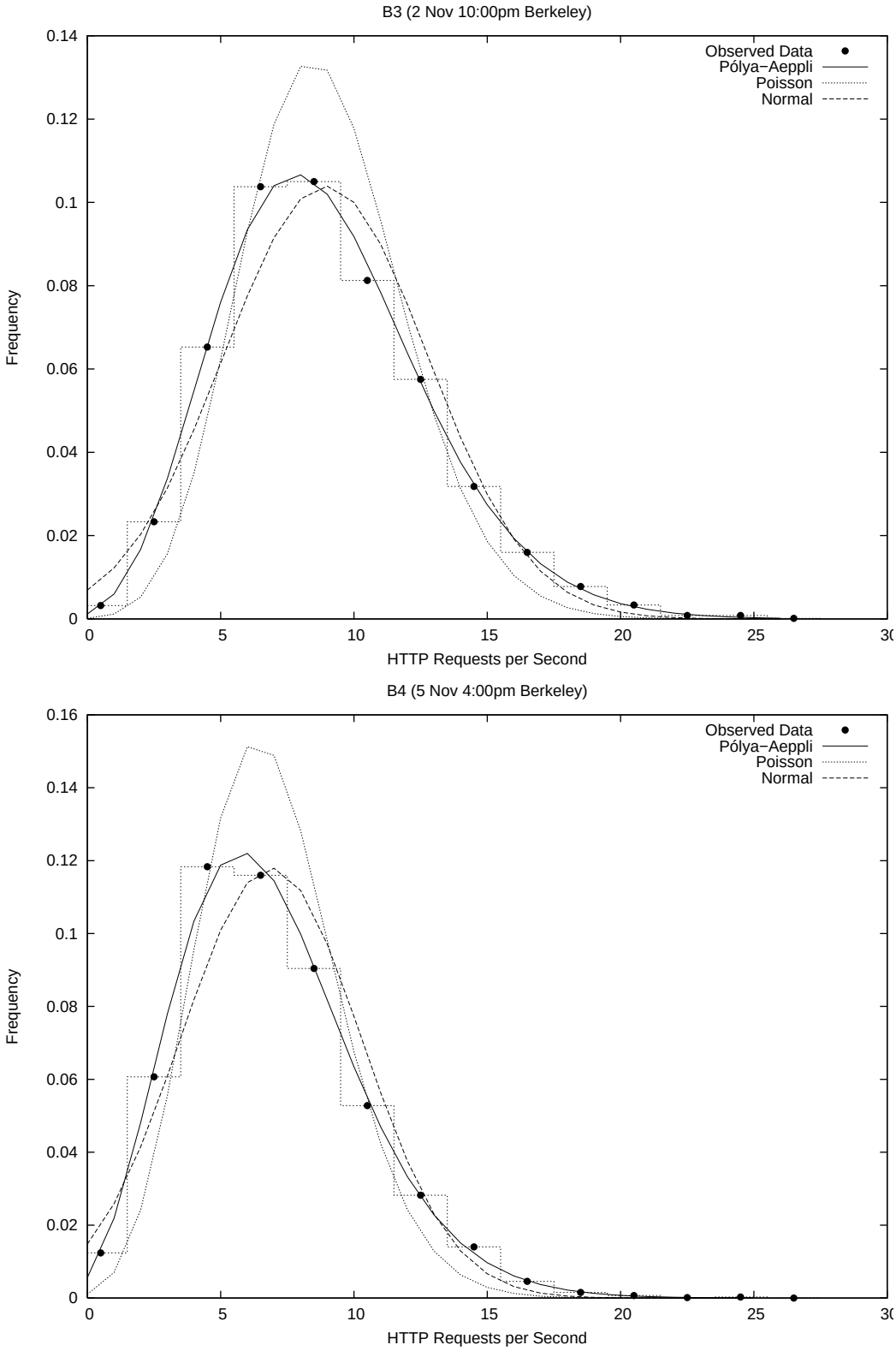


Figure J.1 (Part 2) Histogram HTTP Request Arrival Per Second Compared to Pólya-Aeppli, Normal and Poisson Distributions

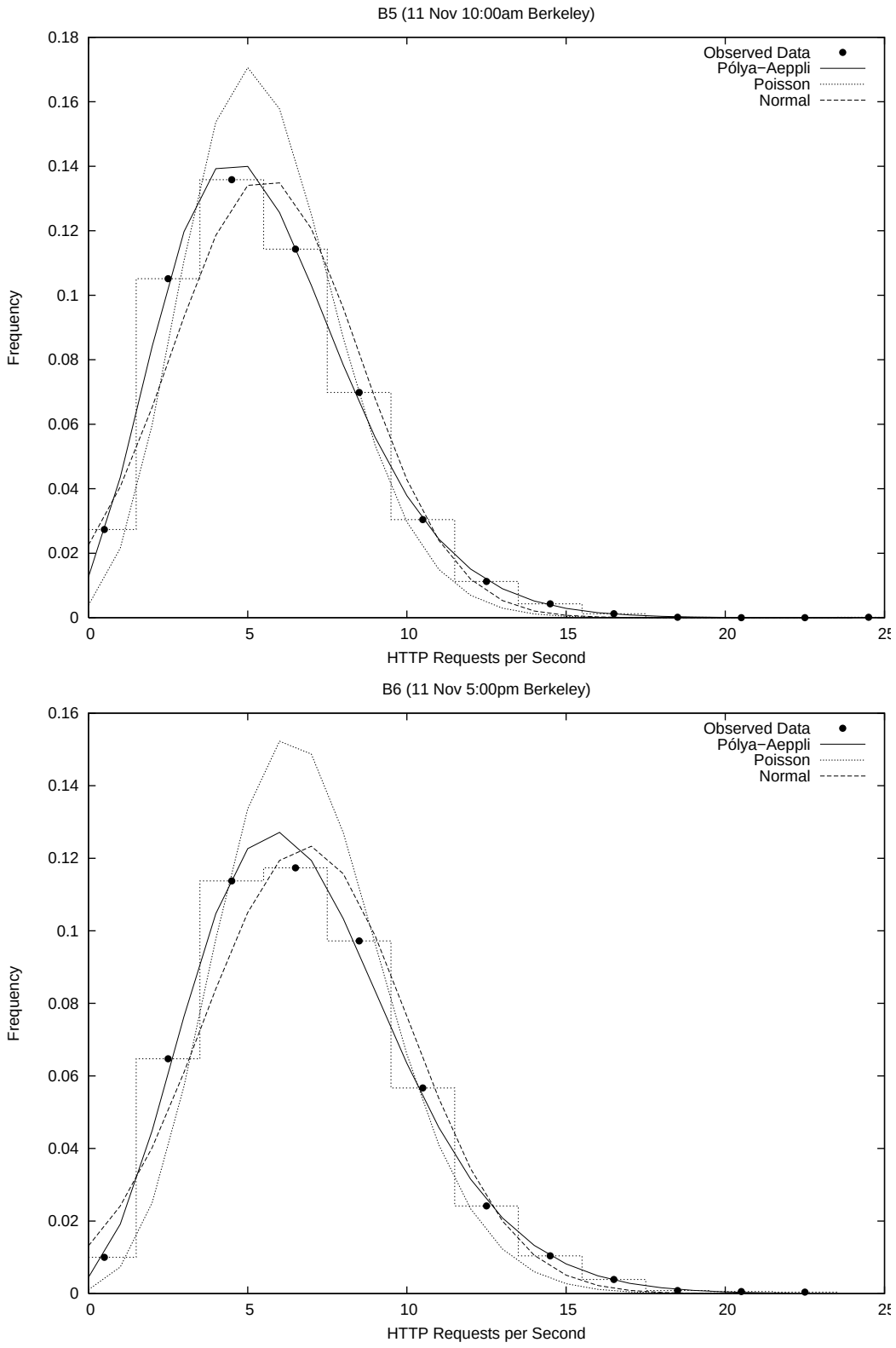


Figure J.1 (Part 3) Histogram of HTTP Request Arrival Per Second Compared to Pólya-Aeppli, Normal and Poisson Distributions

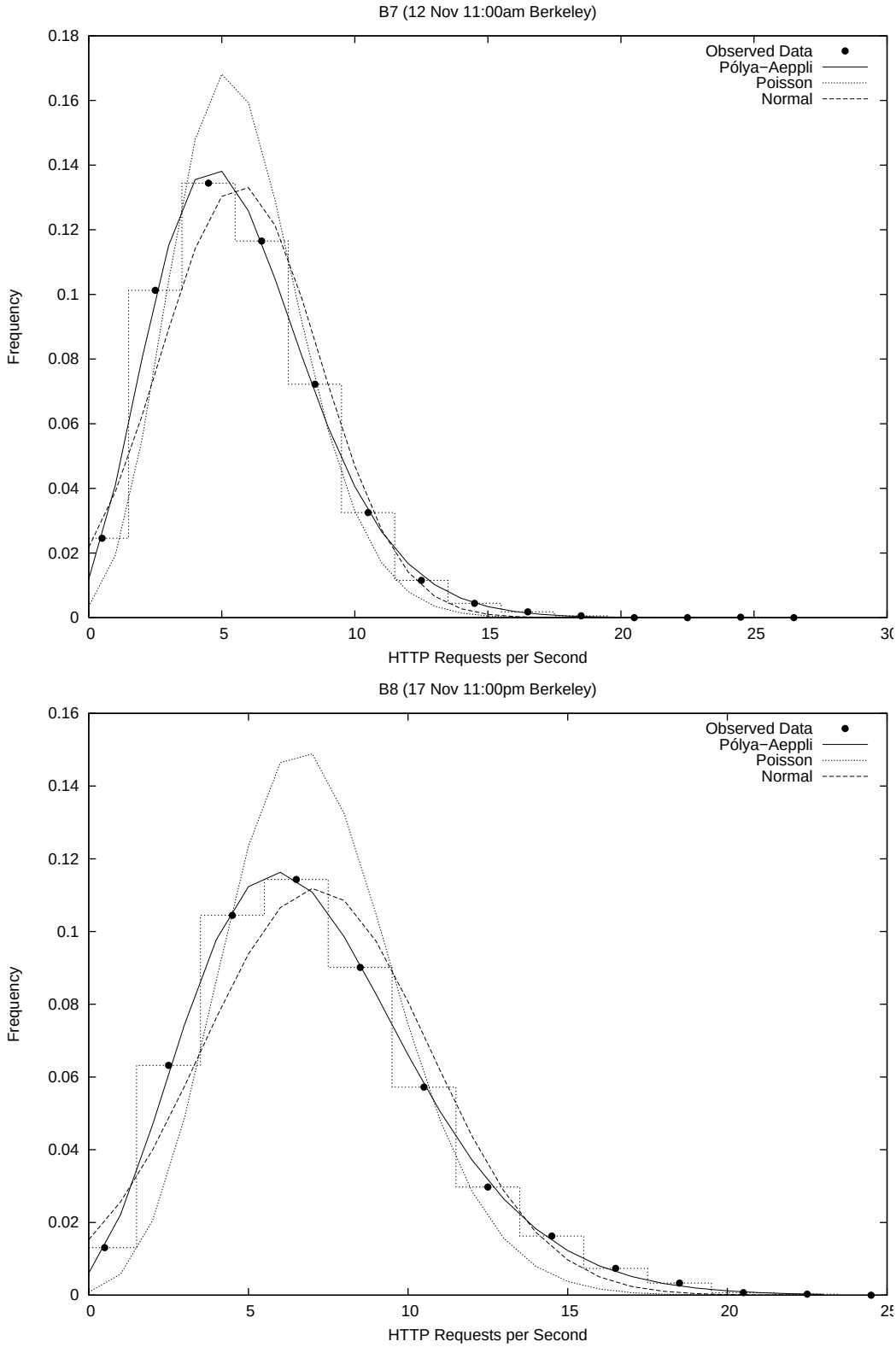


Figure J.1 (Part 4) Histogram of HTTP Request Arrivals Per Second Compared to Pólya-Aeppli, Normal and Poisson Distributions

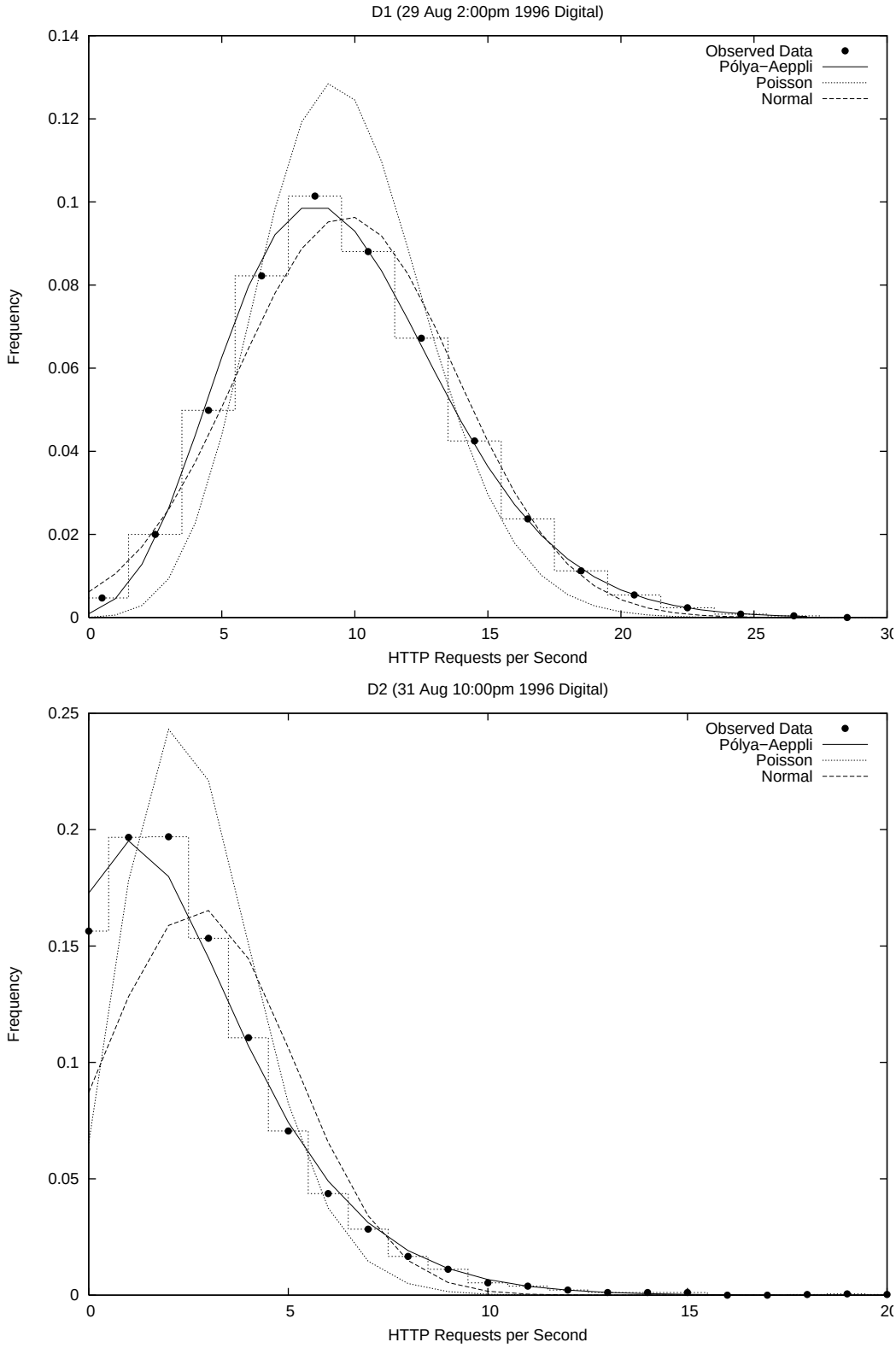


Figure J.1 (Part 5) Histogram of HTTP Request Arrivals Per Second Compared to Pólya-Aeppli, Normal and Poisson Distributions

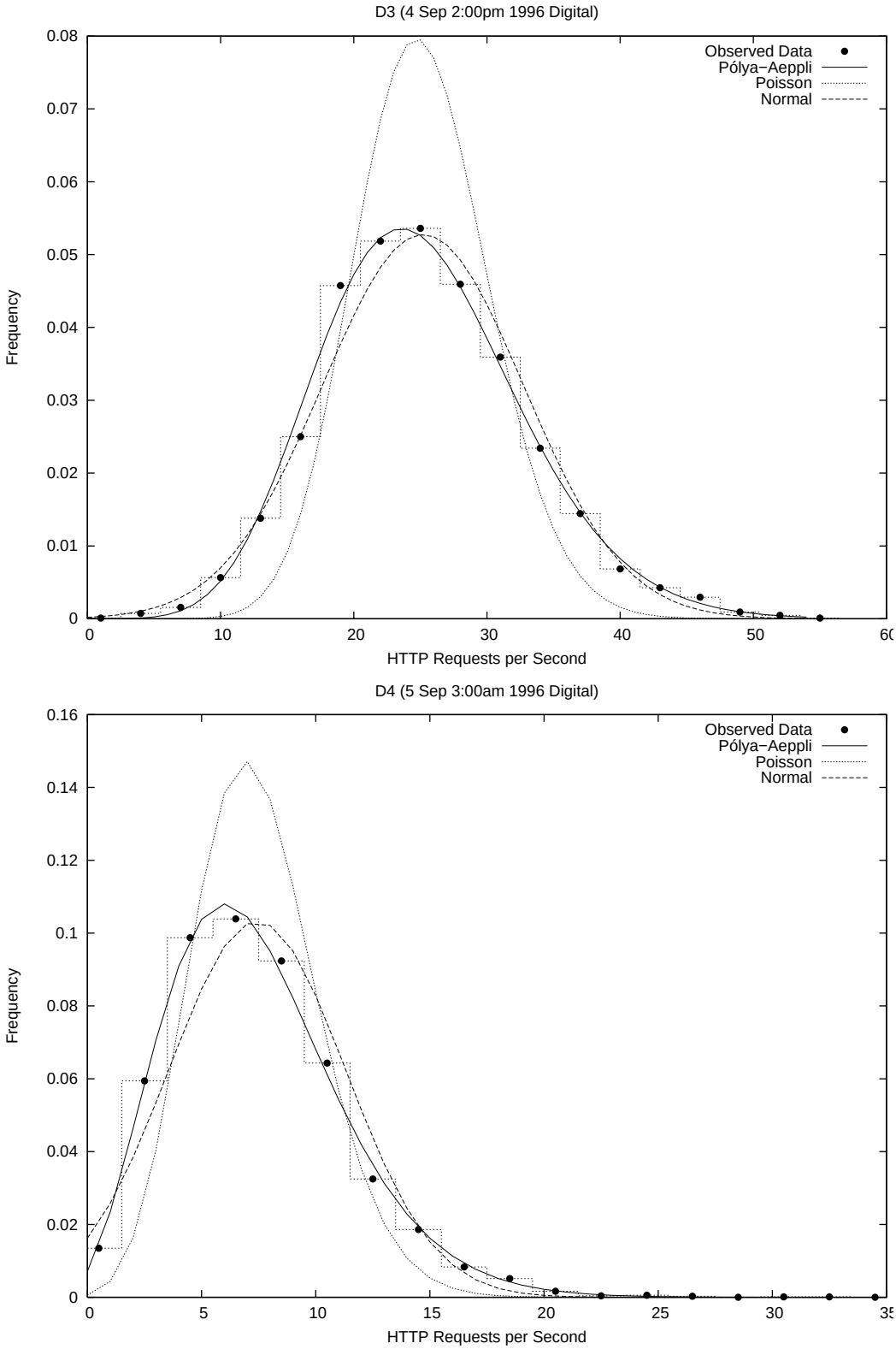


Figure J.1 (Part 6) Histogram of HTTP Request Arrival Per Second Compared to Pólya-Aeppli, Normal and Poisson Distributions

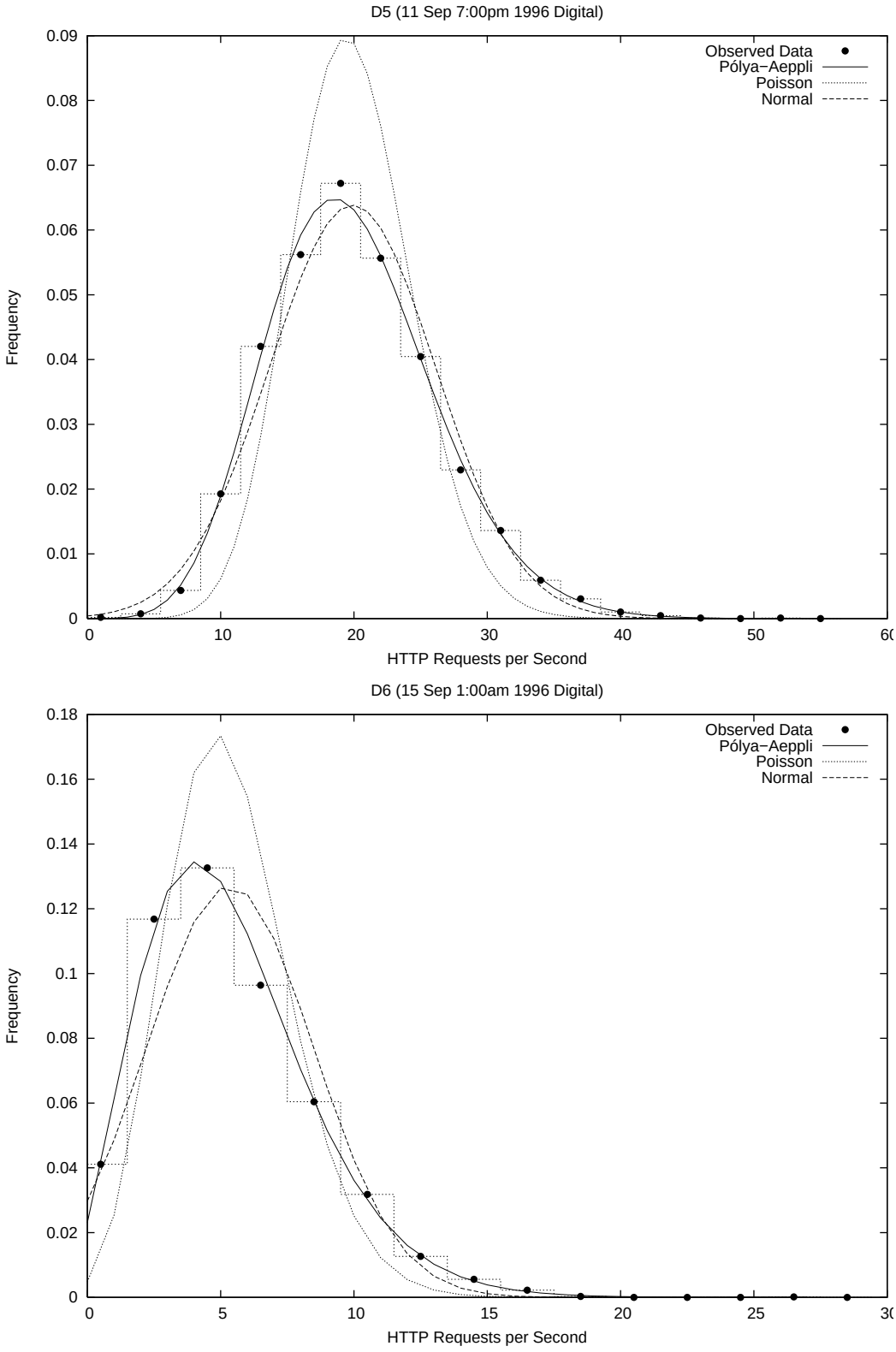


Figure J.1 (Part 7) Histogram of HTTP Request Arrival Per Second Compared to Pólya-Aeppli, Normal and Poisson Distributions

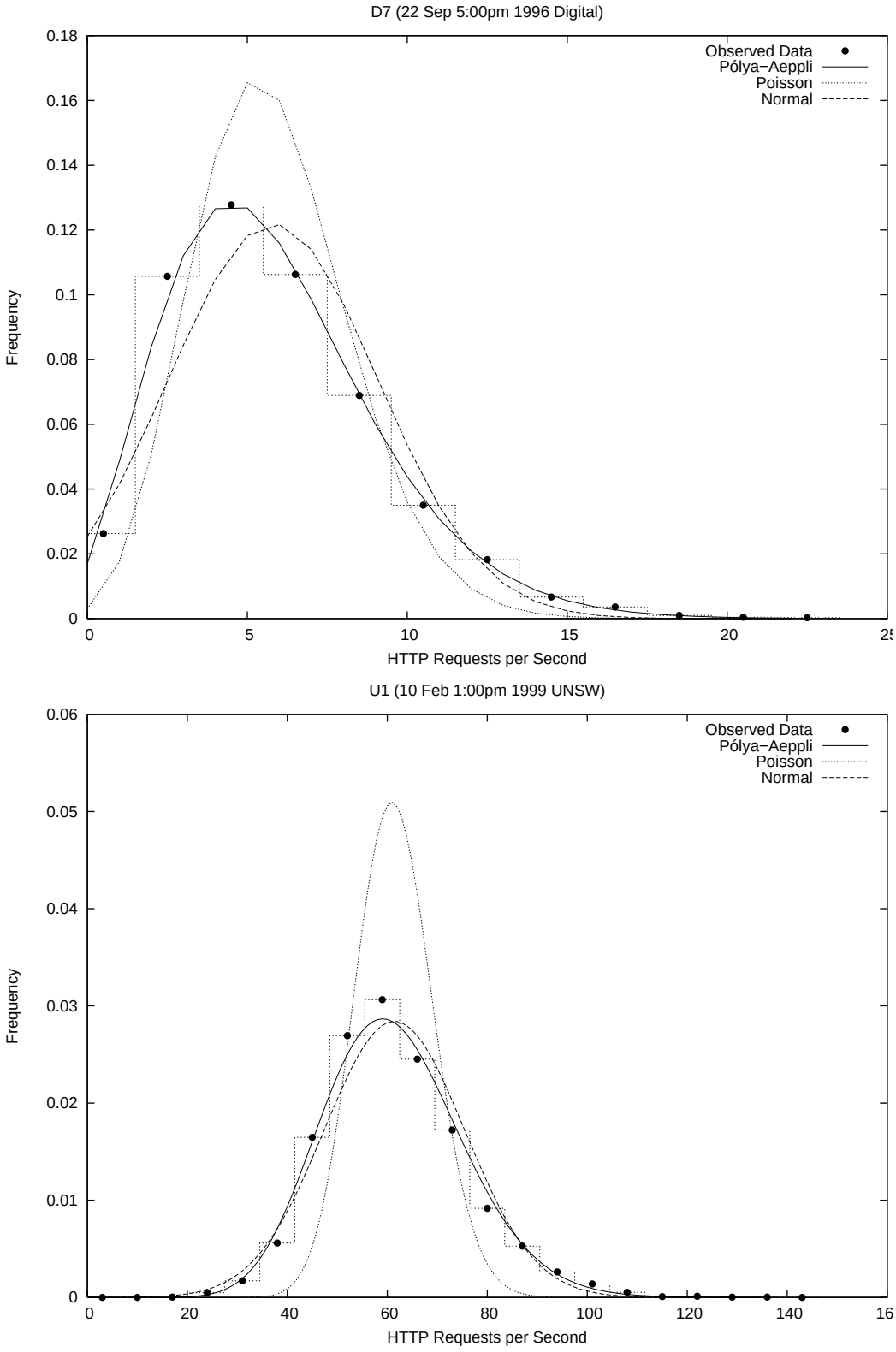


Figure J.1 (Part 8) Histogram of HTTP Request Arrival Per Second Compared to Pólya-Aeppli, Normal and Poisson Distributions

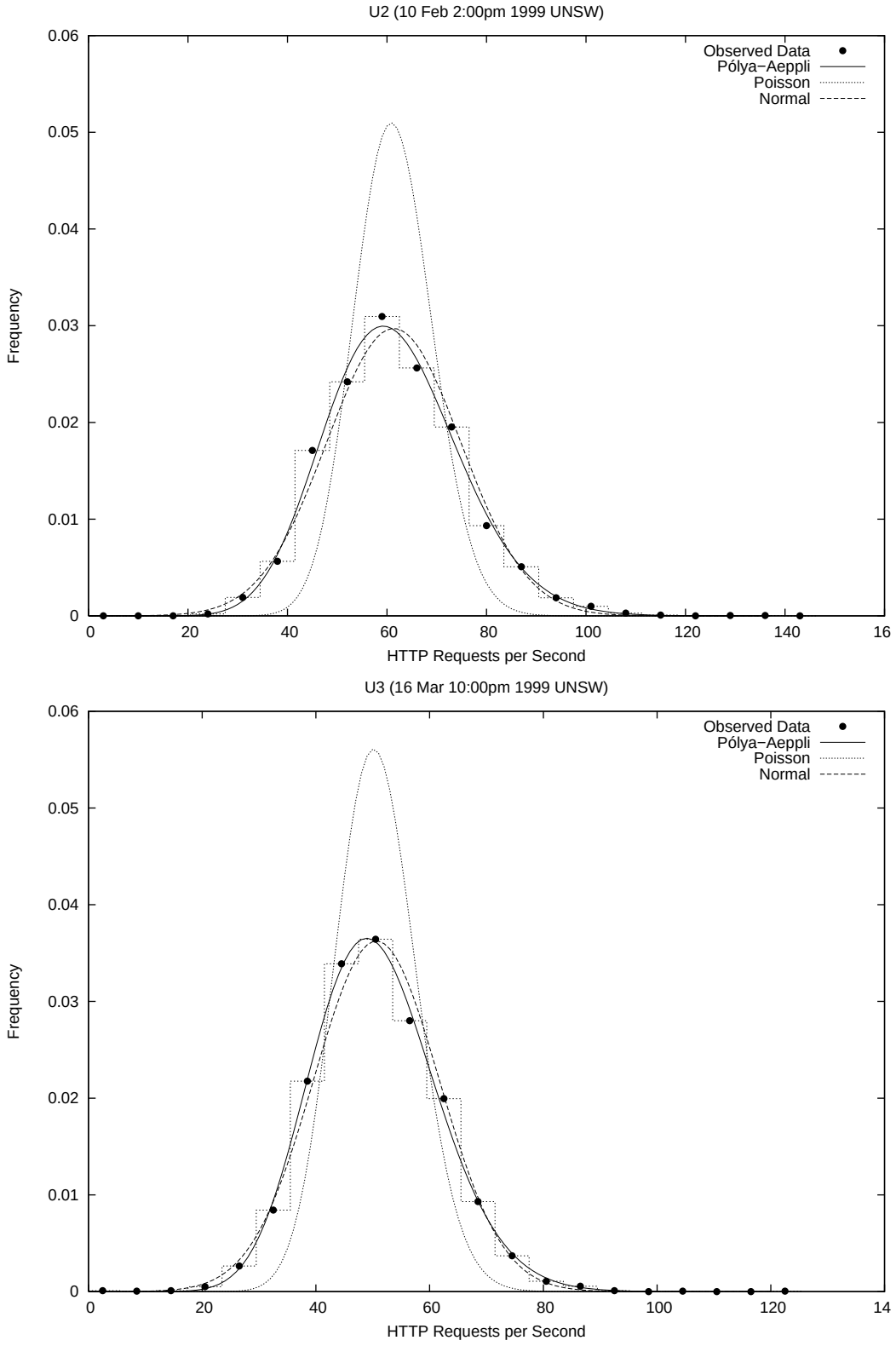


Figure J.1 (Part 9) Histogram of HTTP Request Arrival Per Second Compared to Pólya-Aeppli, Normal and Poisson Distributions

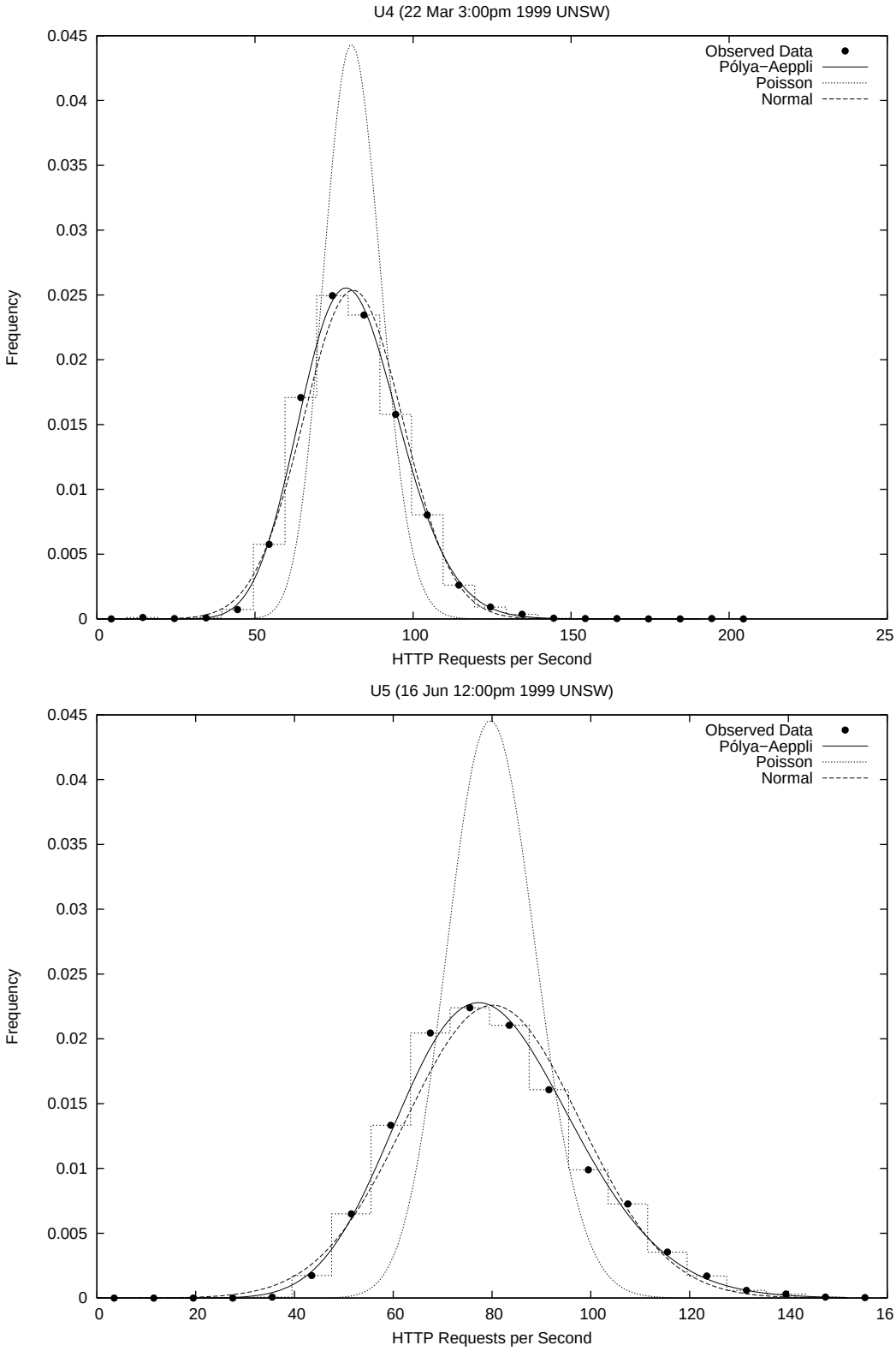


Figure J.1 (Part 10) Histogram of HTTP Request Arrival Per Second Compared to Pólya-Aeppli, Normal and Poisson Distributions

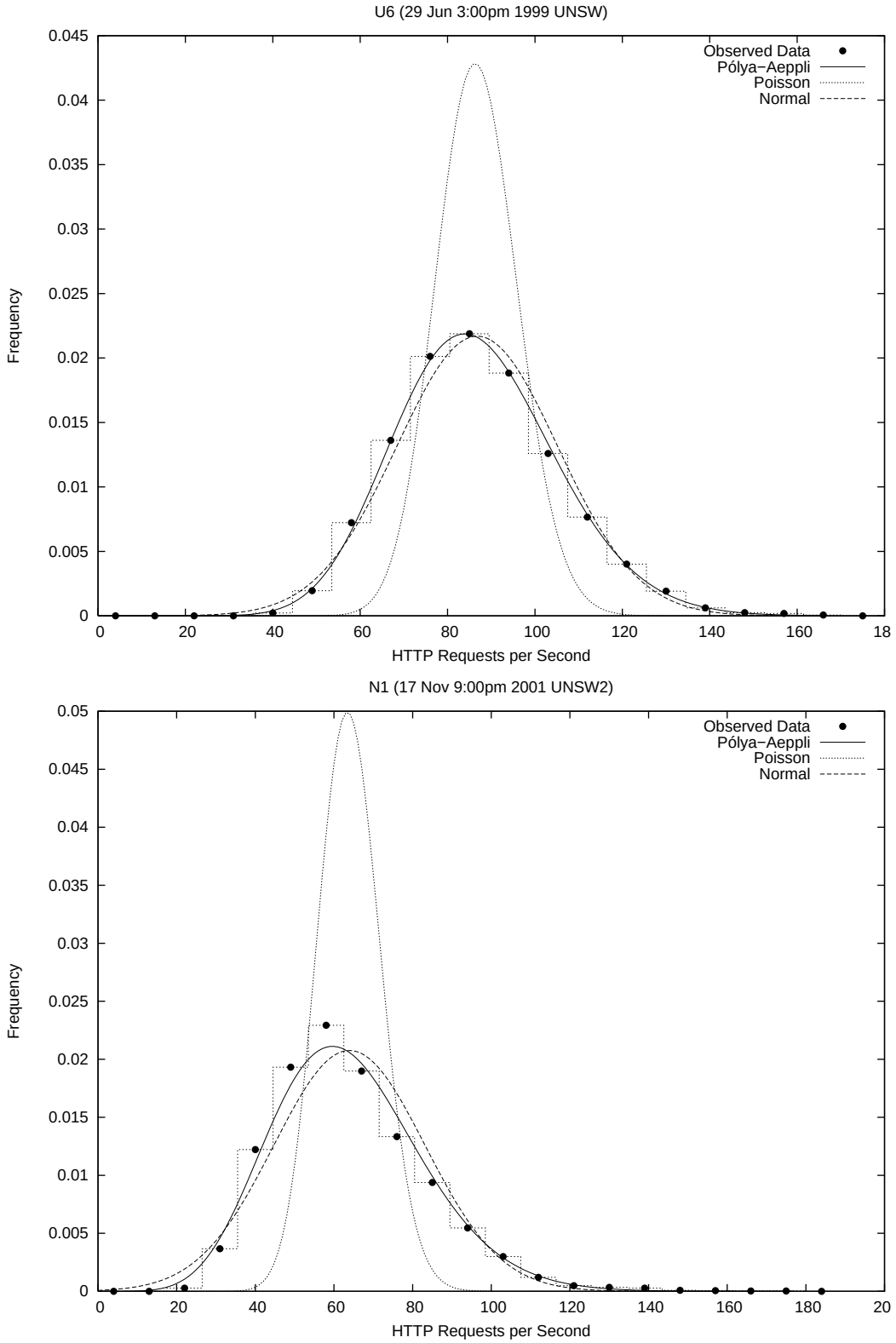


Figure J.1 (Part 11) Histogram of HTTP Request Arrival Per Second Compared to Pólya-Aeppli, Normal and Poisson Distributions

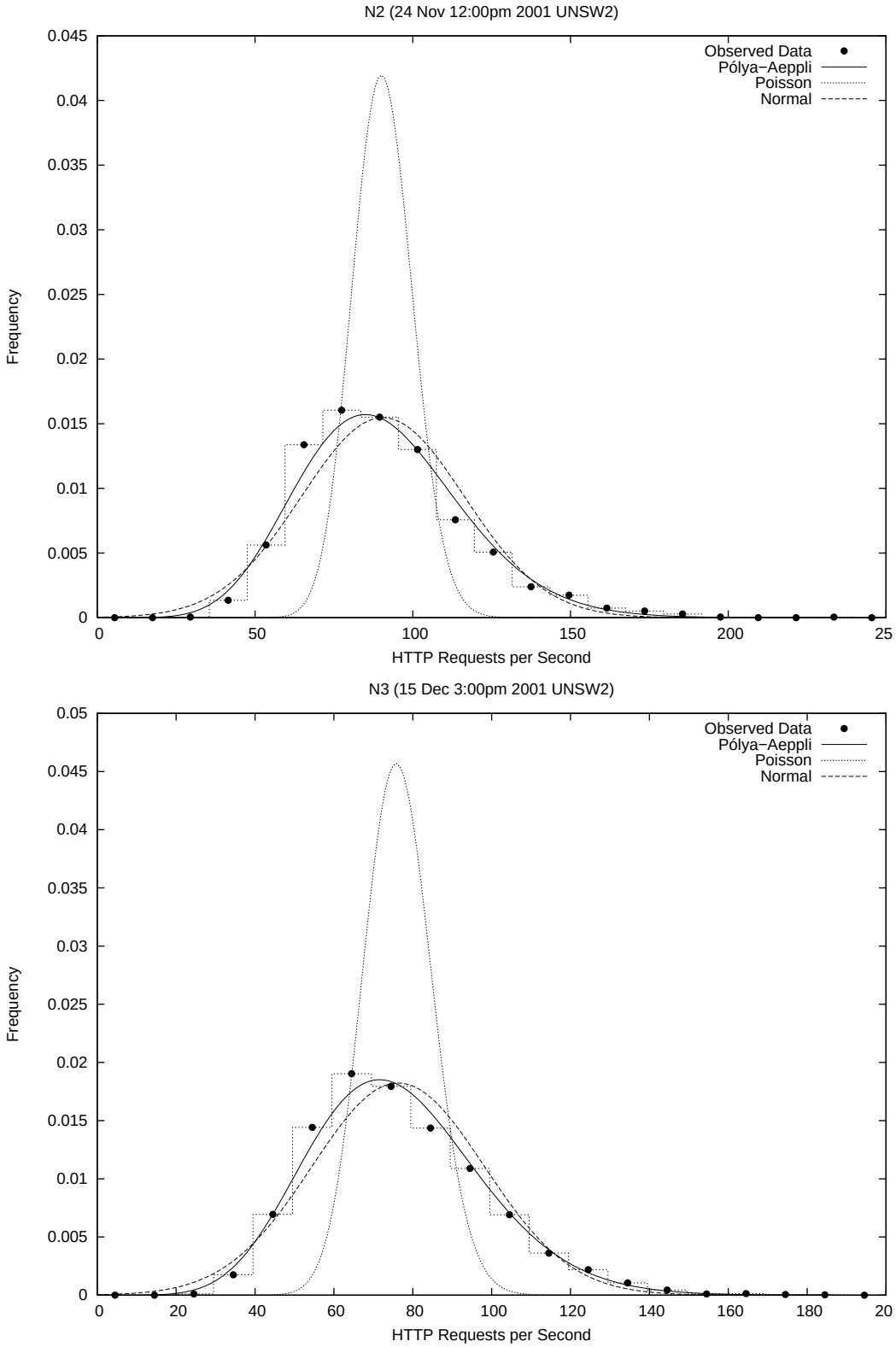


Figure J.1 (Part 12) Histogram of HTTP Request Arrival Per Second Compared to Pólya-Aeppli, Normal and Poisson Distributions

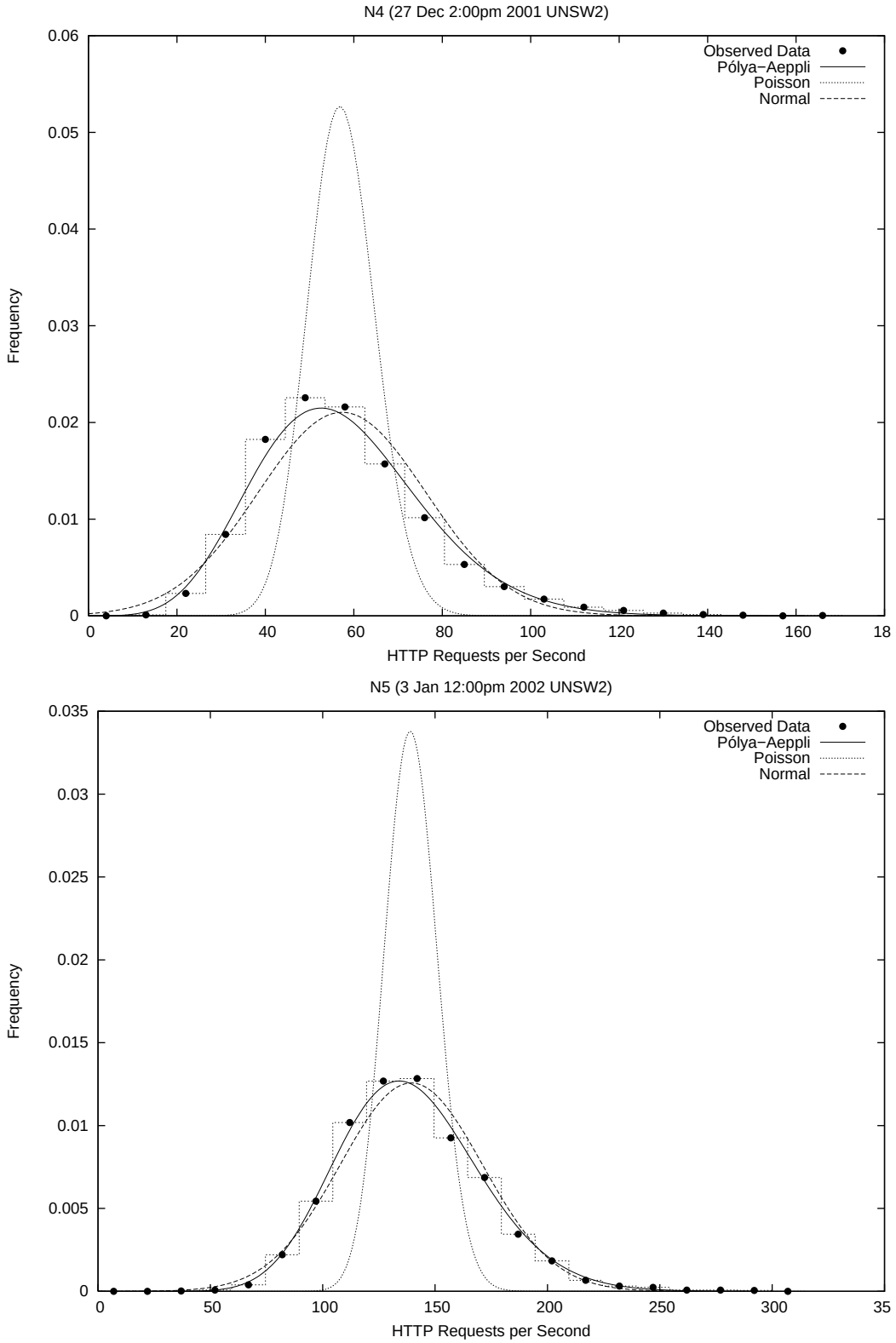


Figure J.1 (Part 13) Histogram of HTTP Request Arrival Per Second Compared to Pólya-Aeppli, Normal and Poisson Distributions

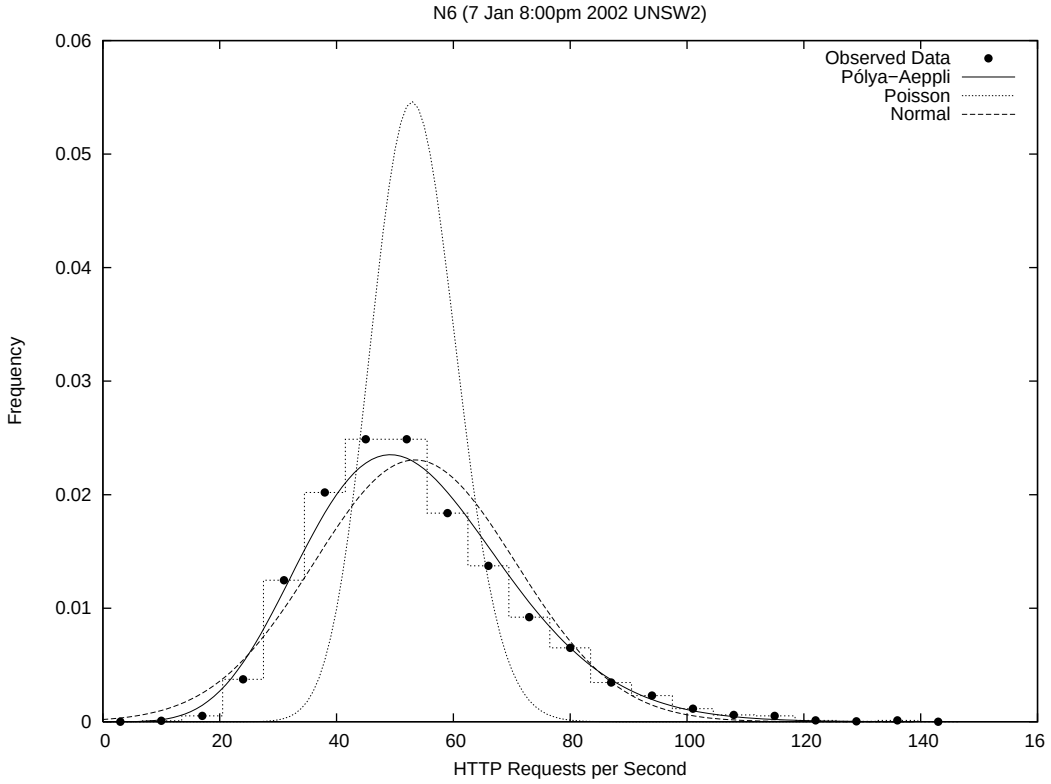


Figure J.1 (Part 14) Histogram of HTTP Request Arrival Per Second Compared to Pólya-Aeppli, Normal and Poisson Distributions

J.2 PP Plot Comparison of HTTP Request Rate with the Pólya-Aeppli Probability Distribution

Figure 5.3 is a PP plot comparison between the observed HTTP request rate with the Pólya-Aeppli, normal and Poisson distributions for the sample hours *B6* and *U6*. The following figure shows the same comparison plot for each of the sample hours of Web traffic listed in Table 2.2. A dotted sloping line is shown on each of the graphs indicating the where the best fit would be plotted. In many cases this is overlapped by the fit of the Pólya-Aeppli distribution.

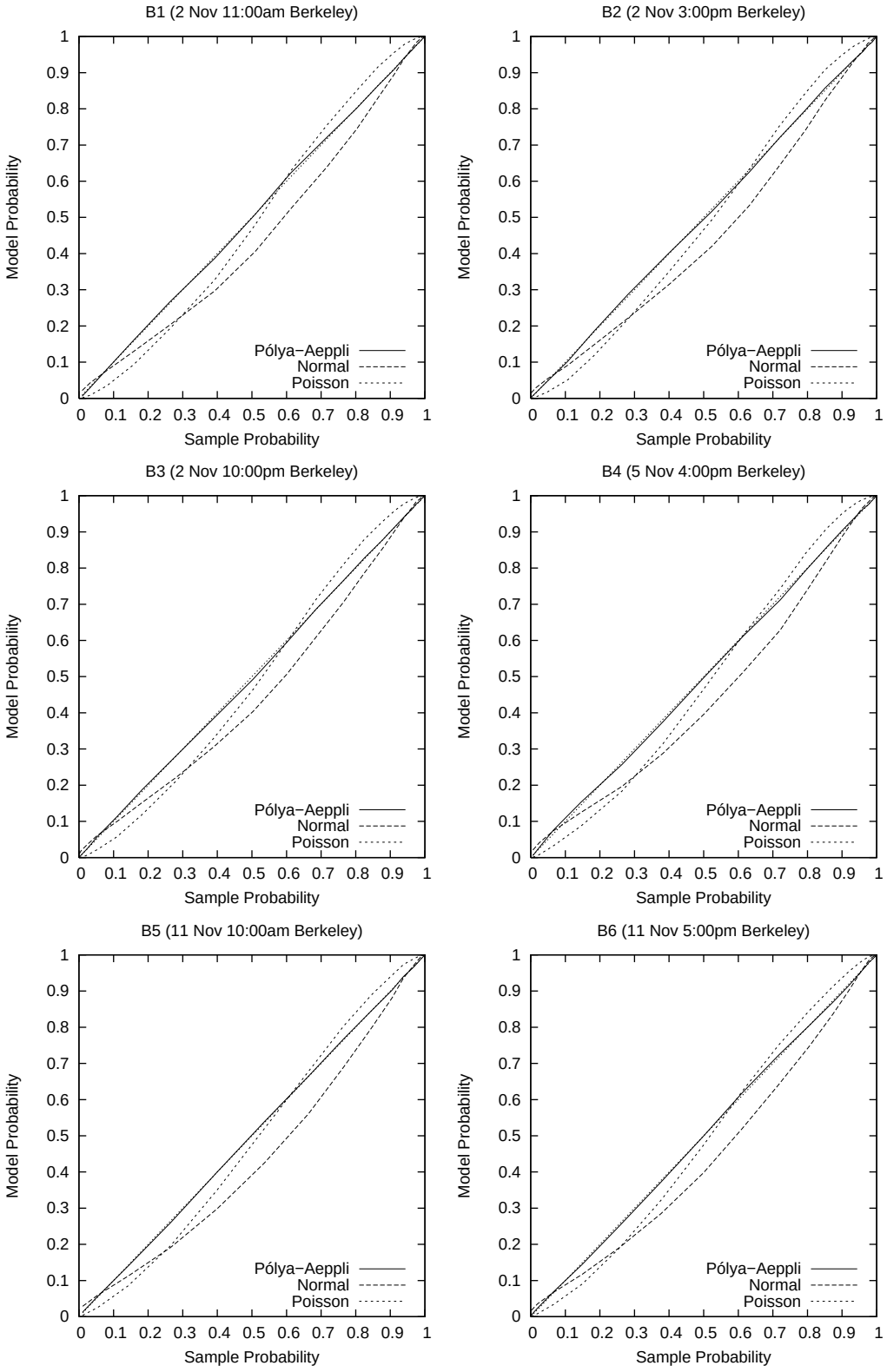


Figure J.2 **(Part 1)** PP Plot of HTTP Request Arrival Per Second Compared to Pólya-Aeppli, Normal and Poisson Distributions

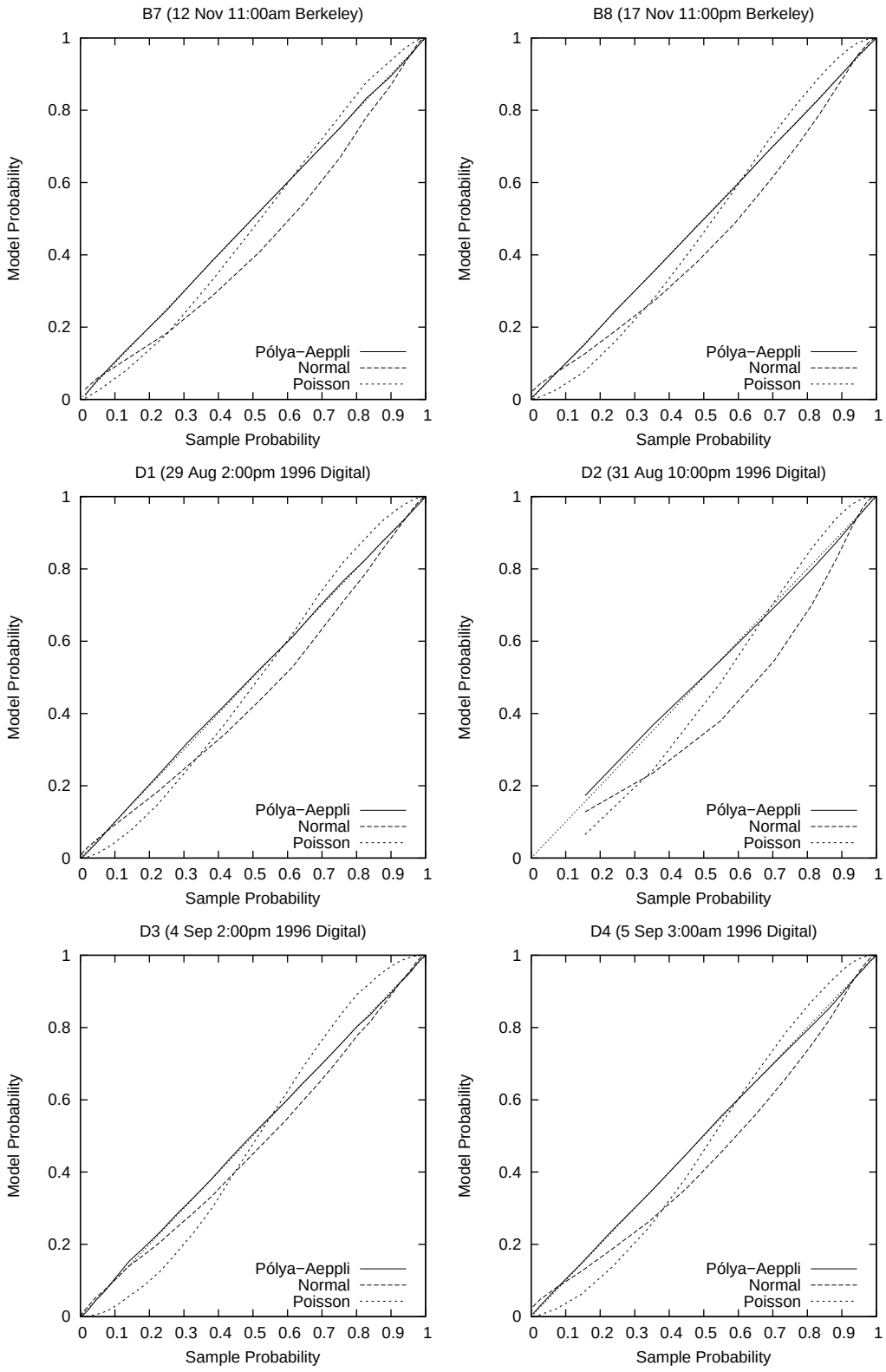


Figure J.2 **(Part 2)** PP Plot of HTTP Request Arrival Per Second Compared to Pólya-Aeppli, Normal and Poisson Distributions

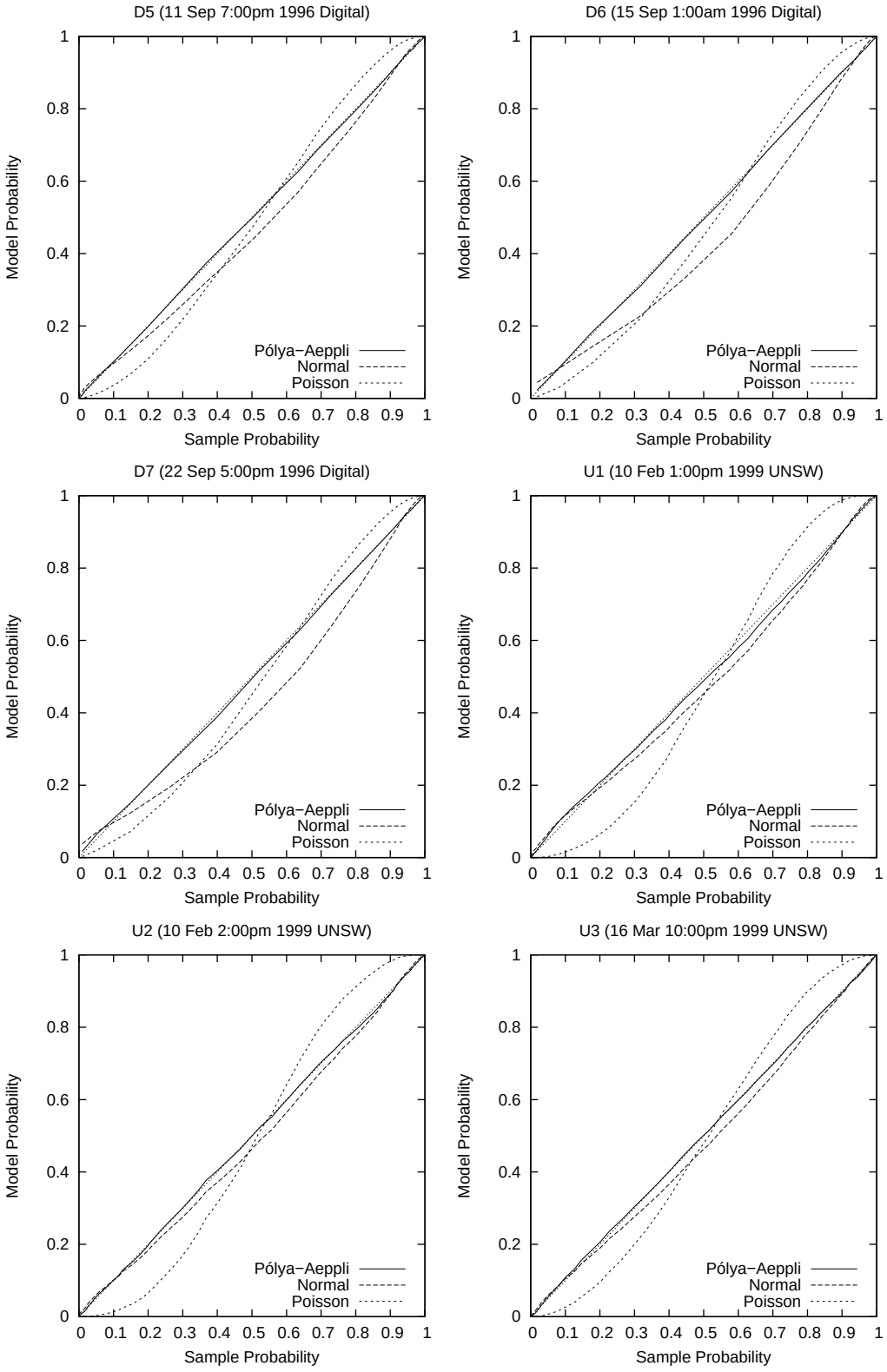


Figure J.2 (Part 3) PP Plot of HTTP Request Arrival Per Second Compared to Pólya-Aeppli, Normal and Poisson Distributions

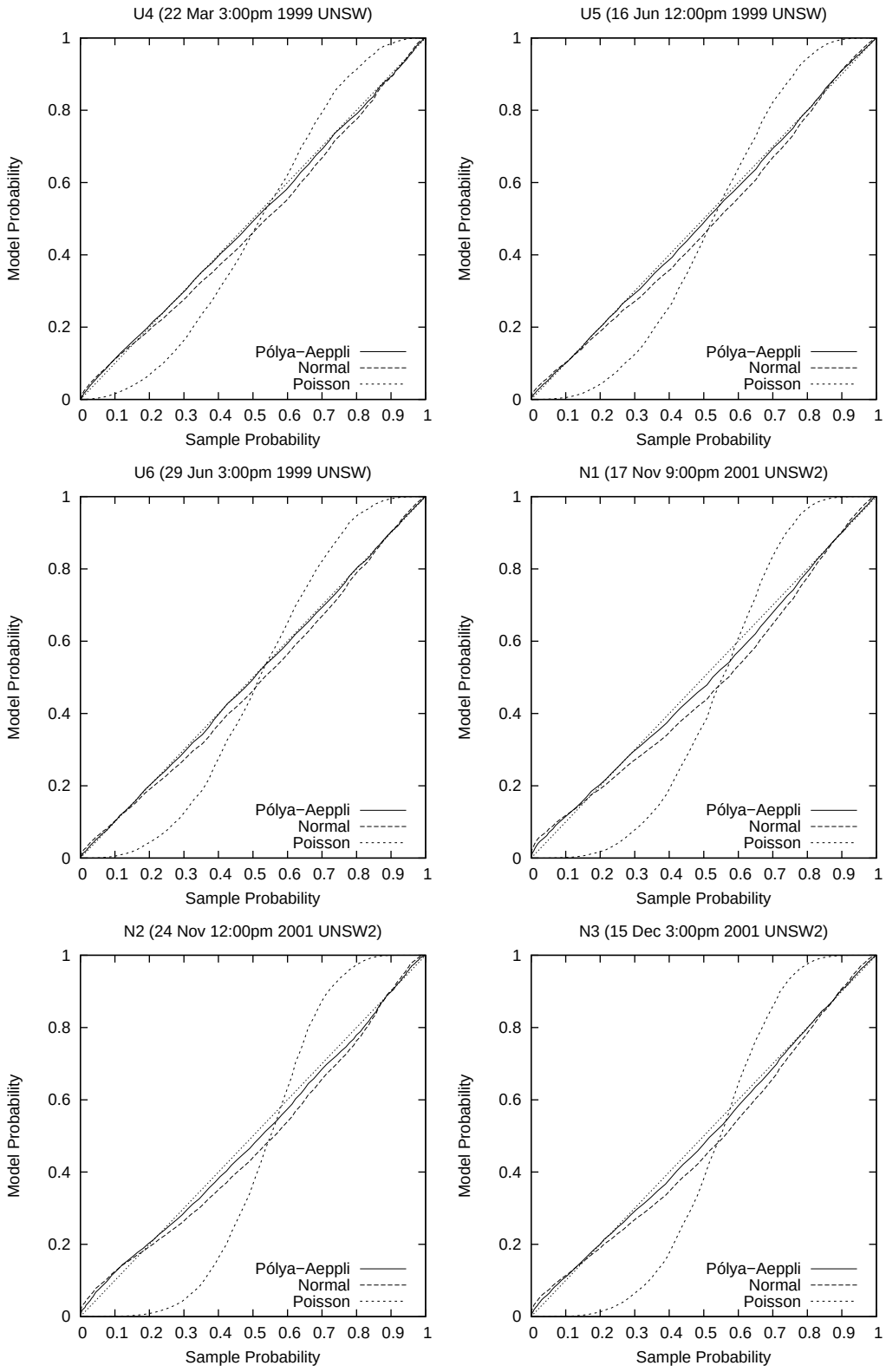


Figure J.2 **(Part 4)** PP Plot of HTTP Request Arrival Per Second Compared to Pólya-Aeppli, Normal and Poisson Distributions

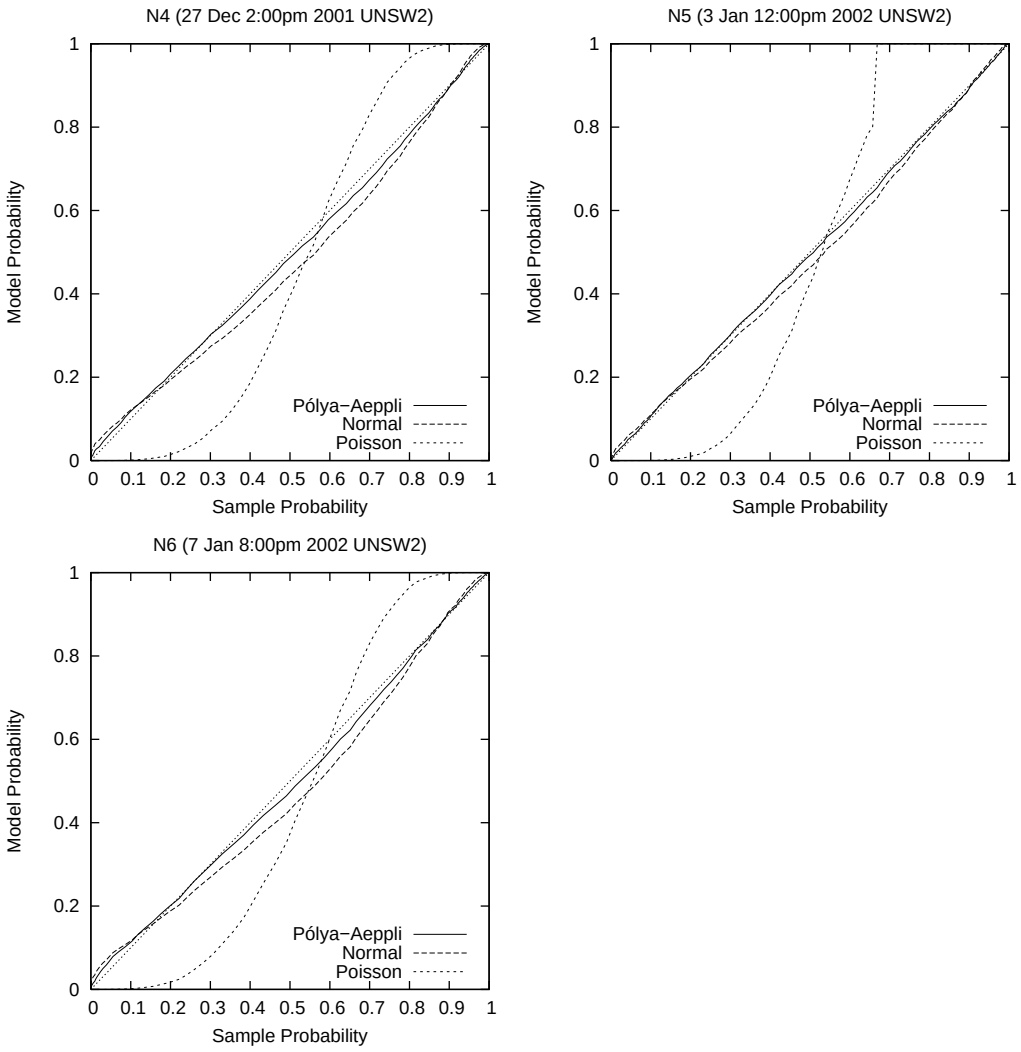


Figure J.2 **(Part 5)** PP Plot of HTTP Request Arrival Per Second Compared to Pólya-Aeppli, Normal and Poisson Distributions

J.3 QQ Plot Comparison of HTTP Request Rate with the Pólya-Aeppli Probability Distribution

Figure 5.4 is a QQ plot comparison between the observed HTTP request rate with the Pólya-Aeppli, normal and Poisson distributions for the sample hours *B6* and *U6*. The following figure shows the same comparison plot for each of the sample hours of Web traffic listed in Table 2.2. The horizontal lines on the plots indicate the 0.01, 0.05, 0.95 and 0.99 quantiles. A dotted sloping line is shown on each of the graphs indicating the where the best fit would be plotted.

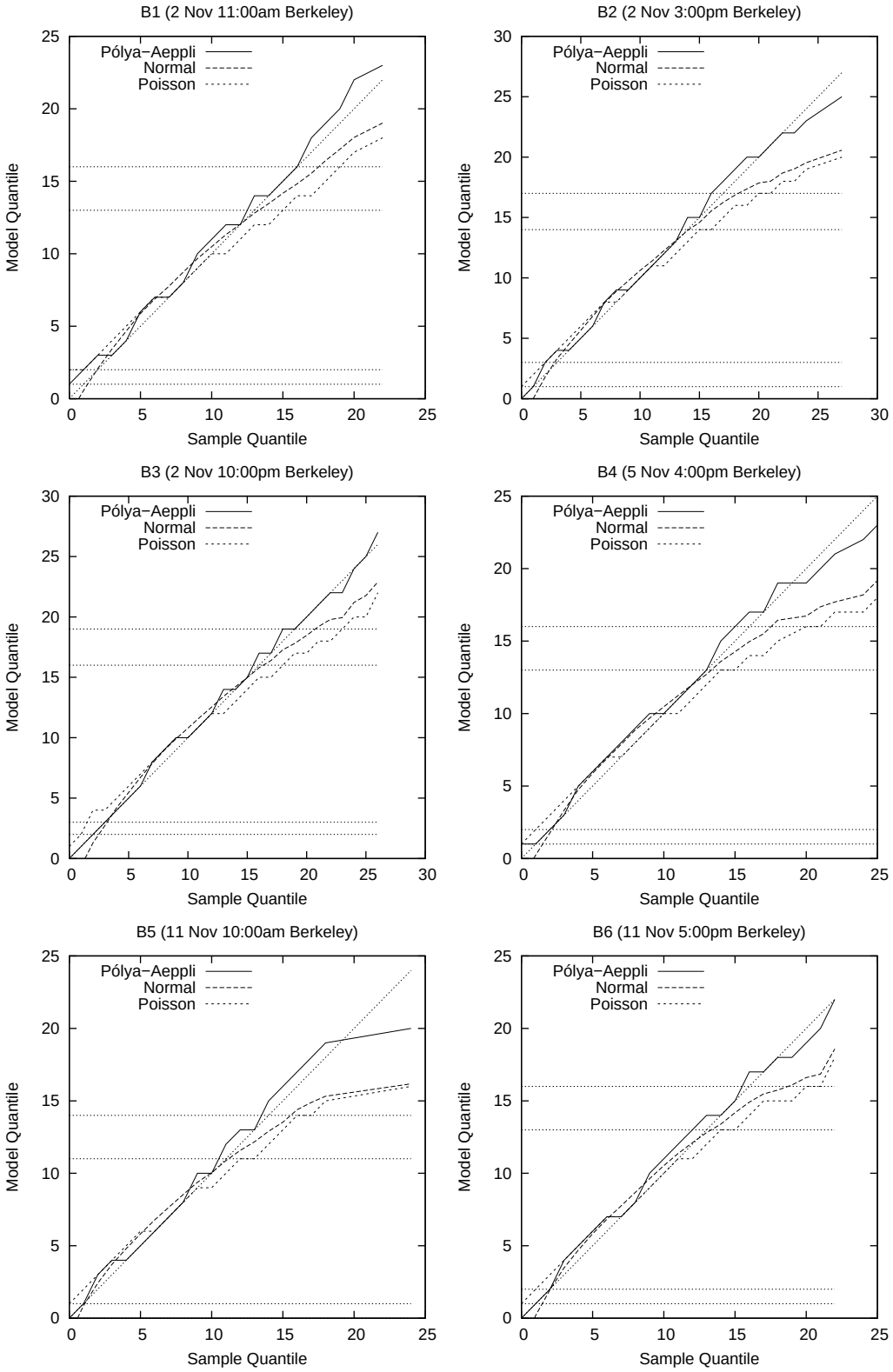


Figure J.3 **(Part 1)** QQ Plot of HTTP Request Arrival Per Second Compared to Pólya-Aeppli, Normal and Poisson Distributions

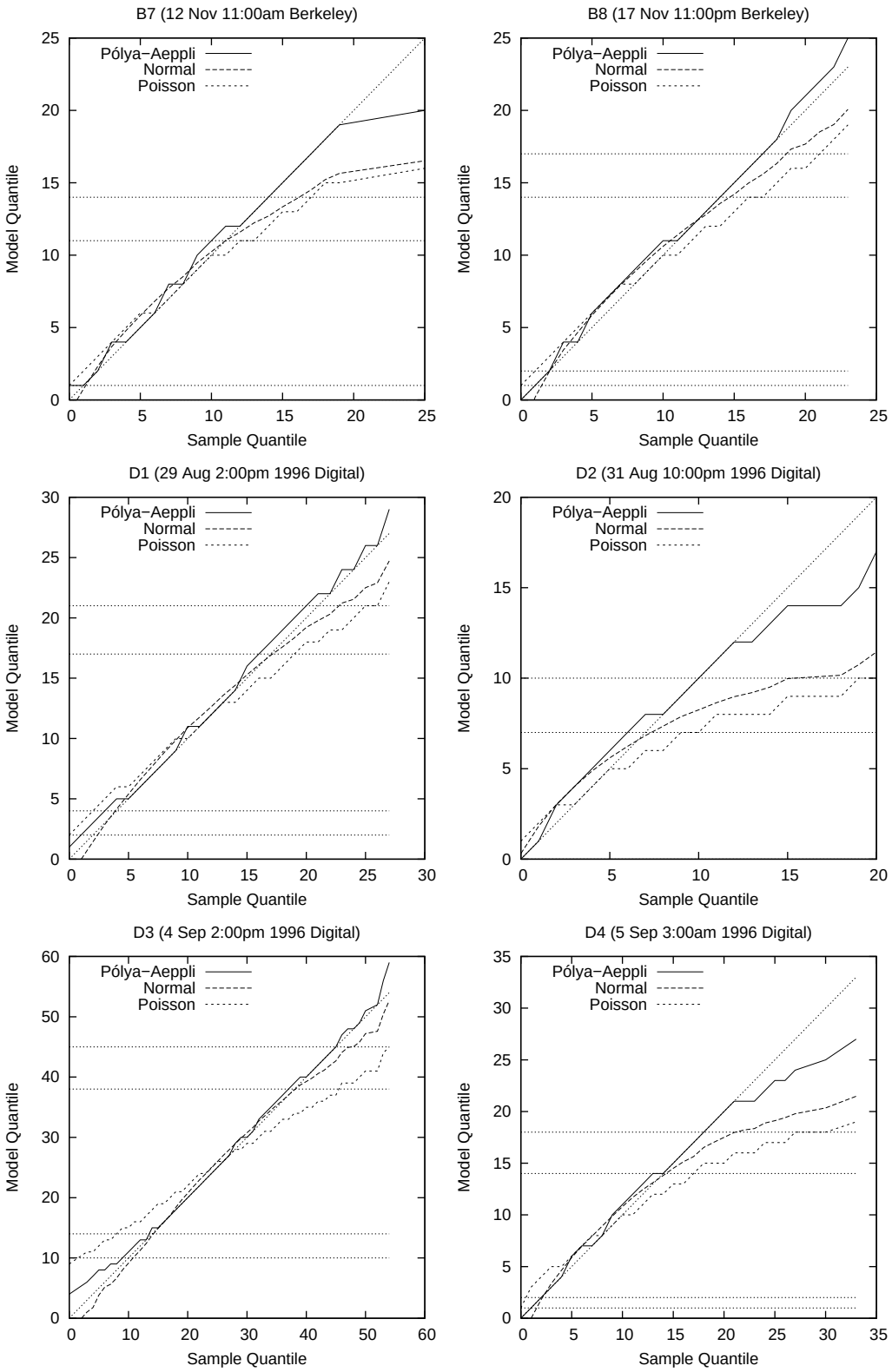


Figure J.3 (Part 2) QQ Plot of HTTP Request Arrival Per Second Compared to Pólya-Aeppli, Normal and Poisson Distributions

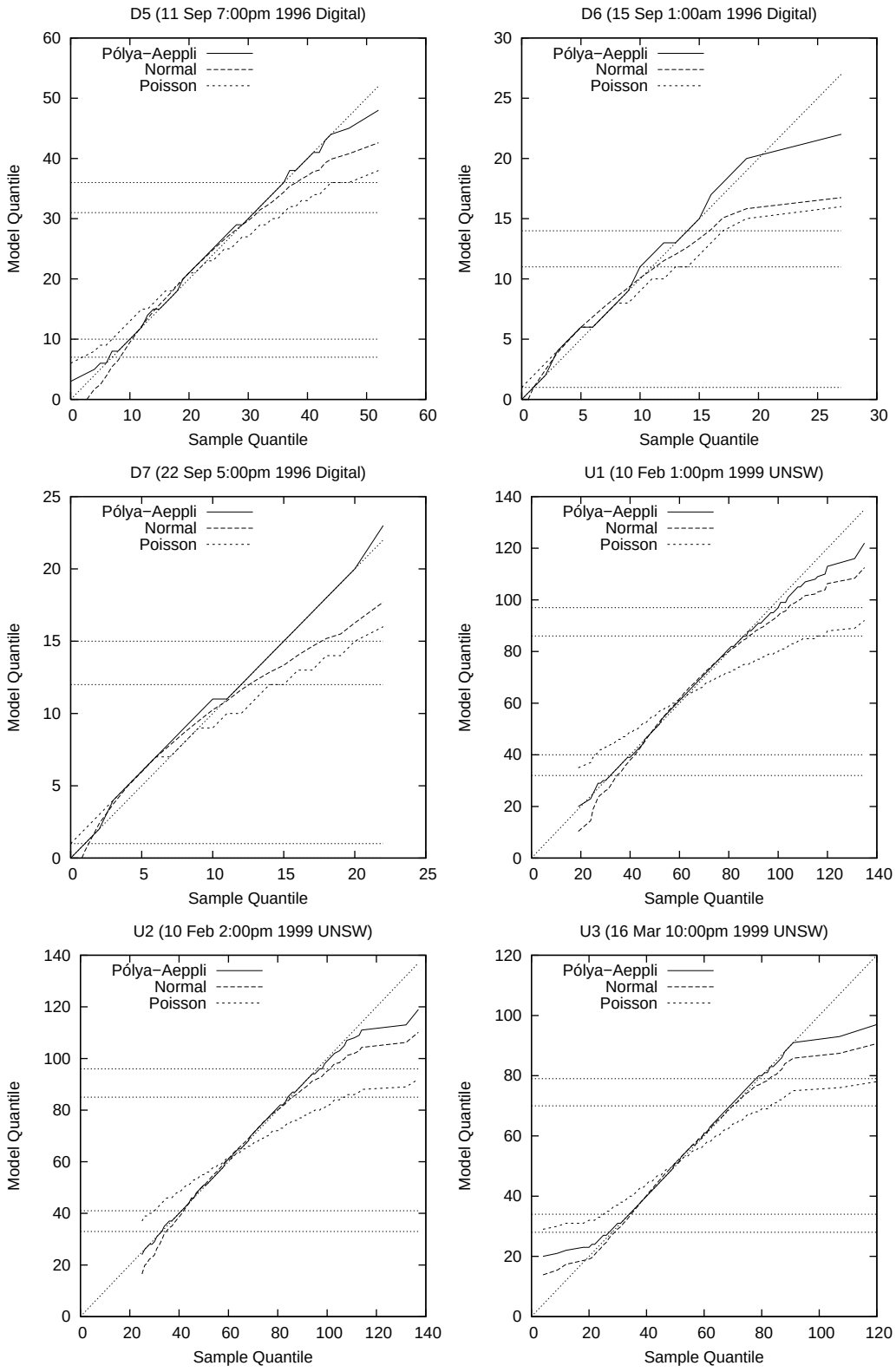


Figure J.3 **(Part 3)** QQ Plot of HTTP Request Arrival Per Second Compared to Pólya-Aeppli, Normal and Poisson Distributions

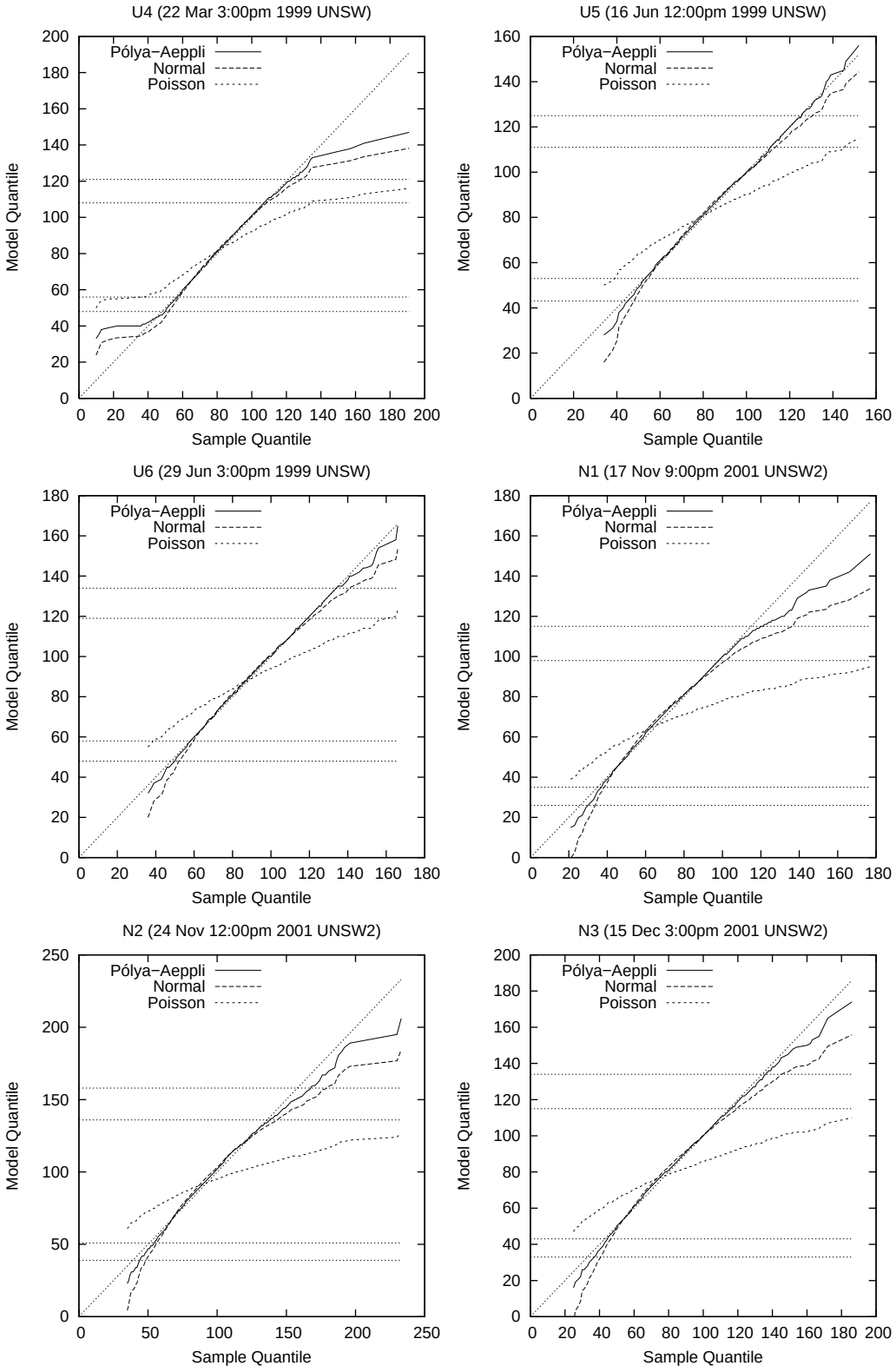


Figure J.3 (Part 4) QQ Plot of HTTP Request Arrival Per Second Compared to Pólya-Aeppli, Normal and Poisson Distributions

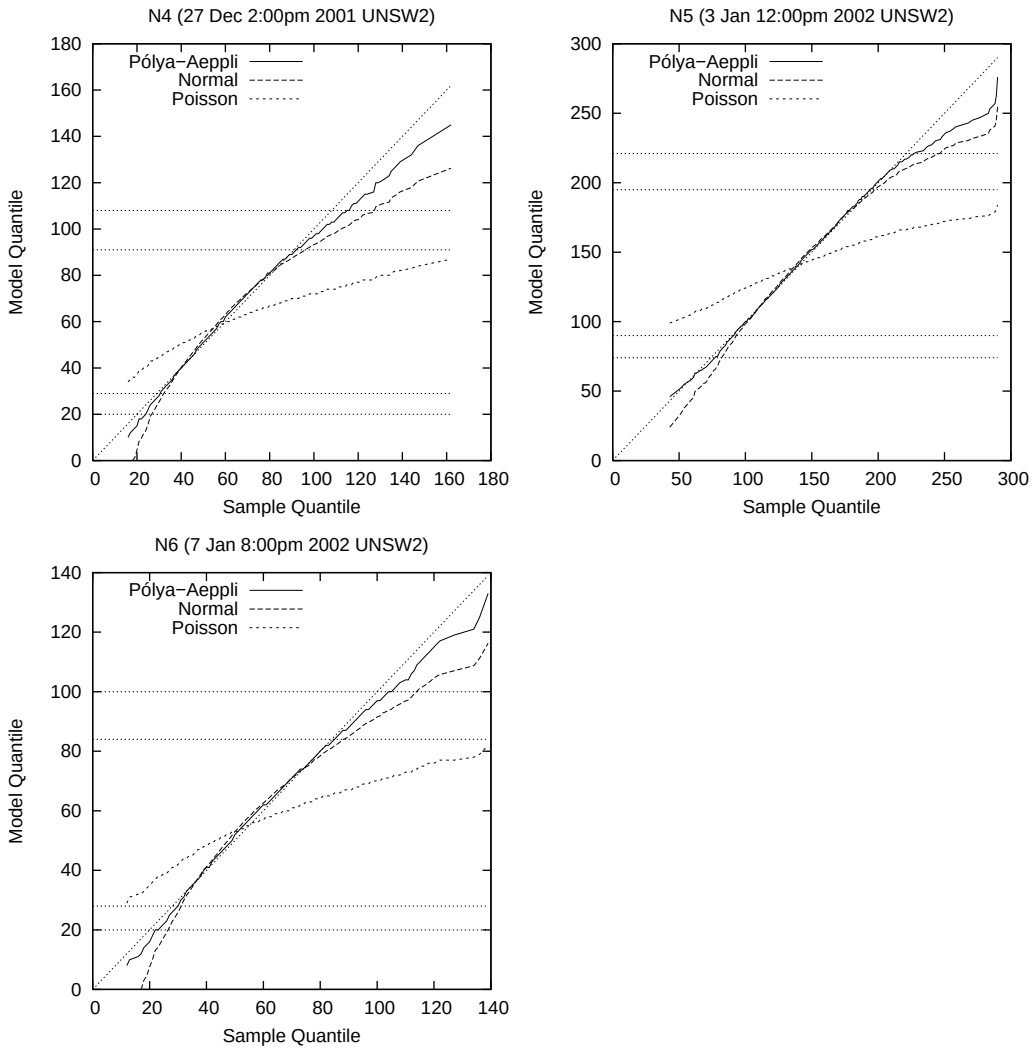


Figure J.3 (Part 5) QQ Plot of HTTP Request Arrival Per Second Compared to Pólya-Aeppli, Normal and Poisson Distributions