Technical Memorandum No. CIT CDS 00-001

Heavy-Tailed Distributions, Generalized Source Coding and Optimal Web Layout Design

Xiaoyun Zhu Jie Yu John Doyle

xyzhu, jyu, doyle@cds.caltech.edu
Control & Dynamical Systems 107-81
California Institute of Technology
Pasadena, CA 91125

Abstract

The design of robust and reliable networks and network services has become an increasingly challenging task in today's Internet world. To achieve this goal, understanding the characteristics of Internet traffic plays a more and more critical role. Empirical studies of measured traffic traces have led to the wide recognition of self-similarity in network traffic. Moreover, a direct link has been established between the self-similar nature of measured aggregate network traffic and the underlying heavy-tailed distributions of the Web traffic at the source level.

This report provides a natural and plausible explanation for the origin of heavy tails in Web traffic by introducing a series of simplified models for optimal Web layout design with varying levels of realism and analytic tractability. The basic approach is to view the minimization of the average file download time as a generalization of standard source coding for data compression, but with the design of the Web layout rather than the codewords. The results, however, are quite different from standard source coding, as all assumptions produce power law distributions for a wide variety of user behavior models.

In addition, a simulation model of more complex Web site layouts is proposed, with more detailed hyperlinks and user behavior. The throughput of a Web site can be maximized by taking advantage of information on user access patterns and rearranging (splitting or merging) files on the Web site accordingly, with a constraint on available resources. A heuristic optimization on random graphs is formulated, with user navigation modeled as Markov Chains. Simulations on different classes of graphs as well as more realistic models with simple geometries in individual Web pages all produce power law tails in the resulting size distributions of the files transferred from the Web sites. This again verifies our conjecture that heavy-tailed distributions result naturally from the tradeoff between the design objective and limited resources, and suggests a methodology for aiding in the design of high-throughput Web sites.

Contents

1 Introduction **3**
- 1.1 Related Work . . . 4
- 1.2 Outline of the Report . . . 5

2 Self-Similar Network Traffic and Heavy-Tailed Distributions **7**
- 2.1 Self-Similar Random Processes . . . 7
- 2.2 α-Stable Distributions . . . 9
- 2.3 Heavy-Tailed Distributions . . . 10
- 2.4 Self-Similarity Through High-Variability . . . 11
- 2.5 Heavy Tails in WWW File Transfers . . . 11

3 Generalized Source Coding **14**
- 3.1 Data Compression . . . 14
- 3.2 Generalized Source Coding . . . 15
- 3.3 Application to WWW . . . 16
- 3.4 Comparison with Data . . . 16

4 A More General PLR Model and Optimal Web Layout Design **18**
- 4.1 One-Dimensional Web Layout Model . . . 18
- 4.2 Web Sites of Multiple Documents . . . 22

5 Simulations of Graph-Based Web Layout Models **25**
- 5.1 Web Sites as Random Graphs . . . 25
  - 5.1.1 The Markov Chain Model . . . 26
  - 5.1.2 Optimization through Splitting and Merging . . . 27
- 5.2 Simulations on Random Graphs . . . 27
  - 5.2.1 Initialization of the Graph Model . . . 27
  - 5.2.2 Heuristic Optimization and Simulation Results . . . 28
  - 5.2.3 Two Special Cases . . . 33
  - 5.2.4 Summary . . . 35
- 5.3 A More Realistic Model . . . 35
  - 5.3.1 Generation of Web Page Geometry . . . 35
  - 5.3.2 Simulation Results . . . 36
- 5.4 Future Improvements of Graph-Based Models . . . 37

6 Concluding Remarks **41**
- 6.1 Summary and Discussion . . . 41
- 6.2 Future Directions . . . 42

A Proof of the Asymptotic Results in Chapter 4 **43**
A.1 For the Gaussian Distribution 43
A.2 For the Cauchy Distribution 44
A.3 Properties of the Operator '$\sim$' 44

Chapter 1

Introduction

The emergence of the Internet over the past twenty years has turned computer networks from sets of locally connected host computers and terminals into a globally distributed information exchanging and processing system. To people throughout the world, the influence of the Internet is tremendous. Communications become easier and faster, information stored in a fragmented manner is now accessible to a global audience, and businesses are conducted in new and more efficient ways. Personal computers have evolved from pure computing devices sitting on people's desks, into windows of communication that connect each individual to the rest of the world. All of these are enabled by the increasing power of the Internet.

On the other hand, as a large-scale, heterogeneous, complex interconnected system, the Internet demonstrates many features that are unparalleled by other engineering systems, and confronts network researchers with immensely challenging problems. One such feature is the unprecedented growth rate in the scale of the Internet. While there were only about 1.3 million Internet hosts in January 1993, the number increased to approximately 72.4 million in January 2000. It has been increasing exponentially with a growth rate of around 80%/year,[1] and there is no evidence indicating that this growth will slow down in the near future. Another important feature of the Internet is its enormous heterogeneity that exists on many levels, including the underlying physical links that carry the information, the protocols that inter-operate over the links and manage the traffic, the mix of applications run at end systems, and the degrees of congestion at different times and locations. Moreover, the dramatic change of the Internet over time lies not only in its scale, but also in the ways it is used. Increasing processor speed and network bandwidth continue spurring development of new applications, such as, multimedia, real-time video and audio, distributed computing, multi-player network gaming, and the World Wide Web. In the meantime, the ascending popularity of these applications has in turn driven the need for more powerful computers and higher speed networks. In particular, the World Wide Web, or the Web for short, originally prototyped in 1991, has quickly become the dominant force that is driving the explosive growth of the Internet [40].

The enormous number of users and rich class of applications on the Internet have made its robustness a highly desirable property, even more important than its efficiency. We have all experienced system break downs, "requested URL not found", broken pipes, slow downloads, and so on. In particular, for the Web users, a survey in 1997 showed that slow access and inability to find relevant information are the two most frequently reported problems [41]. And network congestion is at least partially responsible for the slow access problem. Congestion control has become the most challenging task that is facing network

[1] Source: Internet Software Consortium (http://www.isc.org/)

researchers today. It has many facets, including modeling of Internet traffic, generation of synthetic traffic traces that can be used in simulations, analysis of queueing and network performance under different traffic patterns, and design of network protocols both for the end systems and at the network level. All these problems are intimately related. As a consequence, understanding the characteristics of network traffic becomes more and more critical to designing robust and reliable networks and network services.

Since WWW connections currently dominate Internet traffic, the Web traffic is chosen as the object of study in this report. The goal is to provide a plausible physical explanation for the empirically observed characteristics in measured Web traffic based on the theory of *robust design* of uncertain systems.

1.1 Related Work

High quality measurements have been carried out on traffic in various computer networks since early 1990s. Statistical analysis on the empirical data has shown strong indication of the *self-similar* nature of the traffic in both local area networks(LANs) [28] and wide area networks(WANs) [39], quite unlike the traditionally assumed Poisson traffic models. This means that real traffic data exhibits long-range dependence and high burstiness over a wide range of time scales. In contrast, Poisson models have a characteristic burst length that would be smoothed out by averaging [47].

These discoveries have inspired extensive and interesting research in the modeling of network traffic, its relationship to network protocols and its impact on network performance [29, 38, 26, 14, 2]. For instance, analysis and simulation results have shown a significant difference in queueing performance between traditional traffic models and self-similar models [37, 8, 17, 19]. As a consequence, ignoring the self-similar nature of the traffic at the modeling stage may lead to overly optimistic performance predictions and thus cause potential problems for buffer design and admission control in real high speed networks.

Moreover, evidence was presented in [12] that the subset of network traffic due to WWW transfers, recorded using NCSA Mosaic in early 1995, also demonstrated characteristics of self-similarity. Based on a mechanism of constructing self-similar traffic using a large number of ON/OFF sources that have period lengths drawn from *heavy-tailed* distributions [32, 48, 44], the authors in [12] traced the statistical properties of WWW traffic back to the distribution of Web transmission times, which in turn, is closely related to the distribution of the sizes of files transferred. Both distributions from measurement data exhibit heavy tails which decline like power laws with exponents close to 1. A newer data set collected in the same computing environment in 1998 exhibited similar characteristics, except that the power law tail became relatively lighter with an exponent at approximately 1.4 [7].

Although the recognition of heavy-tailed distributions in Web traffic was relatively new, there exists a rich body of literature on power laws in many complex bio-/eco-/techno-/socio-logical systems (CBETS). They have been found in power outages, forest fires, deaths and dollars lost due to man-made and natural disasters, income and wealth distribution of individuals and companies, variations in stock prices and federal budgets, and many other phenomena (see [9, 10] and the references therein). In addition to Web file transfers, more examples of heavy tails in networks and computer systems have been discovered, including sizes of files in a file system [24], CPU times consumed by UNIX processes [27], inter-keystroke times for typing [14], frame sizes for variable-bit-rate video [21], TELNET inter-packet times and lengths of FTP bursts [39], and sizes and durations of bursts and idle periods of individual Ethernet connections [35]. This ubiquity of power laws has motivated some researchers to suggest that they have a common origin in self-organized criticality [5]. It has also been

suggested that it is possible to describe many of the features of computer networks in terms familiar in information theory, statistical physics, and the "new science of complexity" such as information, entropy, phase transitions, fractals, self-similarity, power laws, chaos, and so on. The hope is that these insights may lead to new approaches to network protocol design.

Ideas such as self-organized criticality have received much promotion in the last decade, but there have been as yet no convincing examples outside of carefully constructed laboratory experiments. Furthermore, while the "new science of complexity" has provided intriguing metaphors and popularized the notion that there are limits to reductionism, there has been little deep theoretical insights or practical applications. Recently, Carlson and Doyle [9] introduced a radically different theory for the nature of complexity and the origin of power laws and "phase transitions" in complex systems called Highly Optimized Tolerance (HOT). HOT systems arise when deliberate robust design aims for a specific level of *tolerance* to uncertainty, which is traded off against the cost of the compensating resources. *Optimization* of this tradeoff may be associated with some mixture of explicit planning as in engineering, or mutation and natural selection as in biology, but the word "design" is used loosely to encompass both. HOT systems in biology and engineering share many common features, including (1) high efficiency, performance, and robustness to uncertainties the systems are designed to deal with, (2) potential hypersensitivity to design flaws and unanticipated perturbations, (3) nongeneric, specialized, structured, and modular configurations, and (4) power laws.

The most important feature of the HOT theory for this report is in providing an alternative explanation for the origin of power laws in Web traffic. The connection between HOT and WWW files was first discussed in [9], and the connections with source coding were first made explicit in [16], which also looked at forest fires and compared HOT with standard SOC results. In this report a variety of more detailed models of varying tractability will be introduced, which present a consistent and coherent application of the HOT ideas to Web file layout. This approach, hopefully, can be applied to study power laws in other HOT systems as well.

1.2 Outline of the Report

Chapter 2 begins with a review of the nature of self-similar random processes and heavy-tailed distributions as a foundation for understanding the basic statistical framework behind self-similar network traffic. It then takes a further look into the previous work that established the connection between self-similar aggregate traffic and individual ON/OFF sources whose periods exhibit heavy-tailed distributions [48, 44]. The most relevant work to this report was the application of the above theory to WWW traffic and the observation of a similar connection in empirical data [12]. This study emphasized the importance of the heavy tails in Web file transfers, because they can be used to explain the self-similarity in WWW traffic, the largest contributor to the overall Internet traffic nowadays.

Now comes the central question: Where do heavy-tailed distributions of WWW file transfers come from? To answer this question, this report takes the basic approach of viewing the design of Web layout as a source coding problem, much like in standard information theory [11, 43], but with a new twist. A classical application for source coding is data compression. The model introduced in [16] that captures important elements common to both data compression and WWW design is reviewed in Chapter 3. It is referred to as the PLR model because it contains probabilities(P), losses(L) and resources(R). The interesting part is: what makes the Web layout design different from data compression? In data compression the aim in the design is to minimize the average length of codewords in order to reduce the cost of storage or transmission, subject to the constraint of decodability. Similarly, an

important goal in Web layout is to minimize the delay in download times and latency, but the design variable is not the selection of codewords, but is instead the layout of the Web site itself. This would of course be in addition to any data compression that would be done to individual files. Such layout can be optimized for a given distribution of user interest, and a constraint, say, on the total number of files. As expected, average download times are minimized by having the high hits be small files, allowing larger files for rarely requested portions of the Web site. The resulting optimal distributions are very unlike those in standard Shannon theory and exhibit power law tails. The comparison between the prediction from the PLR model and the 1995 data is also shown in Chapter 3.

While the above PLR model provides some insights into the origin of power law distributions, it requires several rather unrealistic assumptions, which can be relaxed in various ways with some loss in the transparency of the solutions. With this spirit a more general PLR model is introduced at the beginning Chapter 4. It is this general model and its application to Web layout design that are the focus of this report. With the general PLR model the optimization problem is typically combinatoric. The global optimum can be achieved analytically or numerically on a special and simplified setting, or suboptimal solutions can be obtained using heuristics on more complex models. Chapter 4 takes the first approach, while the second approach is pursued in Chapter 5. We try to connect to the 1995 data first in Chapter 4. In 1995 the Web was in a nascent form, where much of the Web content was the result of putting preexisting documents on the Web. Since most such documents are essentially one-dimensional objects, Web layout in this context can be thought of to a first approximation simply as the problem of chopping up a document into pieces with links between. Design of such a Web layout can again be formulated as a generalized source coding problem, but the model is slightly more realistic than the PLR model with some additional features of a real Web site. However, focusing initially on the simple setting of one-dimensional documents that do not make extensive use of hyperlinks keeps the model still analytically tractable. The results are quite consistent with the simpler PLR model, as well as with the 1995 data.

In Chapter 5 we explore numerically the changes that would be expected as documents are designed specifically for the Web, including more complex layout for Web sites, more effective use of hyperlinks, and more internal structure in individual Web pages. To maximize the throughput, such a Web layout can be modified by splitting or merging files, with a tradeoff between ease of navigation, which would favor fewer files, and having small files to download. A heuristic optimization on random graphs is formulated, with user navigation modeled as Markov chains. Simulations on different classes of graphs always suggest that a Web site optimally designed to minimize the average latency a user experiences in browsing leads to power law distributions in the sizes of file transfers. In addition, the exponents of the power law tails are usually higher than those from the one-dimensional model, which is consistent with the difference found between the 1995 and 1998 data sets. Possible improvements on the graph-based Web layout models are proposed at the end of the chapter.

Finally, Chapter 6 summarizes the main results and suggests future research directions.

Chapter 2

Self-Similar Network Traffic and Heavy-Tailed Distributions

Measured network traffic can be characterized by discrete-time random processes, or time series $\mathbf{X} = \{X_k\}_{k=1}^{\infty}$. Typically the time line is uniformly divided into successive, nonoverlapping intervals of period Δt, and each random variable X_k represents the number of event occurrences during each time interval. For a *Poisson process* where independent events occur at random instants of time at an average rate of λ events per unit time, the above $\mathbf{X}$ is the increment process of the corresponding Poisson counting process with period Δt. All X_k are independent and identically distributed (i.i.d.) with

$$P[X_k = n] = \frac{(\lambda \Delta t)^n}{n!} e^{-\lambda \Delta t}, \qquad n \geq 0. \tag{2.1}$$

The Poisson process model has been successfully applied to describe call arrivals in the public switched telephone networks (PSTN) for at least fifty years, which led to the development of modern queueing theory for performance analysis and capacity planning. However, the Poisson framework is no longer valid for modeling traffic in the quickly emerging data networks, which exhibits high *burstiness* over a wide range of time scales. Instead, self-similar process models were proposed by many researchers to characterize traffic traces from a variety of packet-switching networks, including LANs [28], WANs [39], and WWW [12].

To understand self-similar network traffic and its possible causes, the next few sections review the theory of self-similar random processes and α-stable distributions, as well as the notion of heavy-tailed distributions, which attempts to draw a clear picture of the connections between self-similarity and power laws. All the theorems and lemmas are given without proofs. See [42] for an extensive and in-depth analysis.

2.1 Self-Similar Random Processes

The definition of *self-similarity* varies with the class of random processes it deals with. The standard one is for continuous-time nonstationary processes that have stationary increments, which will not be discussed here. The one that is more relevant to the traffic sequences defined above is for stationary time series. The define it, we need to first define *renormalization group transformations.*

Definition 2.1 *Let* $\mathbf{X} = \{X_k\}_{k=1}^{\infty}$ *be a discrete-time random process. Fix a number* $H > 0$.

For any $n \geq 1$, define the transformation $T_n : \mathbf{X} \to T_n\mathbf{X} = \{(T_nX)_k\}_{k=1}^{\infty}$, where

$$(T_nX)_k = \frac{1}{n^H} \sum_{i=(k-1)n+1}^{kn} X_i, \qquad \textit{for all } \ k \in \mathbb{N}. \tag{2.2}$$

The transformations $T_n, n \geq 1$, are called **renormalization group transformations** *with critical exponent H.*

The family of transformations $\{T_n\}_{n=1}^{\infty}$ forms a semi-group since $T_{mn} = T_mT_n$, with T_mT_n being the transformation T_n followed by the transformation T_m. It is called the *renormalization group with exponent H*. A time series $\mathbf{X} = \{X_k\}_{k=1}^{\infty}$ is said to be a *fixed point* of the renormalization group if for all $n \geq 1$,

$$T_n\mathbf{X} \stackrel{d}{=} \mathbf{X}, \tag{2.3}$$

where "$\stackrel{d}{=}$" denotes equality in finite-dimensional distributions.

Definition 2.2 *A discrete-time, zero-mean, stationary random process $\mathbf{X} = \{X_k\}_{k=1}^{\infty}$ is called (exactly)* **self-similar** *or* **fractal** *with scaling parameter H if it is a fixed-point of the renormalization group $\{T_n\}_{n=1}^{\infty}$ with exponent $H \in [0.5, 1)$. It is said to be* **asymptotically self-similar** *if (2.3) only holds as $n \to \infty$.*

The scaling parameter H is a measure of the degree of self-similarity. The higher the H, the "more" self-similar the process. The reason will become clear in the following example.

Example 2.3 *Fractional Gaussian noise: A stationary Gaussian random process $\mathbf{X} = \{X_k\}_{k=1}^{\infty}$ with mean zero and variance σ^2 is called a fractional Gaussian noise with Hurst parameter $H \in [0.5, 1)$ if the autocorrelation satisfies*

$$R(k) = E[X_iX_{i+k}] = \frac{\sigma^2}{2}(|k+1|^{2H} - 2|k|^{2H} + |k-1|^{2H}). \tag{2.4}$$

Fractional Gaussian noise is the stationary increment process of fractional Brownian motion, a continuous-time self-similar process. For $H = 0.5$, fractional Gaussian noise reduces to Gaussian white noise. For $0.5 < H < 1$, we have the following lemma.

Lemma 2.4 *Let $R(k)$ and $S(\lambda)$ be the autocorrelation function and the spectral density of a fractional Gaussian noise with Hurst parameter $0.5 < H < 1$. Then*

$$\begin{array}{lll} R(k) \sim \sigma^2 H(2H-1)k^{2(H-1)}, & as & k \to \infty, \\ S(\lambda) \sim \sigma^2 C_H |\lambda|^{1-2H}, & as & |\lambda| \to 0, \end{array} \tag{2.5}$$

where $f(x) \sim g(x)$ denotes $\lim_{x \to \infty/0} f(x)/g(x) = 1$, and C_H is a constant for fixed H, independent of λ.

The above lemma tells us both $R(k)$ as $k \to \infty$ and $S(\lambda)$ as $|\lambda| \to 0$ behave like *power laws*, whose exponents are determined by the Hurst parameter H. The higher the H, the slower the decay of the autocorrelation $R(k)$ as $k \to \infty$. In fact, with $0.5 < H < 1$, $\sum_{k=-\infty}^{\infty} R(k) = \infty$, in which case we say that the random process $\mathbf{X}$ has *long-range dependence*, a phenomenon referred to by Mandelbrot as the "Joseph Effect." [34] Accordingly, the spectral density $S(\lambda)$ diverges at the origin.

Is fractional Gaussian noise a self-similar process? The answer is yes based on the following lemma.

Lemma 2.5 *Fractional Gaussian noise is the only Gaussian fixed point of the renormalization group, with the critical exponent equal to the Hurst parameter H.*

Another set of fixed-points of the renormalization group involve sequences of i.i.d. random variables with α-stable distributions, which will be discussed in the next section.

2.2 α-Stable Distributions

Definition 2.6 *A random variable X is said to have a* **stable distribution** *if for any $n \in \mathbb{N}$, there exist $C_n \in \mathbb{R}^+$ and $D_n \in \mathbb{R}$ such that*

$$X_1 + X_2 + \cdots + X_n \stackrel{d}{=} C_n X + D_n, \tag{2.6}$$

where $X_1, X_2, \ldots, X_n$ are independent copies of X. Moreover, X is called **strictly stable** *if $D_n = 0$. For any stable random variable, there is a number $\alpha \in (0, 2]$ such that $C_n = n^{\frac{1}{\alpha}}$. The number α is called the index of stability or characteristic exponent. A stable random variable X with index α is called α-stable.*

Example 2.7 *The Gaussian distribution $N(\mu, \sigma^2)$ is an α-stable distribution with $\alpha = 2$. The Cauchy distribution with the density function $\frac{\sigma}{\pi((x-\mu)^2+\sigma^2)}$ is an α-stable distribution with $\alpha = 1$.*

Lemma 2.8 *The sequence $\mathbf{X} = \{X_k\}_{k=1}^{\infty}$ of i.i.d. strictly α-stable random variables is a fixed point of the renormalization group $\{T_n\}_{n=1}^{\infty}$ with critical exponent $H = \frac{1}{\alpha}$.*

As shown in the previous section, if nontrivial correlations between X_k are allowed, we obtain another set of fixed points of $\{T_n\}_{n=1}^{\infty}$ which are fractional Gaussian noises with Hurst parameter $H \in [0.5, 1)$. The intersection of these two cases is Gaussian white noise with $\alpha = 2$ and $H = \frac{1}{2}$.

We have seen that both Gaussian ($\alpha = 2$) and non-Gaussian ($\alpha < 2$) stable distributions can produce self-similarity or long-range dependence in random processes. The follow lemmas explain the differences between these two distributions.

Lemma 2.9 *Let X be an α-stable random variable. Then as $x \to \infty$,*

$$\begin{cases} P[|X| > x] \sim \frac{1}{\sqrt{\pi}\sigma x} e^{-\frac{x^2}{4\sigma^2}}, & \text{for} \quad \alpha = 2; \\ P[|X| > x] \sim A_\alpha x^{-\alpha}, & \text{for} \quad 0 < \alpha < 2, \end{cases} \tag{2.7}$$

where $A_\alpha > 0$ is a constant for fixed α, independent of x.

The above lemma says the tail of an α-stable distribution with $0 < \alpha < 2$ behaves like a power law ($x^{-\alpha}$), which indicates much higher variability than the Gaussian case with $\alpha = 2$.

Lemma 2.10 *Let X be an α-stable random variable with $0 < \alpha < 2$. Then*

$$\begin{cases} E|X|^p < \infty, & \text{for any} \quad 0 < p < \alpha; \\ E|X|^p = \infty, & \text{for any} \quad p \geq \alpha. \end{cases}$$

A direct result of the above lemma is that an α-stable random variable has infinite variance when $0 < \alpha < 2$, and infinite mean when $0 < \alpha \leq 1$, which again is in sharp contrast to the Gaussian case where both mean and variance are finite.

Stable distributions can be generated through the central limit theorem. In fact, stable distributions are the only distributions that can be obtained as limits of normalized sums of i.i.d. random variables, as stated in the following theorem.

Theorem 2.11 *A random variable X has a stable distribution if and only if there is a sequence of i.i.d. random variables $\{Y_k\}_{k=1}^{\infty}$, $d_n > 0$, and $a_n \in \mathbb{R}$, for all $n \geq 1$, such that*

$$\frac{Y_1 + Y_2 + \cdots + Y_n}{d_n} + a_n \stackrel{d}{\Rightarrow} X, \tag{2.8}$$

where $\stackrel{d}{\Rightarrow}$ denotes convergence in distribution.

The special case of the limit being a Gaussian distribution ($\alpha = 2$) occurs when the Y_k have finite variance, which is also known as the ordinary central limit theorem. When Y_k have infinite variance, their normalized sums converge to an α-stable distribution with $0 < \alpha < 2$, which, as we have seen, exhibits a power law tail, or high variability. The distributions with infinite variance are discussed in the next section.

2.3 Heavy-Tailed Distributions

We use the term *heavy-tailed* distributions for the general class of distributions with hyperbolic (power law) tails, i.e.,

$$P[X > x] \sim cx^{-\alpha}, \qquad as \quad x \to \infty, \quad 0 < \alpha < 2, \tag{2.9}$$

where $c > 0$ is a constant independent of x, regardless of the behavior of the distributions at small scales. This kind of distributions have infinite variance, a phenomenon that Mandelbrot referred to as the "Noah Effect" [33]. In practice, it means that the corresponding random variable can take on extremely large values with nonnegligible probability.

The best way to visualize a heavy-tailed distribution is to plot its complementary cumulative distribution on a log-log scale and look for approximately linear behavior in the upper tail. The slope of the straight line $-\alpha$ measures the "heaviness" of the tail, or the intensity of the "Noah Effect."

As we have seen in the previous section, α-stable distributions with $0 < \alpha < 2$ are heavy-tailed distributions. The converse is not necessarily true. One counter example is the *Pareto* distribution which is hyperbolic over its entire range. Its probability density function (PDF) is

$$p(x) = \alpha k^{\alpha} x^{-\alpha-1}, \qquad \alpha, k > 0, \quad x \geq k, \tag{2.10}$$

and its complementary cumulative distribution function (CCDF) is given by

$$P[X > x] = (\frac{x}{k})^{-\alpha}, \tag{2.11}$$

where k is the location parameter corresponding to the minimum value for X. The above distribution is heavy-tailed when $\alpha < 2$.

A well-known result about the Pareto distribution is that it is the only distribution that satisfies the following *scaling property* [33, 3]:

$$P[X > x | X > x_0] = (\frac{x}{x_0})^{-\alpha}, \qquad x > x_0. \tag{2.12}$$

The above relation means that the distribution conditioned on $X > x_0$ is also a Pareto distribution with the same index α and a different location parameter x_0. For general heavy-tailed distributions, (2.12) is only satisfied asymptotically, as x and x_0 become large enough.

This appealing property guarantees that the distribution is robust with respect to truncation from below.

Although heavy-tailed distributions may not be α-stable in general, properly normalized sums of i.i.d. heavy-tailed distributions converge to α-stable distributions with $0 < \alpha < 2$ through the central limit theorem. The latter, can in turn, give rise to random processes that are long-range dependent, or bursty over a wide range of time scales. Hence, possible links exist between heavy-tailed distributions and self-similar network traffic through α-stable distributions with infinite variance. Connections can also be made between fractional Gaussian noise traffic that is also self-similar and heavy-tailed distributions at the source level, as has been observed through empirical studies, which will be reviewed in the next two sections.

2.4 Self-Similarity Through High-Variability

In the previous sections, we discussed random processes that are self-similar or long-range dependent (the "Joseph Effect") and heavy-tailed distributions with high variability or infinite variance (the "Noah Effect"). The former has been observed in empirical studies of a large collection of Internet traffic measurements. The latter can be found in a rich body of literature on networks and computer systems.

In the search for possible physical causes of self-similar Internet traffic, an interesting connection between the above two phenomena was established by Willinger and Taqqu et al. in [48]. Based on a model originally introduced by Mandelbrot in [32], this paper claimed that the superposition of a large number of ON/OFF sources (also known as "packet trains" [25]) whose ON-periods ("train lengths") and OFF-periods ("intertrain distances") exhibit heavy tails produces aggregate network traffic that exhibits self-similarity. In fact, as both the number of sources and the block aggregation size become large enough, the cumulative traffic approaches a fractional Gaussian noise, the only stationary Gaussian process that is self-similar. Moreover, a simple relation exists between the exponent of the heavy tail ($1 < \alpha < 2$) and the Hurst parameter ($0.5 < H < 1$) of the self-similar traffic, that is

$$H = \frac{3 - \alpha}{2}. \tag{2.13}$$

Generalizations of this result to more realistic settings were discussed in [44] and a rigorous proof was given. Validated by the Ethernet LAN traffic data collected at the source level, the above theory provided a plausible and simple explanation for the occurrence of self-similarity in measured network traffic in terms of the heavy-tailed nature of the traffic generated by individual sources or source-destination pairs. In contrast, traditional ON/OFF source models typically assume exponential or geometric distributions (or more generally, distributions with finite variance) for their ON- and OFF-periods. It was only in recent years when people discovered the discrepancy between the aggregate traffic generated by multiplexing a large number of these sources and the real traffic measured from various working networks. Therefore, the recognition of *heavy-tailed* distributions is indeed the essential point of departure from traditional to self-similar traffic modeling.

2.5 Heavy Tails in WWW File Transfers

Empirical studies on the subset of Internet traffic due to WWW transfers have also found evidence indicating self-similarity. One of the earliest works was presented by Crovella et

al. in [12], where a simple explanation for the self-similarity of WWW traffic was proposed, using the mechanism discussed in the previous section. In fact, the aggregate traffic can be viewed as the superposition of many ON/OFF processes, whose ON-periods correspond to transmission times of individual Web files and OFF-periods correspond to quite times in between. Measurement data from 130,140 such file transfers to 591 users on 37 machines at Boston University in early 1995 showed that the distribution of the transmission times of individual files (ON-periods) has an upper tail which declines like a power law with exponent close to 1, demonstrating extremely high variability. Moreover, the distribution of the sizes of files transferred is also heavy-tailed with a similar exponent, which is not surprising since the transmission time of a document can be roughly modeled as the sum of a cost proportional to the document size plus a fixed overhead, the latter of which can be neglected for large documents. Meanwhile, the quite times (OFF-periods) in the same data, mainly due to user think time, also exhibited heavy-tailed behavior, but with a higher exponent $\alpha \approx 1.5$. Based on the result in [44] which claimed that between the distributions of the ON- and OFF- periods, the one with the heavier tail actually determines the degree of self-similarity in aggregate traffic, the authors in [12] concluded that the distribution of Web file transfers is more likely to be responsible for the observed level of traffic self-similarity. Therefore, the focus of this report is to develop a series of analytical and simulation models which provide plausible explanations for the origin of heavy-tailed distributions of Web file transfers, which in turn, may be the cause for self-similar network traffic. In addition, we neglect the difference between the file sizes and the corresponding transmission times to simplify the models. The term "Web file transfers" is used in general to refer to the sizes of files transferred during Web sessions.

In Figure 2.1 the data set for the WWW file transfers mentioned above is displayed on a log-log scale (base 10), along with a data set for codewords from data compression of a file using standard methods, which we will explain in the next chapter. A line of slope $\alpha = -1$ is shown for comparison. Each data set consists of (l_i, P_i) pairs of individual events with sizes ordered as $l_i \geq l_{i+1}$ and the cumulative frequencies $P_i = P[l \geq l_i]$. l_i can be thought of as the loss or cost proportional to the sizes of files (WWW) or lengths of codewords (DC) to be transmitted on the Internet. In sharp contrast to the data from data compression, which has an exponential distribution ($P_i \propto e^{-\lambda l_i}$), the WWW data exhibits a heavy-tailed distribution ($P_i \propto l_i^{-\alpha}$, $\alpha \approx 1$). Since the Web workloads and user behavior have changed significantly since early 1995, a new set of data was collected in 1998 in the same computing environment with 306 users on 29 machines. Again the distribution of the file transfers displays a heavy tail, although the exponent for the power law increased from approximately 1 in 1995 to 1.4 in 1998 [7]. These two data sets will be referred to as "the 1995 data" and "the 1998 data", respectively, in later chapters.

It is worth pointing out that all the "cumulative distribution" plots in this report are complementary cumulative distributions on a log-log scale (base 10) for the convenience of studying heavy tails. The y-axis can be cumulative probabilities or cumulative frequencies, in the latter case $\max_i P_i > 1$, like in Figure 2.1.

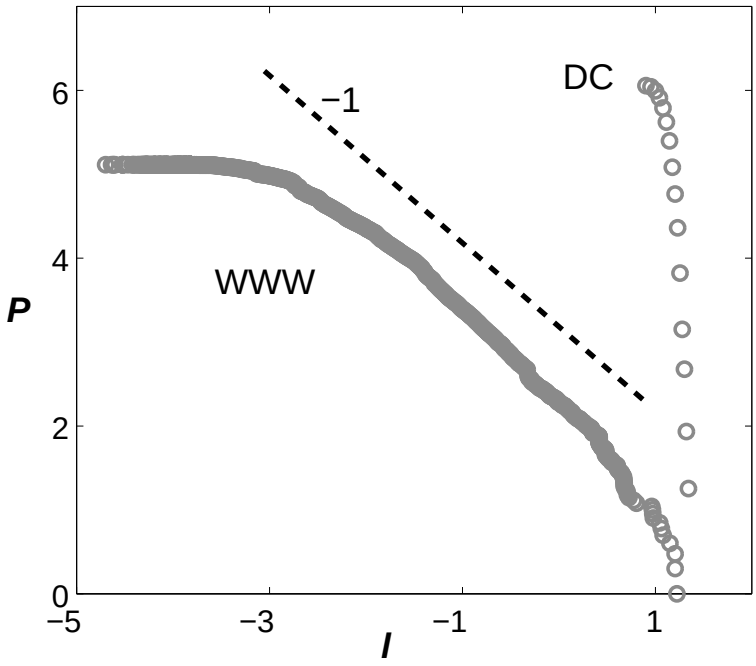

Figure 2.1: Cumulative distributions P_i vs. l_i of Web file transfers (in megabytes) and codewords from data compression.

Chapter 3

Generalized Source Coding

Standard source coding can be generalized using an abstract model, called the PLR model, where resources are allocated to limit average loss, e.g., the average file download time, subject to uncertainties in file content (DC) and file access (WWW), modeled probabilistically. First the PLR setup will be described abstractly, following [16], and then it is specialized to the data compression and WWW cases. For more details, implications for more general complex systems, and comparison with self-organized criticality, see [16].

For a set of abstract events with index i, $1 \leq i \leq N$, whose probabilities of occurrence are p_i, we assume there is a relationship $l_i = f(r_i)$, which describes how the allocation of resources r_i limits the sizes l_i of events. The only coupling occurs through an overall constraint on the resources available: $\sum r_i \leq R$, which only coarsely accounts for the connectivity and spatial structure of real design problems. A natural design objective in data compression and WWW is to minimize the following expected cost

$$J = \{\sum p_i l_i \mid l_i = f(r_i),\ \sum r_i \leq R\}. \tag{3.1}$$

3.1 Data Compression

While our results are completely standard and well-known in the data compression case, we will briefly review them anyway, as our notation is nonstandard to allow for generalizations to the WWW case, and otherwise could lead to some confusion. A message is a vector $\mathbf{X}$, composed of a sequence of symbols $\mathbf{X} = X_1 X_2 ....$ The X_k are independent, identically distributed random variables, chosen from N source symbols, which occur with probabilities p_i, $\sum_i p_i = 1$, and $1 \leq i \leq N$. An optimized code, consisting of codewords of length l_i, minimizes the expected length of transmission J given in equation (3.1). For data compression the resource limitation $\sum_i 2^{-l_i} \leq 1$ is just Kraft's inequality [11], which is equivalent to unique decodability. In the PLR formulation, this becomes $l_i = f(r_i) = -\log_2(r_i)$, and $\sum r_i = R \equiv 1$. The optimal solution is $r_i = p_i$, $l_i = -\log_2(p_i)$, and the optimal cost is the Shannon-Kolmogorov entropy $J_0 = -\sum p_i \log_2(p_i)$ [11].

In the context of more general design problems, what is special about data compression is that $p_i = 2^{-l_i}$ does not have heavy tails. "Small" resource allocations are associated with long codewords, but since the number of codewords grows *exponentially* with length, rare symbols have codewords that grow in size only logarithmically. The data in Figure 2.1 uses standard Huffman coding [11] to compress the postscript file of [9]. The comparison between the data and the prediction from the PLR model will be shown in Section 3.4.

3.2 Generalized Source Coding

A more general resource versus loss function $l_i = f_\beta(r_i)$ of the following form

$$f_\beta(r_i) = \begin{cases} -\log(r_i), & \beta = 0; \\ \frac{c}{\beta}({r_i}^{-\beta} - 1), & \beta > 0 \end{cases} \tag{3.2}$$

allows treatment of more general design problems. This choice reduces to data compression ($\beta = 0$, $c = 1$), and keeps $f_\beta(1) = 0$ and ${f_\beta}'(r_i) = -c{r_i}^{-\beta-1}$, $\forall \beta \geq 0$. Note that these conditions uniquely determine $f_\beta(r_i)$ to within the constant c, which reflects a choice of units in the event sizes. The $f_\beta(1) = 0$ normalizes the loss/resource so that devoting a full unit resource eliminates loss. The choice of R reflects the total resources and the units in which resources are measured. The key quantity is the exponent β characterizing the relationship for large l_i and small r_i. It is easy to show using Lagrange multipliers that the optimal solution is

$$r_i = R p_i^{\frac{1}{1+\beta}} \left(\sum_j p_j^{\frac{1}{1+\beta}} \right)^{-1}, \tag{3.3}$$

or

$$l_i = \begin{cases} -\log(Rp_i) + \log(\sum_j p_j), & \beta = 0; \\ \frac{c}{\beta}\left[\left(R p_i^{\frac{1}{1+\beta}} \right)^{-\beta} \left(\sum_j p_j^{\frac{1}{1+\beta}} \right)^{\beta} - 1 \right], & \beta > 0. \end{cases} \tag{3.4}$$

And the optimal cost is given by

$$J_\beta = \begin{cases} -\sum_i p_i \log(Rp_i) + (\sum_i p_i)\log(\sum_i p_i), & \beta = 0; \\ \frac{c}{\beta}\left[R^{-\beta} \left(\sum_i p_i^{\frac{1}{1+\beta}} \right)^{1+\beta} - \sum_i p_i \right], & \beta > 0. \end{cases} \tag{3.5}$$

In the special case when $R = c = 1$ and $\sum_i p_i = 1$, the above cost function becomes

$$J_\beta = \begin{cases} -\sum_i p_i \log(p_i), & \beta = 0; \\ \frac{1}{\beta}\left[\left(\sum_i p_i^{\frac{1}{1+\beta}} \right)^{1+\beta} - 1 \right], & \beta > 0 \end{cases} \tag{3.6}$$

which reduces to the Shannon-Kolmogorov entropy for $\beta = 0$.

Without loss of generality we assume $r_i < 1$, since events with $r_i = 1$ ($l_i = 0$) do not contribute to the cost and can be eliminated, reducing R by 1. Inverting (3.4) yields the (noncumulative) probabilities of events of size l_i

$$p_i(l_i) = c_1(l_i + c_2)^{-(1+1/\beta)} \tag{3.7}$$

for $\beta > 0$, where c_1 and c_2 are unit-dependent constants depending on c and R.

Just as Shannon theory represents an idealized approximation to realistic data compression problems, applications of the PLR model to WWW represent very coarse descriptions of the underlying phenomena. The PLR WWW model is even more idealized than data compression, so we will later explore more complete WWW models. Nonetheless, in each case there are natural interpretations of the fundamental PLR quantities.

3.3 Application to WWW

In WWW models, we associate design with subdivision of large documents into files. The l_i are file lengths, and the p_i are determined by user interest in the files. The cost $J = \sum p_i l_i$ is the average delay a user experiences in downloading files because the transmission time of each reasonably large file is approximately proportional to the file size. For really small files this may not be true due to the nonnegligible fixed overhead for the transmission, so there is a small scale cutoff on l_i. The resources r_i divide the document into files, where large r_i correspond to small files and small r_i correspond to large files. The tradeoff between the lower cost of shorter files, and the need for efficient Web management and user navigability which favors larger files leads to a resource constraint on the number of files.

The most subtle aspect of the PLR model is determining β to reflect the resource versus loss tradeoff. We begin by assuming that a Web site is created by taking a one-dimensional document and splitting it into files which are then connected by links. Subdividing a one dimensional document into files with zero dimensional cuts leads to an inverse relationship between the file size and the resource density, $l_i \propto {r_i}^{-1}$, so that $\beta = 1$. To go beyond this rough dimensional argument and construct a microscopic model for which the PLR formalism (and ultimately also the dimensional argument) applies exactly, we must also insure that the p_i are not changed by varying the r_i. Suppose we split the document into N regions of equal size L and assume a fixed probability p_i of a user hit occurring in the i^{th} region. We then design cuts to subdivide the i^{th} region into n_i equal files each of size l_i. Thus, the size is $l_i = n_i^{-1} L$, while the resource allocation in the i^{th} region is $r_i \propto n_i$, where proportionality reflects possibly differing units. This yields $l_i \propto r_i^{-1}$, again giving $\beta = 1$. In [16], the more general case where $\beta = d$ is considered, which reflects the natural relationship between the size of $d-1$ dimensional resources traded off against the size of the resulting d dimensional events.

3.4 Comparison with Data

The data sets for data compression and WWW shown in Figure 2.1 are replotted in Figure 3.1 for comparison with the PLR model. Each data point (l_i, P_i) shows the cumulative frequency P_i versus the event size l_i, where l_i are ordered as $l_i \geq l_{i+1}$. To generate corresponding data $(l_i, \hat{P}_i)$ for the PLR model, we choose β based on the resource vs. loss relationship for a given system ($\beta = 0, 1$ for DC and WWW, respectively). To relate the noncumulative p_i for the model to the cumulative P_i for the data, we evaluate (3.7) for each l_i in the data set. The difference approximation to $p = dP/dl$

$$p_i = (P_{i+1} - P_i)/(l_i - l_{i+1}) \tag{3.8}$$

can be inverted to approximate each P_{i+1} in terms of all the p_i and l_i, i.e.,

$$\begin{aligned} \hat{P}_{i+1} &= \sum_{j \leq i} p_j (l_j - l_{j+1}) \\ &= \sum_{j \leq i} c_1 (l_j + c_2)^{-(1+1/\beta)} (l_j - l_{j+1}). \end{aligned} \tag{3.9}$$

The constants c_1 and c_2 in (3.9) (equivalently R and c) are set by the cutoff size of the smallest events in the data set, and the total frequency of events, and thus effect the position but not the general shape of the curves. The data compression problem has only the constant c_1, since for DC $c_2 = 0$.

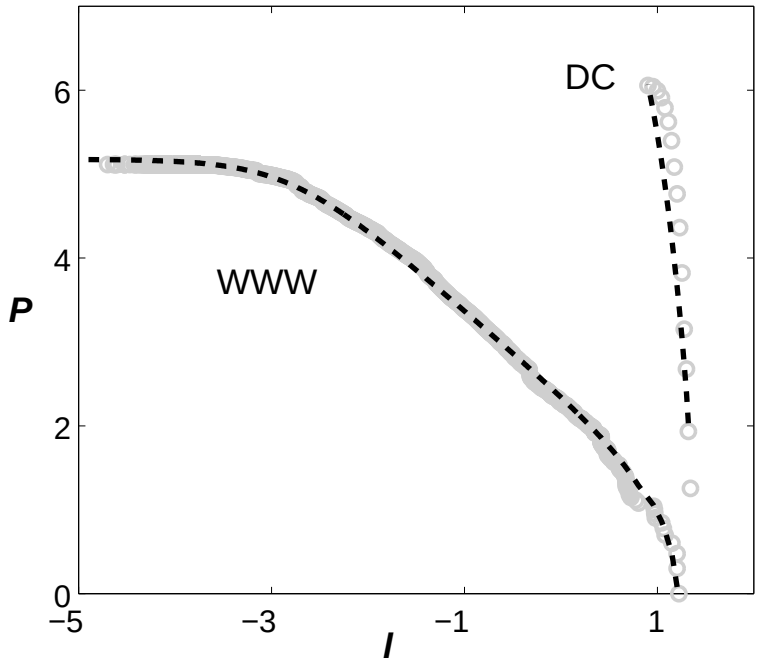

Figure 3.1: Comparison of DC and WWW data (l_i, P_i) with the results of the PLR model $(l_i, \hat{P}_i)$, $i \geq 2$.

In Figure 3.1 the data sets $(l_i, \hat{P}_i)$, $i \geq 2$ from the PLR model (dashed line) for $\beta = 0,\ 1$ are compared directly and unambiguously with the original data (l_i, P_i) (grey circles). The agreement is excellent, especially for the WWW data. The slight discrepancy in the DC case is a consequence of finite block sizes (in this case 16 bits) and integer codeword lengths. Both are essential practical issues not captured in the idealized Shannon theory discussed in Section 3.1.

The PLR model provides some insights into the origin of power laws in HOT systems [9, 10] in general, and the special characteristic in data compression that leads to nonheavy-tailed optimal distributions. Its application to Web file transfers provides a plausible explanation for the self-similar behavior of WWW traffic, given the results discussed in the previous chapter. On the other hand, due to its simplicity, the PLR model is an extreme abstraction of the complicated Web design process in the real world. Of course, it is unlikely that individual Web sites in 1995 were optimally designed, so the agreement of model with data could be coincidental. This issue will be discussed at the end of the report, after we examine more realistic Web layout models and see if similar agreement still holds.

Chapter 4

A More General PLR Model and Optimal Web Layout Design

The PLR model defined in (3.1) assumes that only the loss l_i depends on the resource r_i, while the probability of each event p_i remains fixed, which allows an analytical solution to be obtained from the optimization. This constraint is rather restrictive, which may be violated in more practical Web layout design. The following PLR model

$$J = \{\sum p_i l_i \mid l_i = f(r_i),\ p_i = g(r_i),\ \sum r_i \leq R\} \tag{4.1}$$

generalizes (3.1) to the cases where the probability p_i may also be affected by the resource allocation r_i. This fairly broad setting makes the optimization problem extremely hard to solve analytically. However, this model can also be simplified by making some further assumptions. In these special cases, optimal solutions for (4.1) can be calculated either numerically or even analytically. The one-dimensional Web layout model introduced in this chapter is an example of this specialization. More complicated and realistic models of Web site design can be built on top of this, in which case global solutions to (4.1) are not guaranteed to be found. Simulations of the optimization procedure using heuristic algorithms on graph-based Web layout models will be presented in the next chapter.

The model proposed in this chapter is motivated by the PLR model for WWW reviewed in the previous chapter, but is actually closer in detail to the models studied in [9]. By focusing on the simplest possible setting—a one-dimensional Web layout problem, the model allows for analytical treatment. At the same time, it captures some additional features of a real Web site beyond what is in the PLR model in [16]. In particular, this new model includes a hierarchical structure for the Web site and a more reasonable model of user navigation behavior. We begin with the case of building a Web site out of a single document, then discuss the more complex scenario with multiple documents.

4.1 One-Dimensional Web Layout Model

The one-dimensional Web layout model considers the early stage of Web site design when people basically just put preexisting documents onto the Web. When such a document is fairly big, it is unlikely that all users are interested in every portion of the document. Then it is necessary to split the document into smaller files and connect them in a simple way so that the average latency a user experiences while downloading the relevant information is minimized.

A real random variable X with the sample space $[0, L]$ can be used to model user interest in various portion of the document, where L is to total length of the document with certain units, e.g., bytes. The PDF $p(x)$ indicates the popularity of the information at location x. The specific functional form of $p(x)$ may vary with each individual document. Here we would make a reasonably generic assumption that the document is organized in such a way that more popular information is placed before less popular information, which means $p(x)$ is strictly decreasing with x, as shown in Figure 4.1.

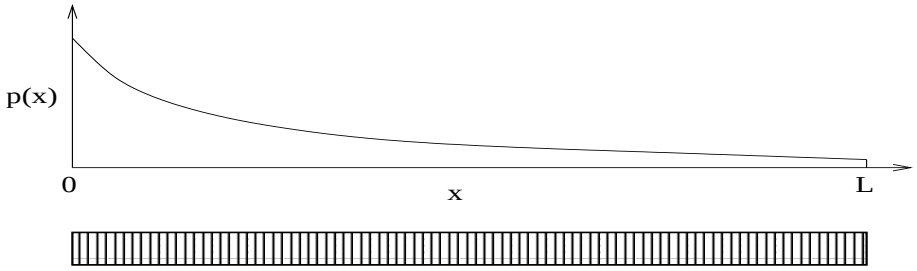

Figure 4.1: Document of size L.

Now divide the document into N files, with the locations of the cuts being $c_1, c_2, \ldots, c_{N-1}$. Denote the size of each file as l_i and the average popularity of each file as p_i. Then for any i, we have

$$\begin{cases} l_i = c_i - c_{i-1}, \\ p_i = \int_{c_{i-1}}^{c_i} p(x)dx, \end{cases} \tag{4.2}$$

where $c_0 = 0$, $c_N = L$. This division is illustrated in Figure 4.2. Note that $\sum_{i=1}^{N} l_i = L$, and $\sum_{i=1}^{N} p_i = 1$.

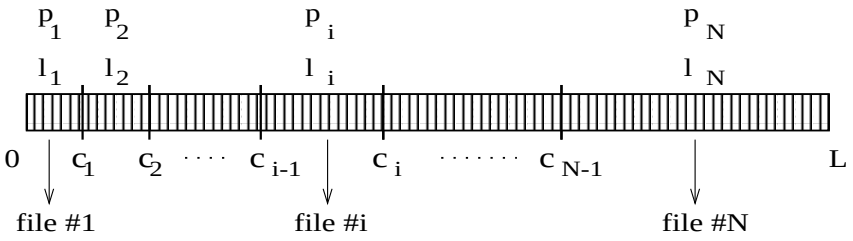

Figure 4.2: Dividing the document.

A one-dimensional Web site can be made by sequentially linking the above N files, so that each file corresponds to a Web page, as shown in Figure 4.3. The design problem is to minimize the average download time of the Web site by optimally dividing the original document, which can be modeled using the general framework in (4.1), but in a simplified setting. At one extreme, it is always desirable to divide the document into as many small files as possible to lower the cost, which, however, sacrifices the ease of manageability and navigability of the Web site. It also causes the Web server to process more requests for the same amount of information. Moreover, when a file is sufficiently small, the overhead associated with the transmission dominates the time required to transfer the file, which diminishes the advantage of making small files. So the number of files N is assumed to be fixed as a constraint on limited resources, like the constraint $\sum_i r_i \leq R$ in the PLR model. When N is not too large, we assume all the resulting files are big enough that the download time of each file can be approximated by the file size, with proper arrangement of the units.

To take into account the navigation behavior of users, we also assume that any user who visits the Web site has to start from the first page, and go through pages in ascending order. This means a user interested in page #i must access pages #1 through #i.

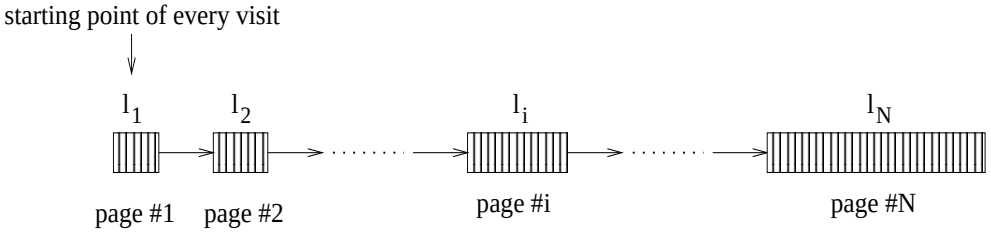

Figure 4.3: A one-dimensional Web site.

In summary, page #i will be targeted with probability p_i from each user's visit. Then all the pages from #1 through #i are transferred, with the total download time equal to the total size of the files, which is $\sum_{j=1}^{i} l_j = c_i$. Thus, minimizing the average download means minimizing the following cost function

$$J_1 = \sum_{i=1}^{N} p_i c_i \tag{4.3}$$

with respect to the cut locations $c_1, \ldots, c_{N-1}$.

An equivalent way to formulate the problem is the following. Denote the hit rate of file #i among all file transfers as p_i^t, then $p_i^t = \sum_{j=i}^{N} p_i$, because file #$i$ will be transferred whenever a user is interested in file #j with $j \geq i$. Note that p_i^t are not probabilities since $\sum_{i=1}^{N} p_i \neq 1$. However, we can say that for each file transfer, the download time is l_i with frequency p_i^t. Hence, the average download time becomes

$$J_2 = \sum_{i=1}^{N} p_i^t l_i \tag{4.4}$$

which can be minimized with respect to the sizes of individual files $l_1, \ldots, l_N$.

It is easy to show that $J_1 = J_2$. So the above two optimization problems are exactly the same. To derive the optimality condition, let $\frac{\partial J_1}{\partial c_i} = 0$, $i = 1, \ldots, N-1$. Then the following recursion relations hold:

$$p(c_i) l_{i+1} = p_i, \qquad i = 1, \ldots, N-1. \tag{4.5}$$

Since $p(x)$ is a strictly decreasing function, it is easy to verify that $l_i < l_{i+1}$ based on (4.2), which means the optimal file sizes will strictly increase with the index. Unfortunately the exact values of the optimal l_i can only be solved numerically for general $p(x)$. For analysis purposes we need to make further assumptions on $p(x)$, for instance, we can assume $p(x)$ belongs to certain classes of distributions that are plausible for modeling user interest and have distinct characteristics. One such class is the exponential distribution. This occurs when users navigate through the document with a memoryless process, i.e., $P[X > x_0 + x | X > x_0] = P[X > x]$. It means that the probability of a user going through a segment of length x is independent of the starting point, and the exponential function is the only continuous PDF that meets this criterion. It will be interesting to see what the distribution of file transfers looks like on an optimized Web site. So assume that

$$p(x) = ke^{-\lambda x} \tag{4.6}$$

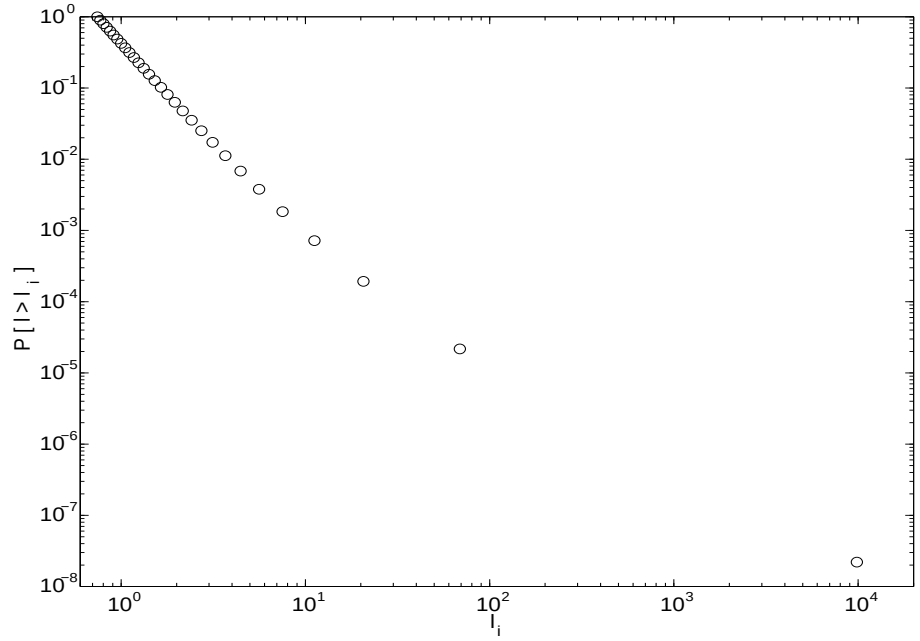

Figure 4.4: P_i^t vs. l_i for Web file transfers. ($p(x)$ is exponential.)

where $0 \leq x \leq L$, and $k = \frac{\lambda}{1-e^{-\lambda L}}$. Let L be big enough so that $e^{-\lambda L} \ll 1$, and $k \doteq \lambda$. By substituting (4.6) into (4.5), we get the recursion equations for the optimal file lengths:

$$l_{i+1} = \frac{e^{\lambda l_i} - 1}{\lambda}, \qquad i = 1, \ldots, N-1. \tag{4.7}$$

The above relation indicates that the sizes of the files grow exponentially towards the end of the document. It is predictable that the last file in the sequence will be significantly larger than the first file. Obviously it is beneficial to group less popular information into big files while keeping the sizes of popular files small. The above equations can be solved numerically with the constraint $\sum_{i=1}^{N} l_i = L$. With the optimal solutions for l_i, we can compute $c_i = \sum_{j=1}^{i} l_j$, and furthermore, $p_i^t = \int_{c_{i-1}}^{L} p(x)dx = e^{-\lambda c_{i-1}} - e^{-\lambda L}$. Now normalize p_i^t such that $\sum_{i=1}^{N} p_i^t = 1$. If we consider the size of the file transferred as a random variable l, then the sample space is $\{l_1, l_2, \ldots, l_N\}$, and p_i^t is the probability mass function(PMF) of l over each sample. The cumulative probability $P_i^t = P[l \geq l_i]$ vs. the size l_i is shown in Figure 4.4, with $L = 10^4$, $N = 20$ and $\lambda = 0.1$.

The cumulative distribution of transferred file sizes displays an asymptotic behavior that is consistent with a heavy-tailed distribution. The slope of the curve in a log-log scale increases as the file size l_i grows bigger, and approaches -1 asymptotically as l_i becomes sufficiently large. This characteristic of the distribution is quite robust with respect to different parameter values N and λ as long as L is large enough.

To prove this asymptotic behavior analytically, we need to consider the limit case where $L = \infty$. Then the probability of transferring file $\#i$ becomes $p_i^t = e^{-\lambda c_{i-1}}$. It follows that as $l_i \to \infty$,

$$\frac{\log p_{i+1}^t - \log p_i^t}{\log l_{i+1} - \log l_i} = \frac{\log e^{-\lambda c_i} - \log e^{-\lambda c_{i-1}}}{\log \frac{e^{\lambda l_i} - 1}{\lambda} - \log l_i} = \frac{-\lambda l_i}{\log(e^{\lambda l_i} - 1) - \log \lambda l_i} \sim \frac{-\lambda l_i}{\lambda l_i - \log \lambda l_i} \sim -1,$$

where $f(x) \sim g(x)$ denotes $\lim_{x\to\infty} \frac{f(x)}{g(x)} = 1$, as defined in Chapter 2. The above argument shows that the slope of p_i^t vs. l_i in a log-log plot approaches -1 as l_i approaches infinity,

which means $p_i^t \sim cl_i^{-1}$ where c is a constant. So the probability of transferred file size has a heavy tail with exponent 1. In reality the file sizes are always finite. But we can view each l_i as being infinite when $l_i \gg l_1$. Even for medium sized l_i, the slope $\frac{-\lambda l_i}{\lambda l_i - \log \lambda l_i}$ is fairly close to -1. And because $l_i \ll l_{i+1}$, $p_i^t \gg p_{i+1}^t$, the cumulative distribution of transferred file sizes P_i^t has exactly the same asymptotic behavior, which is verified by the numerical result demonstrated in Figure 4.4.

Some people may argue that user navigation of a Web site is not necessarily memoryless, which means $p(x)$ may not be exponential. So two other commonly used distribution functions were also tested: the Gaussian distribution ($p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$) and the Cauchy distribution ($p(x) = \frac{\lambda/\pi}{1+(\lambda x)^2}$). These two were chosen because they are representative of distributions with different types of tails. Both are α-stable distributions. The Gaussian distribution has $\alpha = 2$ implying a finite variance, while the Cauchy distribution has infinite variance and infinite mean with $\alpha = 1$. So the Cauchy distribution is itself heavy-tailed, while the Gaussian distribution is far from being heavy-tailed. With these distributions, a simple recursion relation between l_i as the one in (4.7) is not available. However, similar asymptotic results on the tail of the probability of file transfers can be shown. See Appendix A for the proof of these results. Here we solve the optimization numerically using equation (4.5).

Again, we set $L = 10^4$, and $N = 20$. In both cases, the resulting cumulative distributions P_i^t vs. l_i display heavy tails, as shown in Figure 4.5. The Gaussian case is similar to the exponential one, where most of the resulting l_i cluster around size 1 and a few bigger ones spread out quickly towards L. Again the distribution is not exactly a power law, but the slope of the curve approaches -1.4 in the region of large l_i. With the Cauchy distribution the resulting l_i are evenly distributed on a logarithmic scale and their cumulative probabilities can be very well approximated by a power law with exponent 1.

To summarize, the above results suggest that the optimal layout of a one-dimensional Web site with the objective of minimizing the average download time subject to a constraint on the number of files leads to the heavy-tailed distribution of transferred file sizes with exponent $\alpha \approx 1$. This confirms the observation from the PLR model in Chapter 3, and is consistent with the 1995 data. The result is quite robust with respect to the specific choice of the initial $p(x)$ and whether it is heavy-tailed or not. Although the behavior of the model under distributions other than those tested above is subject to verification, we do believe that the link between the heavy-tailed distribution of Web file transfers and the optimization behind Web layout design and the tradeoff between performance objective and resource constraints is fairly generic. Of course the above study is not conclusive since the model of Web site layout and user behavior is so simplified. The rest of the report addresses the issue of investigating more complex models for Web layout design.

4.2 Web Sites of Multiple Documents

It is possible to extend the model in the previous section to Web sites consisting of multiple documents. Imagine such a Web site is designed by dividing each document into a one-dimensional chain in the optimal way described in the previous section. The goal is to study the statistics of the overall traffic from all the documents. Assume that each document is equally likely to be visited. And because previous study indicates that the optimal design for each document is qualitatively quite independent of the specific choice for the initial $p(x)$, for simplicity we would further assume that the popularity of the information in each document is i.i.d. with an exponential distribution $p(x) = \lambda_0 e^{-\lambda_0 x}$. The analysis of a single document has shown that the key to the resulting heavy-tailed distribution is the recursion relation

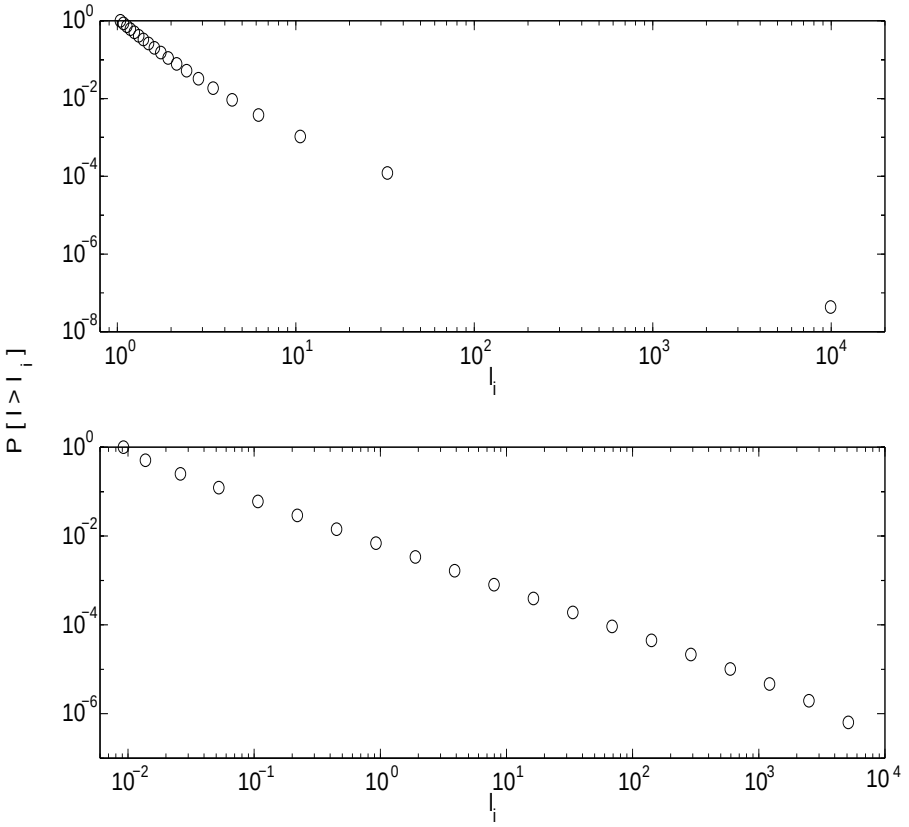

Figure 4.5: P_i^t vs. l_i for Web file transfers. (Top: $p(x)$ is Gaussian, $\mu = -30$, $\sigma = 20$; Bottom: $p(x)$ is Cauchy, $\lambda = 100$.)

(4.7) between sizes of subsequent files. Therefore, if the distribution of the smallest file l_1 in each chain is known, it can be propagated through the recursion equation and obtain the overall distribution of all the files. In general the sizes of small files do not have heavy-tailed distributions, so the distribution of l_1 is assumed to be an exponential function on an interval I_1 with parameter λ_1,[1] i.e., its PDF $f_1(l_1) = k_1 e^{-\lambda_1 l_1},\ l_1 \in I_1$.

Let $h(x) = \frac{e^{\lambda_0 x}-1}{\lambda_0}$. Then $l_i = h(l_{i-1}) = h^{i-1}(l_1)$, for all $i > 1$, where $h^m(\cdot)$ denotes the m_{th} composite function of $h(\cdot)$. The inverse function of $h(x)$ is $g(x) = \frac{\log(\lambda_0 x+1)}{\lambda_0}$, so $l_1 = g(l_2) = g^{i-1}(l_i)$. Then the PDF of $l_i (i > 1)$ is

$$f_i(l_i) = f_1(l_1)\frac{\partial l_1}{\partial l_i} = f_1(g^{i-1}(l_i))\frac{\partial g^{i-1}(l_i)}{\partial l_i}, \tag{4.8}$$

which is a highly complicated function of l_i. However, because each l_i is an exponential function of the previous one, it is conceivable that the cumulative distributions of l_i spread out exponentially as i increases.

The size of the file transferred from this Web site can be considered as a new random variable l_m, whose distribution will be a mixture of all the distributions $f_i(l_i)$. And the mixing probability p_i^t is also a function of the particular value of l_i, which can be calculated as we described in the previous section. By the theory of mixture probability [20], the cumulative distribution $P_m(x) = P[l_m > x]$ is given by

[1] Note that λ_1 is for the distribution of l_1 while λ_0 is for the distribution of user interests in each document.

$$P_m(x) = \sum_{i=1}^{N} \int_x^{\infty} p_i^t(l_i) f_i(l_i) dl_i. \tag{4.9}$$

The above distribution is computed numerically and the resulting P_m vs. l_m is displayed in Figure 4.6 (dashed line). The upper tail of the distribution is very well approximated by a power law with exponent 1.1. To illustrate the cause of this heavy tail, the cumulative distributions of l_i conditioned on transferring file $\#i$ in the chain are also shown (solid lines). The plot is cut off at $x = 10^{10}$ to show to details of how the solid curves add up to give the dashed curve. Here we used $N = 10$, $\lambda_0 = 0.25$, and $\lambda_1 = 1$. The shape of the distribution is quite insensitive to the choice of N and λ_1. As λ_0 gets bigger, the distributions of $l_i (i > 1)$ spread out more so that the exponent of the power law tail gets closer to 1. If λ_0 is too small, no big files will be generated from this iteration, which is of no interest to this study.

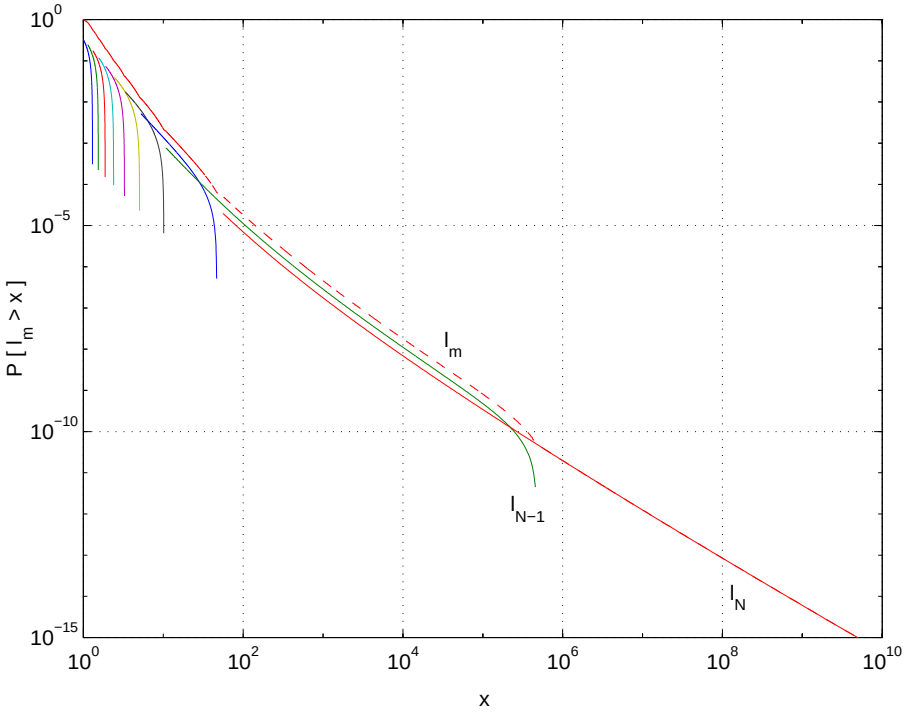

Figure 4.6: Cumulative distribution of l_m.

Chapter 5

Simulations of Graph-Based Web Layout Models

The one-dimensional Web site model is good for analytical study, but is only representative of fairly primitive Web layout design. Recently, the layout of Web sites has become much more complicated and versatile and the design process has gone far beyond just organizing a single document, which means the one-dimensional assumption is no longer suitable. The objective of this chapter is to build a series of models with increasing details of real Web layout and minimize the average download time, i.e.,

$$J = \sum_i p_i l_i \tag{5.1}$$

where l_i are the sizes of individual files on a Web site, and p_i are the corresponding probabilities that each file gets downloaded. Again constraints are posed due to limited resources. The resulting optimization problem fits into the general PLR model in (4.1). Its combinatoric nature implies that analytical solutions will typically not be available. Instead, heuristic optimization schemes have been developed and are presented in this chapter, which mimic some aspects of the tuning and refining involved in real Web site design, and suggest simple methods by which this could be made more systematic.

The next section introduces random graphs as more realistic models for general Web sites. Graph-based models have been employed by many other researchers for topologies of computer networks [46, 15] and the WWW [6, 36, 23, 45]. In [6] the Web was viewed as a large directed graph whose vertices are documents and edges are links pointing from one document to another. In [23] the graph model was applied on a single Web site, which is closer to its application in this chapter — the study of the layout of each individual Web site. The optimization of the average download time is done by heuristically reorganizing (splitting and merging) files based on statistics of user interest. Simulations of different classes of graphs including chains, trees, and incrementally generated networks are described in Section 5.2. In Section 5.3 a more sophisticated model with geometries in individual Web pages is presented as well as simulations with a slightly more involved optimization procedure.

5.1 Web Sites as Random Graphs

Consider a Web site modeled as a directed graph (V, E), where each node $V_i (i = 1, \ldots, N)$ represents a Web page and each directed edge E_{ij} represents a hyperlink pointing from page V_i to page V_j. This model still contains a fair amount of simplification from real Web sites, for instance,

1. The geometry inside each file is ignored, so each file is an abstract node in the graph;

2. Each Web page contains only a single file. All the embedded objects (e.g., images) are considered to be part of that file. Hence, no distinction will be made between a file and a Web page;

3. The effect of caching is not taken into account so that every hit on a Web page leads to a transfer of the corresponding file.

All these assumptions can be removed by adding more complexity to the model. The model described in Section 5.3 is an example of relaxing the first condition. The implication of the no caching assumption will be discussed later along with simulation results.

5.1.1 The Markov Chain Model

A discrete-time Markov chain is used to model user navigation behavior, where all the nodes in the graph are considered as possible states for the random process, and the probability of going from V_i to V_j defines the transition probability p_{ij}. Let $M = [p_{ij}]$ be the transition probability matrix, then $M^n = [p_{ij}(n)]$ gives the n-step transition probabilities. The following theorem is from the standard theory of Markov chains.

Theorem 5.1 *[30] For an irreducible, aperiodic, and positive recurrent Markov chain,*

$$\lim_{n\to\infty} p_{ij}(n) = p_j, \qquad \textit{for all } j,$$

where p_i are the unique nonnegative solutions of the following equations:

$$p_j = \sum_i p_i * p_{ij}, \tag{5.2}$$

$$\sum_i p_i = 1. \tag{5.3}$$

A Markov chain that satisfies the conditions stated in the theorem is called ergodic, which is true for the Markov chains defined in all the models in this chapter. Asymptotically, an ergodic Markov chain settles into a stationary random process with state probabilities p_i independent of the time instant n. Each p_i also corresponds to the long-term proportion of time spent on state V_i, in other words, the probability of page V_i being accessed, or downloaded, which is exactly the p_i in the cost function (5.1). Suppose the transition probability matrix M is know, then p_i can be computed by solving equations (5.2) and (5.3). More specifically, let $p = [p_1, \ldots, p_N]^T$ (column vector), we can write (5.2) in its vector form, which is

$$p' = p'M \qquad \text{or} \qquad p = M'p. \tag{5.4}$$

This means p is the right eigenvector of M' with an eigenvalue 1. For real Web sites, p_i can be directly calculated from log data with sufficiently large sample size. More advanced technology even allows the server to trace every link that a user follows, which provides information on the transition probabilities p_{ij}, or, the matrix M, from which p_i can be obtained.

To further define the Markov process, page V_1 is set as the entry point of every user's navigation, assuming that a user starts from the front page and then proceeds to subsequent pages through hyperlinks. So after downloading page V_i, the user follows hyperlink E_{ij} to

visit page V_j with probability p_{ij}. All $p_{i1}(i > 1)$ are viewed as probabilities of exiting the Web site from page V_i, which means the user either stops navigating or goes on to other Web sites. Here instead of defining another node as the exiting state, we let all users go to node V_1 when they exit because the overall statistics are an aggregation of many users' navigation behavior, and there is no way to tell the difference between user A going back to V_1 and starting over again and user B entering at V_1 and beginning to navigate.

5.1.2 Optimization through Splitting and Merging

Suppose the original Web layout has N files connected as a graph. The length of each file is l_i, and the probability of each file being downloaded is p_i. A naive way to reduce the average download time is to split high hit files and merge unpopular ones. In particular, if a file is popular and large at the same time, split it into two smaller files. Repeatedly doing this will certainly lower the cost defined in (5.1). However, the total number of files N increases by 1 with every split. Eventually the number of files becomes too large, which leaves the Web site too hard to manage and users frustrated by going through too many clicks. Therefore, as a coarse approximation of the tradeoff between the cost and the above concerns, an upper limit is posed on the total number of files, as in the one-dimensional Web layout model. To meet this constraint, every time one file gets split up, two other files that neighbor[1] each other and are rarely visited are chosen to be merged into a larger file. This is the basic idea behind the heuristics used in the optimization. Either splitting or merging changes the topology of the graph, the corresponding Markov chain, and the distribution of the file sizes. The details involved in splitting and merging depend on individual models, and will be discussed in the next section on specific simulations.

5.2 Simulations on Random Graphs

5.2.1 Initialization of the Graph Model

The first step of the simulation is to create the initial topology of the graph. Two parameters are chosen to provide simple control over the key structural characteristics of a Web site: N_{node}, the number of nodes in the graph, determines the scale of the Web site under study, and N_{link}, the number of incoming links at each node, governs the degree of connectivity between the Web pages. The graph model is built in an incremental way to mimic the generation of real Web sites, where people initially put up some Web pages with hyperlinks in between, then gradually add more pages and connect them with the existing pages, and keep growing the Web site in an iterative fashion. Each node V_i has coordinates (x_i, y_i), which are generated uniformly in a 10×10 square in $\mathbb{R}^2$. This is for the convenience of modeling the logic relation between different pages. The relevance of information between pages V_i and V_j is represented by their distance in Euclidean space, the same idea used in Waxman's network model [46]. Each graph usually starts with a few nodes. Every time a new node is generated, it is connected with N_{link} of the existing nodes that are closest to it with bidirectional links. This process is repeated until all N_{node} nodes are generated and connected. Figure 5.1 shows one graph model generated by the above scheme. V_1 corresponds to the starting point of the navigation.

Notice that most nodes are linked to nearby neighbors, while there are occasionally rather long links connecting neighbors that are distant from each other on the graph. This property is similar to the small world phenomena present in the WWW, discussed in [1]. The reason is

[1] Two nodes are called "neighbors" if they are connected by hyperlinks.

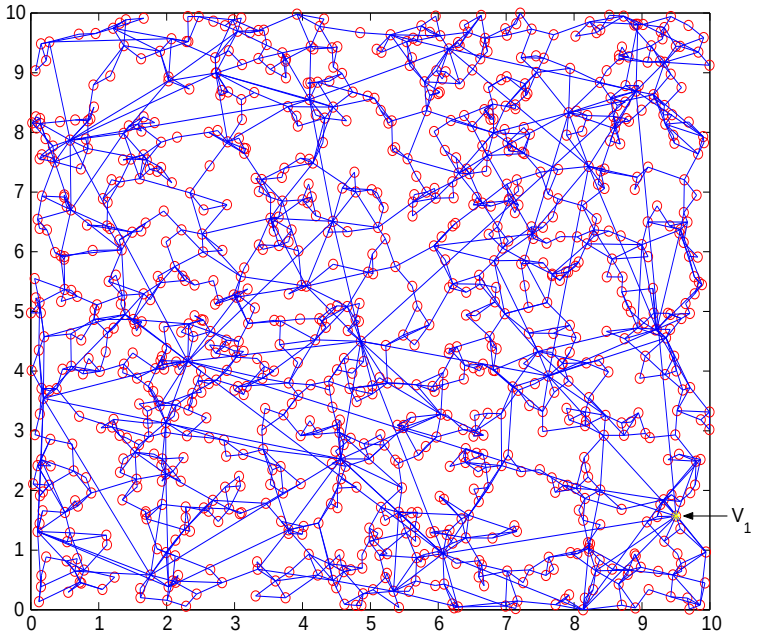

Figure 5.1: Graph model of a random Web site with $N_{node} = 1000$ and $N_{link} = 2$.

that the graph is built incrementally. So these neighbors that are far away are those generated earlier when there were only a few nodes on the graph, while all the others were generated subsequently. This indicates some intrinsic hierarchy that is present in many Web sites, where the front page connects to a small number of key pages with higher level information, each of them surrounded locally by a large number of lower level pages with more detailed information.

Real Web site layout could be fairly different from one to another. Besides the two parameters N_{node} and N_{link} chosen to control the scale and connectivity of a Web site, the randomness in the locations of the nodes and consequently the connections between them are used to provide the degree of freedom in the topology. The difference in the degree of connectivity between $N_{link} = 1$ and $N_{link} = 3$ is clearly demonstrated in Figure 5.2.

After interconnections between the Web pages are determined, the user navigation pattern is modeled by assigning the initial transition probabilities p_{ij} of the Markov chain. For each node V_i, simply assume that there is one common exiting probability p_e except for V_1, i.e., $p_{i1} = p_e (i > 1)$. The remaining probability $1 - p_e$ will be evenly distributed among all the outgoing links $E_{ij} (j > 1)$. Set $p_{ij} = 0$ if there is no link between two nodes V_i and V_j. Note that the particular Markov chains defined here do not have self-loops, i.e. $p_{ii} = 0$. After uniquely determining the whole transition probability matrix M, the download probabilities p_i can be computed through eigenvalue decomposition, or solving (5.4).

The last step of the initialization is to decide the file size l_i for each page. Three cases have been tested: the exponential distribution, the uniform distribution in $[l_{min}, l_{max}]$, and the degenerate case where $l_i = L$, for all i. In the next section, the simulation results with initial exponential distributions for l_i are shown. The results with the other two kinds of distributions are fairly similar, hence will be omitted here.

5.2.2 Heuristic Optimization and Simulation Results

The heuristic optimization algorithm goes as follows:

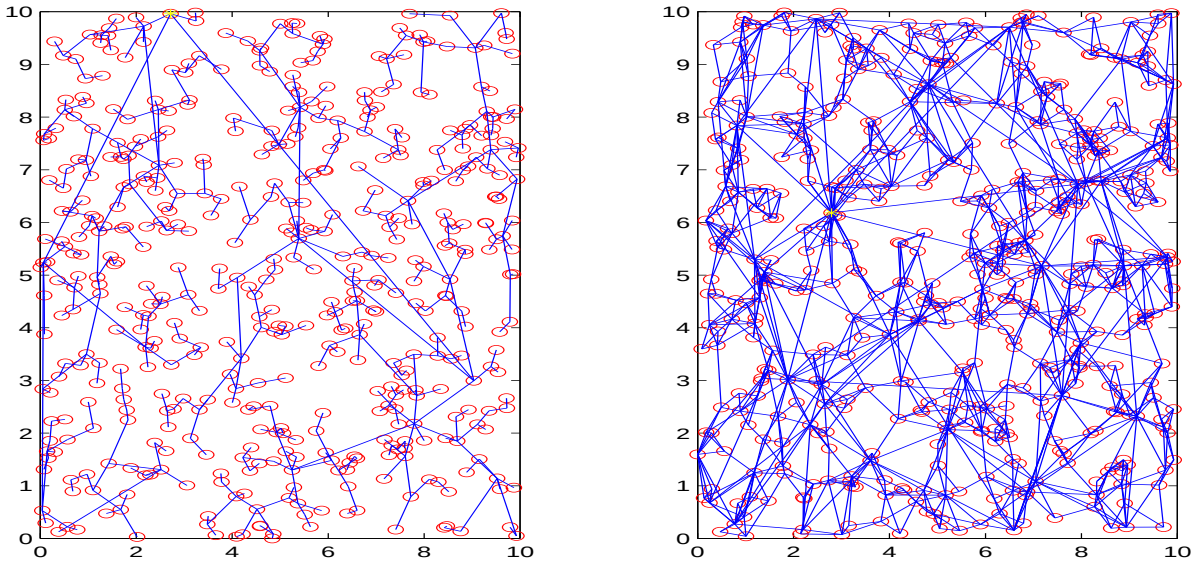

Figure 5.2: Graph models of two random Web sites with $N_{node} = 500$ and different degrees of connectivity. Left: $N_{link} = 1$; Right: $N_{link} = 3$.

Splitting Pick page V_{i^*} with the highest $p_i l_i$. Cut the page in the middle to produce two new pages $V_{i'}$ and $V_{i''}$ with sizes $l_{i'} = l_{i''} = l_i/2$. Any outgoing link E_{ij} becomes $E_{i'j}$ and $E_{i''j}$ without changing the outgoing probability, i.e., $p_{i'j} = p_{i''j} = p_{ij}$. Meanwhile, any incoming link E_{ji} becomes $E_{ji'}$ and $E_{ji''}$ with half incoming probability, i.e., $p_{ji'} = p_{ji''} = p_{ji}/2$. The download probability vector p remains almost the same except that $p_{i'} = p_{i''} = p_i/2$. It is easy to prove that the updated M and p still satisfy equation (5.4).

Merging After splitting one page, find page V_{k^*} with the lowest p_k and its least popular neighbor $V_{k^{**}}$. Combine them into one page V_k with size $l_k = l_{k^*} + l_{k^{**}}$. Copy all the the outgoing links and incoming links for V_{k^*} and $V_{k^{**}}$ into the new page, then merge redundant links and combine probabilities, i.e., $p_{jk} = p_{jk^*} + p_{jk^{**}}$, and $p_{kj} = \frac{p_{k^*}}{p_{k^*}+p_{k^{**}}} p_{k^*j} + \frac{p_{k^{**}}}{p_{k^*}+p_{k^{**}}} p_{k^{**}j}$. The self-loop produced by merging should be removed and all $p_{kj}, j \neq k$ need to be adjusted appropriately so that each row of M still sums up to 1. Recompute the download probability vector p using equation (5.4).

Iteration Update $J = \sum_i p_i l_i$. Repeat the above splitting-merging procedure until the improvement of J is within a certain tolerance level or the number of iterations reaches a preset maximum.

The algorithm described above is the most naive version so that it contains the key operations and is easy to understand. The real implementation involves a fair number of variations of the above algorithm. For example, the criterion for choosing the page to split or the pages to merge can be different; while splitting, the location of the cut does not have to be in the middle, instead, it can be chosen at random; there are other ways to reconfigure the hyperlinks after splitting or merging the pages; parameter values can be varied from one test to another; etc..

Simulations on a large number of random graph models with different implementations of the proposed algorithm have shown that the simple optimization scheme works quite well in continuously improving the cost J. To study the distribution of file transfers, let L^t be

the random variable representing the transferred file sizes. Then for every data set (l_i, p_i), calculate the cumulative probability $P_i^t = \sum_{j \le i} p_j$, with sizes of individual files ordered as $l_i \ge l_{i+1}$, i.e. $P_i^t = P[L^t \ge l_i]$.

The simulation result on a random Web site model with $N_{node} = 1000$ and $N_{link} = 2$ is shown in Figure 5.3, Figure 5.4 and Figure 5.5. Figure 5.3 illustrates the effectiveness of the naive splitting-merging heuristic in reducing the cost J iteration by iteration. The improvement is quite dramatic at the beginning, then it slows down and finally reaches some steady state after 350 iterations.

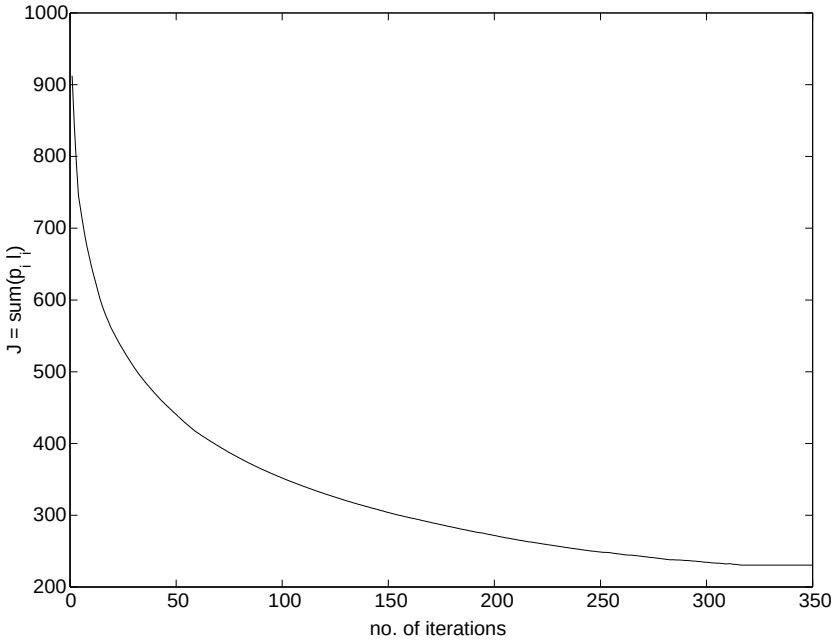

Figure 5.3: Cost function J vs. no of iterations, $N_{node} = 1000$, $N_{link} = 2$.

Figure 5.4 shows the cumulative distribution of file transfers P_i^t vs. l_i before and after the optimization. Motivated by the model (3.7) for the noncumulative p_i in Chapter 3, the following hyperbolic function

$$P_i^t = c_1 (l_i + c_2)^{-1/\beta} \tag{5.5}$$

was used to fit the cumulative distribution P_i^t after the optimization. Note that the exponent in the equation becomes $-1/\beta$ instead of $-(1 + 1/\beta)$ in (3.7) due to the integration. The resulting model (dashed line in Figure 5.4) with $c_1 = 7.85 \times 10^5$, $c_2 = 318.28$ and $\beta = 0.42$ well approximates the simulation result over many orders of magnitude in the file size, which verifies that the distribution after the optimization displays a power law tail. Although the resulting value for β implies an exponent $\alpha = 1/\beta > 2$, which means the distribution is not really heavy-tailed as defined in Chapter 2, it does have an upper tail that is much heavier than the initial distribution before the optimization, which drops off exponentially at large l_i. In fact, the optimization process pushes the whole distribution curve down at small and medium file sizes to reduce the average download time. As a tradeoff, the optimized Web site would produce more large file transfers than the original Web site.

Interestingly, we find that the distribution of the sizes of *unique* files on the Web site, as opposed to the traffic from the Web site, also has a nice power-law tail after the optimization, as displayed in Figure 5.5. Let L^u denote the random variable of unique file size on the Web

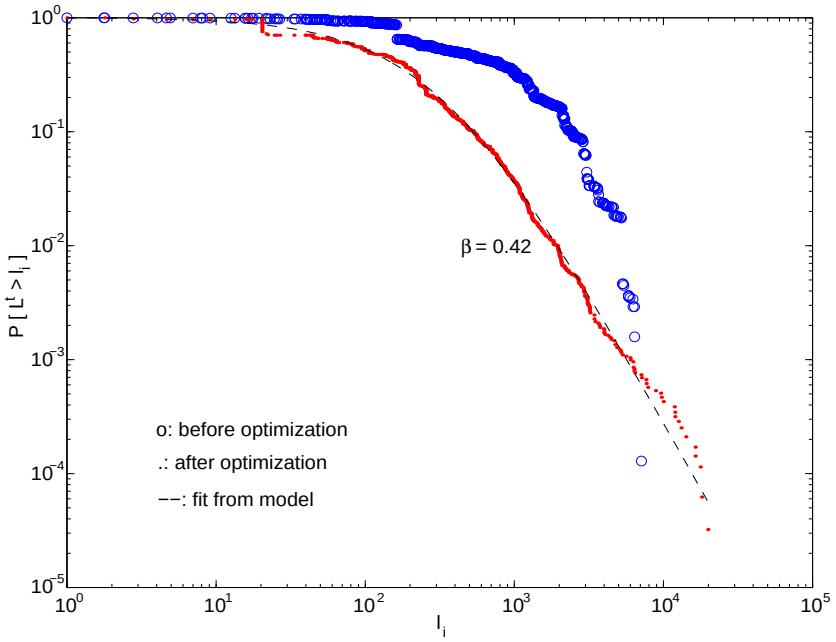

Figure 5.4: P_i^t vs. l_i for Web file transfers before optimization and after 350 iterations, $N_{node} = 1000$, $N_{link} = 2$.

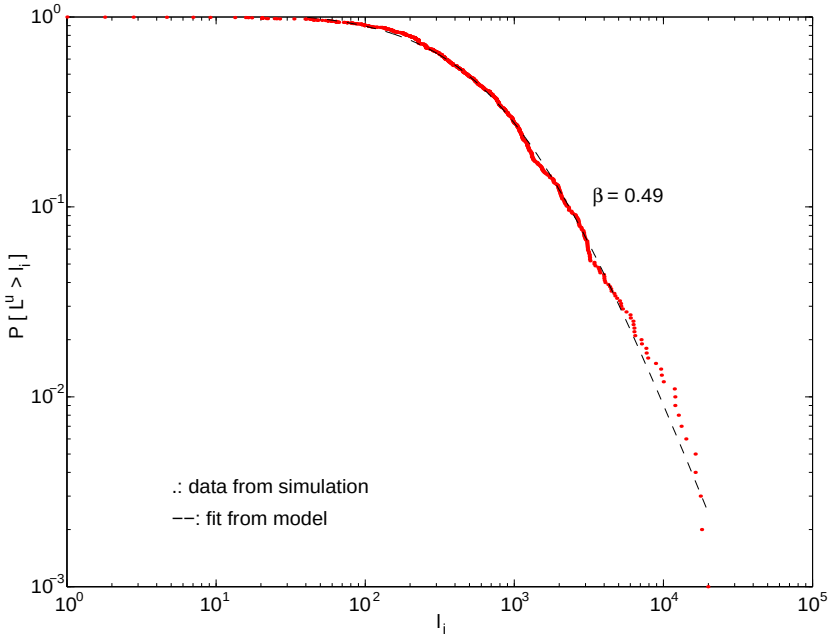

Figure 5.5: P_i^u vs. l_i for unique files after 350 iterations, $N_{node} = 1000$, $N_{link} = 2$.

site, then P_i^u denotes its complementary cumulative distribution, i.e., $P_i^u = P[L^u \geq l_i]$. Again the model in (5.5) can be used to fit P_i^u vs. l_i. The resulting parameters are $c_1 = 1.40 \times 10^6$, $c_2 = 1.04 \times 10^3$, and $\beta = 0.49$, indicating a "heavier" tail than the distribution for the transferred files. The heavy-tailed distribution for unique file sizes has also been observed from both the 1995 data and the 1998 data in [12, 7]. In [12] it was shown that the set of file transfers is intermediate in characteristics between the set of file requests and the set of

unique files. This is due to the use of caching in Web applications. At one extreme, if no caching is provided, then the set of transferred files should be the same as the set of files that are requested, which is a simplification taken in this report. On the other hand, in the case of perfect (infinite) caching, the set of files transferred should be identical to the set of unique files. In reality, caching does exist but is never infinite, so the distribution of resulting file transfers should be between the distributions of file requests and unique files. Back to the simulation results shown in Figure 5.4 and Figure 5.5, the distribution of L^t is less heavy-tailed than the distribution of L^u because small files get requested more often than large files. If caching is taken into account, then the distribution of file transfers should have a β value between 0.42 and 0.49 for this particular example. The effect of caching is not the focus of this report, but will be an interesting subject for future research.

These simulation results are quite robust with respect to the choices that can be made during the implementation of the optimization algorithm, as was discussed earlier in this subsection. Of course the objective function J may decay in different trajectories, and the resulting exponents of the power law tails can be slightly different, but the qualitative behavior of the same example of the Web layout model is fairly consistent in all the simulations.

Since the above example was generated at random, it may not be representative of the general characteristics of Web layouts of similar structure, say with the same number of N_{node} and N_{link}. Therefore, the above design and optimization procedure was also simulated on a subnet consisting of K random Web sites, each with $N_{node} = 1000$ and $N_{link} = 2$. All the Web sites are considered equally probable to be accessed. So the overall statistics are just the average of those for individual ones. After the simulation all K data sets of (l_i, p_i) were mixed together and the total cumulative distributions P_i^t of transferred files and P_i^u of unique files were computed. Both distributions contain power law tails, which are shown in the top two plots of Figure 5.6, compared with the distributions before the optimization that have exponential drop off at large file sizes. Again the hyperbolic model (5.5) was used to fit the simulation results, giving $\beta = 0.44$ for P_i^t and $\beta = 0.55$ for P_i^u. This means the distribution of transferred files has an exponent α slightly above 2 and the distribution of unique files has an exponent α slightly below 2. Comparing these exponents with the exponents obtained from the PLR model ($\alpha = 1$) and the one-dimensional Web layout model in Chapter 4 ($\alpha \approx 1$), it is easy to see that the distributions of file transfers from these graph models are less heavy-tailed than those from simpler models. This difference is consistent with the change in the empirically observed behavior of Web file transfers from the 1995 data to the 1998 data. A plausible explanation will be provided after more simulation results are presented.

The algorithm was also tested with different choices of N_{link}. Only the cumulative distributions of the transferred files L^t for $N_{link} = 1$(left) and $N_{link} = 3$(right) are displayed at the bottom of Figure 5.6. The distributions of the unique files L^u are not shown, but they have similar shape as in the $N_{link} = 2$ case (top right), and also are more heavy-tailed than the distributions of L^t with the same N_{link}. In the $N_{link} = 1$ case, P_i^t after the optimization can also be nicely fit using (5.5) with $\beta = 0.52$, indicating a power law tail that is heavier than that for $N_{link} = 2$ (top left). However, for P_i^t with $N_{link} = 3$, no perfect fit using (5.5) can be obtained since the distribution starts to drop off more quickly after $l_i \approx 10^5$. The best fit found leads to $\beta = 0.42$. This implies that the distribution for file transfers is even less heavy-tailed than for $N_{link} = 2$.

Comparison of these three cases suggests that as N_{link} becomes higher and higher, the tails of the cumulative distributions of transferred files become less and less heavy-tailed. The parameter N_{link} captures the degree of connectivity of a Web site layout. The higher the N_{link}, the more connected the Web site. The consequence is the parameter β that fits the power law distribution of file transfers varies with the degree of connectivity. The PLR model

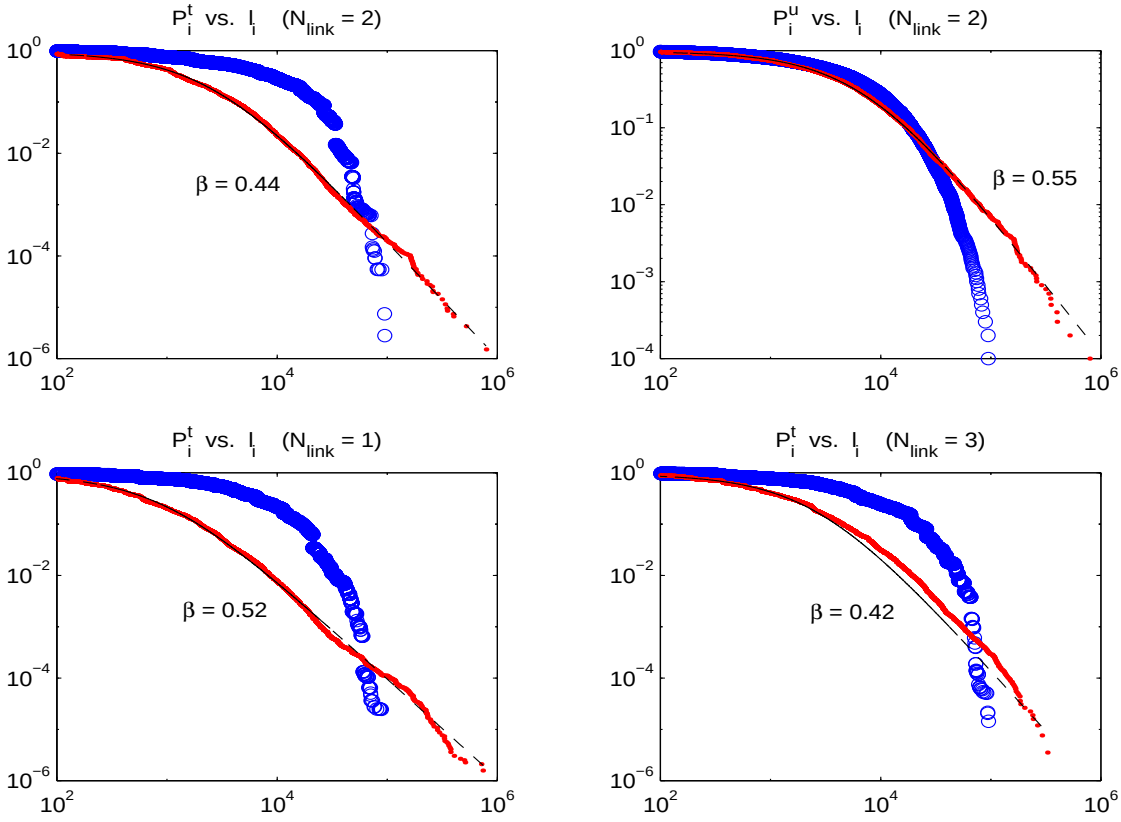

Figure 5.6: Mixed P_i^t vs. l_i for transferred files and mixed P_i^u vs. l_i for unique files from 10 random Web sites, each with $N_{node} = 1000$, before and after the optimization.

characterizes one-dimensional Web layout with a very low degree of connectivity, $\beta = 1$; while the graph model for more connected Web layouts gives $\beta \approx 0.5$, and β becomes smaller as N_{link} increases. This observation provides a possible explanation for why the power law tail from the 1998 data was steeper than that from the 1995 data. The reason could be that from 1995 to 1998 the design of Web layout became more and more mature and sophisticated, including more extensive use of hyperlinks that led to greater connectivity in resulting Web sites, which, in turn, would demonstrate less heavy-tailed distributions in file transfers.

5.2.3 Two Special Cases

In the previous subsection, Web site topologies were modeled as incrementally generated graphs. This subsection studies two special cases of random graphs: chains and trees.

Chains

The simplest possible layout for a Web site is a chain-like structure, where all the Web pages are sequentially connected by unidirectional links, like the one in Figure 4.3. As in the general case, each Web page V_i contains a single file with length l_i, and the user enters the Web site at the first page V_1. At each page V_i, he has the choice of either leaving the Web site with probability p_e or following the hyperlink $E_{i,i+1}$ to browse the next page V_{i+1} with probability $1 - p_e$.

The splitting-merging iteration is quite trivial to implement in this special case. And the

chain structure is preserved during the operations. Again small variations can be added to the standard algorithm, like in the case of general graphs. For example, split a file at a random location instead of in the middle, or pick p_e randomly in a certain range instead of at a fixed value. Simulations of different combinations of the above choices all give similar results. An example is demonstrated in Figure 5.7, where the cumulative distributions of the sizes of file transfers P_i^t vs. l_i before and after the optimization are shown. The distribution from the optimized Web site displays a very nice power law decay with an exponent $\alpha \approx 1$ in the whole range of l_i. Experiments of different N_{node} and p_e show that this slope is fairly independent of the parameter values, which validates the theoretical result from the one-dimensional Web layout model in Chapter 4.

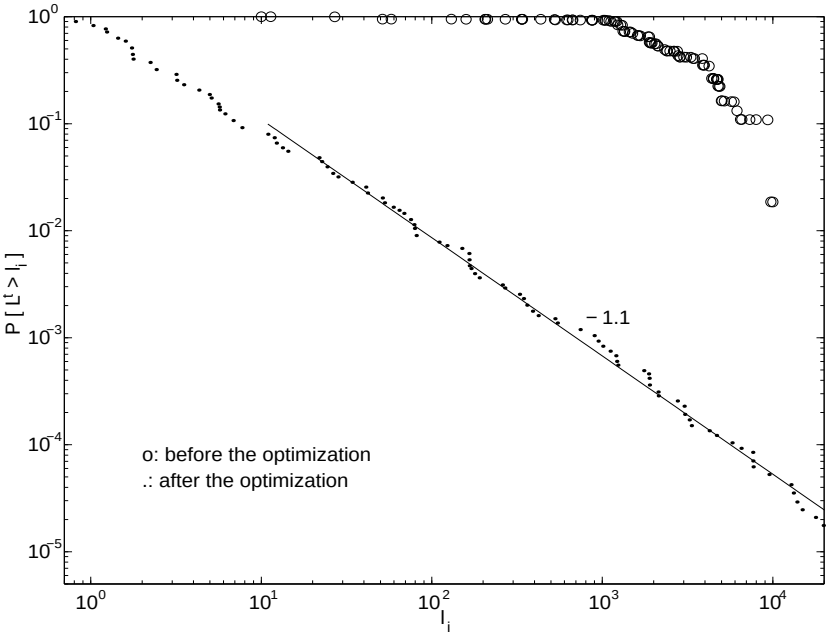

Figure 5.7: P_i^t vs. l_i for file transfers in a chain-like Web site before and after the optimization, $N_{node} = 100$.

Trees

Another special topology is the rooted-tree. It is a simplification of the implicit hierarchical structure that is present in many Web sites nowadays. The same incremental scheme for generating the initial topology in the random graph case is used for the tree structure, except that $N_{link} = 1$ and one more parameter needs to be specified: the maximum out-degree M_{out} for all the nodes. Note that M_{out} and N_{node} together determine the depth and the degree of connectivity of the initial tree.

The standard splitting-merging algorithm for the general graphs has to be modified so that the tree structure can be preserved during the optimization. The simulation result from one example is shown in Figure 5.8, which contains the cumulative distribution of sizes of file transfers P_i^t vs. l_i before and after the optimization. The distribution from the optimized Web site has a little curvature. It follows a power law with α close to 1 for small l_i, and the slope becomes steeper as the file size increases. A linear fit to the curve in the large l_i region gives $\alpha = 1.53$, which still suggests a heavy tail. Again this result in general does not depend on the specific values of the parameters N_{node} and M_{out}, although smaller M_{out}

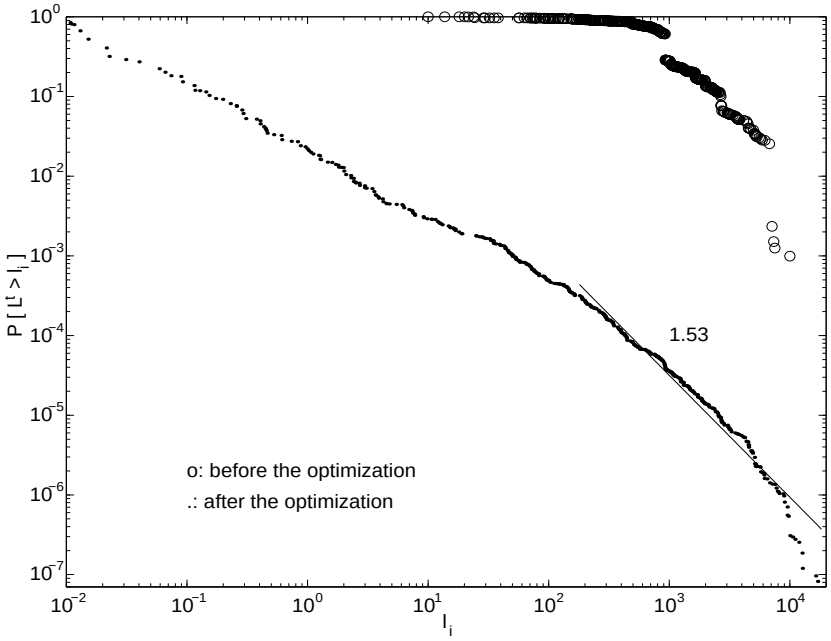

Figure 5.8: P_i^t vs. l_i of file transfers from a tree-like Web site before and after the optimization, $N_{node} = 500$, $M_{out} = 25$.

usually produces a heavier tail. A possible explanation for this is when N_{node} is fixed, a tree with smaller M_{out} spreads out less and goes down more deeply, which is closer to the one-dimensional chain case that produces a power law tail with $\alpha \approx 1$.

5.2.4 Summary

Comparison of the simulation results from chains, trees, and more general random graphs with different N_{link} suggests that the exponent of the power law tail in the distribution of file transfers from the optimized Web site is highly correlated with the initial degree of connectivity of the Web site. Among the three topologies that were studied, the chain structure has the least connectivity, the random graph with large N_{link} has the greatest connectivity, while the tree, similar to the random graph with $N_{link} = 1$,[2] is between those two extremes. At the same time, the exponents of the tails show a similar pattern. We could conclude that the more connected the graphs, the less heavy-tailed are the resulting distributions.

5.3 A More Realistic Model

5.3.1 Generation of Web Page Geometry

The Web layout models presented in the previous section treat every Web page as an abstract node in a random graph without any geometry in it. Real Web pages are (mostly) HTML files containing embedded objects (images, etc.), distributed hyperlinks, and complicated layout. In this section a more realistic Web site model is introduced, which takes into account the internal structure of each Web page. Since it is based on the random graph

[2]The slight difference between these two is that in the tree case, the $N_{link} = 1$ property is preserved throughout the optimization.

model, only the new features that are not in the previous model are described here. So each page V_i still contains a single file with length l_i, but each file is divided into one or more paragraphs to coarsely capture the logical connection between information. Different paragraphs are considered relatively independent, and it is ok to split them and move them into different pages when necessary. The number of paragraphs in each file and the locations of the paragraph breaks are generated randomly. In addition, the hyperlinks that point out of this page can be located anywhere in the file. To avoid confusion, the model used in the previous section is referred to as the graph model, while the new model used in this section will be referred to as the geometric model.

Similar to the graph model, user navigation of the Web site always starts at page V_1. Inside each page V_i, assume that the user reads the file from top down. Again user interest $p(x)$ is assumed to be an exponential function of the location of the information x, with parameter λ, as has been studied in Chapter 4. λ can be easily adjusted to change the decay of user interest. When $\frac{1}{\lambda} \gg l_i$, then user interest is distributed almost uniformly throughout the whole page. If there is a hyperlink E_{ij} at location x_0, the user either follows the link to visit page V_j with probability P_n or keeps going in V_i. P_n is defined as the navigation probability, and treated as a constant throughout the whole Web site. Suppose there are k hyperlinks $E_{ij_1}, \ldots, E_{ij_k}$ in page V_i, and the probabilities of following these links are $p_{ij_1}, \ldots, p_{ij_k}$ respectively. Then $p_{i1} = 1 - \sum_{s=1}^{k} p_{ij_s}$ is defined to be the exiting probability.

The optimization algorithm for the geometric model is more involved than for the graph model, because we need to keep track of the paragraph structure and the locations of the hyperlinks in the original Web pages. Also locations of the cuts while splitting files are determined by assumptions on user interests, more specifically, on the parameter λ and how $\frac{1}{\lambda}$ is compared to l_i. Moreover, unless there is only one paragraph, the splitting point has to be between paragraphs. The relative locations of the hyperlinks in each paragraph will be preserved whenever files are split or merged. Every iteration of the splitting-merging procedure produces a new Markov chain, and there are no simple ways to obtain the new stationary probabilities p_i just by updating the old ones. So p_i will be recomputed using the updated transition probability matrix M at each iteration, which makes the computational cost higher than that of the graph model.

5.3.2 Simulation Results

A large number of simulations of the geometric model with different parameter settings have proven the effectiveness of the simple splitting-merging heuristic in reducing the cost function J, i.e., the average file download time from the Web site. One example of a Web site with 1000 nodes and $N_{link} = 2$ is demonstrated in Figure 5.9.

Figure 5.10 and Figure 5.11 show the cumulative distributions of sizes of transferred files (L^t) and unique files (L^u) from the same simulation as in Figure 5.9. Both distributions display power law tails, with $\beta = 0.52$ for P_i^t and $\beta = 0.81$ for P_i^u by fitting the hyperbolic model (5.5) to the simulation data. The fit here is not as good as those for the graph model because the distributions from the geometric model are less smooth than the previous ones. In addition, the β values from the fit are lower than those from the graph model, indicating that the distributions are more heavy-tailed. However, the qualitative behavior of the two models are fairly similar. In particular, the optimization process always reshapes the distribution of file transfers and results in a more heavy-tailed distribution than the original one without the optimization. Moreover, the distribution for the unique files is always more heavy-tailed than the distribution for the transferred files. And as was discussed in the previous section, if caching is taken into account, then the real distribution of the transferred files should be

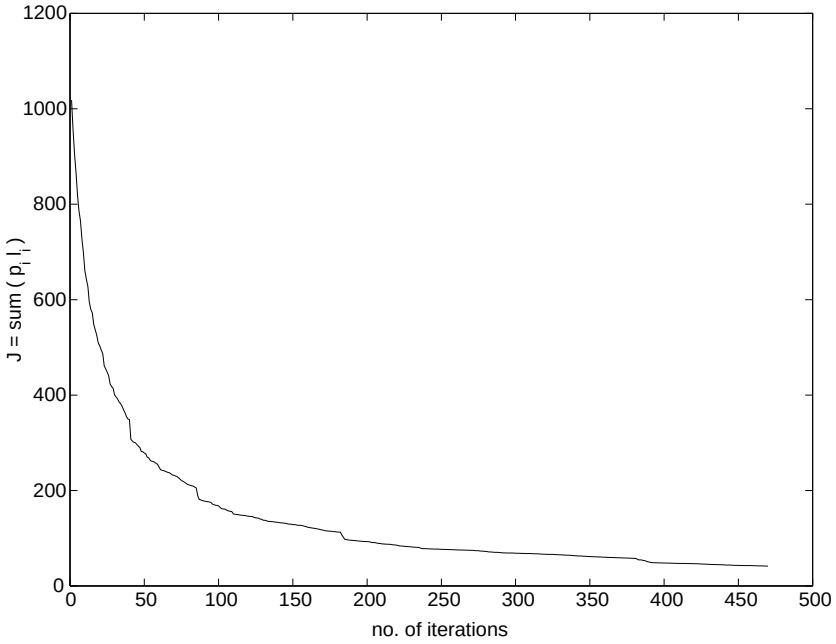

Figure 5.9: Cost function J vs. no. of iterations, $N_{node} = 1000$, $N_{link} = 2$.

something between the P_i^t and P_i^u obtained from this model.

The simulation was also conducted on a subnet consisting of K random Web sites with page geometries, where each Web site has $N_{node} = 500$ and $N_{link} = 2$. The probabilities of visiting individual Web sites are assumed to be equal. The results are shown on the top two plots of Figure 5.12. Again both cumulative distributions for the mixed L^t and L^u contain power law tails and the tail of P_i^u ($\beta = 0.89$) is heavier than that of P_i^t ($\beta = 0.58$). The cumulative distributions of the transferred files from two other simulations with $N_{link} = 1$ and $N_{link} = 3$ are shown in the bottom two plots of Figure 5.12. Both have power law tails with $\beta = 0.58$ and $\beta = 0.56$ respectively. Although the fit using (5.5) is not as good as those for the graph model, as with the single Web site case, it still captures the basic shape of the distributions. However, with the geometric model, the value of β from the fit is fairly stable for Web layouts with different N_{link}, which is not the case with the graph model. It seems that the internal structure of individual Web pages blurred the difference in the connectivity caused by the difference in the number of links. A full understanding of its causes and implications will require more simulations with different kinds of Web layout models, and more importantly, a combination of simulations and an empirical study of real Web sites. What is consistent between the graph model and the geometric model is that the resulting distributions for Web file transfers all have power law tails with exponent $\alpha \approx 2$, or $\beta \approx 0.5$ in the hyperbolic model (5.5). This is much less heavy-tailed than the distributions from the PLR model or the one-dimensional Web layout model where $\beta \approx 1$. The relation between this simulation result and empirical data from [12, 7] as well as a possible explanation have been given in the previous section with the graph model, hence will not be repeated here.

5.4 Future Improvements of Graph-Based Models

Although the geometric model has a higher level of realism than the simple graph model, it is still an abstraction of real Web sites nowadays. Although we do believe that these models capture some important features of Web layout design that provide a plausible explanation of

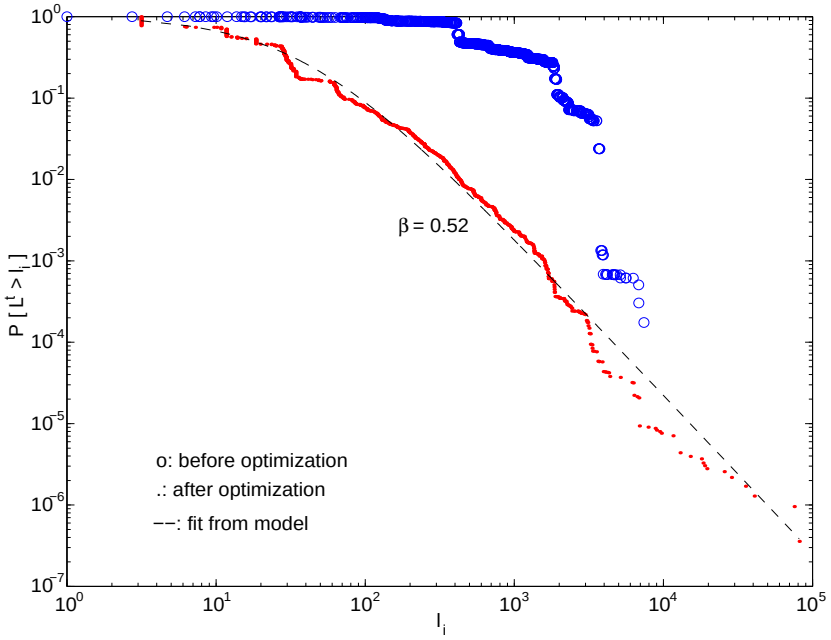

Figure 5.10: P_i^t vs. l_i for Web file transfers before optimization and after 470 iterations, $N_{node} = 1000$, $N_{link} = 2$.

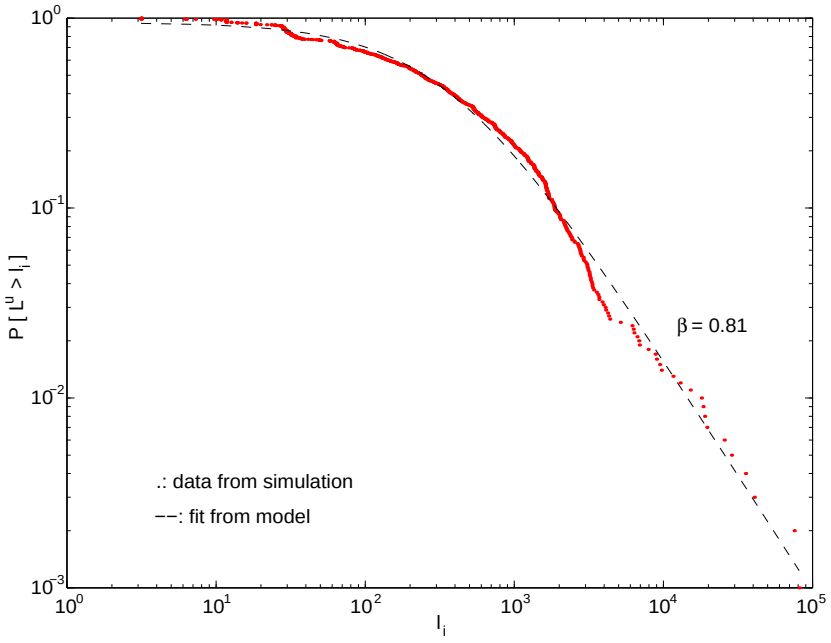

Figure 5.11: P_i^u vs. l_i for Web file transfers after 470 iterations, $N_{node} = 1000$, $N_{link} = 2$.

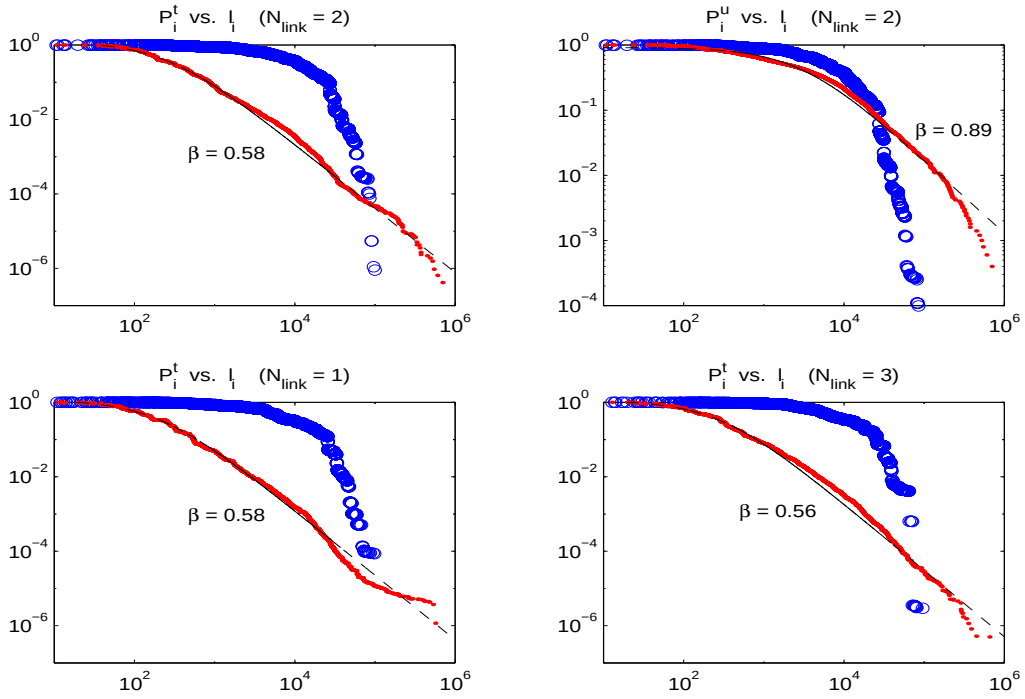

Figure 5.12: Mixed P_i^t vs. l_i for transferred files and mixed P_i^u vs. l_i for unique files from 10 random Web sites, each with $N_{node} = 500$, before and after the optimization.

the origin of heavy-tailed distributions of Web file transfers, there are many more structures and characteristics of real Web sites that are not yet included in our abstract models. The following are some of the issues that are worth looking into.

First of all, the Markov chain model for user navigation, which assumes that the probability a user takes a certain hyperlink E_{ij} from page V_i is independent of the pages the user has visited before V_i, is a simplification of perhaps more intricate user behavior. The validity of this approximation needs to be verified by empirical studies of user access patterns on real Web sites. In the case when user navigation demonstrates strong memory so that the Markov chain assumption cannot be used, the idea proposed in [23] that the average probability that a user visits the next page decays with the number of pages this user has already visited during each Web session can be adopted to modify our model. With this assumption a fixed transition probability matrix M does not exist, so the probabilities p_i of each Web page being visited cannot be calculated directly. However, by repeatedly simulating user access to the Web site, p_i can be well approximated by the average hit rate $\hat{p}_i$ of each page as long as the number of simulations run is sufficiently large.

Second, both of the graph models we have discussed did not distinguish between text (HTML) files and multimedia such as images, audio and video files. When these files are embedded in the HTML files, they are treated as part of a large file so that they can be reorganized during the splitting and merging procedure. However, multimedia files are hard to split and merge. The paragraph structure in the geometric model captures this feature to certain degree. But since each embedded object has a unique URL and contributes to the Web traffic as an individual file, it is desirable to incorporate them into our graph model so that their impact on the heavy-tailed distributions of Web file transfers can be evaluated. This is especially important since the traffic generated by real-time multimedia applications has become an increasingly larger contributor to the overall Web traffic. One thing worth

pointing out is that according to a study by Crovella et al. in [12] on the relationship between distributions of different types of files, although the presence of multimedia formats does add to the weight of the tail in the overall distribution, the distribution of text files itself is heavy-tailed.

Finally, so far we have taken a relatively static view of files on the Web. With the fast development in Web technologies, Web contents nowadays become more and more dynamic. Many Web files are generated on real time when a user is browsing the Web site. It will be a challenging and interesting task for future research to study the impact of dynamic contents on Web site layout design and heavy-tailed Web traffic.

Chapter 6

Concluding Remarks

6.1 Summary and Discussion

This report has presented a series of models of Web design which treated the layout of a Web site as an optimal design problem. The design objective was assumed to be the minimization of the average latency that the user experiences in downloading files while browsing the Web site, subject to constraints on the total number of files. For simplicity and to make contact with the data collected by Crovella et al. in 1995, when the WWW just started to take off, it was assumed that Web sites in their earlier stage consisted of preexisting one-dimensional documents that were simply split into files and linked trivially. With either the PLR model in Chapter 3 or the one-dimensional Web layout model in Chapter 4, it was found consistently that the resulting distribution of Web file transfers had a power law tail with exponent $\alpha \approx 1$. In Chapter 5 more sophisticated Web layouts were explored, which viewed general Web sites as random graphs and simulated the design process using heuristic optimization schemes. Again the resulting distribution of transferred files from the optimized Web site displays a power law tail with exponent $\alpha \approx 2$. The observation that the distributions of Web file transfers become less heavy-tailed with more complex Web layouts is also consistent with the change from the 1995 data to the new data collected in 1998. An implication of this result is that optimal Web layouts and more effective use of hyperlinks may tend to produce much less bursty network traffic, which will be an interesting subject in our future study.

Does the idealized problem described here actually explain the data, or is the remarkable agreement simply a coincidence? Even more important questions are what are the implications of these results for Web design, network traffic, network performance, and protocol design? We will attempt to give tentative answers, but since this work is relatively new, we must caution that any answers are necessarily fairly speculative.

It is unlikely that early (or current) Web designers had any explicit intent to optimally design the layout of their Web sites, so we would expect individual Web sites to deviate from optimal to varying degrees. It is likely however that even the most ad hoc approach to Web layout would naturally make small home pages, with links to increasingly larger files as one moved deeper into the Web site. Furthermore, the natural layout of documents that predates the Web involves hierarchical structures with heavy tails, with titles, abstracts, introductions, summaries, reviews, chapters, appendices, and so on. The 1995 data was an aggregation of a large number of file downloads from a variety of Web sites, none of which are necessarily optimal, but most of which are probably reasonably well designed. Thus it is not surprising that the aggregate behavior might have statistics that appear optimal even if the individual Web sites are not. Furthermore, users would tend to avoid very poorly designed Web sites, adding an element of selection to the data.

Most importantly, even if the agreement with data is entirely coincidental, it is still true that Web sites *should* have these statistics if they are designed to minimize average file download times. This has more serious implications than the agreement with data. In particular, heavy tailed distributions can be viewed as the inevitable outcome of a very natural optimal "source coding" problem that is analogous to standard data compression, but with very different resulting distributions. Given the connection of heavy tails in Web sites and bursty network traffic, this can be thought of as bringing some initial closure to the origins of such traffic, but raises new questions. Bursty traffic is thus not an artifice of user behavior, but has some aspects which are intrinsic to at least the current dominant application, the WWW, and may be even more intrinsic to any application which organizes information for human consumption.

6.2 Future Directions

As discussed at the end of Chapter 5, the Web layout models studied in this report are still at a high level of abstraction from real Web sites today. More realistic models of Web layouts with increasing levels of complexity can be built on top of these relatively simplified models. It is worth the effort to gather empirical data of real Web site topology and user access patterns so that the truthfulness of the simulation models can be tested and enhanced. Moreover, there are many more design objectives and resource constraints other than minimizing the average download time and limiting the total number of files on a Web site. Alternative settings for the optimization problem need to be explored to see whether similar conclusions can be drawn.

Additionally, if Web site layout can be viewed as "source coding" albeit with many strange and unfamiliar properties, then network protocol design and congestion control might profitably be viewed as a form of channel coding, but also presumably with strange and unfamiliar properties. There may be some advantages in exploiting the specific features of the resulting source, as well as advantages in some joint design where the source coding reflects the nature of the network on which it must be transmitted. Recent work along these lines include designing new scheduling policies on Web servers for specifically dealing with heavy-tailed workloads [22, 13]. It may also be possible to utilize the knowledge of the heavy-tailed Web traffic by making more effective use of caching and prefetching at the Web application level. A good example of combining source coding with channel coding is the rate distortion theory in data compression [18]. There is great potential in combining rate distortion techniques and Web layout design for more efficient transmission of information over the Internet. Another promising direction is to integrate generalized source coding with Low et al.'s optimization flow control theory [31, 4] so that joint source and channel coding can be viewed as a global optimization problem. Hopefully this mixed framework can facilitate a new look into different TCP congestion control algorithms under self-similar network traffic in order for possible updates of existing protocols or design of new protocols to be explored.

Appendix A

Proof of the Asymptotic Results in Chapter 4

Consider the ideal case where $L = \infty$. Let $F(x)$ be the CCDF corresponding to the PDF $p(x)$, i.e., $F(x) = \int_x^\infty p(x)dx$. The probability of transferring file $\#i$, whose size is $l_i = c_i - c_{i-1}$, is $p_i^t = F(c_{i-1})$. Based on the recursion relation $p(c_i)l_{i+1} = p_i$, where $p_i = \int_{c_{i-1}}^{c_i} p(x)dx$, we want to show that

$$p_i^t \sim k l_i^{-1} \tag{A.1}$$

as $l_i \to \infty$ for certain $p(x)$, where $k > 0$ is a constant independent of l_i. This is equivalent to

$$\frac{\log p_{i+1}^t - \log p_i^t}{\log l_{i+1} - \log l_i} \sim -1. \tag{A.2}$$

In general, we have

$$\frac{\log p_{i+1}^t - \log p_i^t}{\log l_{i+1} - \log l_i} = \frac{\log F(c_i) - \log F(c_{i-1})}{\log \frac{p_i}{p(c_i)} - \log l_i} = \frac{\log F(c_i) - \log F(c_{i-1})}{\log \frac{F(c_{i-1}) - F(c_i)}{p(c_i) l_i}}.$$

Therefore, for any $p(x)$ that satisfies

$$\frac{\log F(c_i) - \log F(c_{i-1})}{\log \frac{F(c_{i-1}) - F(c_i)}{p(c_i) l_i}} \sim -1, \tag{A.3}$$

as $l_i \to \infty$ ($c_i,\ c_{i-1} \to \infty$), (A.1) holds. Although it is hard to find the exact class of distributions that satisfy (A.3), we can use this condition to test individual distributions. The case where $p(x)$ is an exponential distribution was shown in Chapter 4. Next we prove that (A.3) holds for the normal distribution and the Cauchy distribution as well.

A.1 For the Gaussian Distribution

We assume that $p(x)$ is one side of a normal distribution, and we only show for the standardized case. For more general cases, a similar argument holds. Here $p(x) = \frac{2}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$, $x \geq 0$.

As $x \to \infty$, we have $F(x) \sim \frac{2}{\sqrt{2\pi}x} e^{-\frac{x^2}{2}}$. Then as $l_i \to \infty$,

$$\log F(c_i) - \log F(c_{i-1}) \sim \log \frac{e^{-\frac{c_i^2}{2}}}{\sqrt{2\pi} c_i} - \log \frac{e^{-\frac{c_{i-1}^2}{2}}}{\sqrt{2\pi} c_{i-1}} = -\frac{1}{2}(c_i^2 - c_{i-1}^2) + \log c_{i-1} - \log c_i,$$

and

$$\log\frac{F(c_{i-1})-F(c_i)}{p(c_i)l_i}\sim\log\frac{\frac{1}{\sqrt{2\pi}c_{i-1}}e^{-\frac{c_{i-1}^2}{2}}-\frac{1}{\sqrt{2\pi}c_i}e^{-\frac{c_i^2}{2}}}{\frac{1}{\sqrt{2\pi}}e^{-\frac{c_i^2}{2}}l_i}=\log[\frac{1}{c_i}(\frac{c_i}{c_{i-1}}e^{\frac{c_i^2-c_{i-1}^2}{2}}-1)]-\log l_i$$

$$\sim\log\frac{1}{c_{i-1}}e^{\frac{c_i^2-c_{i-1}^2}{2}}-\log l_i=\frac{1}{2}(c_i^2-c_{i-1}^2)-\log c_{i-1}-\log l_i.$$

Hence,

$$\frac{\log F(c_i)-\log F(c_{i-1})}{\log\frac{F(c_{i-1})-F(c_i)}{p(c_i)l_i}}\sim\frac{-\frac{1}{2}(c_i^2-c_{i-1}^2)+\log c_{i-1}-\log c_i}{\frac{1}{2}(c_i^2-c_{i-1}^2)-\log c_{i-1}-\log l_i}\sim-1.$$

A.2 For the Cauchy Distribution

Again for simplicity assume $\lambda=1$. So $p(x)=\frac{2}{\pi(1+x^2)},\ x\geq 0$.

As $x\to\infty$, we have $p(x)\sim\frac{2}{\pi}x^{-2}$, and $F(x)=1-\frac{2}{\pi}\arctan x\sim\frac{2}{\pi}x^{-1}$. Then as $c_i,\ c_{i-1}\to\infty$,

$$l_{i+1}=\frac{p_i}{p(c_i)}=\frac{F(c_{i-1})-F(c_i)}{p(c_i)}\sim\frac{\frac{2}{\pi}c_{i-1}^{-1}-\frac{2}{\pi}c_i^{-1}}{\frac{2}{\pi}c_i^{-2}}=c_i(\frac{c_i}{c_{i-1}}-1)=\frac{c_i}{c_{i-1}}l_i.$$

Hence,

$$\frac{\log F(c_i)-\log F(c_{i-1})}{\log\frac{F(c_{i-1})-F(c_i)}{p(c_i)l_i}}=\frac{\log F(c_i)-\log F(c_{i-1})}{\log c_i-\log c_{i-1}}\sim-1.$$

A by-product of this proof is the following:

$$\frac{l_{i+1}}{l_i}\sim\frac{c_i}{c_i-1}\Longrightarrow\frac{l_{i+1}}{l_i}\sim\frac{c_i+l_{i+1}}{c_{i-1}+l_i}=\frac{c_{i+1}}{c_i}.$$

Additionally

$$\frac{l_{i+2}}{l_{i+1}}\sim\frac{c_{i+1}}{c_i},$$

therefore,

$$\frac{l_{i+1}}{l_i}\sim\frac{l_{i+2}}{l_{i+1}}\sim\cdots=\text{constant}.$$

The above argument explains why the optimal l_i in the Cauchy case are distributed uniformly on a logarithmic scale, as shown in Figure 4.5.

A.3 Properties of the Operator '$\sim$'

The operator '$\sim$' defines an equivalent class of functions. In particular, if $f(x)\sim g(x)$ as $x\to a$, then $\lim_{x\to a}\frac{f(x)}{g(x)}=1$, where $a=0$, or ∞. [42]

The following are some useful properties of the equivalent class defined by '$\sim$'.

1. $f(x)\sim g(x)\Longleftrightarrow\frac{f(x)}{g(x)}\sim 1$.
2. $f(x)\sim g(x)\Longleftrightarrow g(x)\sim f(x)$.
3. $f(x)\sim g(x)$ and $g(x)\sim h(x)\Longrightarrow f(x)\sim h(x)$.

4. $f(x) \sim g(x) \Longrightarrow h(x)f(x) \sim h(x)g(x)$, for $h(x) \neq 0$.

5. $f_1(x) \sim g_1(x)$ and $f_2(x) \sim g_2(x) \Longrightarrow \frac{f_1(x)}{f_2(x)} \sim \frac{g_1(x)}{g_2(x)}$.

6. If $f_1(x) \sim g_1(x)$ and $f_2(x) \sim g_2(x), g_1(x) + g_2(x) \neq 0$, and either $\lim_{x\to a} \frac{g_1(x)}{g_2(x)}$ or $\lim_{x\to a} \frac{g_2(x)}{g_1(x)}$ exists, then $f_1(x) + f_2(x) \sim g_1(x) + g_2(x)$.

7. If $f(x) \sim g(x)$, and $\exists c > 0$, such that $|\log g(x)| \geq c$, then $\log f(x) \sim \log g(x)$.

Bibliography

[1] R. Albert, H. Jeong, and A.-L. Barabási. Diameter of the World-Wide Web. In *Nature*, volume **401(6749)**, pages 130–131, 1999.

[2] M.F. Arlitt and C.L. Williamson. Web server workload characterization: The search for incariants. In *Proceedings of ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, pages 126–137, 1996.

[3] B. Arnold. *Pareto Distributions*. International Cooperative Publishing House, Maryland, 1983.

[4] S. Athuraliya, D.E. Lapsley, and S.H. Low. An enhanced random early marking algorithm for Internet flow control. In *Proceedings of Infocom '2000, Isreal*, 2000. To appear.

[5] P. Bak. *How Nature Works: The Science of Self-Organized Criticality*. Copernicus, New York, 1996.

[6] A.-L. Barabási, R. Albert, and H. Jeong. Scale-free characteristics of random networks: The topology of the world wide web. In *Preprint submitted to Elsevier Preprint*, 1999.

[7] P. Barford, A. Bestavros, A. Bradley, and M.E. Crovella. Changes in web client access patterns: Characteristics and caching implications,. In *World Wide Web: Special Issue on Characterization and Performance Evaluation*, volume **12**, pages 15–18, 1999.

[8] F. Brichet, J. Roberts, A. Simonian, and D. Veitch. Heavy traffic analysis of a storage model with long range dependent on/off sources. In *Queueing Systems*, volume **23(1-4)**, pages 197–215, 1996.

[9] J.M. Carlson and J.C. Doyle. Highly Optimized Tolerance: A mechanism for power laws in designed systems. In *Physics Review E*, volume **60**, pages 1412–1428, 1999.

[10] J.M. Carlson and J.C. Doyle. Highly Optimized Tolerance: Robustness and design in complex systems. In *Physics Review Letters*, volume **84(11)**, pages 2529–2532, 2000.

[11] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. John Wiley and Sons, New York, 1991.

[12] M.E. Crovella and A. Bestavros. Self-similarity in World Wide Web traffic: Evidence and possible causes. In *IEEE/ACM Transactions on Networking*, volume **5(6)**, pages 835–846, 1997.

[13] M.E. Crovella, R. Frangioso, and M. Harchol-Balter. Connection scheduling in Web servers. In *Proceedings of the 1999 USENIX Symposium on Internet Technologies and Systems, Boulder, Coloraro*, 1999.

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

[15] M.B. Doar. A better model for generating test networks. In *Proceedings of Globecom '96*, 1996.

[16] J.C. Doyle and J.M. Carlson. Highly Optimized Tolerance and generalized source coding. In *Physics Review Letters*, 2000. submitted.

[17] N.G. Duffield and N. O'Connell. Large deviations and overflow probabilities for the general single-server queue, with applications. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume **118**, pages 363–374, 1995.

[18] M. Effros. Distortion-rate bounds for fixed- and variable-rate multiresolution source codes. In *IEEE Transactions on Information Theory*, volume **45(6)**, pages 1887–1910, 1999.

[19] A. Erramilli, O. Narayan, and W. Willinger. Experimental queueing analysis with long-range dependent packet traffic. In *IEEE/ACM Transactions on Networking*, volume **4(2)**, pages 209–223, 1996.

[20] B.S. Everitt and D.J. Hand. *Finite Mixture Distributions*. Chapman and Hall, London, New York, 1981.

[21] M. Garrett and W. Willinger. Analysis, modeling, and generation of self-similar VBR video traffic. In *Proceedings of SIGCOMM '94*, pages 269–280, 1994.

[22] M. Harcol-Balter, M. Crovella, and S.-S. Park. The case for SRPT scheduling in Web servers. In *MIT-LCS-TR-767*, 1998.

[23] B.A. Huberman, P.L.T. Pirolli, J.E. Pitkow, and R.M. Lukose. Strong regularities in World Wide Web surfing. In *Science*, volume **280(5360)**, pages 95–97, 1998.

[24] G. Irlam. ufs'93 [Updated file size survey results]. In *USENET newsgroup comp.os.research, message 2ddp3b$jn5@darkstar.ucsc.edu*, Nov. 29, 1993.

[25] R. Jain and S.A. Routhier. Packet trains: Measurements and a new model for computer network traffic. In *IEEE Journal on Selected Areas in Communications*, volume **4**, pages 986–995, 1986.

[26] Park K., G.T. Kim, and M.E. Crovella. On the relationship between file sizes, transport protocols, and self-similar network traffic. In *Proceedings of 4th International Conference in Network Protocols*, pages 171–180, 1996.

[27] W.E. Leland and T.J. Ott. Unix process behavior and load balancing among loosely-coupled computers. In O.J. Boxman, J.-W. Cohen, and H.C. Tijms, editors, *Teletraffic Analysis and Computer Performance Evaluation*, pages 191–208, Amsterdam, 1986. Elsevier Science Publishers B. V.

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

[29] W.E. Leland and D.V. Wilson. High time-resolution measurement and analysis of LAN traffic: Implications for LAN interconnection. In *Proceedings of IEEE INFOCOM, Bal Harbour, FL*, pages 1360–1366, 1991.

[30] A. Leon-Garcia. *Probability and Random Processes for Electrical Engineering*. Addison-Wesley, 1994.

[31] S.H. Low and D.E. Lapsley. Optimization flow control I: Basic algorithm and convergence. In *IEEE/ACM Transactions on Networking*, volume **7(6)**, pages 861–874, 1999.

[32] B.B. Mandelbrot. Long-run linearity, locally gaussian processed, h-spectra and infinite variances. In *International Economic Review*, volume **10**, pages 82–113, 1969.

[33] B.B. Mandelbrot. *The Fractal Geometry of Nature*. Freeman, New York, 1983.

[34] B.B. Mandelbrot and J.R. Wallis. Noah, Joseph, and operational hydrology. In *Water Resources Research*, volume **4**, pages 909–918, 1968.

[35] K. Meier-Hellstern, P.E. Wirth, Y.-L. Yan, and D.A. Hoeflin. Traffic models for ISDN data users: Office automation application. In A. Jensen and V.B. Iversen, editors, *Teletraffic and Datatraffic in a Period of Change, Proc. of ITC13, Copenhagen*, pages 167–172, Amsterdam, 1991. Elsevier Science Publishers B. V.

[36] A.O. Mendelzon and T. Milo. Formal models of Web queries. In *Proceedings of the Sixteenth ACM Symposium on Principles of Database Systems, Tucson, Arizona*, 1997.

[37] I. Norros. A storage model with self-similar input. In *Queueing Systems*, volume **16**, pages 387–396, 1994.

[38] V. Paxson. Empirically-Derived analytic models of wide-area TCP connections. In *IEEE/ACM Transactions on Networking*, volume **2**, pages 316–336, 1994.

[39] V. Paxson and S. Floyd. Wide-area traffic: the failure of poisson modeling. In *IEEE/ACM Transactions on Networking*, volume **3(3)**, pages 226–244, 1995.

[40] V. Paxson and S. Floyd. Why we don't know how to simulate the Internet. In *Proceedings of the 1997 Winter Simulation Conference*, 1997.

[41] J.E. Pitkow and C.M. Kehoe. GVU's WWW Users Surveys [online]. 1997. Available at www.gvu.gatech.edu/user_surveys.

[42] G. Samorodnitski and M.S. Taqqu. *Stable Non-Gaussian Random Process: Stochastic Models with Infinite Variance*. Chapman and Hall, New York, 1994.

[43] C.E. Shannon. A mathematical theory of communication. In *Bell System Technical Journal*, volume **27(3)**, pages 379–423, 623–656, 1948.

[44] M.S. Taqqu, W. Willinger, and R. Sherman. Proof of a fundamental result in self-similar traffic modeling. In *Computer Communication Review*, volume **27**, pages 5–23, 1997.

[45] H.A. Wan and C.-W. Chang. Web page design and network analysis. In *Internet Research: Electronic Networking Applications and Policy*, volume **8(2)**, pages 115–122, 1998.

[46] B.M. Waxman. Routing of multipoint connections. In *IEEE Journal on Selected Areas in Communications*, pages 1617–1622, 1988.

[47] W. Willinger and V. Paxson. When mathematics meets the Internet. In *Notices of the AMS*, volume **45(8)**, pages 961–970, 1998.

[48] W. Willinger, M.S. Taqqu, R. Sherman, and D.V. Wilson. Self-similarity through high variability: statistical analysis of Ethernet LAN traffic at the source level. In *IEEE/ACM Transactions on Networking*, volume **5(1)**, pages 71–86, 1997.