Evaluating Web Server Performance with an Extension of the N-Burst Model

Ronit Nossenson
Dept. of Computer Science
Technion, Haifa, 32000, Israel
e-mail: ronitt@cs.technion.ac.il

Hagit Attiya
Dept. of Computer Science
Technion, Haifa, 32000, Israel
e-mail: hagit@cs.technion.ac.il

May 19, 2002

Abstract

This paper describes and evaluates a queuing model for Web server called the N-Burst/ME/1 model. This model captures many of the issues that affect Web server as observed by empirical studies. Two important aspects of the systems we consider are that the requests arrival process is an ON/OFF process with truncated heavy-tailed ON times distribution, and that the service-time is drawn from a truncated heavy-tailed distribution. Real data and simulation show that this model is an important step toward a realistic queuing model of a Web server. Our model is an extension of Schwefel and Lipsky [21]. We also present an application of the model in the analysis of a known method for implementing a distributed Web server: Distributed Packet Rewriting [5].

1 Introduction

Two important aspects of a Web server system as a stochastic queuing system are the arrival process and the service process. The arrival process of client requests to the server has to be characterized: are the clients identical? what is the distribution of the interarrival times? do clients submit requests all the time? The service provided by the server has to be characterized as well: does the server supplies one kind of service or several types of services? what is the distribution of the service time?

A human-initiated request arrival process is usually described well by a standard Poisson model. However, document requests in a Web server are not entirely initiated by the user. When a user requests a page, the browser program automatically generates a series of additional requests to download inline images (e.g., logos, icons and buttons). This was indicated by empirical studies performed on the Web.

According to [9, 22], the document request arrival process in Web servers is an ON/OFF process, with several requests during the ON period followed by an OFF period that is significantly longer than the interarrival times during the ON period. The ON and the OFF times were found to be heavy-tailed distributions. Such distributions are characterized by extremely high variability, and may have infinite variance and infinite mean.

According to [15], the document request arrival process in Web servers shows self-similar behavior: it is bursty over a wide range of time-scales. Such process exhibit long-range dependence, that
is, values at any instant are typically non-negligibly positively correlated with values at all future instants. Studies such as [12, 16, 19] show that performance parameters are radically different when simulations use either real data or synthetic data that incorporates long-range dependence.

When considering the service-time properties of Web servers, we restrict our attention to static service in which clients (browsers) send requests for files, and servers reply with the requested files. In such systems, the service time distribution is characterized by the files’ transmission-duration distribution. This distribution was shown to be heavy-tailed [10]. As a result, there is no uniformity in the service-time of clients’ requests (i.e., most of the requests are small and frequent but once in a while a very long request arrives and keeps the server busy for a long time).

The above results imply that a stochastic queuing model of a Web server should include an ON/OFF requests arrival process with heavy-tailed ON and OFF times distributions and a heavy-tailed service-time distribution. A model combining these two aspects is hard to analyze since it is not a Markov process and it does not fit into any known solution of a queue with general arrival and general service distribution.

In this paper, we suggest a new stochastic queuing model of a Web server called the N-Burst/ME/1 model. This model combines an N-Burst arrival process with a matrix exponential truncated heavy-tailed service-time distribution. The N-Burst arrival process [21] is a superposition of N identical independent ON/OFF sources with matrix exponential truncated heavy-tailed ON time distribution and exponential OFF time distribution. Such a process is asymptotically self-similar. Matrix exponential truncated heavy-tailed distribution [13] mimics a heavy-tailed distribution: its reliability function shows heavy-tailed behavior for a limited range, and drops off exponentially thereafter. In this way, it is possible to simplify calculations by relying on Markov processes.

Replacing real heavy-tailed distributions with matrix exponential truncated heavy-tailed distributions obviously decreases the model reliability. However, using such distributions with reliability functions which show heavy-tailed behavior for long enough range, can be a reasonable compensation as shown by real data and simulation.

Boxma and Cohen [6] consider one important aspect of Web server queueing system: real heavy-tailed service-time distribution. However, they assume a Poisson arrival process, which as previously mentioned, is not accurate when considering a Web server.

Schwefel and Lipsky [21] consider the other important aspect of Web server queuing system: an N-Burst arrival process. However, they assume an exponential service-time distribution, while the distribution of the service time in Web server is known to be heavy-tailed. Real data and simulation (see Section 5) show that assuming exponential service-time instead of heavy-tailed service-time leads to a severe underestimation of the required system resources (number of servers, buffer size, etc).

The N-Burst/ME/1 model presented in this paper is the first analytic model that agrees with empirical results regarding both the document request arrival process and the service-time distribution in Web servers.

We compute the queue-length distribution, the waiting-time distribution, and the buffer-overflow probability for the N-Burst/ME/1 model. These performance parameters do not change smoothly with increasing server speed. Instead, jumps called blow-ups points, appear (similar to [21]). Our model, which provides extended knowledge about the service structure, allows a more delicate analysis: for each blow-up point, we have levels of probability for over-saturating the server. These probability levels can help analyzing situations where the server is not over-saturate in its average
behavior, but is over-saturated temporarily. The critical points that define these probability levels can be useful in evaluating system parameters, for example, placement of buffers in the memory hierarchy.

Evaluating Web server performance with our model requires several assumptions. We verify these assumptions and their consistency with reality. We suggest a methodology for choosing parameters in order to make the model more accurate. We use a simulation to compare between the N-Burst/ME/1 model and the previous N-Burst/M/1 model [21]. Simulation and real data analysis show that our model predicts reality in a much more realistic (i.e., pessimistic) way.

The paper ends with a demonstration of how to use the N-Burst/ME/1 model to analyze a well-known implementation of a distributed Web server, called Distributed Packet Rewriting [5]. A similar analysis can be applied to One-IP [11], another implementation of distributed Web servers.

The remainder of this paper is organized as follows. The next section describes definitions that we use. Section 3 describes the N-Burst/ME/1 model, and calculates its queue-length distribution, its waiting-time distribution, and its buffer-overflow probability. In Section 4 we verify the assumptions of the model and suggest how to choose its parameters. Section 5 describes the simulation of our model including a comparison with real data and with the model of [21]. Section 6 presents an application of the model. A discussion of future research work appears in Section 7.

2 Definitions

In this section we describe some central concepts that we use in this paper. It includes definitions of heavy-tailed distributions and matrix exponential distributions.

2.1 Heavy-tailed distributions

Definition 2.1 A random variable $X$ follows a heavy-tailed distribution (with tail index $\alpha$, $0 < \alpha < 2$) if

$$P[X > x] \sim x^{-\alpha}, \text{ as } x \to \infty$$

Where $a \sim b$ means that $\lim_{x \to \infty} a/b = c$ for some constant $c$.

A simple heavy-tailed distribution is the Pareto distribution with shape parameter $\alpha$ and location parameter $k$. It has the following cumulative distribution function (see [14]).

$$F(x) = P[X \leq x] = 1 - (k/x)^{-\alpha}, \text{ } k > 0, \text{ } x \geq k,$$

Heavy-tailed distributions are characterized by extremely high variability, which increases sharply as $\alpha$ decreases. Such a distribution has infinite variance; if $\alpha \leq 1$, then it also has infinite mean.

2.2 Matrix exponential distributions

One of the most studied continuous distributions is the exponential distribution. The exponential distribution is memoryless, that is, the value of an exponential random variable in a given moment of time $t$ does not depend on its past values. A stochastic process is called Markovian if its behavior depends only on its current state. For continuous time Markov chains this property is achieved
only if the distribution of the state-holding times is memoryless, i.e., exponential. When modelling network traffic, an extension of the pure Markovian concept is necessary. By replacing a single state of a Markov chain with a box that contains a network of states, called *phases*, any distribution for the state time can be approximated arbitrarily closely [17, 18].

Figure 1 demonstrates this transformation for state number 4 in the Markov chain. The exponential state times are generalized by replacing the single state (here, number 4) with a bigger box, containing a network of phases.

The following discussion of matrix exponential distributions is based on results from [17, 18].

The distribution of the time between a single customer entering a subsystem (such as the box in Figure 1) and leaving it can be determined by using matrix algebra. We consider the following two basic matrices:

**P** The *transition probabilities* matrix within the subsystem: $P = (P_{ij})$

**M** The *completion rate* matrix is the diagonal matrix of the single state leaving rates $\mu_k$.

We also consider the following two derived matrices:

**V** The *service time* matrix is $V = [M(I - P)]^{-1}$. The element $V_{ij}$ is the overall mean time spent at phase $j$ until the customer leaves the system, given that it started at phase $i$.

**B** The *service rate* matrix is $B = M(I - P)$. It is the inverse of $V$.

The components $p_i$ of the *entrance vector* $p$ are the probabilities that a new customer will directly go to phase $i$ upon entering the system. Since this vector $p$ is constant, a new customer enters the system independently of any previous customer.
The reliability function of the time that a customer spends in the box is:

\[ R(x) = p \cdot \exp(-xB) \cdot \varepsilon' \]  

Thus, the matrix exponential representation, \((p, B)\), defines the distribution of Equation 1. Such distributions are called matrix exponential (in short, ME).

The \( n \)th moments of these distributions follow from Equation 1:

\[ E\{X^n\} = n! \cdot pV^n \cdot \varepsilon' \]

There are a few basic types of distributions that are essential for modelling network traffic [20]. We describe two common examples together with their ME representation: the hyper-exponential distribution and the hypo-exponential distribution.

**Definition 2.2** Let \( X_i, \ldots, X_n \) be \( n \) independent exponential random variables each with parameter \( \lambda_i \), \( i = 1, \ldots, n \) where \( \lambda_i \neq \lambda_j \) for \( i \neq j \). Suppose that there are \( n \) positive constants \( p_i, i = 1, \ldots, n \) such that \( \sum_{i=1}^{n} p_i = 1 \). If the random variable \( X = X_i \) with probability \( p_i \), then \( X \) is a hyper-exponential random variable with \( n \) exponential phases.

Let us analyze the ME-representation of a hyper-exponential distribution with two phases (HYP-2) as shown in Figure 2. The P-matrix is zero, since there are no transitions within the box. Thus, \( B = M \). The probability vector \( p \) is \([p_1, (1 - p_1)]\), and the matrices are:

\[ B = \begin{bmatrix} \mu_1 & 0 \\ 0 & \mu_2 \end{bmatrix}, \quad V = \begin{bmatrix} \frac{1}{\lambda_1} & 0 \\ 0 & \frac{1}{\lambda_2} \end{bmatrix} \]

By these matrices and Equation 1, the HYP-2 distribution has the reliability function:

\[ R(x) = p_1 e^{-\mu_1 x} + (1 - p_1) e^{-\mu_2 x} \]

The mean is:

\[ E\{x\} = \frac{p_1}{\mu_1} + \frac{1 - p_1}{\mu_2} \]

**Definition 2.3** Let \( X_i, \ldots, X_n \) be \( n \) independent exponential random variables each with parameter \( \lambda_i \), \( i = 1, \ldots, n \) where \( \lambda_i \neq \lambda_j \) for \( i \neq j \). The random variable \( X \) has hypo-exponential distribution if \( X = \sum_{i=1}^{n} X_i \).
A special case of the hypo-exponential distribution arises when $\forall i$, $\lambda_i = \lambda$ and is known as the $Erlang(n, \lambda)$ distribution.

Let us analyze the ME-representation of a hypo-exponential distribution with two phases (HYPO-2) as shown in Figure 3. The probability vector $p$ is $[1, 0]$, and the matrices are:

$$
P = \begin{bmatrix}
0 & 0 \\
1 & 0 \\
0 & 1 \\
0 & 0
\end{bmatrix}, 
M = \begin{bmatrix}
\mu_1 & 0 \\
0 & \mu_2
\end{bmatrix}, 
B = \begin{bmatrix}
\mu_1 & -\mu_1 \\
0 & \mu_2
\end{bmatrix}, 
V = \begin{bmatrix}
\frac{1}{\mu_1} & \frac{1}{\mu_2} \\
0 & \frac{1}{\mu_2}
\end{bmatrix}
$$

By these matrices and Equation 1, the HYPO-2 mean is:

$$
E\{x\} = \frac{1}{\mu_1} + \frac{1}{\mu_2}
$$

3 The N-Burst/ME/1 model

In this section we describe the N-Burst/ME/1 model. The results presented here are the queue-length distribution, the buffer-overflow probability, and the waiting-time distribution. We give here a more detailed description of the N-Burst model [21], discuss the blow-ups points together with the probability levels of over-saturating the server, and defined the critical points which separate these levels.

3.1 The N-Burst arrival process

The N-Burst arrival process [21] is a superposition of traffic streams from $N$ independent, identical sources of ON/OFF type (as shown in Figure 4): each source emits requests at a Poisson-rate $\lambda_p$ during its ON-time, and transmits nothing during its OFF-time. Let $k$ be the mean rate of the individual source (the average for the ON- and OFF-times together), then the $N$ sources collectively generate requests at mean rate $\lambda = Nk$.

The distribution of the duration of the ON periods is a special ME distribution $\langle p,B \rangle$ as described in Section 3.2. The choice of the actual ON time distribution has a great impact on the performance. The distribution of the OFF time duration is assumed to be exponential.

3.2 Matrix exponential truncated heavy-tailed distributions

Greiner et al. [13] introduce a family of hyper-exponential distributions called matrix exponential truncated heavy-tailed (in short, ME-THT), see Figure 5. They showed that these distributions with growing number of phases asymptotically approach heavy-tail distributions.
Let us analyze the ME representation of such a distribution with $T$ phases. The P-matrix is zero, since there are no transitions within the box. Thus $B = M$. The probabilities

$$p_i = \frac{[\theta^{i-1}(1 - \theta)]}{(1 - \theta^T)}, \quad i = 1, \ldots, T$$

of entering phase $i$ decay geometrically with the factor $\theta$, $0 < \theta < 1$. The state holding times grow geometrically with the factor

$$\gamma = \frac{1}{\theta^{1/\alpha}}$$

The matrices are:

$$B = \mu \begin{bmatrix} \gamma & \cdots & 0 \\ 0 & \ddots & \vdots \\ 0 & \cdots & \gamma^{T-1} \end{bmatrix}, \quad V = \frac{1}{\mu} \begin{bmatrix} \gamma^0 & \cdots & 0 \\ 0 & \ddots & \vdots \\ \cdots & \cdots & \gamma^{T-1} \end{bmatrix}$$

By these matrices and Equation 1, this distribution has the reliability function:

$$R(x) = \frac{1 - \theta}{1 - \theta^x} \sum_{i=0}^{T-1} \theta^i \exp \left( - \frac{\mu}{\gamma^i} x \right)$$

In order to have heavy tail behavior with tail-exponent $\alpha$ and mean $x$, we need to have [13]:

$$\mu = \frac{1 - \theta}{1 - \theta^T} \frac{1 - (\theta \gamma)^T}{1 - \theta \gamma} \frac{1}{x}$$

The heavy-tail range, $\text{Rng}(R(x))$, is the mean of the largest exponential phase. That largest phase is mainly responsible for the drop-off in the reliability function:

$$\text{Rng}(R(x)) = \frac{\gamma^{T-1}}{\mu}$$

The variable $\theta$ can be chosen freely in the open interval $(0, 1)$. The larger the value of $\theta$ is, the more phases are necessary to obtain the same heavy-tail range.
3.3 Blow-up regions and critical points

As previously mentioned, in this model we assume that the service-time has ME-THT distribution. Assume the service-time distribution $(p_i, B_i)$ has $T$ phases and average service rate $\tau$. To simplify notation, we refer to the service-phase rates as $\tau_j = \frac{\mu_j}{\sum_j \mu_j}, j = 1, \ldots, T$ (see Figure 5).

Since truncated heavy-tailed distributions (in the N-Burst arrival process) allow very long bursts with non-negligible probability, performance parameters of the Schwefel and Lipsky’s model do not change smoothly with increasing server speed. Instead, blow-up points appears. This behavior influences our model too.

We say that $i, i = 1, \ldots, N$, sources are long-term active sources if these sources are in ON state for a long period, and the other $N - i$ sources are in average behavior during this period. In this case, the mean arrival rate is temporarily set to

$$\Lambda_i = i\lambda_p + (N - i)k$$

The $i$ long-term active sources (each with rate $\lambda_p$) leave $(N - i)$ sources with average behavior (each with rate $k$). The $N$ blow-up points are located at $\tau = \Lambda_i$ [21].

When considering a THT-ME service-time distribution instead of exponential service-time distribution, more insight can be achieved. For each blow-up point, we use the service-phase rates $\tau_j, j = 1, \ldots, T$, and the probabilities of the entrance vector $p_i$ to define the levels of probability for over-saturating the server. The points that separate these levels are called critical points.

Let us demonstrate how to calculate the first level of probability for over-saturating the server. This level means that there is a positive probability for over-saturating the server. Consider the last service phase-rate $\tau_T$ and the critical points $\tau_T = \Lambda_i, i = 1, \ldots, N$. These points represent equality between the phase with the slowest rate and the temporary arrival rate. Passing these points means that there is a positive probability that $i$ long-term active sources lead to over-saturation of the server. Furthermore, assuming that $\tau_T < \Lambda_i < \tau_{T-1}$, then $i$ long-term active sources lead to over-saturation of the server with probability $p_T$. If $\tau_j < \Lambda_i < \tau_{j-1}, 1 < j < T$, then this probability grows to $\sum_{k=j}^T p_k$. Note that if $\tau_T > \Lambda_i$ then with probability 1, $i$ long-term active sources will not
lead to over-saturation of the server.

Another $N$ interesting critical points are $\tau_i = \Lambda_i$ which define the last level of probability for over-saturation of the server. These points represent equality between the phase with the fastest rate and the temporary arrival rate. Passing these points (that is, $\tau_i \leq \Lambda_i$) means that $i$ long-term active sources lead to over-saturation of the server with probability $1$.

Consider the average server rate $\tau = p_s \cdot B_s \cdot \tau'$. Similar to [21], the region of the parameter-space for which

$$\Lambda_{i_0-1} < \tau < \Lambda_{i_0}, \ i_0 \in \{1, \ldots, N\}$$

is called the blow-up region $i_0$. Within such a blow-up region, $i_0$ long-term active sources (each with rate $\lambda_s$) together with the remaining $(N - i_0)$ sources with average behavior are sufficient to produce a positive queue-length growing average rate: $\Lambda_{i_0} - \tau$. The parameter $i_0$ has a great influence on the delay probability in the N-Burst/ME/1 queue.

An important region of the parameter-space of the N-Burst/ME/1 model is when

$$\Lambda_{i_1-1} < \tau_T < \Lambda_{i_1}, \ i_1 \in \{1, \ldots, N\}$$

Within this region, $i_1$ is the minimum number of long-term active sources that together with the remaining $(N - i_1)$ sources with average behavior are sufficient to produce an over-saturation period with positive probability. Note that if $i_1 < i_0$ then $i_1$ long-term active sources do not produce a positive queue-length growing average rate. However, if the queue length is bounded then the parameter $i_1$ influences the buffer-overflow distribution as described below.

Beyond the last critical point, $\tau_T > \Lambda_N = N \lambda_s$, over-saturation of the server can never happen. The delay is rather small and the actual burst-length distribution does not have visible impact, i.e., self-similarity by heavy-tailed ON-times does not lead to very different Quality of Service (QoS) than standard exponential burst-lengths.

### 3.4 The queue length distribution

As previously mentioned, heavy-tail distributed burst-lengths can cause very long over-saturation periods in the N-Burst/ME/1 queue. Those over-saturation periods are caused by $i_1$ (or more) long-term active sources and they have a major impact on the performance. Assume the ON time distribution is $\langle p_s, B_s \rangle$ with $R(x) \sim x^{-\alpha_s}$. The reliability function, $R_i(x)$, that none of $i$ simultaneously long-term active bursts finishes before time $x$ has a heavy-tail with exponent $i \alpha_s - (i-1)$ [21]. Within such an over-saturation period, the queue-length will grow with average rate $\Lambda_i - \tau$, which is positive for $i \geq i_0$. Assuming an empty queue at the beginning of the over-saturation period, the probability that queue-length $q_0$ is reached within that period is approximately

$$p_i(q_0) \Delta R_i\left(\frac{q_0}{\Lambda_i - \tau}\right) \sim \left(\frac{\Lambda_i - \tau}{q_0}\right)^{i \alpha_s - i + 1}$$

The assumption of an empty queue at the start of the over-saturation period is reasonable when the utilization, $\rho$, is not too close to 1 (the system is empty for a non-negligible fraction of the time).

The above discussion about $p_i(q_0)$ assume average service behavior, that is, service rate $\tau$. Additional information about the service structure leads to more insight about the system. Assume there are $i$ simultaneously long-term active bursts, and the service rate is temporally $\tau'$. Within such an over-saturation period, the queue-length will grow with rate $\Lambda_i - \tau'$, which can be positive
for $i \geq i_1$ and small $\tau'$. Note that the interesting situations are when $i_1 \leq i < i_0$ and $\tau'$ is small enough so $\Lambda_i - \tau'$ is positive. Assuming an empty queue at the beginning of the over-saturation period, the probability that queue-length $q_0$ is reached within that period is approximately

$$p'_i(q_0) \triangleq R_i(\frac{q_0}{\Lambda_i - \tau'}) \sim \frac{(\Lambda_i - \tau')^{i_{a_n} - i + 1}}{q_0}$$

Multiplying $p'_i(q_0)$ with the probability that the service rate is $\tau'$, approximates the probability that within that period the server rate is $\tau'$ and queue-length $q_0$ is reached. If the buffer-size is bounded then this observation influences the buffer-overflow distribution as described below.

By [21],

$$\frac{p_{i+n}(q_0)}{p_i(q_0)} \sim \frac{1}{q_0^{\alpha(a_n-1)}}$$

which implies that large queue-length $q_0 \gg 0$ are most likely caused by an over-saturation periods with $i_0$ long-term active bursts. In other words, the duration of the over-saturation periods with $i_0$ long-term active sources dominates for large $q_0$ since it has the 'heaviest' tail with exponent

$$\beta \triangleq i_0(\alpha - 1) + 1, \quad \alpha > 1$$

Therefore, it is not the heavy-tail exponent $\alpha$ of the individual ON-period, but rather the tail exponent $\beta$ of the duration of the over-saturation period, that determines the queuing behavior [21].

The queue tail-behavior in the N-Burst/ME/1 model is:

$$Pr(q^{(c)} = k) \sim \frac{1}{k^\beta} \text{ and } R_i(k) \triangleq Pr(q^{(c)} > k) \sim \frac{1}{k^{\beta-1}}$$

Where $q^{(c)}$ denotes the queue-length at request arrival times.

Note that a steady-state distribution for $q^{(c)}$ exists whenever $\rho < 1$. However, since the queue-length distribution has a heavy-tail it can have infinite moments. In particular, its mean is infinite when $\beta - 1 \leq 1$ which is equivalent to

$$1 < \alpha \leq 1 + \frac{1}{i_0}$$

### 3.5 The buffer-overflow probability

The buffer-overflow probability (BOP) follows directly from the queue-length distribution,

$$BOP(B) \triangleq Pr(q^{(c)} > B) \sim \frac{1}{B^{\beta-1}}$$

That is, a heavy-tail distribution. As in [21], the practical implication is that additional buffer-space reduces the BOP very slowly, in particular, slower than exponential decrease.

The result about the BOP assume average service behavior, that is service rate $\mu$. This assumption is hidden in the tail exponent parameter $\beta$. Again, more accurate modelling of the service-time distribution leads to more insight about the system. As mentioned before, $i \geq i_1$ long-term active sources together with the remaining $(N - i)$ sources with average behavior are sufficient to produce an over-saturation period with positive probability. Assume that there are $i$ simultaneously long-term active bursts, and that the service rate is temporally $\tau'$. As previously mentioned, within such
an over-saturation period, the queue-length will grow with rate $\Lambda_i - \tau'$, which can be positive for $i \geq i_1$ and small $\tau'$. Note that (again) the interesting situations are when $i_1 \leq i < i_0$ and $\tau'$ is small enough so $\Lambda_i - \tau'$ is positive. As a result, there is a positive probability for buffer-overflow even with less than $i_0$ long-term active sources. One can calculate the probability exactly by multiplying the BOP with tail exponent parameter

$$\beta' \overset{\Delta}{=} i(a - 1) + 1, \quad a > 1, \quad i_1 \leq i < i_0$$

with the probability that the service rate is $\tau'$.

### 3.6 The waiting time distribution

The waiting-time is related to the queue-length, $q^{(c)}$, at request-arrival: a request that arrives at the queue with $q_1^{(c)}$ requests in front of it, in the N-Burst/ME/1 model, experiences an hypo-exponential

$$\sum_{j=1}^{T} \text{Erlang}((q^{(c)}_j + 1) \cdot p_j, \tau_j)$$

distributed delay with mean

$$\sum_{j=1}^{T} \frac{((q^{(c)}_j + 1) \cdot p_j)}{\tau_j}$$

which approaches a deterministic distribution for large $q_1^{(c)}$. Thus, the waiting-time (WT) has the same tail-behavior,

$$Pr(WT > t) \approx Pr(q^{(c)} > t\tau) \sim \frac{1}{t^{\beta-1}}$$

Assuming finite buffer size, $B$, the maximal delay is distributed as followed:

$$\sum_{j=1}^{T} \text{Erlang}((B + 1) \cdot p_j, \tau_j)$$

### 4 Assumptions verification and selection of parameters

This section deals with two important issues: verification of the model assumptions and a discussion of how to choose the model parameters in the most realistic way.

#### 4.1 Assumptions verification

Evaluating Web server performance with the N-Burst/ME/1 Model requires several assumptions. Let us analyze these assumptions and their consistency with reality.

The first assumption set of the model refers to the arrival process. We assume that client requests for Web documents arrive according to the N-Burst arrival process (Section 3.1). This statement hides the following four assumptions which are discussed below.

1. The arrival process is an ON/OFF process.
2. The distribution of the duration of the ON periods is ME-THT.
(3) The OFF time distribution is exponential.

(4) The arrival process is a superposition of $N$ identically distributed sources.

The first assumption agrees with the empirical models of Crovella and Bestavros [9] and Shuang Deng [22]. By these studies, which were based on different data sets, an ON/OFF process can be used to model the WWW request arrivals, with several requests during the ON period followed by an OFF period that is significantly longer than the interarrival time during the ON period. Figure 6 presents real data of a Web server request arrival process. It is easy to see that the process is indeed an ON/OFF process. Details about the real data collection can be found below.

The second assumption is justified since the ON time was found to have a heavy-tailed Weibull distribution [22]. Although replacing this real heavy-tailed distribution with ME-THT distribution decreases the model reliability, using ME-THT distribution with reliability functions which show heavy-tailed behavior for long enough range, can be a reasonable compensation. The heavy-tail range of such distribution is defined as the mean of the largest exponential phase, which is mainly responsible for the drop-off of the reliability function. As previously mentioned, the variable $\theta$ can be chosen freely in the range $0 < \theta < 1$, and for larger value of $\theta$, more phases are necessary to obtain the same heavy-tail range as for lower $\theta$.

The third assumption is justified although the OFF time was found to have a heavy-tailed Pareto distribution [22]. The OFF times are heavy-tailed but with lighter tail than the distribution of the ON times [9]. Therefore, the ON times are more likely responsible for the observed level of traffic self-similarity, rather than the OFF times [9]. Thus, the OFF time distribution is less important.

The fourth assumption suggests that the arrival process is a superposition of $N$ identically distributed sources. However, there is a heavy-tailed distribution of clients among domains [2]. That is, a function with a small number of large values (corresponding to popular Internet service
providers, large research institutes and corporations), and a long tail assuming small values. A good distribution that fits the non-uniformity of this function is the Zipf-like distribution [23]. This distribution is a parametric distribution where the probability of selecting the i-th item is proportional to $1/i^{1-x}$. The Zipf's function is commonly adopted in various social contexts to model the distribution of people choices. As a result, a more sophisticated arrival process is required, in which the N arrival processes are not identically distributed. This model is much more complicated and we consider it as a future research.

The other set of assumptions refers to the duration of files transfer (that is, the service-time) in Web server system. Again, we replace real heavy-tailed distribution [10] with ME-THT distribution to simplify our calculations, and again, using reliability functions which show heavy-tailed behavior for long enough range, can be a reasonable compensation.

4.2 Parameters selection

Another important question which we need to answer here is how to choose the model parameters in a realistic way. All these parameters can be evaluated using the server access log file. This evaluation should be done for each client separately and the final parameters should be taken as the average.

Let us discuss which parameters are interesting. For the ON time distribution, it is important to evaluate the tail-exponent $\alpha$ and the mean $x$. In addition, using ME-THT distribution with heavy-tail range which is twicer longer than the longest ON duration that appears in the server log-file should be reasonable. Since we assume exponentially OFF time distribution, for the OFF time distribution, we need to evaluate only the mean.

The tail-exponent and the mean of the service-time distribution should be evaluated. We should choose ME-THT distribution with heavy-tail range which is twicer longer than the longest transmission duration that appears in the server log-file.

5 Simulation results

This section describes the simulations performed to evaluate the N-Burst/ME/1 model. These simulations consist of two parts: (i) Comparison between the N-Burst/ME/1 model and the N-Burst/M/1 model; and (ii) Comparison with real data.

We performed the simulations in MATLAB. The source files of the simulations together with implementations of N-Burst arrival process, hyper-exponential distribution and ME-THT distribution are available at [1].

The comparison between the N-Burst/ME/1 model and the N-Burst/M/1 model was performed by testing two service processes in a simulation with an N-Burst arrival process: ME-THT service and M (exponential) service, both with the same mean.

Figure 7 presents the arrival process which is shared by the two models. For these arrivals, the departure process in the N-Burst/ME/1 model and the departure process in the N-Burst/M/1 model are compared. It can be seen from this figure that the departure process in the N-Burst/ME/1 model is consistently more pessimistic than the departure process in the N-Burst/M/1 model.

Figure 8 presents (for the same simulation) the number of clients in the system process. The number of clients in the system consists of the client in service (if there is one) and the clients in the queue (if there are any). Again, it can be seen from this figure that the N-Burst/ME/1 model
is consistently more pessimistic. According to this simulation the N-Burst/ME/1 model predicts that the maximum number of clients in the system will be twice the maximum number of clients in the system predicted by the N-Burst/M/1 model.

Note that in both models the service mean is equal. The huge difference in the prediction is a result of the difference in service distributions. The ME-THT distribution approximates a heavy-tailed distribution. Thus, once in a while a very long service time is needed. When this happens, the system queue begins to fill and the number of clients in the system increases dramatically. This phenomena does not exist with an exponential distribution.

The second part of the simulations compares between these models and real Web data (arrival-time and service-time). This comparison is not simple. Two instances of the same arrival process might be very different and it is difficult to compare the results (the service-times are in response to different situations). Furthermore, when we consider two instances of different arrival processes (real arrival process and N-Burst arrival process) the comparison is impossible. Thus, we assume that the real arrival process is similar to the N-Burst arrival process and use the real arrival data only. For the real arrival process we compare between the real service-time process, the ME-THT service-time process (called the ME model) and the M (exponential) service-time process (called the M model).

We use a trace collected in Boston University [3, 7, 8], which is the latest public trace (known to us) that combines two essential properties for our simulation. First, the ability to separate data regarding different servers is important to our simulation. In this trace, the anonymity was achieved by hashing each column of the data set and not each line. This preserves the ability to compare members of a column for equality without leaking information regarding those members identities. The second essential property is that the trace allows to calculate the duration of each request. Usually, traces supply for each request its start time but not its termination time. The duration of request is in fact its service-time, thus, it is important to our simulation.

Another issue is the real data arrival process. Since the trace was made in the client side, the resulted request arrival process is in fact an arrival process of requests from this specific client to
several servers. We do not have any information about these servers, thus, we cannot consider all these requests, and we should choose the request arrival process to one of these servers. Naturally, we consider the requests made to the most popular server. Here we have another problem: Since this arrival process is the result of requests from one client only, the resulting traffic is very sparse. Since in this situation the system is almost always empty, comparison between the models is meaningless. Thus, we randomly choose ten short segments which are located far away from each other (the data was collected during a long period of time). Then we refer to the superposition of these segments. That is, we mimic ten different clients with the same distribution on the arrival processes.

The results of this simulation are presented in Figure 9. It presents the number of clients in the system process. It can be seen from this figure that the ME model is consistently more pessimistic and realistic than the M model. According to this simulation the ME model predicts that the maximum number of clients in the system is 4 times the maximum number of clients in the system predicted by the M model. Furthermore, it can be seen that the real clients in the system process is much closer to the prediction of the ME model.

The client in the system process is very important in the evaluation of the system Quality of Service (QoS) parameters. For example, if we are interested in a system which each server serves at most ten clients simultaneously, according to the real data and the ME model we need four servers. However the M model predicts that we need only one server. Another important parameter is the buffer size. Again, according to the real data and to the ME model the buffer size should be four times bigger than the size predicted by the M model.

6 An application of the N-Burst/ME/1 model

In this section we demonstrate how to use our model to analyze a distributed Web server, that is, an architecture consisting of multiple Web servers with some mechanism to spread the clients' requests among the servers. We refer here to a known technique for distributed Web server: Dis-
A similar analysis can be used to analyze One-IP [11], another known technique for distributed Web servers.

A Web client accesses documents via a browser. The browser sends a request to a Web server, which responds with the requested documents. A Web server can respond to multiple Web clients. Web clients and servers communicate using the HyperText Transfer Protocol (HTTP). For each HTTP request, a TCP connection between a client and a server is established: the client sends a request to a server and the server responds with the requested information. The TCP connection is then closed. Typically, the server location for a URL is identified by the hostname contained in the URL [4]. To establish a TCP connection, the browser asks the Domain Name Service (DNS) to map the server hostname to an IP address.

Distributed Packet Rewriting (DPR) [5] uses a forwarding approach: each host in the system provides both Web service and packet routing functions. That is, each server can redirect a connection to another server in a client transparent manner. We focus on a technique suggested in [5] to select the Web server: the destination server is determined through a hash function applied to the client IP address. The hash function can be determined by analyzing the distribution of client IP addresses in the access log so that client requests are approximately evenly distributed to all servers.

In DPR, all hosts share each other IP addresses as their non-primary addresses. That is, each server is configured to respond to all of the possible original destination addresses in addition to its own primary address. These shared addresses are publicized in the DNS. All client requests are sent to these addresses, in a Round-Robin manner. The hash function ensures that every client request will be processed by exactly one server and that future incoming packets for the same request will be directed to the same server.
Assuming the hash function evenly distributes the client requests among the servers, let us analyze the fluid view of DPR. A client request is served locally at the server it reaches with probability \( \frac{1}{n} \) and served by another server with probability \( \left( \frac{n-1}{n} \right) \). In the later case, we add the forwarding (also called rewriting) time to the total service-time of this request. Assume the rate of the rewrite operation is \( \mu_1 \) and the rate of the "service" operation is \( \mu \). The average request total service-time is \( \frac{n-1}{n} \cdot \frac{1}{\mu_1} + \frac{1}{\mu} \).

The N-Burst/ME/1 model of DPR appears in Figure 10. Let us analyze the ME-representation of this distribution. The service replacement box contains one state that represents the rewriting operation work and two state boxes that represent the "service" operation each with ME-representation \( \langle \bar{p}, \bar{B} \rangle \). The matrices are:

\[
B = \begin{bmatrix}
\mu_1 & -\frac{\mu_1 \cdot \bar{p}}{\mu_1} & 0 \\
0 & \frac{\bar{p}}{\mu_1} & \bar{B} \\
0 & 0 & \bar{B}
\end{bmatrix},
\quad V = \begin{bmatrix}
\frac{1}{\mu_1} & \bar{p} \cdot \bar{B}^{-1} & 0 \\
0 & \bar{B}^{-1} & 0 \\
0 & 0 & \bar{B}^{-1}
\end{bmatrix}
\]

By these matrices and Equation 1, the mean is:

\[
E\{x\} = \frac{n-1}{n} \cdot \frac{1}{\mu_1} + \frac{\bar{p}}{\mu} \cdot \left[ \begin{array}{ccc}
\gamma_0 & \cdots & 0 \\
0 & \gamma_{T-1}
\end{array} \right] \cdot \varepsilon^t
\]

Note that when using the average service rate, \( \mu_1 \), of \( \langle \bar{p}, \bar{B} \rangle \) (that is, \( \bar{p} \cdot \bar{B} \cdot \varepsilon^t \)), as the rate of the "service" operation \( \mu \), this result consists with the fluid view. However, the N-Burst/ME/1 model reveals much more information: the service work is divided into components and the contribution of each component can be calculated.
Furthermore, using the analysis of the N-Burst/ME/1 model, one can calculate the delay distribution and its mean: a request that arrives at the queue with \( q_1^{(c)} \) (the "a" stands for 'at arrival times') requests in front of it, experiences an hypo-exponential distributed delay (see Section 2.3):

\[
\text{Erlang}\left((q_1^{(c)} + 1) \cdot \frac{n-1}{n}, \mu_1\right) + \sum_{j=1}^{T} \text{Erlang}\left((q_1^{(c)} + 1) \cdot \tilde{p}_j, \frac{\mu}{\gamma_j-1}\right)
\]

with mean

\[
\frac{(q_1^{(c)} + 1) \cdot \frac{n-1}{n}}{\mu_1} + \sum_{j=1}^{T} \frac{(q_1^{(c)} + 1) \cdot \tilde{p}_j}{\frac{\mu}{\gamma^j-1}}
\]

which approaches a deterministic distribution for large \( q_1^{(c)} \).

The analysis of the N-Burst/ME/1 model provides knowledge about the blow-up points and the critical points that can be useful for system design purposes. For example, one can design the system in such a way that the service rate will be beyond the last critical point, that is, \( \frac{\mu}{\gamma^T} > \Lambda_N = N\lambda_p \), so, over-saturation of the server can never happen.

Using the knowledge about blow-up points and critical points can achieve some benefits in the decision about buffer location in the memory hierarchy. For example, if the blow-up region \( i_0 \) was calculated together with its \( T \) critical points, one can locate parts of the buffer according to the probabilities of over-saturation of the server.

7 Future research

The first natural future research step is to extend the N-Burst/ME/1 model into the N-NEQ-Burst/ME/1 model in which the N arrival processes are not equal. Such a model takes into account all aspects of a queuing system at a Web server that are known today, in particular, the non-uniform distribution of clients among domains [2].

Another future research step is to replace the ME-THT distributions with real heavy-tailed distributions. This step is difficult since this process can not be represented as a continuous time Markov chain, where the distribution of state-holding times is memoryless.

Acknowledgment

The authors wish to thanks Hadas Shachnai for useful discussions.

References


