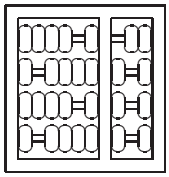

INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

Fractal Traffic Modeling and Policing using Envelope Processes

Flávio M. Pereira, Nelson L. S. Fonseca and Dalton S. Arantes

Technical Report - IC-06-03 - Relatório Técnico

February - 2006 - Fevereiro

Fractal Traffic Modeling and Policing using Envelope Processes

Flávio de M. Pereira[†], Nelson L. S. da Fonseca[*] and Dalton S. Arantes[†]

[†]Department of Communications
School of Electrical and Computer Engineering
State University of Campinas
P. O. Box 6101
13083-970 Campinas/SP — Brazil

[*]Institute of Computing
State University of Campinas
P. O. Box 6176
13084-971 Campinas/SP — Brazil

February 21, 2006

Abstract

In this paper, an envelope process called *Fractional Bounded Arrival Process* (FBAP) is proposed for self-similar traffic representation. A queueing analysis for FBAP traffic is developed, and upper bounds for the backlog and for the delay are obtained. The policing of FBAP traffic is also investigated.

Results are then extended to the multifractal traffic case, for which an envelope process called *Multifractal Bounded Arrival Process* (MFBAP) is proposed. Comments on the queueing analysis and on the policing for MFBAP traffic are also outlined.

1 Introduction

Recent studies indicate that many types of network traffic, ranging from local to wide area network traffic, exhibit *fractal* scaling properties [1–13]. In other words, statistical properties of the traffic do not vary within a range of time scales, and a dominant time scale cannot be identified [2, 3, 6].

It is well known that fractality is closely related to high levels of burstiness and long range dependence, which have pervasive effects on network performance, and cannot be represented by

traditional traffic models [14–24]. Thus, investigations on the modeling and on the control of fractal traffic are of great importance.

Many authors claim that the fractal properties of network traffic are consistent with *statistical self-similarity* [2–6, 9, 10, 25]. Although multiple definitions of self-similarity can be found in the literature, most papers assume that a process $\mathbf{Z}(t)$ with stationary increments is self-similar[1] if the scaling law

$$\mathbf{Z}(t) \stackrel{d}{=} m^{-H}\mathbf{Z}(mt),\ m, t \geq 0, \tag{1}$$

holds for any $0 < H < 1$. The symbol $\stackrel{d}{=}$ denotes equality in distribution. Notice that self-similarity can be regarded as a particular case of fractality, for which the scaling law is constant for all time scales. Thus, a self-similar process can also be called a *monofractal* process.

Studies indicate that self-similar processes are able to capture the long range dependencies of the traffic [2–6, 9, 10, 25]. In particular, the latter is revealed when $H > 0.5$, for which case the autocorrelation function exhibits a power law decay, i.e.,

$$r(\lambda) \sim c\,|\lambda|^{2H-2}\,,\ \lambda \to \infty,\ 0.5 < H < 1.$$

Self-similarity in network traffic has been a topic of intense research, and a number of traffic models has been proposed for self-similar traffic [13]. In particular, Norros [27] introduced a traffic model called *fractional Brownian traffic*, which is defined as

$$\mathbf{A}(t) = \mu t + \sigma\mathbf{Z}(t)\,,\ t \geq 0, \tag{2}$$

where $\mathbf{A}(t)$ represents the traffic up to time t, $\mu > 0$ is the traffic mean rate, $\sigma \geq 0$ is a standard deviation coefficient and $\mathbf{Z}(t)$ is a *fractional Brownian motion* with zero mean, self-similar parameter $0.5 \leq H < 1$ and incremental variance $\operatorname{var}\mathbf{Z}(t) = |t|^{2H}$. A queueing analysis for the fractional Brownian traffic has been provided in [14]. Moreover, recent studies indicate that the process $\mathbf{A}(t)$ can also be used to represent traffic with α-stable marginal distributions, by assuming $\mathbf{Z}(t)$ to be a *linear fractional stable motion* [25, 28, 29].

Due to its stochastic nature, the fractional Brownian traffic has a hard mathematical tractability. An alternative is the use of deterministic functions, called *envelope processes*, which bound the traffic from above. An envelope process is called *deterministic* if the probability that the actual traffic exceeds it is equal to zero. Otherwise, the bounding function and the corresponding probability of violation constitute a *statistical* envelope. An envelope process can also be either *cumulative* or *incremental*, depending on whether it constrains the total amount of traffic up to a given instant of time, or the amount of traffic over a given interval of time.

In order to represent self-similar traffic, a cumulative statistical envelope process was proposed in [20]. Such an envelope, called *fractional Brownian motion (fBm) envelope process*, is defined as

[1] Some authors refer to a process satisfying (1) as a self-affine process, and use the term self-similarity in a stricter sense [26].

$$\widehat{A}(t) = \mu t + k\sigma t^H,\ t \geq 0. \tag{3}$$

The parameters μ and σ are chosen to match the parameters of the original fractional Brownian traffic. The additional parameter k determines the probability that $\mathbf{A}(t)$ exceeds $\widehat{A}(t)$ at time t, i.e.,

$$\mathbb{P}\left\{\mathbf{A}(t) > \widehat{A}(t)\right\} = \mathbb{P}\left\{\mathbf{Z}(1) > k\right\} = \overline{\Phi}(k), \tag{4}$$

where $\overline{\Phi}(\cdot)$ denotes the gaussian residual distribution function. Clearly, the value of k must be chosen so that the probability of violation is negligible.

Notice that the fBm envelope process assumes a gaussian marginal distribution for the traffic, an hypothesis which is not always true [25,28,30]. Besides, it fails to provide an adequate representation for the incremental traffic, particularly when traffic increments are assumed to be stationary. In this case, the probability of violation is given by

$$\begin{aligned} \mathbb{P}\left\{\mathbf{A}(t+\tau) - \mathbf{A}(t) > \widehat{A}(t+\tau) - \widehat{A}(t)\right\} &= \mathbb{P}\left\{\mu\tau + \gamma\mathbf{Z}(\tau) > \widehat{A}(t+\tau) - \widehat{A}(t)\right\} \\ &= \mathbb{P}\left\{\mathbf{Z}(1) > k\frac{(t+\tau)^H - t^H}{\tau^H}\right\}, \end{aligned} \tag{5}$$

which is not consistent with the underlying assumption of stationarity.

In this paper, an incremental envelope called *Fractional Bounded Arrival Process* (FBAP) is proposed for self-similar traffic representation. The FBAP envelope is able to represent both the incremental and the cumulated self-similar traffic, and generalizes the fBm envelope process by assuming no specific marginal distribution for the traffic. A queueing analysis for FBAP traffic is also developed, and upper bounds for the backlog and for the delay are obtained for a queueing system with constant service rate.

Although self-similar processes are often able to accurately represent network traffic, recent studies revealed that more sofisticated models are sometimes required [6, 11, 12, 24]. Actually, traffic may exhibit more complex scaling laws, which deviate from self-similarity for time scales of the order of hundreds of milliseconds and below [24]. Sometimes, such a behavior is also verified at larger time scales [24]. When the scaling law of the traffic significatively deviates from self-similarity, *multifractality* shall be taken into consideration.

Multiple definitions of multifractality are available in the literature [6,12,24,26]. In [26], a process $\mathbf{Z}(t)$ with stationary increments is called a multifractal process if the following scaling law holds:

$$\mathbf{Z}(mt) \stackrel{d}{=} \mathbf{C}(m)\mathbf{Z}(t),\ m, t \geq 0, \tag{6}$$

where $\mathbf{Z}(t)$ and $\mathbf{C}(m)$ are independent random functions. A *generalized scaling index* $\mathbf{H}(m)$ can be defined as

$$\mathbf{H}(m) = \log_m \mathbf{C}(m).$$

From (6),

$$\mathbf{Z}(mt) \stackrel{d}{=} m^{\mathbf{H}(m)}\mathbf{Z}(t),\ m,t \geq 0.$$

The use of multifractal processes for network traffic modeling is still a subject of intense research. In [11], Riedi and Véhel analyzed various TCP traffic traces, and showed that they actually exhibit multifractal scaling laws. In [12], Feldmann et al. proposed the use of multiplicative processes (also known as cascades) for network traffic representation. In [24], Molnár et al. analyzed the tail of an infinite capacity queueing system with a constant service rate, and fed by a generic multifractal traffic.

In the present paper, an envelope process called *Multifractal Bounded Arrival Process* (MFBAP) is proposed for multifractal traffic modeling. Comments on a queueing analysis for MFBAP traffic are also outlined.

The policing of fractal traffic is also investigated in this paper. Although many studies focus on fractal traffic modeling, few papers are actually dedicated to the policing of such a sort of traffic. In [31], a new policing mechanism called *Fractal Leaky Bucket* (FLB) algorithm has been proposed for policing self-similar traffic. The FLB algorithm is a window-based policing mechanism, which constrains traffic to a deterministic cumulative envelope given by

$$\widehat{A}(t) = \mu t + \psi t^H,\ t \geq 0. \tag{7}$$

In order to obtain a negligible probability of discarding well-behaved traffic, the FLB parameters are chosen to match the corresponding fBm envelope process parameters, where ψ is given by the product $k\sigma$.

Since the FLB algorithm constrains traffic to a cumulative envelope process, it is unable to bound the burstiness of the traffic. Consequently, such a policing mechanism does not adequately support the provision of performance bounds. Moreover, it is based on a self-similar traffic model, which may not apply for a multifractal traffic case.

An alternative is the use of the traditional Leaky Bucket algorithm for policing fractal traffic. In such an approach, a single mechanism can be used for policing both fractal and non-fractal traffic. Moreover, results available in the literature which assume the traffic to be Leaky Bucket constrained can be used to establish performance bounds for traffic flows.

In this paper, the use of the traditional Leaky Bucket algorithm for policing FBAP traffic is analyzed. Mathematical relations between the FBAP parameters and the Leaky Bucket parameters are presented. Moreover, such an approach is compared to the use of the FLB algorithm, in terms of capacity of supporting the provision of backlog bounds in a queueing system with constant service rate.

The rest of the paper is organized as follows. In Section 2, the FBAP model is introduced. In Section 3, the queueing analysis for FBAP traffic is developed, and upper bounds for the backlog and for the delay are obtained for a queueing system with constant service rate. In Section 4, traffic policing by using the Fractal Leaky Bucket algorithm is analyzed. In Section 5, the policing of FBAP traffic using the Leaky Bucket algorithm is considered. In Section 6, a comparison between

the Fractal Leaky Bucket algorithm and the Leaky Bucket algorithm is provided. In Section 7, the MFBAP model is introduced. In Section 8, some comments on a queueing analysis and on the policing of MFBAP traffic are outlined. Finally, in Section 9, conclusions are drawn.

2 The Fractional Bounded Arrival Process

In this section, a new envelope process which is capable of representing both the cumulated and the incremental traffic is proposed. Let the cumulated traffic of a self-similar source be given by (2), where $\mathbf{Z}(t)$ is an arbitrary self-similar process. This paper assumes that a process $\mathbf{Z}(t)$ is self-similar if the equality

$$\mathbf{Z}(t) \stackrel{d}{=} m^{-H}\mathbf{Z}(mt),\ m,t \geq 0, \tag{8}$$

holds for some $0 < H < 1$. The symbol $\stackrel{d}{=}$ denotes equality in distribution. Notice that, from such a definition of self-similarity, it is possible to conclude that $\mathbf{Z}(t) = t^H\mathbf{Z}(1)$. Assuming the increments of $\mathbf{Z}(t)$ to be stationary, i.e., $\mathbf{Z}(t+\tau) - \mathbf{Z}(t) \stackrel{d}{=} \mathbf{Z}(\tau)$, for $\forall t, \tau \geq 0$, the increments of the process $\mathbf{A}(t)$ in the interval $[t; t+\tau]$ are then given by

$$\begin{aligned} \Delta\mathbf{A}(t; t+\tau) &= \mathbf{A}(t+\tau) - \mathbf{A}(t) \\ &= \mu\tau + \gamma\mathbf{Z}(\tau). \end{aligned} \tag{9}$$

The parameter τ represents the *time scale* over which the amount of traffic is evaluated. Notice that the process $\Delta\mathbf{A}(t; t+\tau)$ is also stationary, and can thus be denoted by $\Delta\mathbf{A}(\tau)$ for the sake of simplicity. An envelope process for $\Delta\mathbf{A}(\tau)$ can be defined as

$$\Delta\widehat{A}(\tau) = \mu\tau + k\gamma\tau^H,\ \forall\tau \geq 0. \tag{10}$$

Notice that $\Delta\widehat{A}(\tau)$ can be considered a *universal traffic model*, since no marginal distribution is *a priori* assumed for the traffic. The process $\Delta\widehat{A}(\tau)$ will be called a *Fractional Bounded Arrival Process* (FBAP), since it can be regarded as a generalization of the well-known *Linear Bounded Arrival Process* (LBAP), which is obtained for $H = 0$ [32, 33]. Since the latter has widely been studied in the litterature, this paper focuses on the self-similar case, for which $0 < H < 1$.

The probability that the traffic increments exceed $\Delta\widehat{A}(\tau)$ is

$$\begin{aligned} \mathbb{P}\left\{\Delta\mathbf{A}(\tau) > \Delta\widehat{A}(\tau)\right\} &= \mathbb{P}\left\{\mathbf{Z}(1) > k\right\} \\ &= \overline{F}_{\mathbf{Z}}(k). \end{aligned} \tag{11}$$

This result differs from the one which would be obtained if the fBm envelope process were considered, which is given by (5). Since traffic increments were assumed to be stationary, the FBAP envelope provides a better representation with a probability of violation that is constant in time.

The parameters of a statistical FBAP envelope can be obtained as follows. Let $\mathbf{A}[n]$ be the time series for which the envelope shall be obtained, i.e.,

$$\mathbf{A}[n] = \mathbf{A}(nT)\,,\ n \in \mathbb{N}, T \geq 0.$$

The discrete increments corresponding to $\mathbf{A}[n]$ are then given by

$$\begin{aligned} \Delta\mathbf{A}_{(i)}[n] &= \mathbf{A}[(n+1)i] - \mathbf{A}[ni] \\ &= \mu T i + \gamma \mathbf{Z}[i]\,,\ n, i \in \mathbb{N} \end{aligned} \tag{12}$$

where $\mathbf{Z}[i]$ is a discrete process corresponding to the continuous process $\mathbf{Z}(t)$. For the sake of simplicity, the index (i) will be omitted from the notation whenever i is equal to 1. The sample-path realization of $\Delta\mathbf{A}_{(i)}[n]$ is denoted by $\Delta A_{(i)}[n]$.

Assuming that the type of the marginal distributions of $\Delta A_{(i)}[n]$ is known, the parameters μ and γ can be obtained by fitting such distributions to the actual traffic. For example, if the marginal distributions of $\Delta A_{(i)}[n]$ are gaussian, these parameters are given by

$$\mu = \frac{1}{T}\mathbb{E}\left\{\Delta A[n]\right\} \tag{13}$$

$$\gamma = \frac{1}{T^H}\sqrt{\mathrm{Var}\left\{\Delta A[n]\right\}}. \tag{14}$$

For the estimation of the parameter H, several methods are available [34, 35]. Finally, the parameter k should be chosen so that a target probability of violating the envelope is obtained, i.e.,

$$\varepsilon = \overline{F}_{\mathbf{Z}}(k),$$

where $\overline{F}_{\mathbf{Z}}$ now denotes the residual distribution function of $\mathbf{Z}[1]$, which is given by $\Delta\mathbf{A}[n] - \mu T$.

The use of the FBAP model is illustrated in Fig. 1. The trace to be represented is a *fractional Gaussian noise* (fGn) time series consisting of 10^6 samples, which was generated using the method proposed in [36]. For the sake of simplicity, T is assumed to be equal to 1. The mean and the standard deviation of the time series are 0.800 and 0.160, respectively. The corresponding self-similarity parameter H was estimated by using the Wavelet method [35], and a value of 0.794 was obtained. For the FBAP envelope, the parameters μ and γ are given by (13) and (14), respectively. The parameter H is chosen to match the value estimated for the trace, and the parameter k is equal to 3.719, for which a probability of violation of 10^{-4} is obtained.

In order to validate the FBAP envelope, an *empirical envelope* for the time series $\Delta\mathbf{A}_{(i)}[n]$ can be defined as

$$\Delta A^{(e)}_{(i)}[n] = \mu T i + Z^{(e)}[i]. \tag{15}$$

The parameter μ is given by (13), and $Z^{(e)}[i]$ is given by

$$Z^{(e)}[i] = \left\{ x \in \mathbb{R} : \mathbb{P}\left\{ \Delta\mathbf{A}_{(i)}[n] - \mu T i > x \right\} = \varepsilon \right\}, \tag{16}$$

where ϵ is the target probability of violation, and $\mathbb{P}\left\{ \Delta\mathbf{A}_{(i)}[n] - \mu T i > x \right\}$ is, for each time scale, directly obtained from the time series.

From Fig. 1, it is possible to verify that the FBAP model accurately represents the traffic, even for large time scales. The difference between the FBAP envelope and the empirical envelope are probably due to imprecisions in the generation of the series and in the estimation of the fGn parameters.

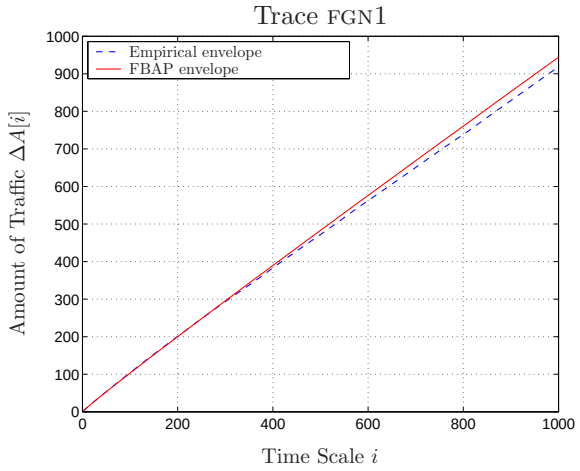

Figure 1: Validation of the FBAP model for a fractional Gaussian noise trace.

Strictly speaking, real network traffic is neither stationary, nor self-similar. Therefore, the FBAP envelope for a given real traffic trace can be obtained as follows. Let ϵ be the target probability of violation for the FBAP envelope. Also, let the empirical envelope be given by (15) and (16), where μ is given by (13). Assuming the trace to be sufficiently long, the probability indicated in (16) can be approximated by the relative frequency of the event $\left\{ \Delta\mathbf{A}_{(i)}[n] - \mu T i > x \right\}$, whenever the actual distribution is not known. Alternatively, $Z^{(e)}[i]$ can be given by $\max_n \Delta\mathbf{A}_{(i)}[n] - \mu T i$, in which case the empirical envelope degenerates into the one described in [31].

The product $k\gamma$ and the parameter H can be obtained by adjusting the function $\Delta\widehat{A}[i] - \mu i T = k\gamma(iT)^H$ so that it bounds $Z^{(e)}[i]$ from above up to a sufficiently large time scale. However, such an approach generally leads to a loose FBAP envelope. A more realistic approach is choosing the values of $k\gamma$ and H so that an arbitrarily small violation of the empirical envelope (15) is tolerated. For example, this can be achieved by solving

$$\{k\gamma, H\} = \arg \min_{\substack{0 \le H \le 1 \\ k\gamma \ge 0}} \left\{ \frac{1}{2} \sum_{i=1}^{N} \left[f\left(\log Z^{(e)}[i] - \log k\gamma - H \log i \right) \right]^2 \right\}, \tag{17}$$

where

$$f(x) = \begin{cases} x, & x < 0 \\ wx, & x \geq 0,\ w > 1. \end{cases}$$

The value of w must be chosen so that a feasible compromise between the violation of (15) and the looseness of the envelope is obtained. In some cases, a better representation is achieved by associating (17) with another estimation procedure that performs better for a given trace. For example, the product $k\gamma$ can be given by

$$k\gamma = \max_n \Delta A[n], \tag{18}$$

and the parameter H be estimated by solving

$$H = \arg \min_{0 \leq H \leq 1} \left\{ \frac{1}{2} \sum_{i=1}^{N} \left[f\left(\log Z^{(e)}[i] - \log k\gamma - H \log i \right) \right]^2 \right\}. \tag{19}$$

The use of the FBAP envelope for real traffic modeling is illustrated in Fig. 2–13. The traces BCpAug89, BCpOct89, BCOct89Ext, BCOct89Ext4, lbltcp3, lblpkt4–5 and decpkt2–4 correspond to actual network traffic traces, which are known to present some sort of scaling law and are often used in fractal traffic studies [1, 2, 4, 37, 38].

The original traces were preprocessed to create sequences which have no more than 1×10^6 samples, which represent the amount of traffic (in bytes) that flows through the network during a given interval of time. An interval of time of 10ms was assumed for the traces BCpAug89, BCpOct89, BCOct89Ext and BCOct89Ext4. For the traces lbltcp3, lblpkt4–5 and decpkt2–4, an interval of 100ms was arbitrarily chosen.

The remaining traces correspond to actual MPEG sequences, which are also known to exhibit some sort of scaling law [5, 8]. Each trace consists of a discrete sequence of 4×10^4 samples, which correspond to the size (in bytes) of a complete MPEG video frame. Since the frames are generated at a rate of 20fps, each sample correspond to the amount of traffic (in bytes) that flows through the network during an interval of time of 50ms.

For the FBAP model, a target probability of violation of 10^{-4} was arbitrarily chosen. The parameter μ was obtained by using (13). The product $k\gamma$ and the parameter H were obtained either by using (17) or by using (18) and (19). In both cases, $w = 10$ was assumed. These approaches are referred to as Method 1 and Method 2, respectively. For each trace, the sequence $Z^{(e)}[i]$ and the adjusted function $\Delta\widehat{A}[i] - \mu Ti$ are shown in Fig. 2–7. Notice that the traces asterix, bond, fuss1, mtv2, lambs, movie, star2, talk1, terminator, BCOct89Ext4, BCpAug89, BCpOct89, decpkt2–4, lblpkt5 and lbltcp3, are actually consistent with statistical self-similarity. Thus, the FBAP model is expected to perform better for these traces than for the other ones, which significatively deviate from self-similarity.

The resulting FBAP parameters, as well as the approach which was used to obtain $k\gamma$ and H, are indicated in Table 1. The empirical envelopes and the resulting FBAP envelopes for each trace

Trace	Method	μ	$k\gamma$	H
BCOct89Ext	2	7.49×10^2	5.12×10^4	0.676
BCOct89Ext4	2	6.22×10^3	1.90×10^5	0.949
BCpAug89	1	1.38×10^5	4.54×10^5	0.759
BCpOct89	2	3.63×10^5	6.24×10^5	0.911
asterix	1	5.59×10^5	1.56×10^6	0.740
atp	2	5.47×10^5	1.54×10^6	0.699
bond	2	6.08×10^5	1.41×10^6	0.659
decpkt2	2	2.38×10^5	3.79×10^5	0.793
decpkt3	2	1.81×10^5	3.04×10^5	0.766
decpkt4	2	2.63×10^5	3.09×10^5	0.789
dino	2	3.27×10^5	9.66×10^5	0.723
fuss1	1	6.78×10^5	1.55×10^6	0.639
fuss2	2	6.28×10^5	1.92×10^6	0.774
lambs	2	1.83×10^5	1.02×10^6	0.710
lblpkt4	1	3.64×10^4	2.57×10^5	0.736
lblpkt5	2	2.61×10^4	2.30×10^5	0.718
lbltcp3	2	3.39×10^4	3.76×10^5	0.814
movie	2	3.57×10^5	1.04×10^6	0.641
mrbean	2	4.41×10^5	1.59×10^6	0.777
mtv1	2	6.15×10^5	2.20×10^6	0.797
mtv2	2	4.95×10^5	2.61×10^6	0.769
news1	2	5.17×10^5	1.47×10^6	0.694
news2	1	3.84×10^5	2.00×10^6	0.733
race	1	7.69×10^5	2.03×10^6	0.732
sbowl	2	5.88×10^5	1.20×10^6	0.727
simpsons	2	4.64×10^5	1.19×10^6	0.702
star2	2	2.33×10^5	1.16×10^6	0.729
talk1	2	3.63×10^5	8.96×10^5	0.717
talk2	2	4.48×10^5	1.22×10^6	0.766
terminator	1	2.73×10^5	5.11×10^5	0.626

Table 1: Parameters of the FBAP envelopes corresponding to the traces analyzed in this paper.

are shown in Fig. 8–13. Notice that the FBAP model actually provides an accurate representation for the traces which are consistent with statistical self-similarity. On the other hand, for those which exhibit a more complex scaling law, a multifractal approach must be considered.

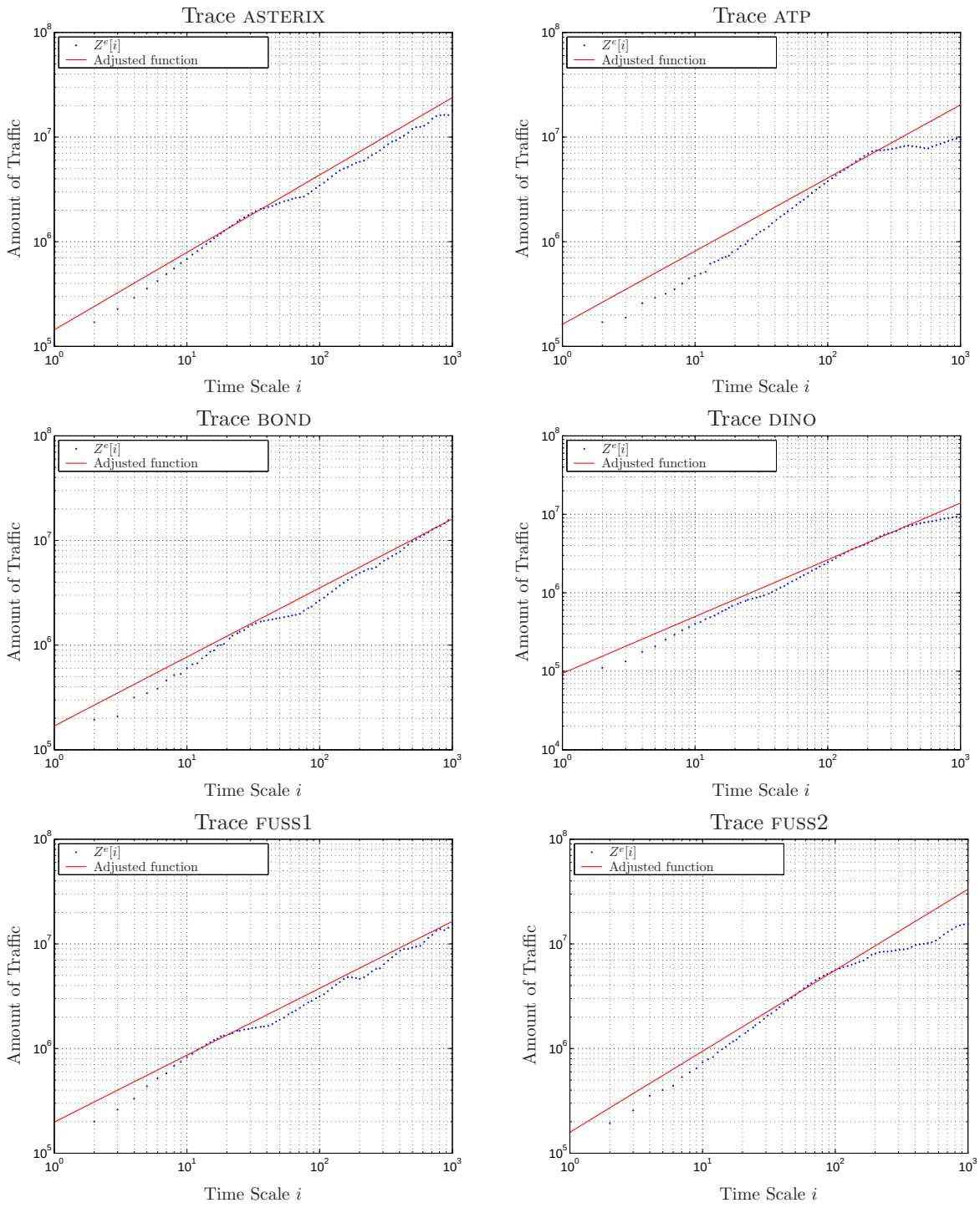

Figure 2: The sequence $Z^{(e)}[i]$ and the adjusted function $\Delta\widehat{A}[i] - \mu Ti$ for the traces ASTERIX, ATP, BOND, DINO, FUSS1 and FUSS2.

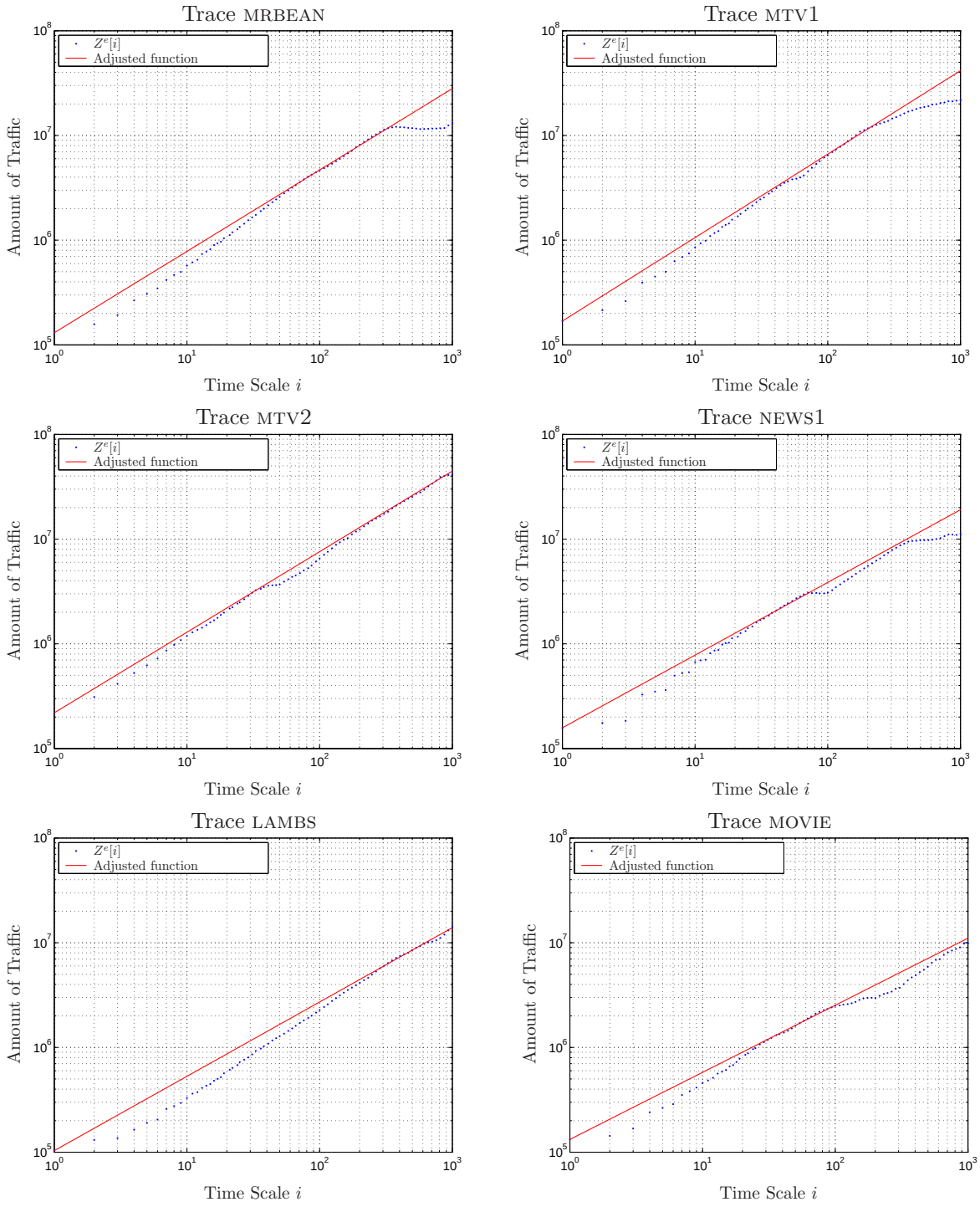

Figure 3: The sequence $Z^{(e)}[i]$ and the adjusted function $\Delta\widehat{A}[i] - \mu Ti$ for the traces MRBEAN, MTV1, MTV2, NEWS1, LAMBS and MOVIE.

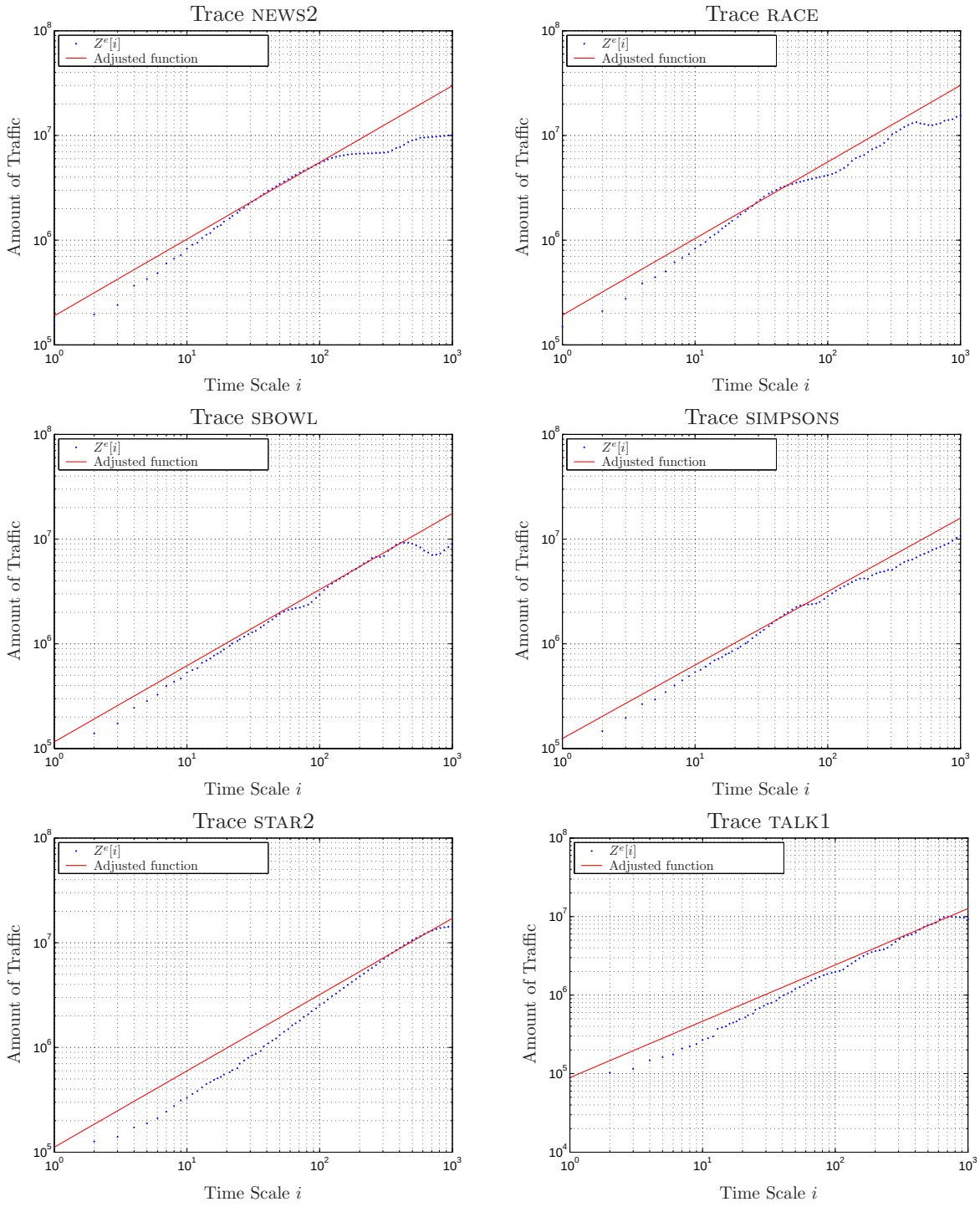

Figure 4: The sequence $Z^{(e)}[i]$ and the adjusted function $\Delta\widehat{A}[i] - \mu T i$ for the traces NEWS2, RACE, SBOWL, SIMPSONS, STAR2 and TALK1.

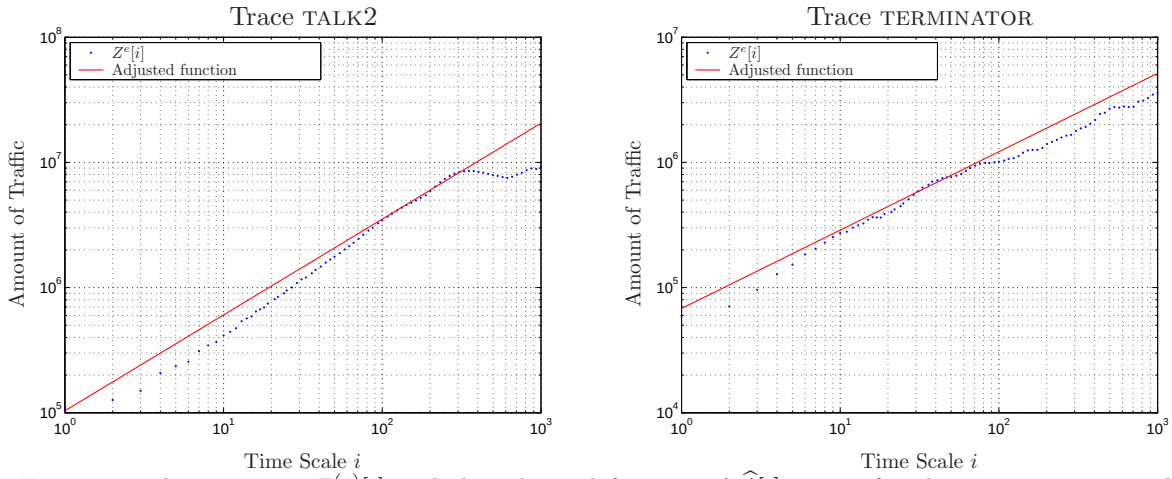

Figure 5: The sequence $Z^{(e)}[i]$ and the adjusted function $\Delta\widehat{A}[i] - \mu Ti$ for the traces TALK2 and TERMINATOR by using the FBAP envelope.

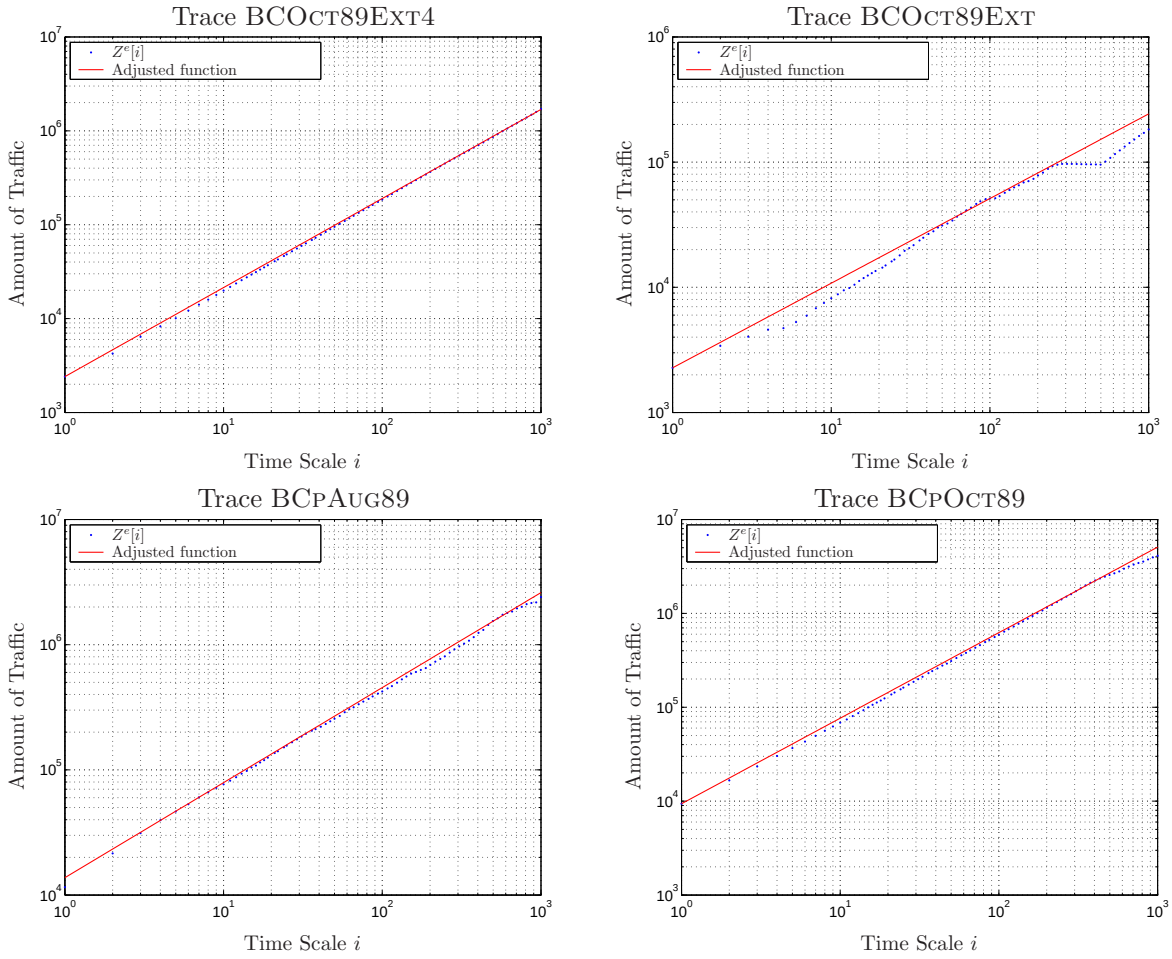

Figure 6: The sequence $Z^{(e)}[i]$ and the adjusted function $\Delta\widehat{A}[i] - \mu Ti$ for the traces BCOct89Ext4, BCOct89Ext, BCpAug89 and BCpOct89 by using the FBAP envelope.

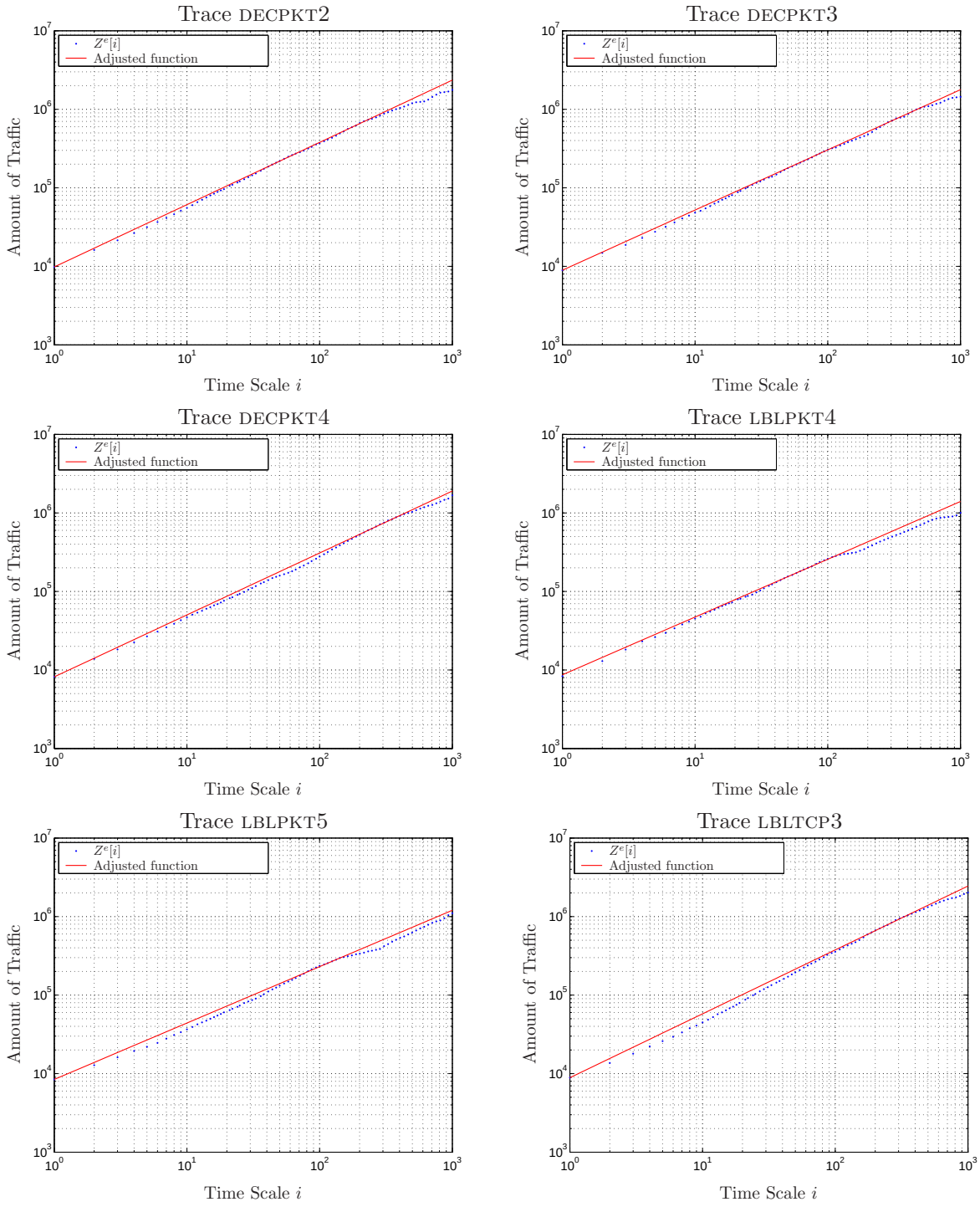

Figure 7: The sequence $Z^{(e)}[i]$ and the adjusted function $\Delta\widehat{A}[i] - \mu Ti$ for the traces DECPKT2, DECPKT3, DECPKT4, LBLPKT4, LBLPKT5 and LBLTCP3 by using the FBAP envelope.

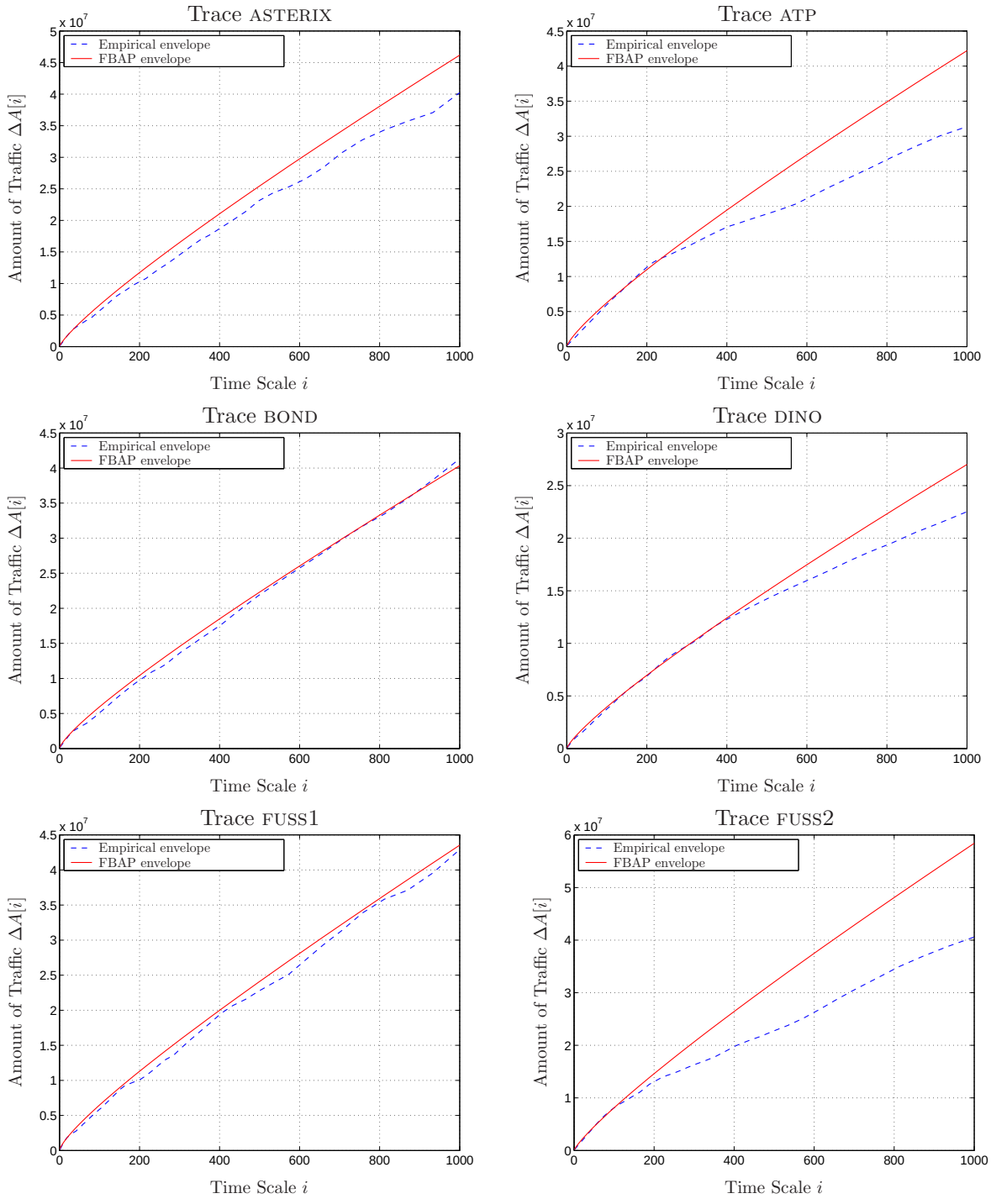

Figure 8: Representation of the traces ASTERIX, ATP, BOND, DINO, FUSS1 and FUSS2 by using the FBAP envelope.

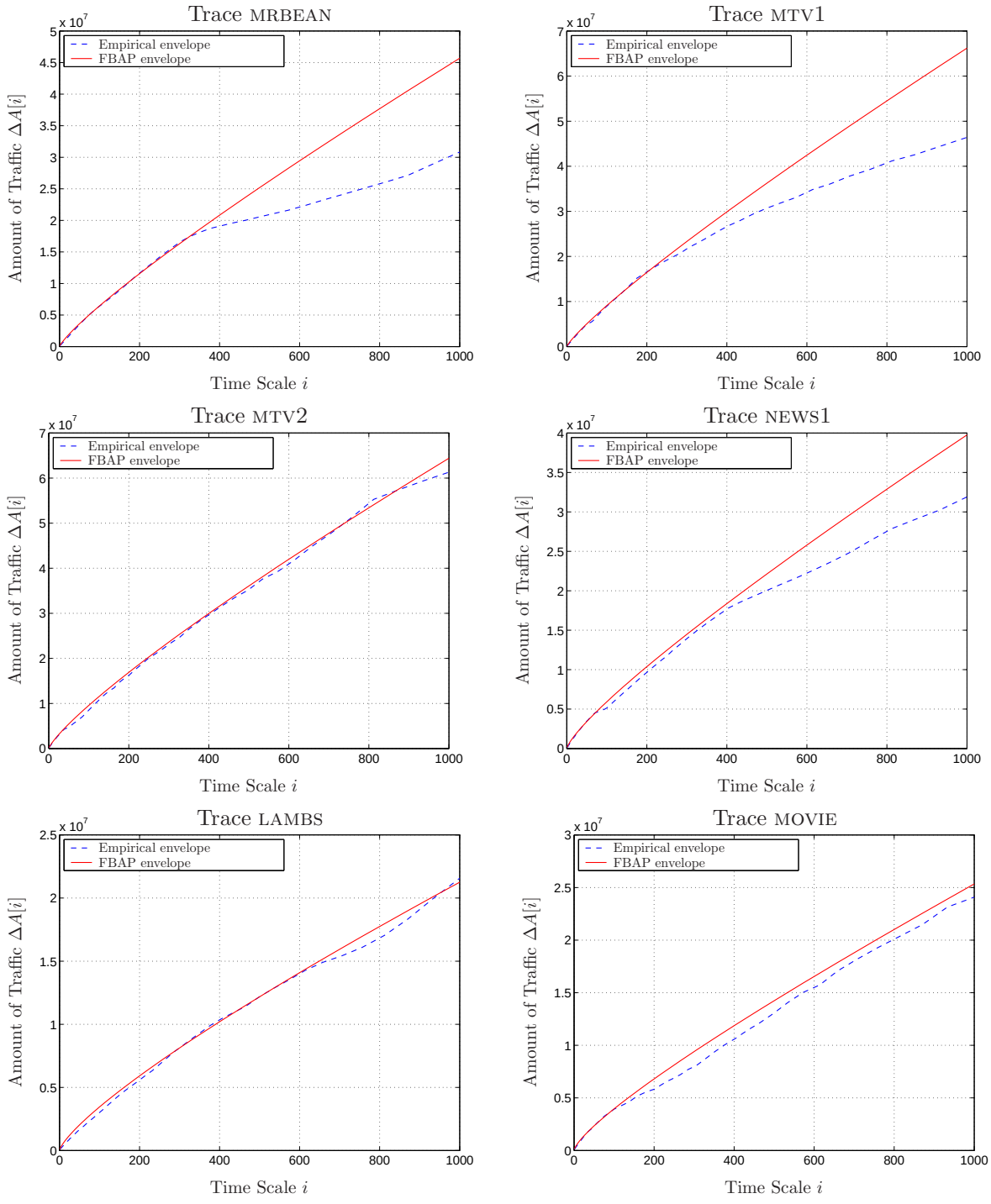

Figure 9: Representation of the traces MRBEAN, MTV1, MTV2, NEWS1, LAMBS and MOVIE by using the FBAP envelope.

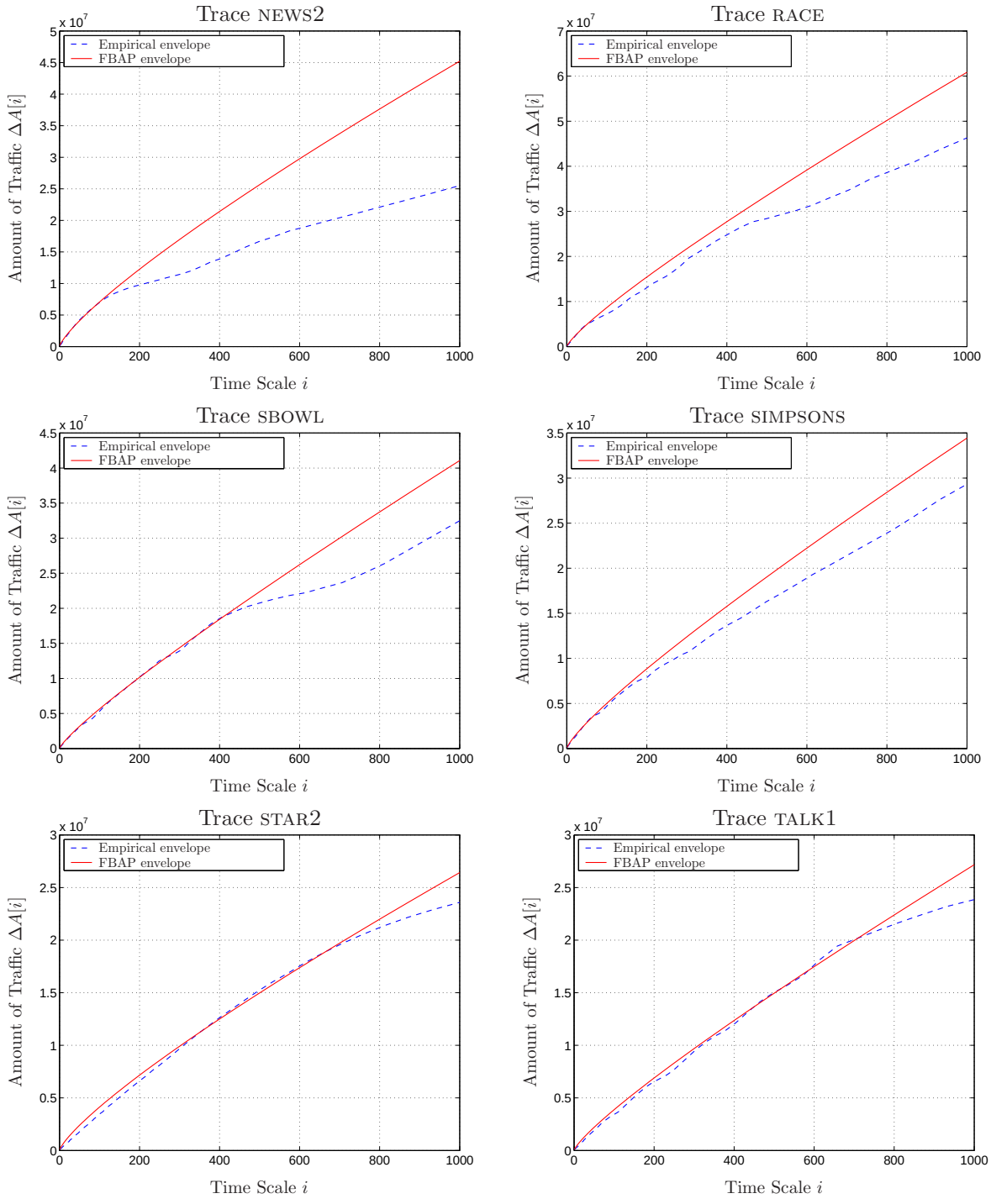

Figure 10: Representation of the traces NEWS2, RACE, SBOWL, SIMPSONS, STAR2 and TALK1 by using the FBAP envelope.

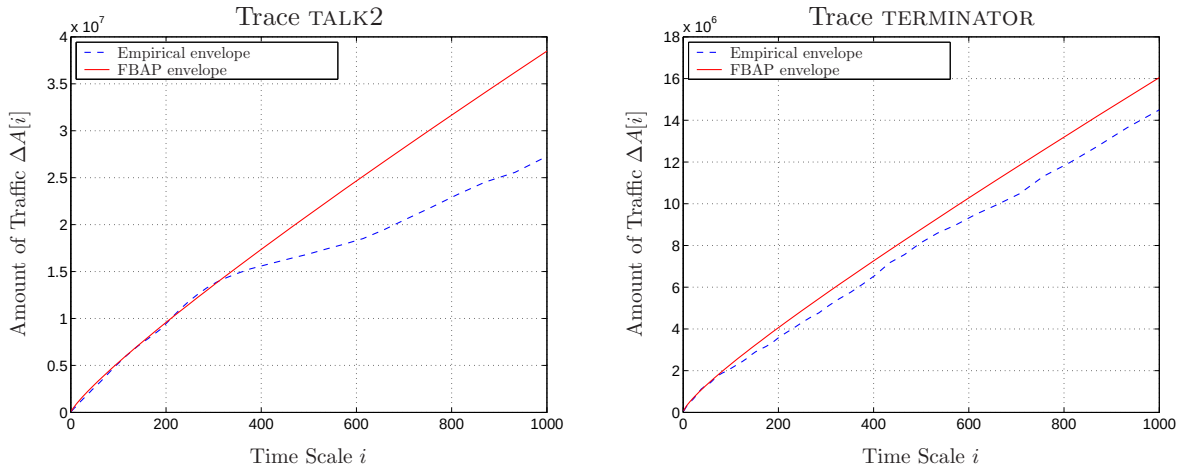

Figure 11: Representation of the traces TALK2 and TERMINATOR by using the FBAP envelope.

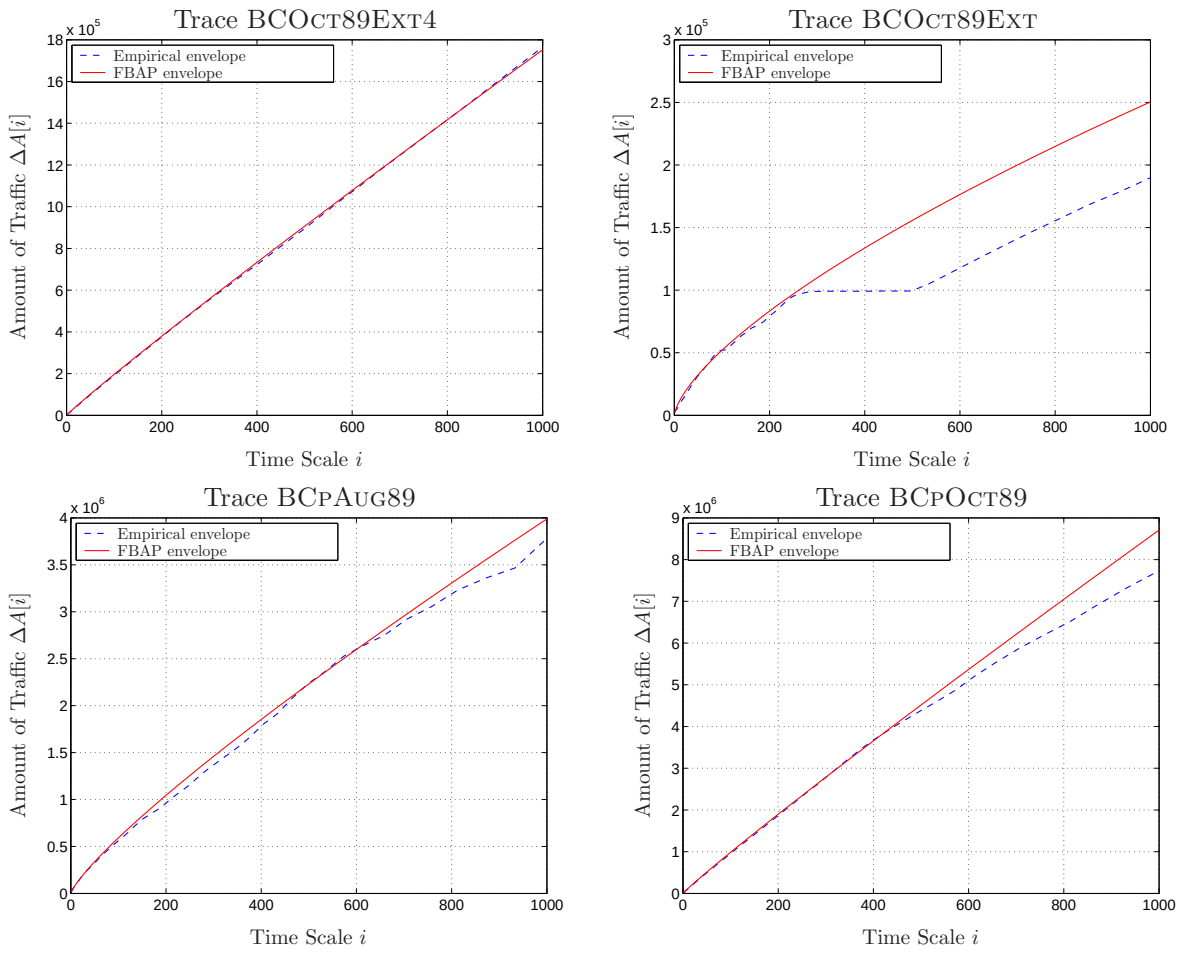

Figure 12: Representation of the traces BCOCT89EXT4, BCOCT89EXT, BCPAUG89 and BCPOCT89 by using the FBAP envelope.

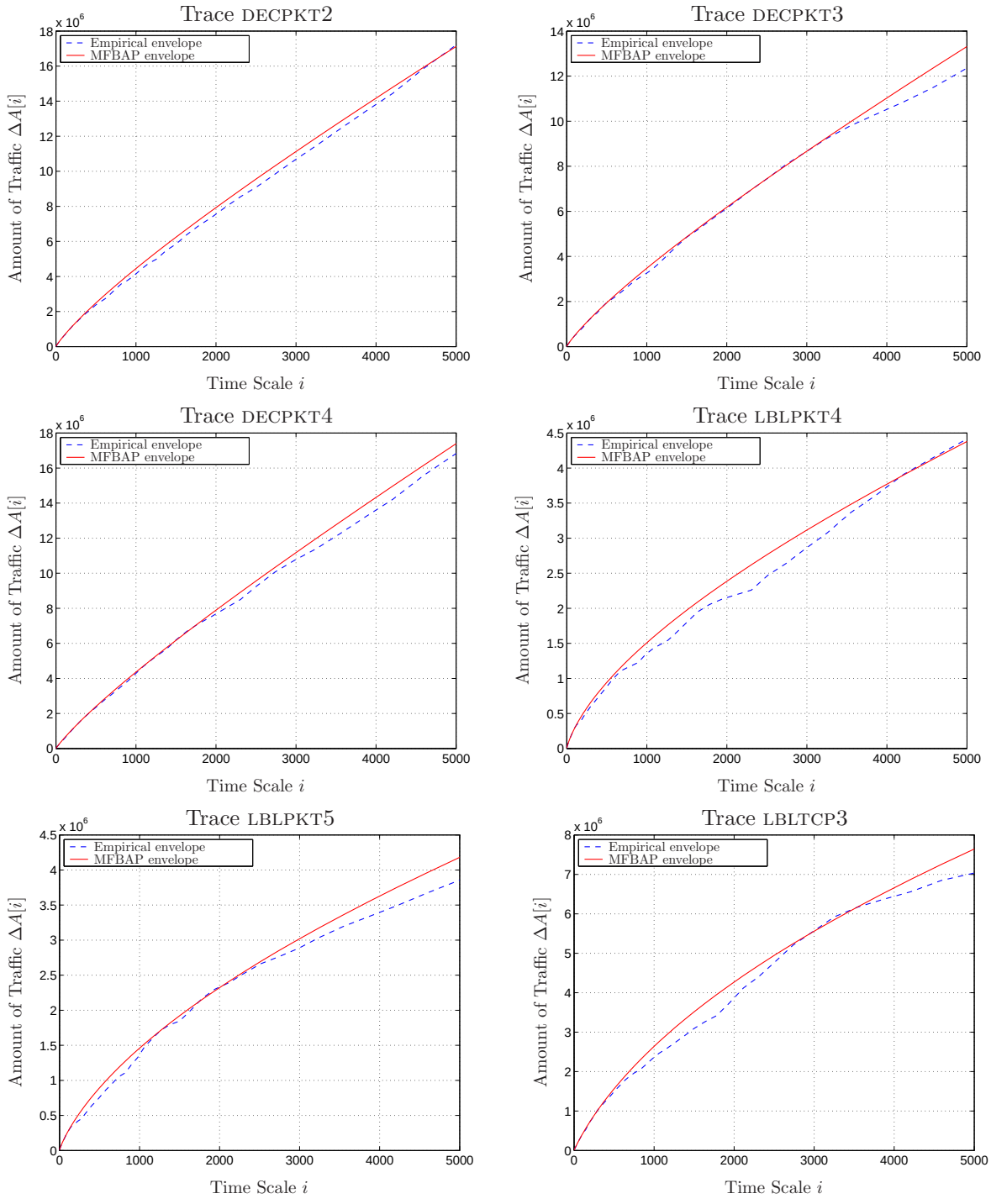

Figure 13: Representation of the traces DECPKT2, DECPKT3, DECPKT4, LBLPKT4, LBLPKT5 and LBLTCP3 by using the FBAP envelope.

3 Queueing analysis for FBAP traffic

In this section, a queueing analysis for FBAP traffic is carried out. Consider a queueing system, in which the input traffic is given by (9), the backlog is denoted by $\mathbf{Q}(t)$, and the server has a constant rate, denoted by g. In order to guarantee the stability of the system, it is assumed that the server rate g is greater than the traffic mean rate μ.

Suppose that the ith busy cycle of the server starts at time t_i. A busy cycle is a period of time in which the input traffic is continuously backlogged. Therefore,

$$\mathbf{Q}(t_i + \tau) = \mathbf{Q}(t_i) + \Delta\mathbf{A}(\tau) - g\tau,\ 0 \leq \tau < \tau_i^{busy}.$$

where τ_i^{busy} is the duration of the ith busy cycle of the server. Since the backlog at the start of a busy cycle is equal to zero,

$$\mathbf{Q}(t_i + \tau) = \Delta\mathbf{A}(\tau) - g\tau,\ 0 \leq \tau < \tau_i^{busy}.$$

Assuming the input traffic to be FBAP, an upper bound for $\mathbf{Q}(t_i + \tau)$ can be defined as

$$\begin{aligned} \widehat{Q}(\tau) &= \Delta\widehat{A}(\tau) - g\tau \\ &= (\mu - g)\tau + k\gamma\tau^H,\ 0 \leq \tau < \tau^{busy}. \end{aligned} \tag{20}$$

where τ^{busy} is given by $\arg_{\tau>0}\left\{\widehat{Q}(\tau) = 0\right\}$. The probability that $\mathbf{Q}(t_i+\tau)$ exceeds $\widehat{Q}(\tau)$ is given by

$$\begin{aligned} \mathbb{P}\left\{\mathbf{Q}(t_i + \tau) > \widehat{Q}(\tau)\right\} &= \mathbb{P}\left\{\Delta\mathbf{A}(\tau) > \Delta\widehat{A}(\tau)\right\} \\ &= \overline{F}_{\mathbf{Z}}(k), \end{aligned} \tag{21}$$

For $0 < H < 1$, it is easy to verify that $\widehat{Q}(\tau)$ is $\cap$-convex for $\tau \geq 0$. An upper bound for $\mathbf{Q}(t)$ can thus be defined as

$$\begin{aligned} Q^* &= \max_{0 \leq \tau \leq \tau^{busy}} \widehat{Q}(\tau) \\ &= (g - \mu)^{\frac{H}{H-1}} (k\gamma)^{\frac{1}{1-H}} H^{\frac{H}{1-H}} (1 - H). \end{aligned} \tag{22}$$

Notice that the delay can be bounded by $D^* = Q^*/g$, which is also the bound for the delay jitter ΔD^*.

The time scale for which Q^* is obtained is called *maximum time scale*, and represents the instant of time, from the start of the busy cycle, at which the unfinished work in the queueing system achieves its maximum value in a probabilistic sense [20]. The probability of violating the backlog bound Q^* at time $t_i + \tau$ is itself upper bounded by

$$\begin{aligned}\mathbb{P}\left\{\mathbf{Q}(t_i+\tau)>Q^*\right\} &\leq \mathbb{P}\left\{\mathbf{Q}(t_i+\tau)>\widehat{Q}(\tau)\right\}\\ &= \overline{F}_{\mathbf{Z}}(k). \end{aligned} \tag{23}$$

Now, let $\mathcal{P}$ denote the probability of violating the bound Q^* at *any* time during the busy cycle. Previous studies [20,31,39,40] assumed that $\mathcal{P}$ was also given by (23), which is not correct. Actually,

$$\begin{aligned}\mathcal{P} &= \mathbb{P}\bigcup_{\tau\geq 0}\left\{\mathbf{Q}(t_i+\tau)>Q^*\right\}\\ &= \mathbb{P}\left\{\sup_{\tau\geq 0}\mathbf{Q}(t_i+\tau)>Q^*\right\}.\end{aligned}$$

In order to evaluate $\mathcal{P}$, consider the inequality

$$\mathbb{P}\left\{\sup_{\tau\geq 0}\mathbf{Q}(t_i+\tau)>Q^*\right\}\geq \sup_{\tau\geq 0}\mathbb{P}\left\{\mathbf{Q}(t_i+\tau)>Q^*\right\}. \tag{24}$$

Therefore,

$$\begin{aligned}\mathcal{P} &\geq \sup_{\tau\geq 0}\mathbb{P}\left\{\mathbf{Q}(t_i+\tau)>Q^*\right\}\\ &= \overline{F}_{\mathbf{Z}}(k).\end{aligned} \tag{25}$$

For the fractional Brownian motion case, it was proved that (24) converges to an equality for a sufficiently large value of Q^*, i.e., a sufficiently small value of $\mathcal{P}$ [15]. In such a case,

$$\mathcal{P}\simeq \overline{F}_{\mathbf{Z}}(k).$$

Finally, it is possible to compute the rate at which the server must operate so that the backlog is bounded by Q^* with a probability of $1-\mathcal{P}$. From (22),

$$g=\mu+(k\gamma)^{\frac{1}{H}}H(1-H)^{\frac{1-H}{H}}\left(Q^*\right)^{\frac{H-1}{H}}. \tag{26}$$

Such a relation can be regarded as the *effective bandwidth* of an FBAP traffic flow, and is similar to the effective bandwidth obtained for the fBm envelope process [31].

The queueing analysis for the FBAP traffic is now illustrated. For the real traffic traces which were analyzed in the previous section, the backlog bounds and the corresponding probability of violation as functions of the service rate are shown in Fig.14–43. The backlog bound is obtained by using (22). The probability of violation is obtained via simulations, and is given by the number of busy cycles for which a violation occurs, divided by the total number of busy cycles. For the sake of simplicity, simulation experiments were conducted assuming the traffic to be a fluid-type traffic.

Moreover, the traffic is assumed to be uniformly distributed during the interval which corresponds to each sample of the traces.

Notice that, for all traces, the backlog bound fastly decreases as the service rate increases. Moreover, for most of the traces, feasible bounds (i.e., bounds which are of the same order of the traffic parameters) are obtained for values of service rate that are of the same order of the mean traffic rate. For the trace BCOct89Ext4, however, the backlog bounds are actually loose, even for high values of service rate, which indicates that the FBAP model is not appropriate in this case.

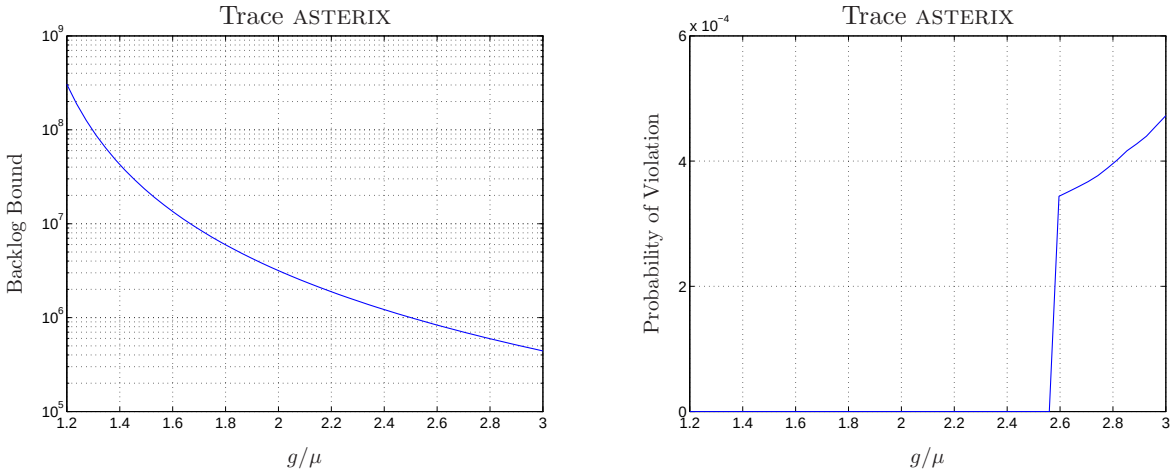

Figure 14: Backlog bound and probability of violation for the trace ASTERIX.

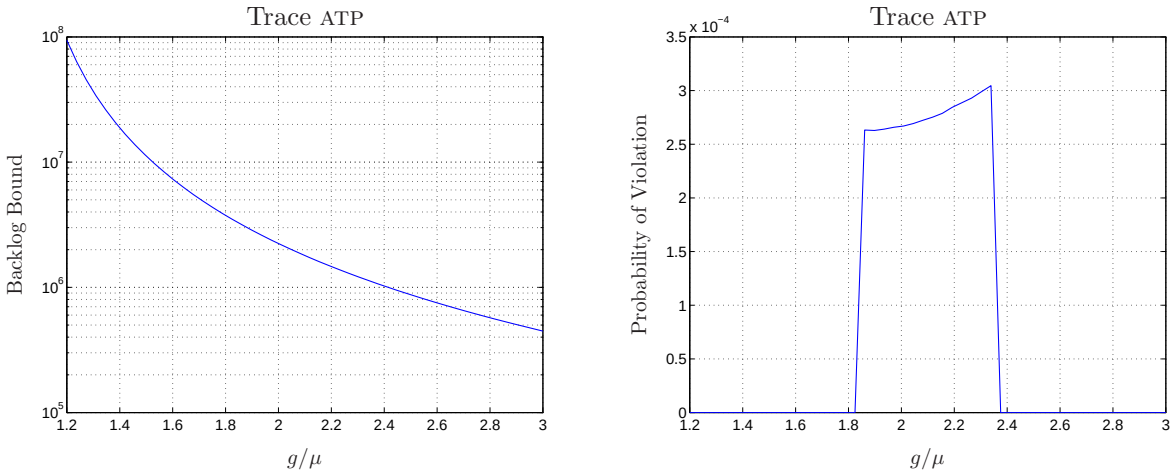

Figure 15: Backlog bound and probability of violation for the trace ATP.

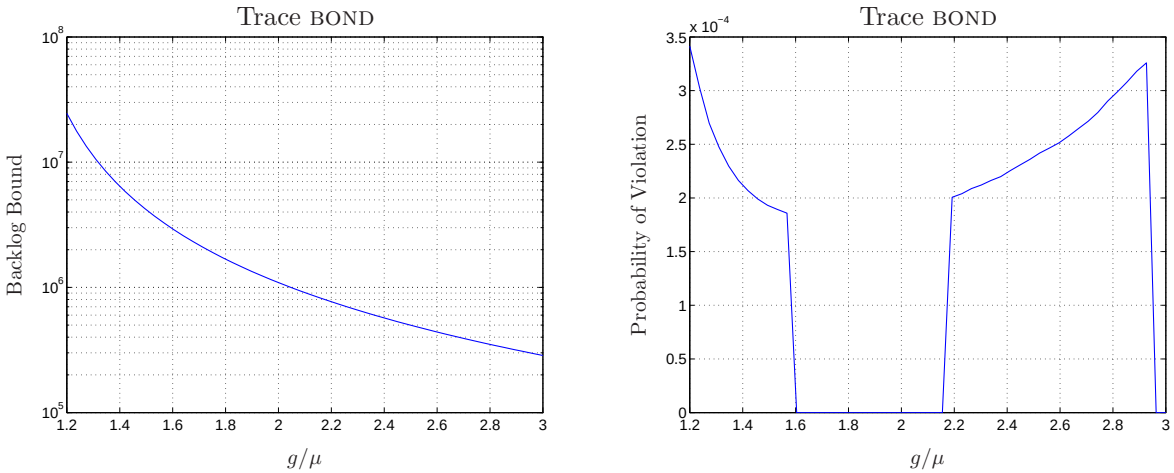

Figure 16: Backlog bound and probability of violation for the trace BOND.

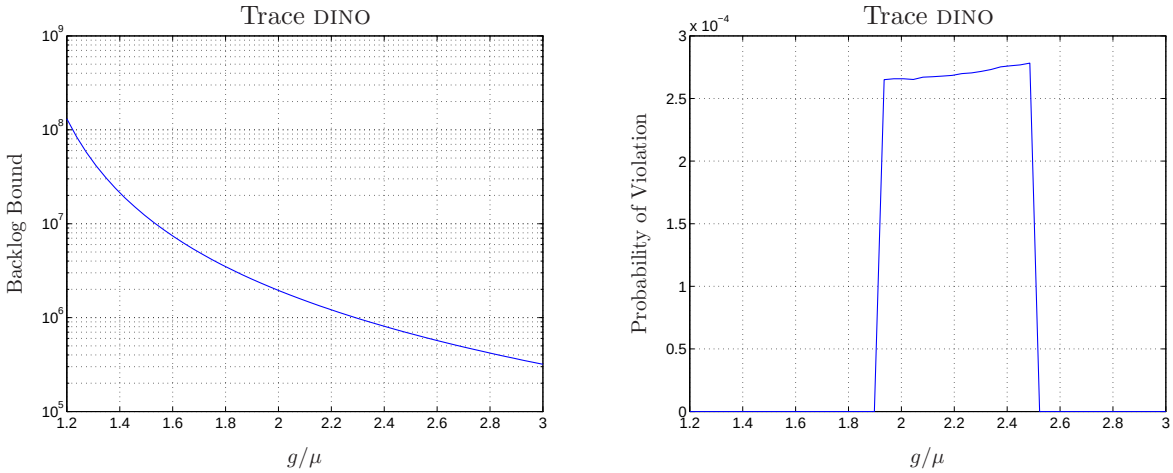

Figure 17: Backlog bound and probability of violation for the trace DINO.

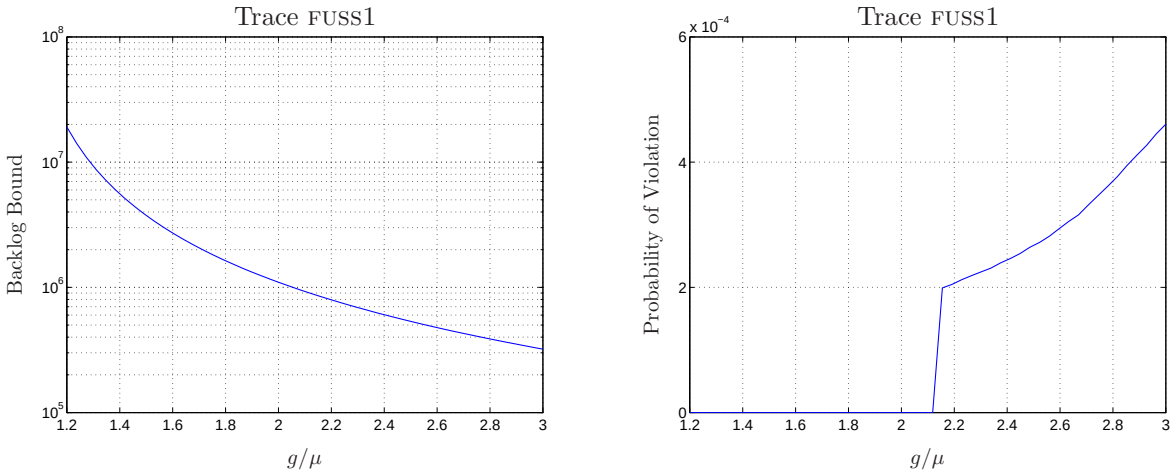

Figure 18: Backlog bound and probability of violation for the trace FUSS1.

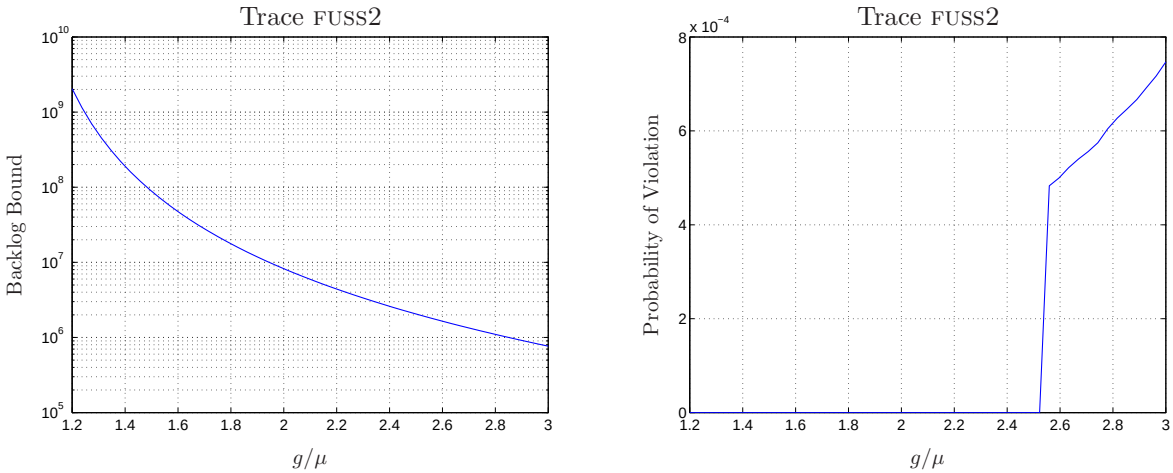

Figure 19: Backlog bound and probability of violation for the trace FUSS2.

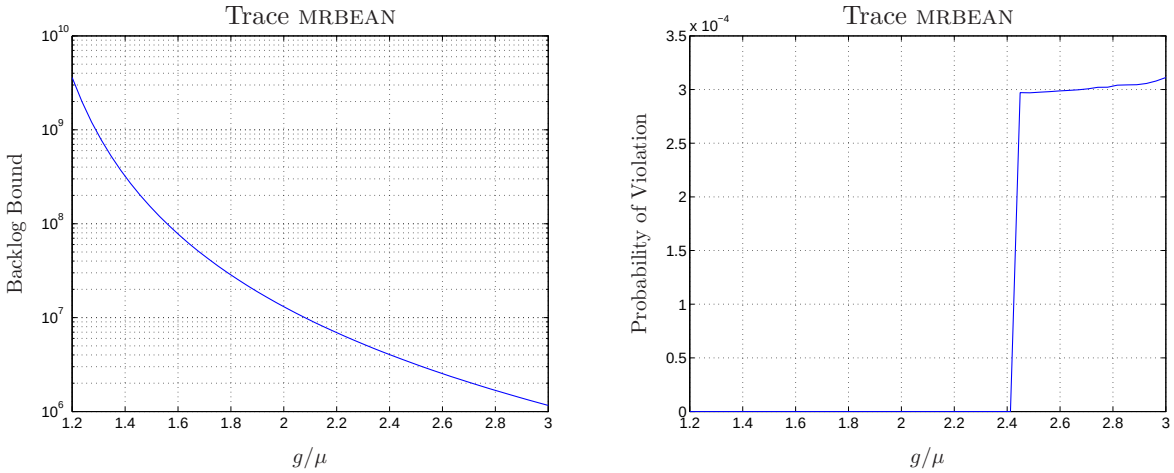

Figure 20: Backlog bound and probability of violation for the trace MRBEAN.

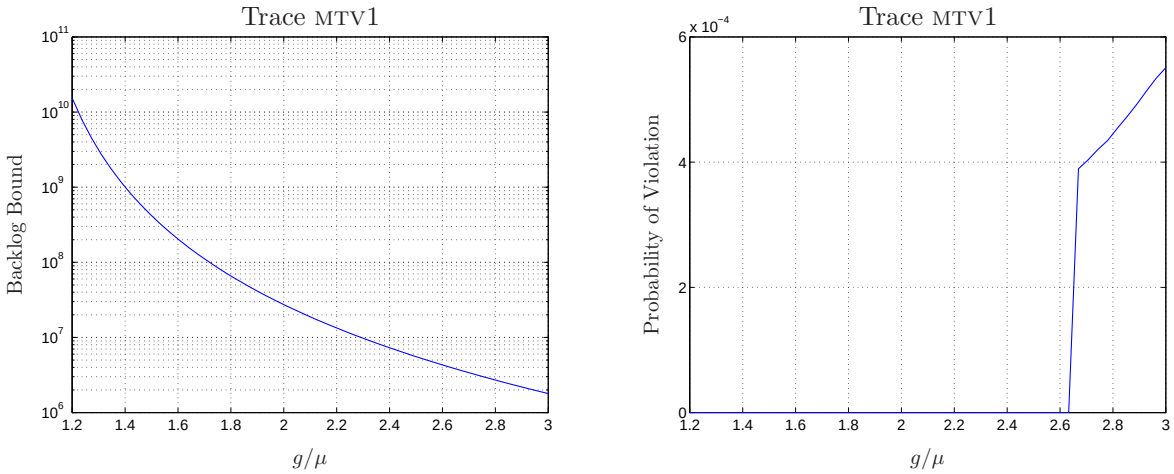

Figure 21: Backlog bound and probability of violation for the trace MTV1.

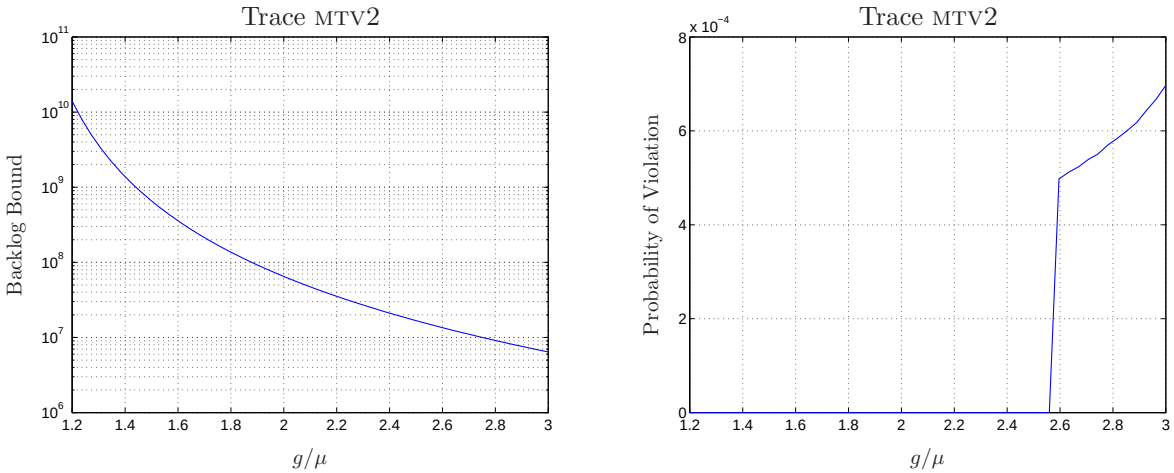

Figure 22: Backlog bound and probability of violation for the trace MTV2.

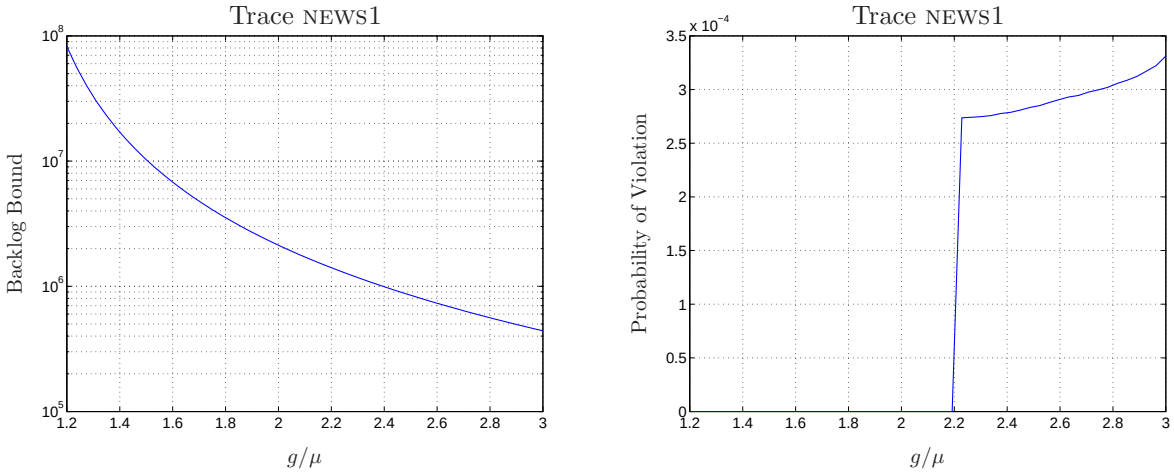

Figure 23: Backlog bound and probability of violation for the trace NEWS1.

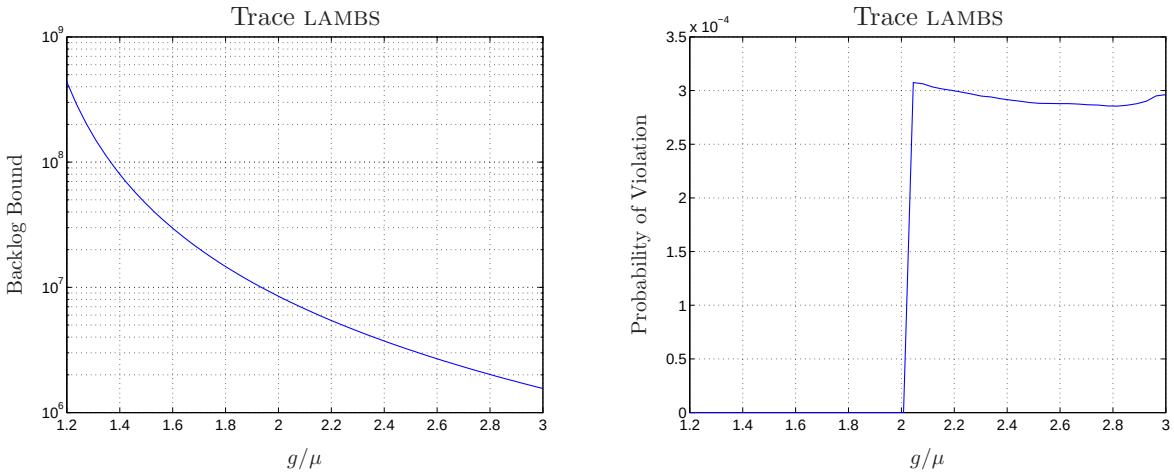

Figure 24: Backlog bound and probability of violation for the trace LAMBS.

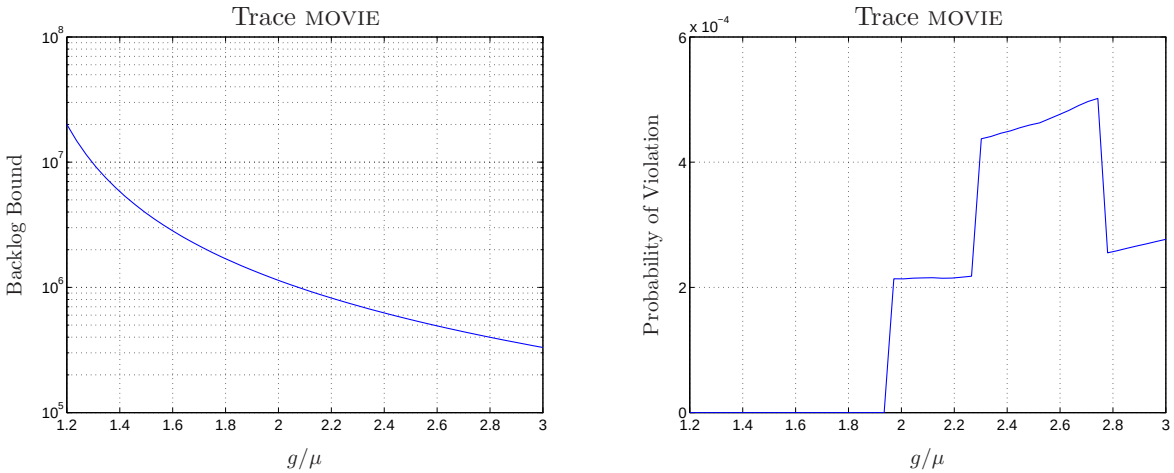

Figure 25: Backlog bound and probability of violation for the trace MOVIE.

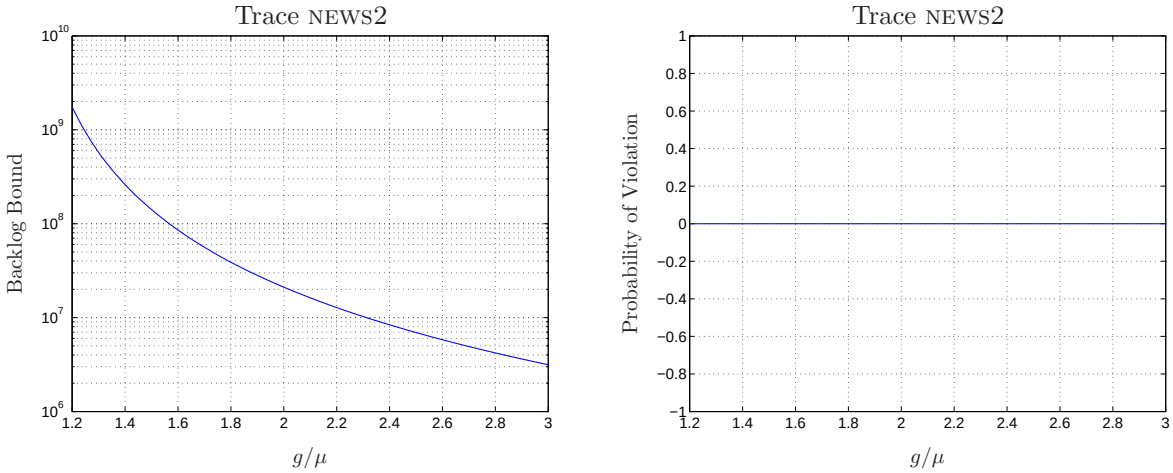

Figure 26: Backlog bound and probability of violation for the trace NEWS2.

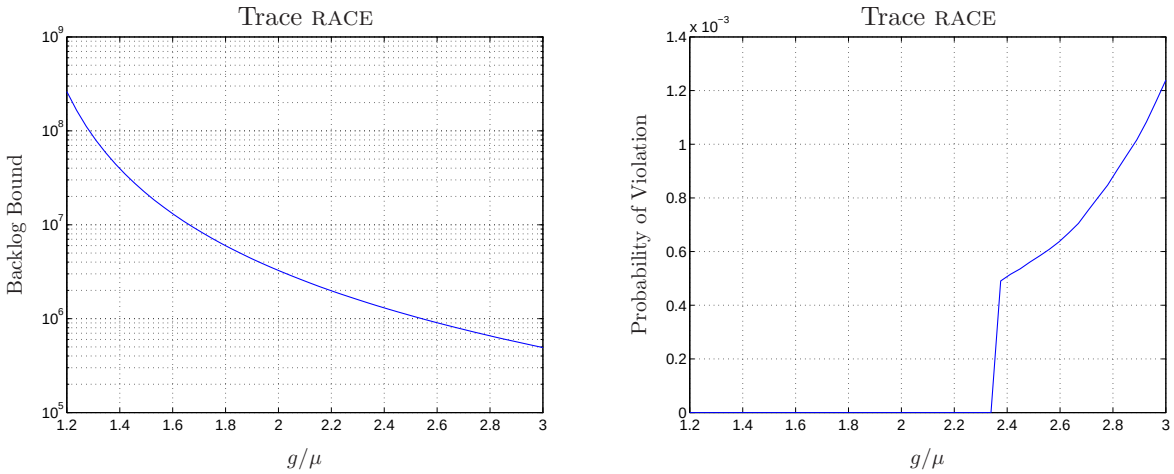

Figure 27: Backlog bound and probability of violation for the trace RACE.

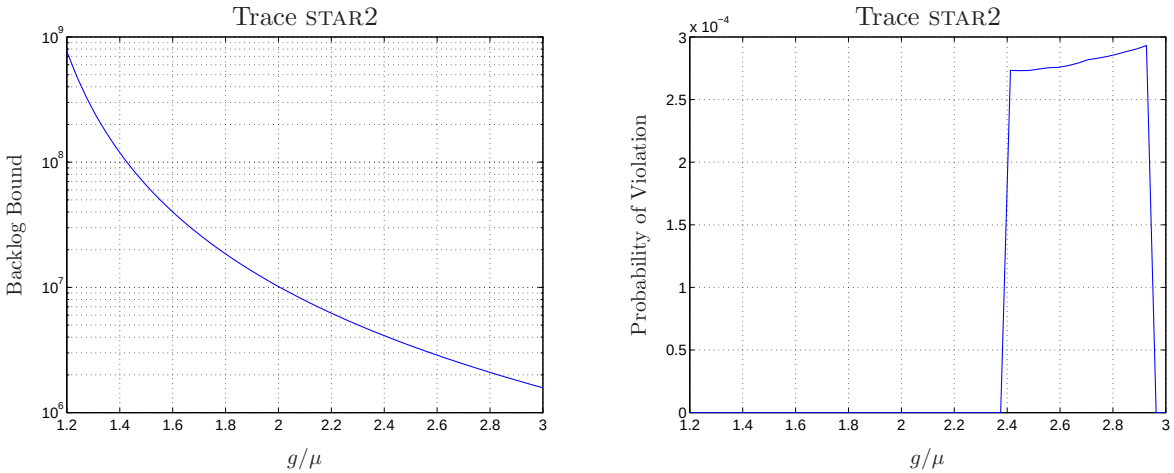

Figure 28: Backlog bound and probability of violation for the trace STAR2.

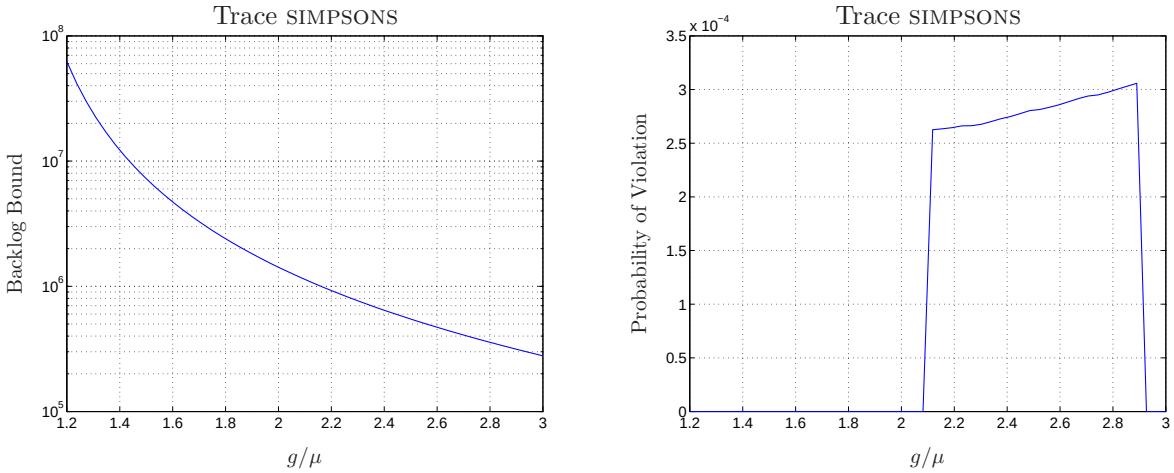

Figure 29: Backlog bound and probability of violation for the trace SIMPSONS.

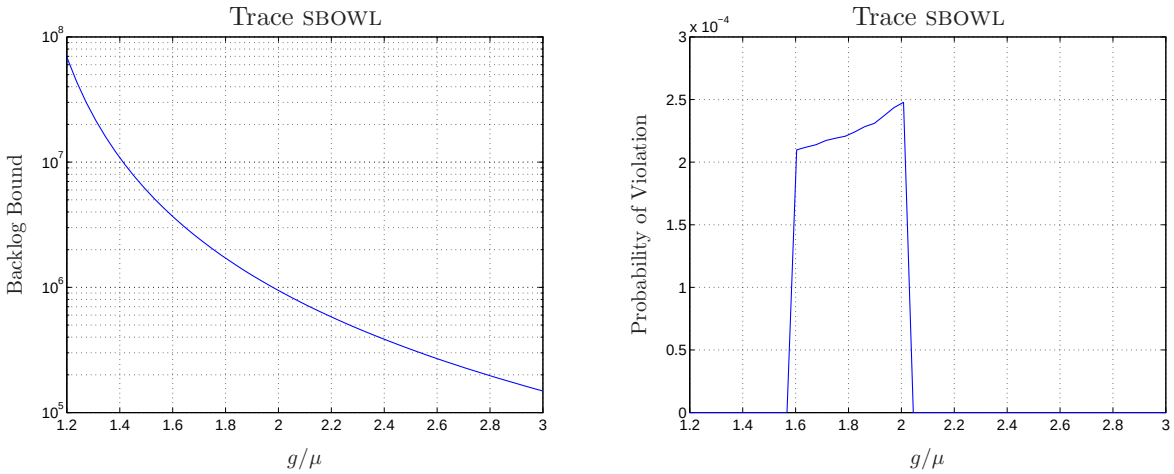

Figure 30: Backlog bound and probability of violation for the trace SBOWL.

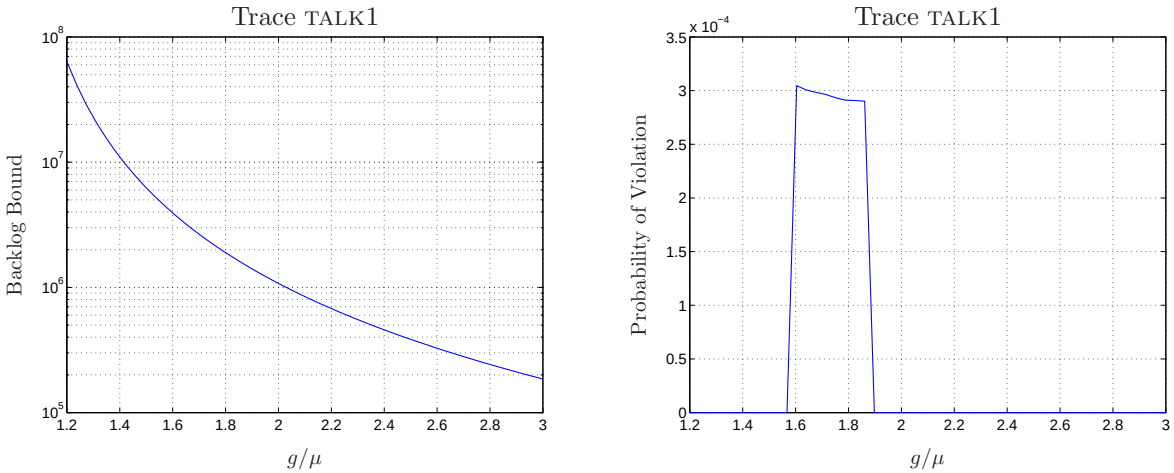

Figure 31: Backlog bound and probability of violation for the trace TALK1.

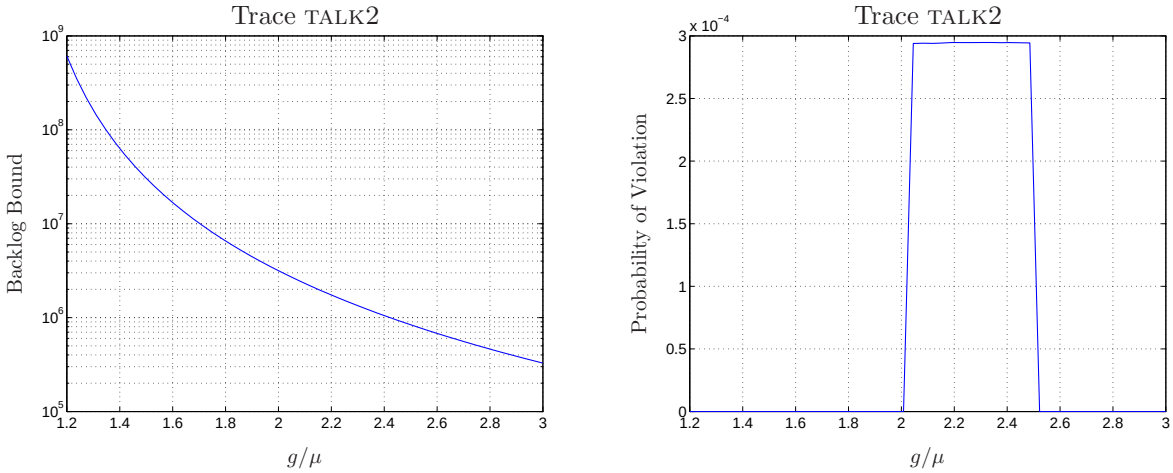

Figure 32: Backlog bound and probability of violation for the trace TALK2.

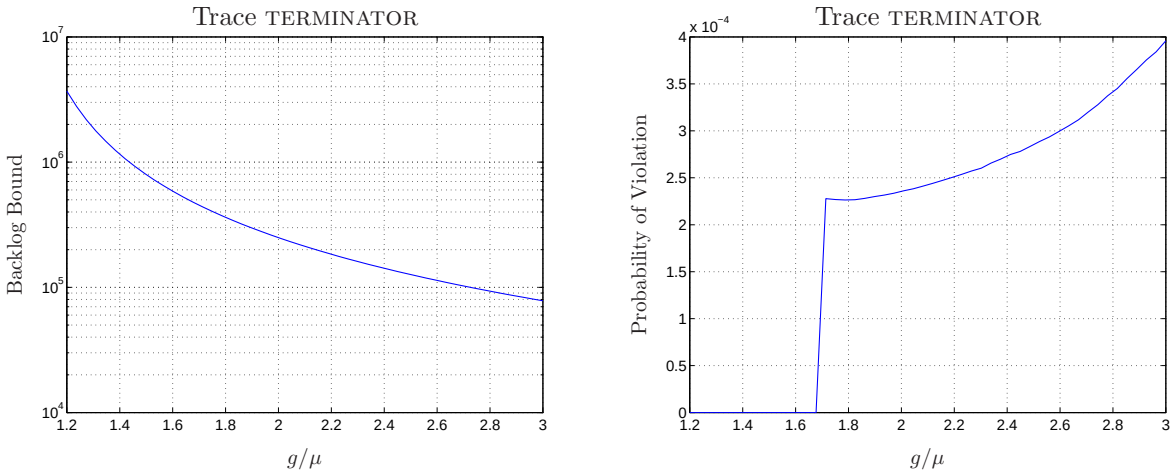

Figure 33: Backlog bound and probability of violation for the trace TERMINATOR.

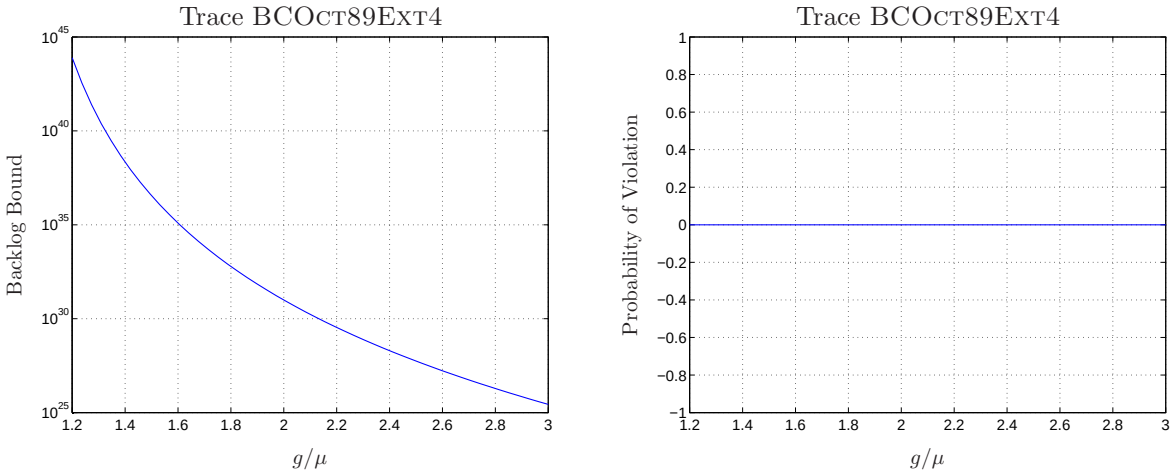

Figure 34: Backlog bound and probability of violation for the trace BCOct89Ext4.

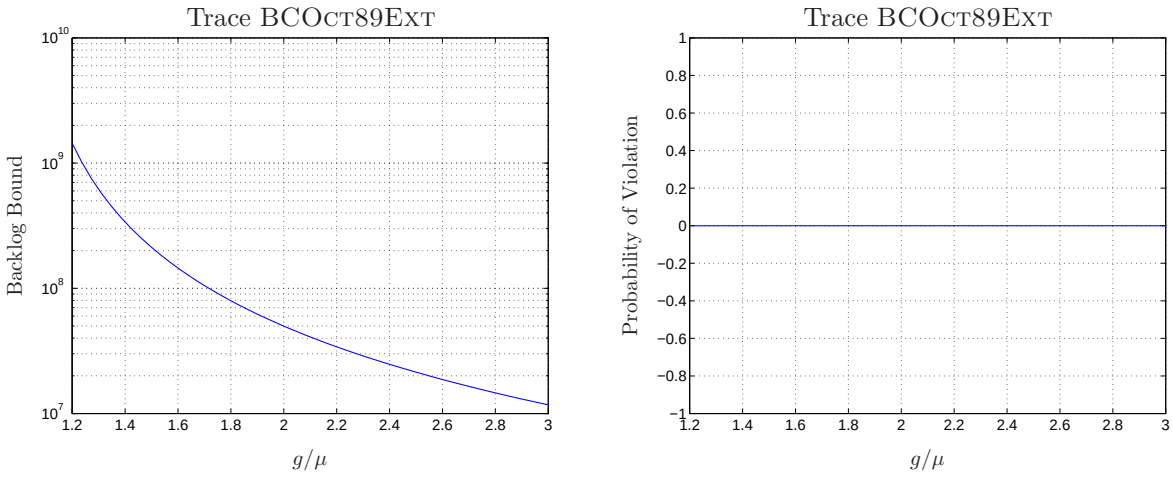

Figure 35: Backlog bound and probability of violation for the trace BCOct89Ext.

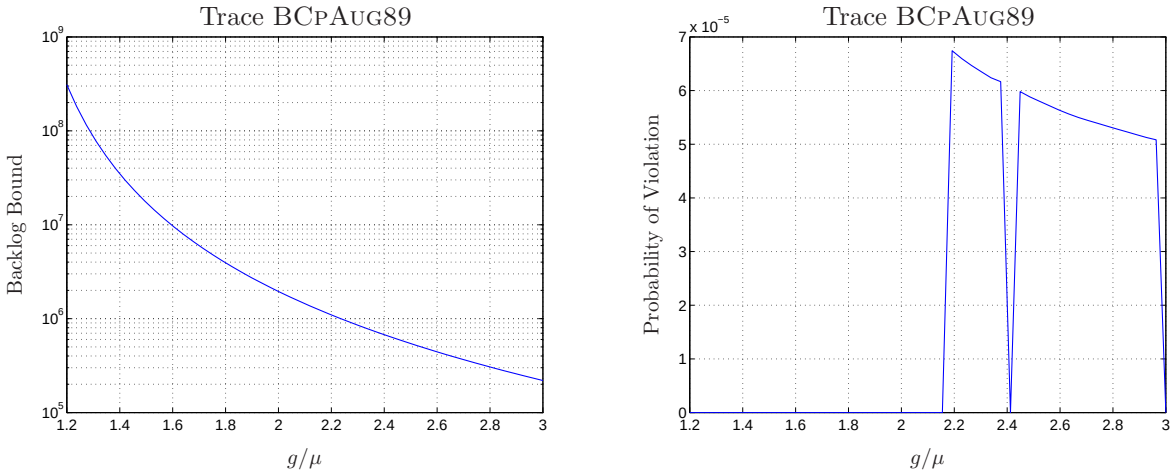

Figure 36: Backlog bound and probability of violation for the trace BCpAug89.

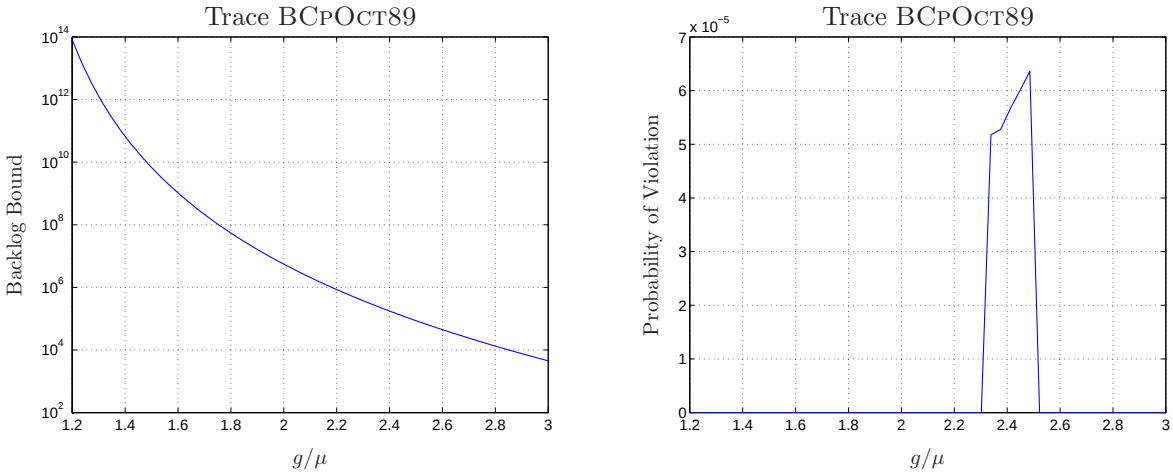

Figure 37: Backlog bound and probability of violation for the trace BCpAug89.

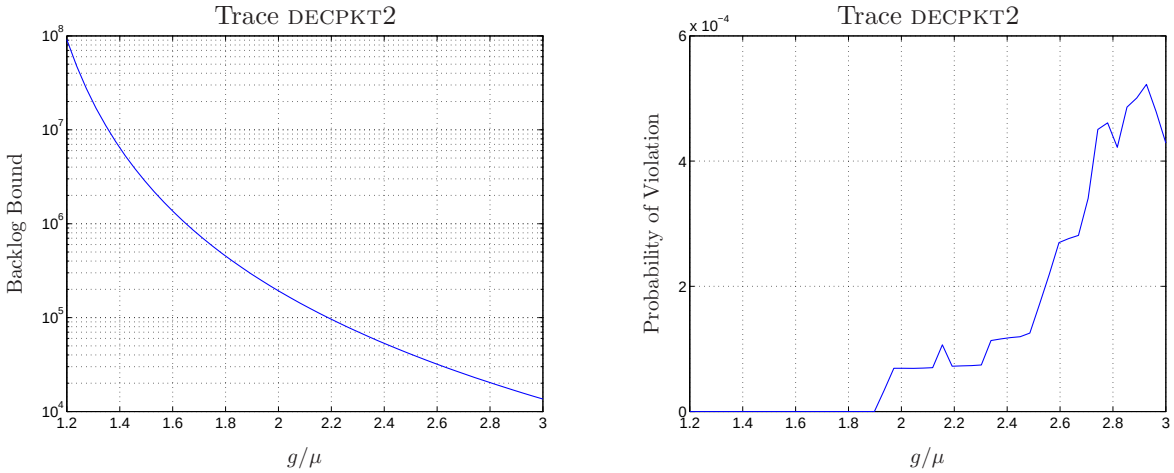

Figure 38: Backlog bound and probability of violation for the trace DECPKT2.

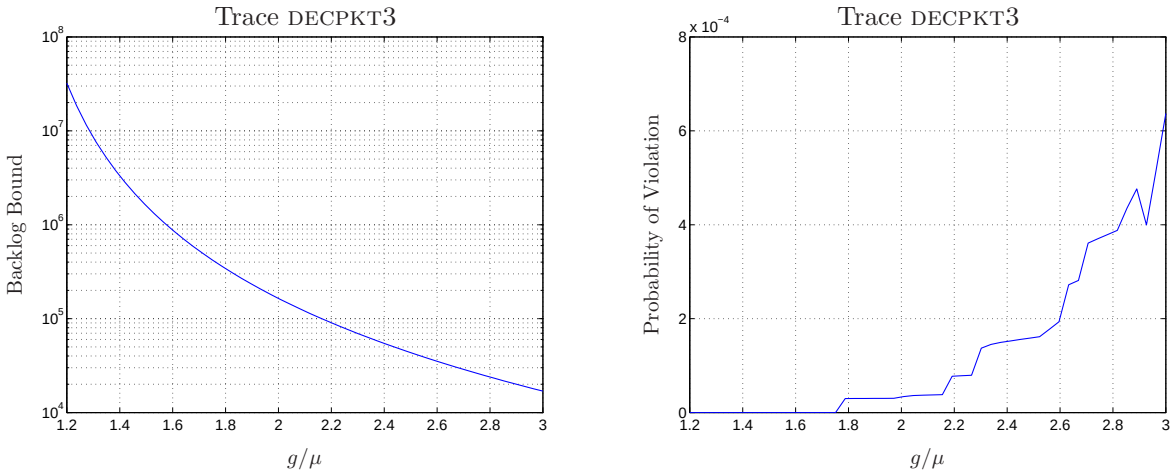

Figure 39: Backlog bound and probability of violation for the trace DECPKT3.

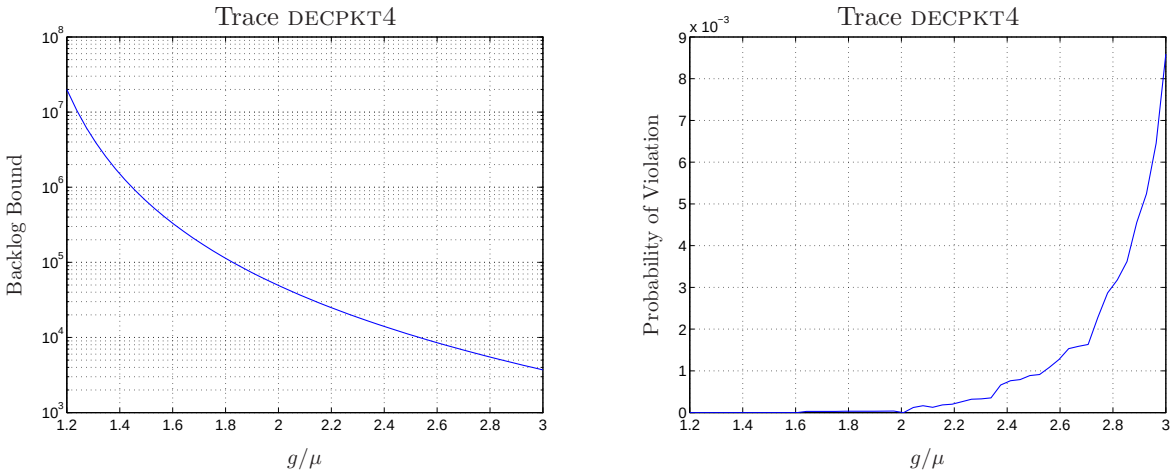

Figure 40: Backlog bound and probability of violation for the trace DECPKT4.

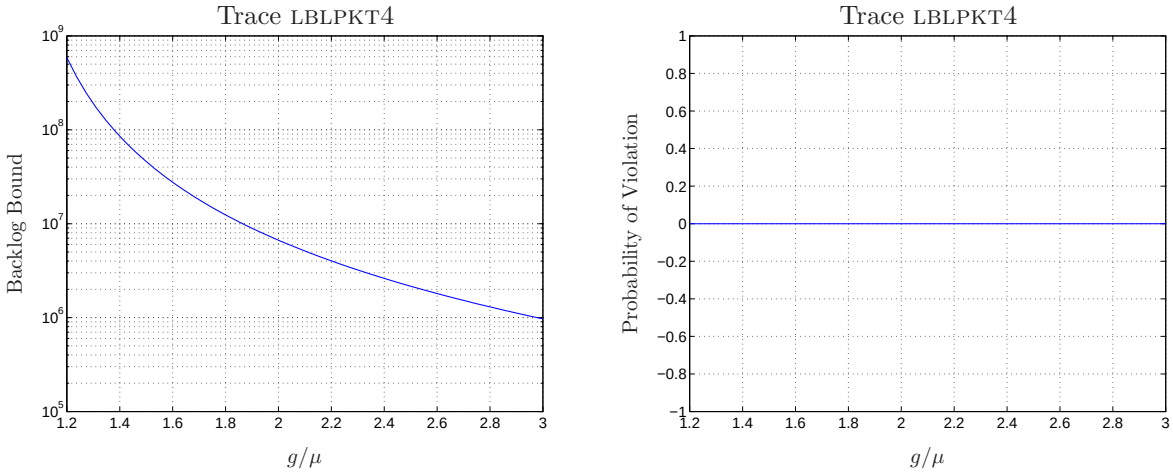

Figure 41: Backlog bound and probability of violation for the trace LBLPKT4.

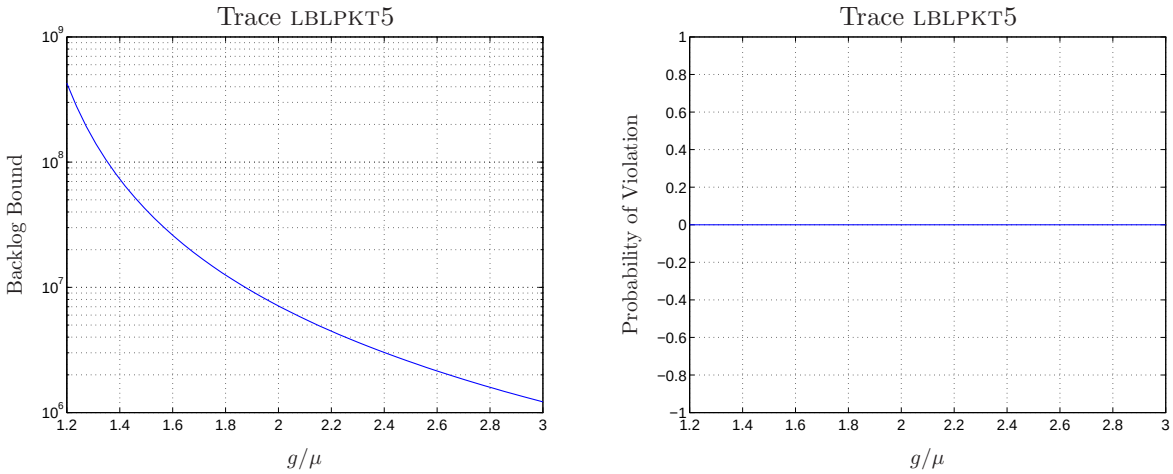

Figure 42: Backlog bound and probability of violation for the trace LBLPKT5.

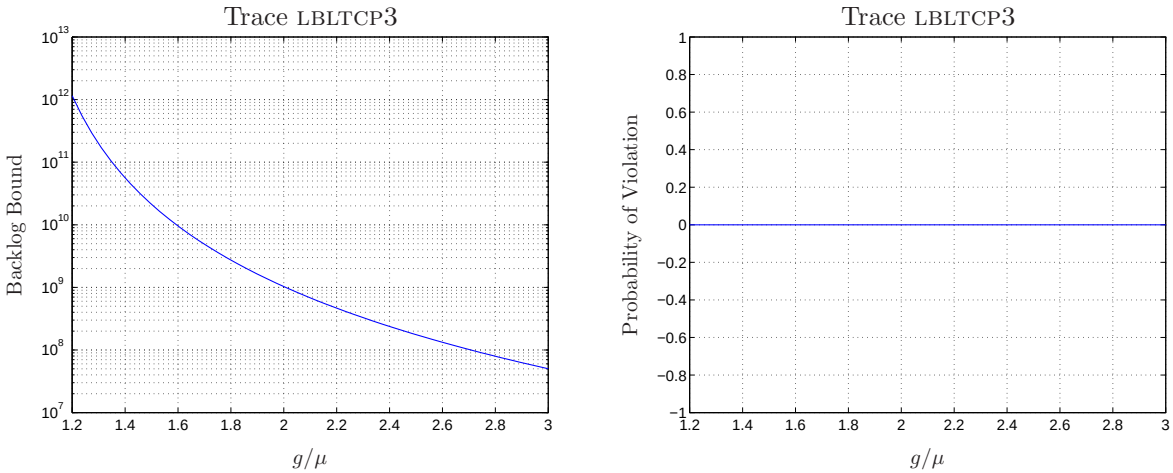

Figure 43: Backlog bound and probability of violation for the trace LBLTCP3.

4 Policing traffic using the Fractal Leaky Bucket algorithm

The results provided in the previous section motivate the development of a policing mechanism that is able to constrain an arbitrary traffic flow to a given deterministic FBAP envelope, so that the performance bounds can be assured. However, obtaining such a mechanism is not trivial. In particular, the nonlinearity of the FBAP envelope implies that

$$\Delta\widehat{A}(m\tau) \leq m\Delta\widehat{A}(\tau). \tag{27}$$

The equality condition is achieved only either for $m = 1$ or for $H = 0$. Such a relation indicates that, in general, a fixed-size window mechanism cannot be applied for policing FBAP traffic, since it is not sufficient to police m consecutive intervals of size τ in order to assure that the whole interval of size $m\tau$ is policed.

An alternative for policing self-similar traffic is the use of the Fractal Leaky Bucket (FLB) algorithm [31]. The FLB algorithm is a window-based policing mechanism, which constrains the traffic to a deterministic envelope given by (7), for which the parameters must be chosen so that an arbitrarily small probability of discarding traffic is obtained.

The operation of the FLB algorithm can be described as follows [31]. Consider a time window with a length of Δ time units. If the arrival process exceeds the declared mean value inside that window (given by $\mu\Delta$), then all packets exceeding the envelope in that period are discarded, and the time window is enlarged by Δ time units. This new window goes into effect at the time when the arrival process violates the declared mean rate. This process is then repeated as long as the incoming traffic violates the declared mean rate inside the window. However, since certain violating packets have already been discarded in the previous window, the number of packets to be discarded is now equal to the number of violating packets in the current window minus the number of packets that have already been discarded in the previous window. When the mean number of arrivals drops below the declared value, the window is shrunk to Δ time units, and the policing process is reinitiated.

Since the FLB algorithm constrains traffic to a cumulative envelope process, it is unable to bound the burstiness of the traffic. Consequently, such a policing mechanism does not adequately support the provision of performance bounds. For example, consider a queueing system, in which the input traffic and the backlog are denoted by $A(t)$ and $Q(t)$, respectively. The server has a constant rate, denoted by g, which is greater than the input traffic mean rate for assuring the stability of the system.

Suppose that the input traffic is policed by the FLB algorithm before entering the system. Nominal input traffic parameters are μ, ψ and H, and FLB parameters are chosen accordingly. The maximum backlog should be deterministically bounded by

$$Q^* = \widehat{A}(t^{max}) - gt^{max}, \tag{28}$$

where t^{max} is given by

$$t^{max} = \arg\max_{t \geq 0} \widehat{A}(t) - gt. \tag{29}$$

Suppose now that the input traffic is given by

$$A(t) = \begin{cases} ct, & 0 \leq t < t^* \\ \mu t + \psi t^H, & t \geq t^*. \end{cases} \tag{30}$$

The discontinuity of (30) implies in the assumption that traffic can be generated at an infinite rate. Such an assumption is only considered for the sake of simplicity, and the analysis to be developed below is valid as long as the traffic can be generated at a rate that is much greater than the service rate g, a condition that is easily verified in practice.

In order to assure that $\widehat{A}(t) \geq A(t)$, $\forall t \geq 0$, let the FLB envelope parameters be given by μ, ψ and H. The input traffic is therefore below the FLB envelope, and no violation occurs.

As it is illustrated in Fig. 44 , the backlog bound (28) is actually satisfied when $c > g$. However, suppose that $\mu < c < g$ and $t^* = t^{max}$. This condition is illustrated in Fig. 45. The total amount of traffic offered to the system up to time t^{max} is given by ct^{max}. Since $c < g$, the queue is empty just before time t^{max}. Notice that the time window of the FLB algorithm at time $t^* = t^{max}$ ranges the interval $[0; t^*)$, since the traffic source operates above the mean rate and below the FLB envelope from time zero. From the definition of the FLB envelope process, the total amount of traffic the policing can accept at time t^{max} is given by $\widehat{A}(t^{max}) - ct^{max}$. This is exactly the amount of traffic the source provides at $t = t^{max}$, if the input traffic is given by (30). The backlog just after time t^{max} is then given by $Q(t^{max}) = \widehat{A}(t^{max}) - ct^{max}$, which is greater than Q^*. Therefore, the backlog bound can be violated, eventhough the FLB envelope is not.

From this simple example, it is possible to conclude that the FLB algorithm is not able to support the provision of a robust backlog bound, since the FLB algorithm is based on a cumulative representation of the traffic, and is thus unable to bound the burstiness of the traffic.

5 Policing self-similar traffic by using the Leaky Bucket algorithm

In this section, the policing of self-similar traffic using the traditional Leaky Bucket algorithm is proposed. The Leaky Bucket algorithm constrains the incremental traffic to the LBAP envelope, which is given by

$$\Delta\widehat{A}_{\text{LBAP}}(\tau) = r\tau + s, \tag{31}$$

where r is the *leaky rate* and s is the *bucket size*. The Leaky Bucket algorithm can be represented by a queueing system, in which the bucket is represented by a queue with size s, continuously served at the leaky rate r. When some amount of traffic arrives at the policing mechanism, an equal amount of tokens is pushed into the queue. If the queue does not overflow, the traffic is considered to be

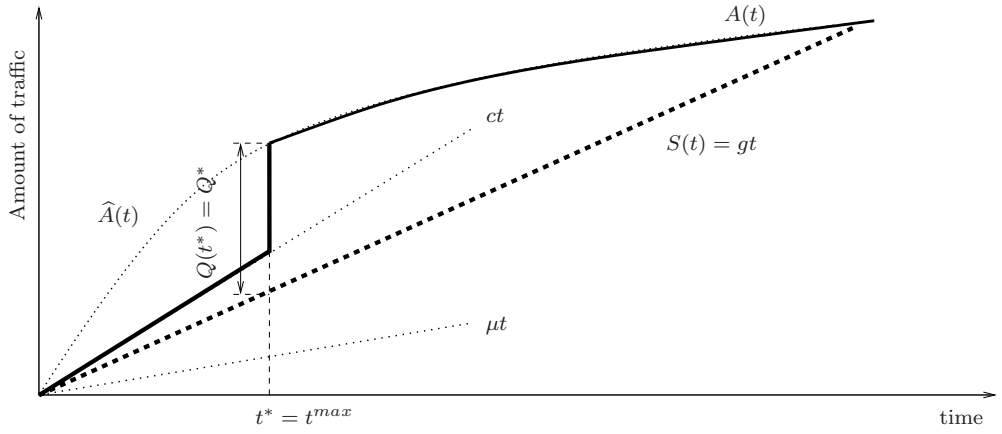

Figure 44: A condition for which the FLB algorithm is able to support the provision of backlog bounds.

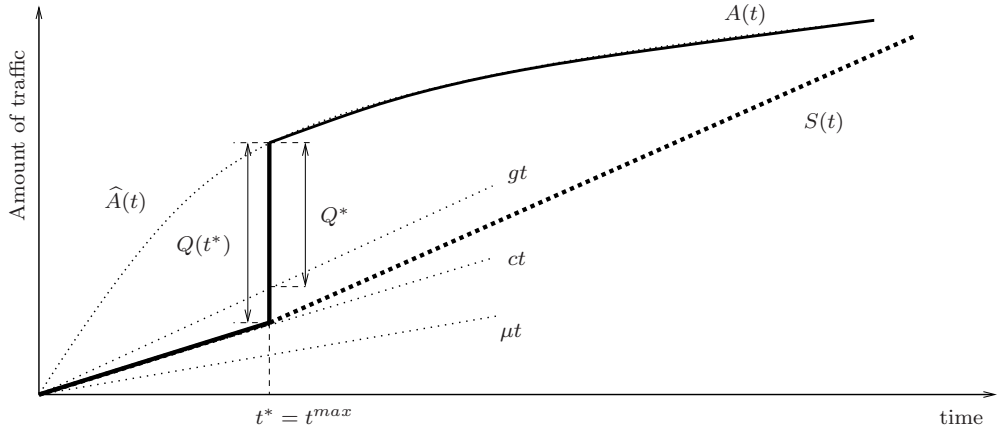

Figure 45: A condition for which the FLB algorithm is unable to support the provision of performance bounds.

well-behaved and is immediately transfered to the output of the policing mechanism. Otherwise, the amount of traffic corresponding to the overflow is discarded.

Let $\Delta\widehat{A}_s(\tau) = \mu_s\tau + k_s\gamma_s\tau^{H_s}$ be the specified FBAP envelope. From the analysis developed in Section 3, a target probability of traffic loss can be obtained if the leaky rate is equal to the equivalent bandwidth of the traffic, and the bucket size is equal to the corresponding backlog bound. From (22),

$$s = (r-\mu_s)^{\frac{H_s}{H_s-1}}\,(k_s\gamma_s)^{\frac{1}{1-H_s}}\,H_s^{\frac{H_s}{1-H_s}}\,(1-H_s)\,. \tag{32}$$

In [31], a similar relation was obtained by considering a different approach. An example of the relation between the leaky rate and the bucket size is shown in Fig. 47. Nominal traffic parameters are $\mu_s = 0.800$, $k_s\gamma_s = 0.595$ and $H_s = 0.800$. Notice that the bucket size rapidly decreases as the leaky rate increases. Moreover, feasible values for the bucket size are obtained even for relatively small values of leaky rate.

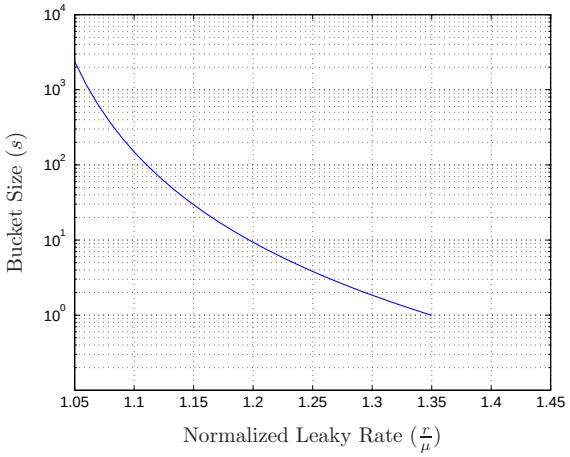

Figure 46: Bucket size as a function of the Leaky Rate for $\mu_s = 0.800$, $k_s\gamma_s = 0.595$, $H_s = 0.800$.

Now, let $\Delta\mathbf{A}_i(\tau) = \mu_i\tau + \gamma_i\mathbf{Z}_i(\tau)$ be the actual input traffic, and $\mathcal{P}_i$ be the probability of traffic loss. When $\mu_i \geq \mu_s$, such a probability is clearly equal to one, since the bucket is likely to overflow. On the other hand, when $\mu_i < \mu_s$, the probability $\mathcal{P}_i$ is bounded by

$$\begin{aligned}
\mathcal{P}_i &= \mathbb{P}\left\{\sup_{\tau\geq 0}\mathbf{Q}_i(t+\tau) > s\right\} \\
&\geq \sup_{\tau\geq 0}\mathbb{P}\left\{\mathbf{Z}_i(1) > \frac{s+(r-\mu_i)\tau}{\gamma_i\tau^{H_i}}\right\} \qquad (33)\\
&= \overline{F}_{\mathbf{Z}_i}\left[\frac{(r-\mu_i)^{H_i}}{\gamma_i H_i^{H_i}\,(1-H_i)^{1-H_i}}s^{1-H_i}\right]. \qquad (34)
\end{aligned}$$

When the incoming traffic matches the specifications, the probability of loss is given by

$$\mathcal{P}_i \geq \overline{F}_{\mathbf{Z}_i}(k_s)$$

When (24) asymptotically converges to the equality, the previous relation asymptotically converges to

$$\mathcal{P}_i \simeq \overline{F}_{\mathbf{Z}_i}(k_s). \tag{35}$$

In order to guarantee the desired probability of discarding well-behaved traffic, the Leaky Bucket parameters must be chosen so that (32) and (35) are satisfied. Notice that, whenever such relations are satisfied, the inequality

$$\Delta\widehat{A}_{\mathrm{LBAP}}(\tau) \geq \Delta\widehat{A}_s(\tau), \tag{36}$$

holds for $\forall\tau \geq 0$. Thus, the FBAP envelope can no longer be used to represent the traffic that leaves the policing mechanism. For queueing analysis purposes, the LBAP envelope must be considered instead. For a queueing system with a constant service rate, however, the backlog (or delay) bound obtained for the Leaky Bucket constrained flow can be equal to the one which would be obtained if the traffic were constrained to a deterministic FBAP envelope instead. Such an equality is achieved when the leaky rate matches the service rate of the system. Therefore, no loss of performance results from the use of the Leaky Bucket algorithm in this case. Moreover, given (36), the less traffic is discarded if the Leaky Bucket is used, which means that it is less penalized in this case.

The use of the Leaky Bucket algorithm for self-similar traffic policing is now illustrated. The input traffic is a synthetic fractional Gaussian noise trace consisting of 10^6 samples, which was generated using the method proposed in [36]. Each sample correspond to the amount of traffic that arrives at the policing mechanism in an interval of a fixed size. Inside each interval, the traffic is assumed to be uniformly distributed, a condition under which the traffic is smoother and violations are more difficult to be detected by the Leaky Bucket algorithm. For the sake of simplicity, simulation experiments were conducted assuming the traffic to be a fluid-type traffic. Moreover, the traffic is assumed to be uniformly distributed during the interval which corresponds to each sample of the traces, a condition under which the traffic is smoother and violations are more difficult to be detected by the Leaky Bucket algorithm. Input traffic parameters are $\mu_i = 1.470$, $\gamma_i = 0.399$ and $H_i = 0.800$.

For the nominal FBAP model, the parameter k_s is equal to 3.719, for which a probability of violation equal to 10^{-4} is obtained. Since the input traffic parameters are fixed, the violations of the mean rate, the scale parameter and the self-similarity parameter are obtained by varying the nominal parameters μ_s, γ_s and H_s. For the Leaky Bucket algorithm, the leaky rate is equal to $1.2\mu_s$, and the bucket size is given by (32).

The probability of loss corresponding to the violation of the mean rate, the scale parameter and the self-similarity parameter are shown in Fig. 47. Such a probability is given by the number of busy cycles for which any loss occurs, divided by the total number of busy cycles.

Notice that the probability of loss increases as the actual traffic parameters grow in relation to the nominal ones, which confirms the adequate operation of the policing mechanism.

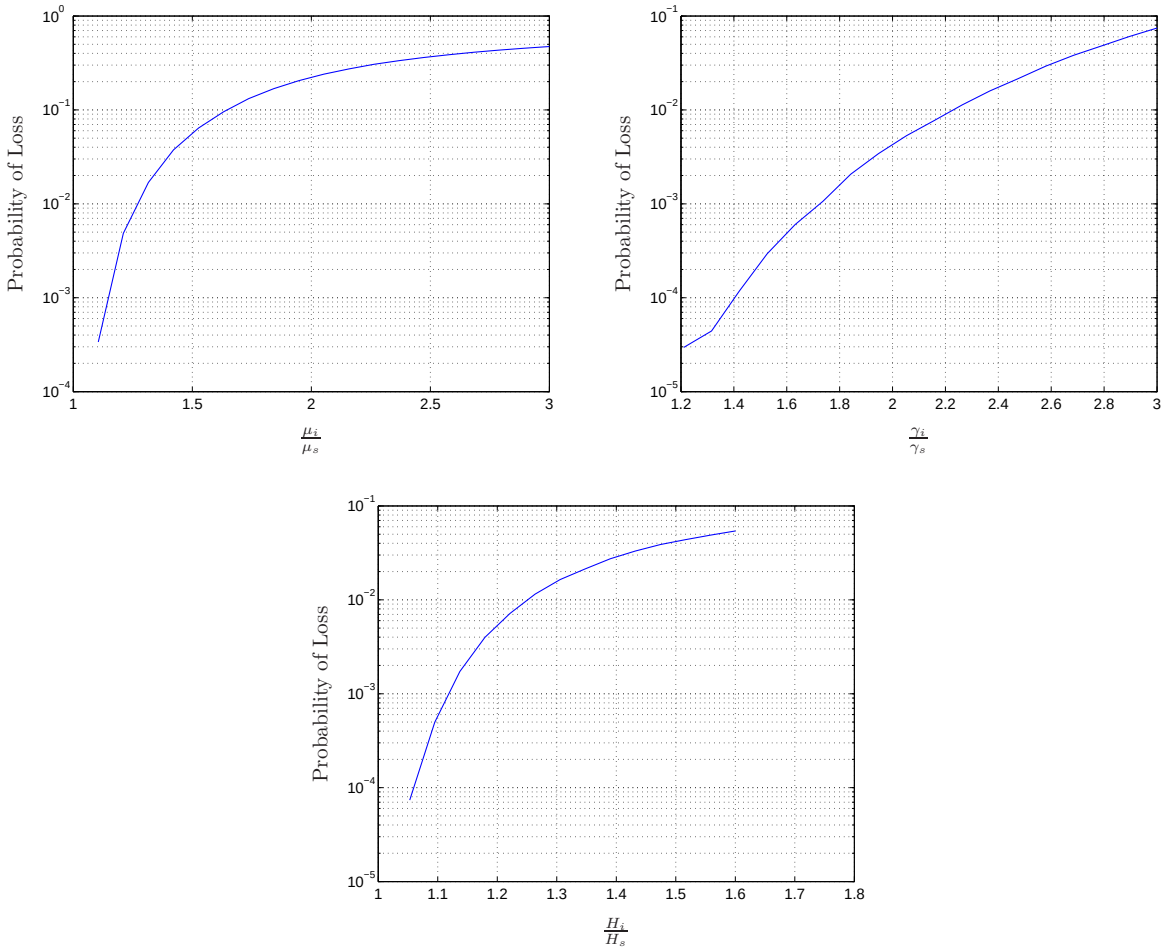

Figure 47: Probability of loss as a function of the violation of the mean rate, the scale parameter and the self-similarity parameter for the proposed example.

6 A comparison between the FLB algorithm and the Leaky Bucket algorithm for policing self-similar traffic

In this section, the use of the Leaky Bucket algorithm for policing self-similar traffic is compared to the use of the FLB algorithm. In particular, it is shown that the former algorithm is able to support the provision of backlog bounds in a queueing system, which is not possible when the latter is used.

For the simulations, the real traffic traces which were described in Section 2 were used. For the sake of simplicity, simulation experiments were conducted assuming the traffic to be a fluid-

type traffic. Moreover, the traffic is assumed to be uniformly distributed during the interval which corresponds to each sample of the traces.

For each trace, the relationship between the backlog bound and the service rate, given by (22), is shown in Fig. 48–77. The maximum backlog obtained in the queueing system when the input traffic is policed by the Leaky Bucket algorithm and the FLB algorithm are also shown. For the Leaky Bucket algorithm, the leaky rate r was chosen to match the service rate, and the bucket size s is given by (32). For the FLB algorithm, the parameters were chosen to match the nominal parameters of the FBAP traffic.

From Fig. 48–77, it is possible to verify that the backlog bound is never violated when the input traffic is policed by the Leaky Bucket algorithm. However, when the FLB algorithm is used, the backlog bound can be violated depending on the traffic profile and on the service rate, which agrees with the analysis developed in the previous sections.

For each trace, the discard ratio corresponding to the FLB algorithm and the Leaky Bucket algorithm is also shown in Fig. 48–77. Notice that the use of the FLB algorithm results in a discard ratio which is constant in relation to the service rate. Such a result was expected, since the FLB algorithm does not take the service rate into consideration. On the other hand, the use of the Leaky Bucket algorithm results in a discard ratio that depends on the service rate, since the leaky rate is chosen to match that rate. Notice, however, that the Leaky Bucket algorithm always discards less traffic than the FLB algorithm. Thus, the former algorithm is able to support the provision of backlog bounds by penalizing less the traffic flow.

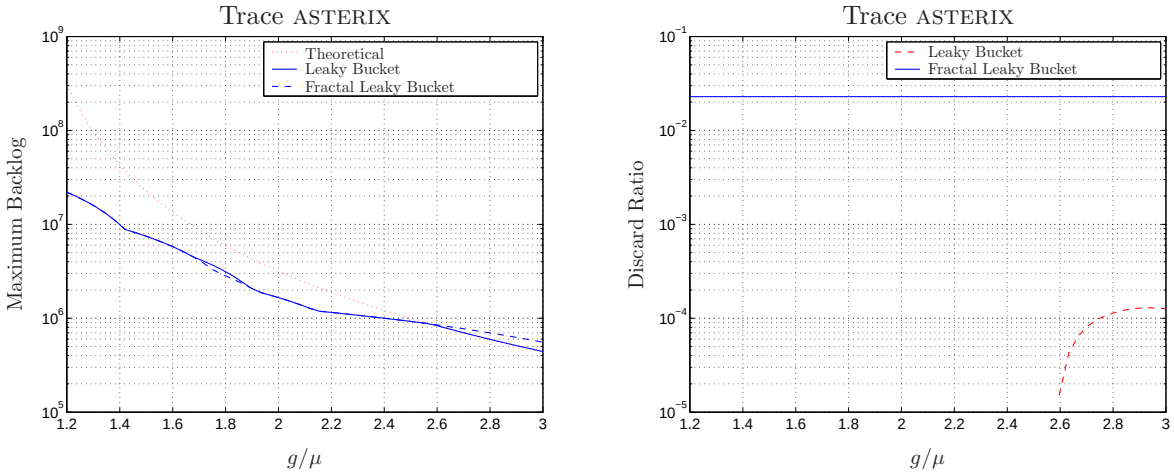

Figure 48: Comparison between the Leaky Bucket and the FLB algorithms for the trace ASTERIX.

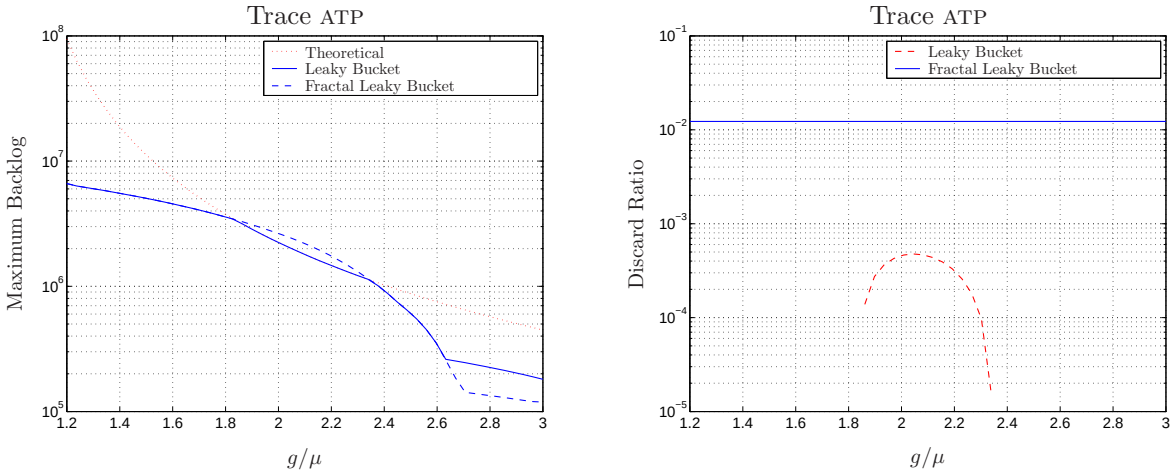

Figure 49: Comparison between the Leaky Bucket and the FLB algorithms for the trace ATP.

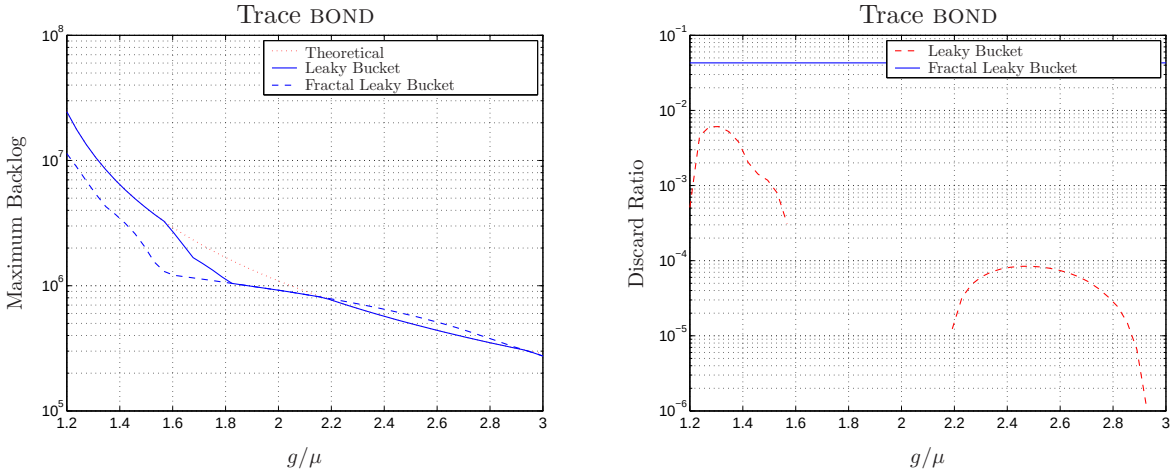

Figure 50: Comparison between the Leaky Bucket and the FLB algorithms for the trace BOND.

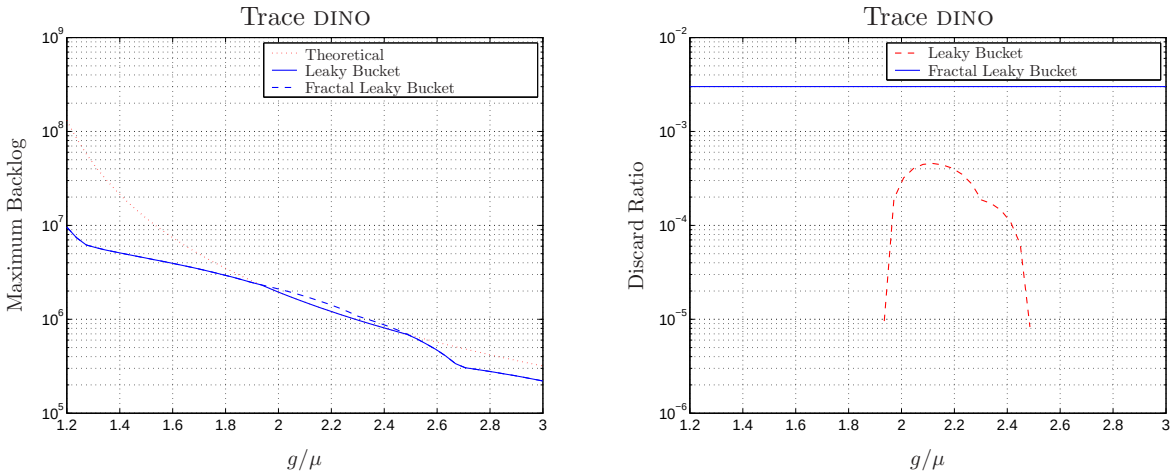

Figure 51: Comparison between the Leaky Bucket and the FLB algorithms for the trace DINO.

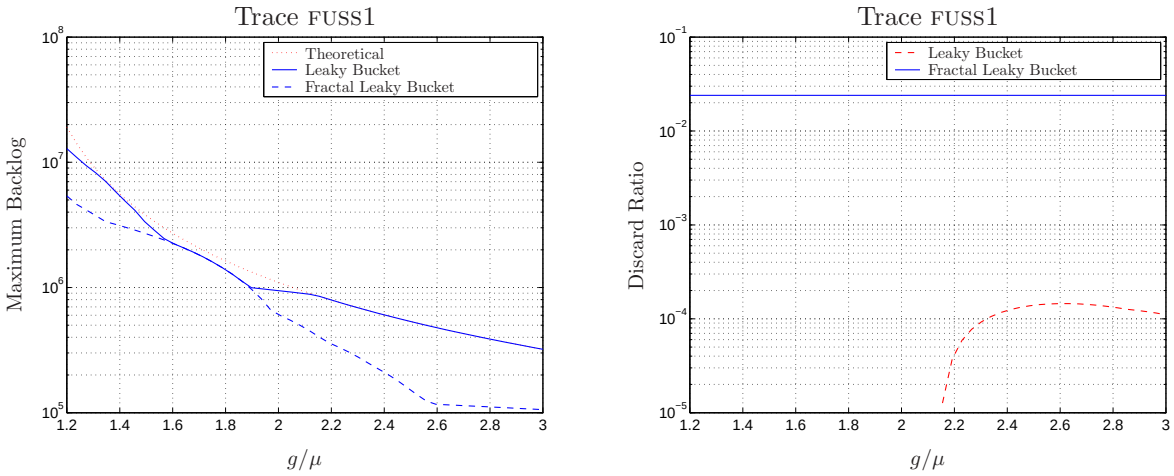

Figure 52: Comparison between the Leaky Bucket and the FLB algorithms for the trace FUSS1.

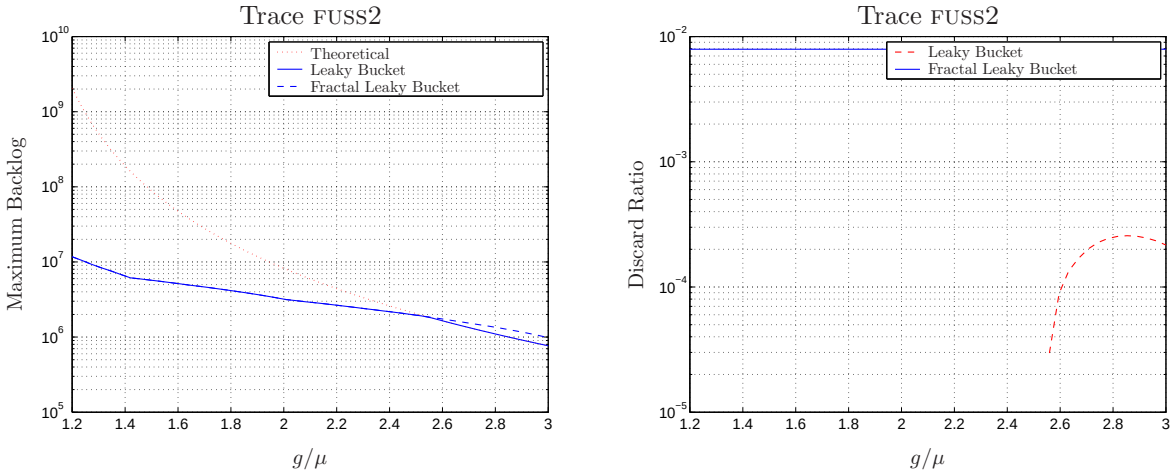

Figure 53: Comparison between the Leaky Bucket and the FLB algorithms for the trace FUSS2.

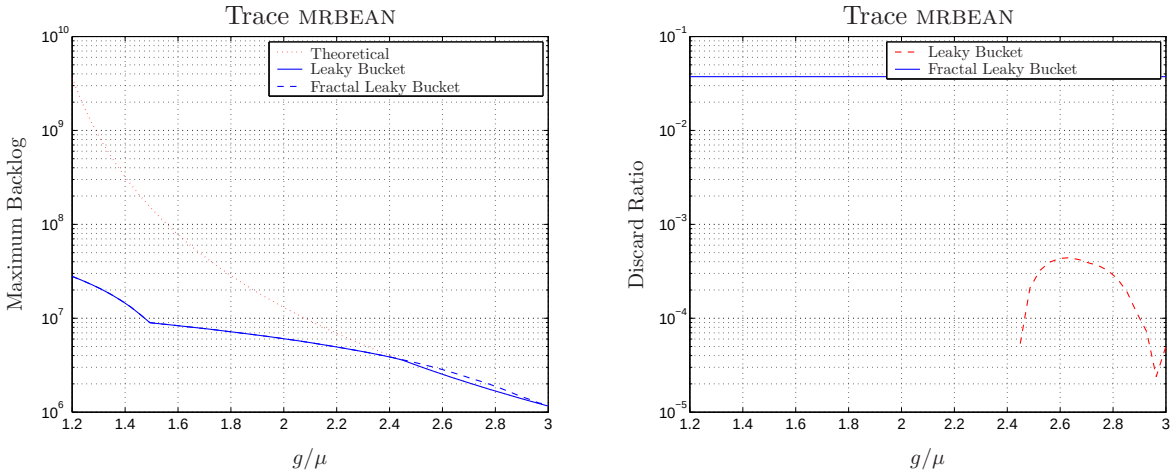

Figure 54: Comparison between the Leaky Bucket and the FLB algorithms for the trace MRBEAN.

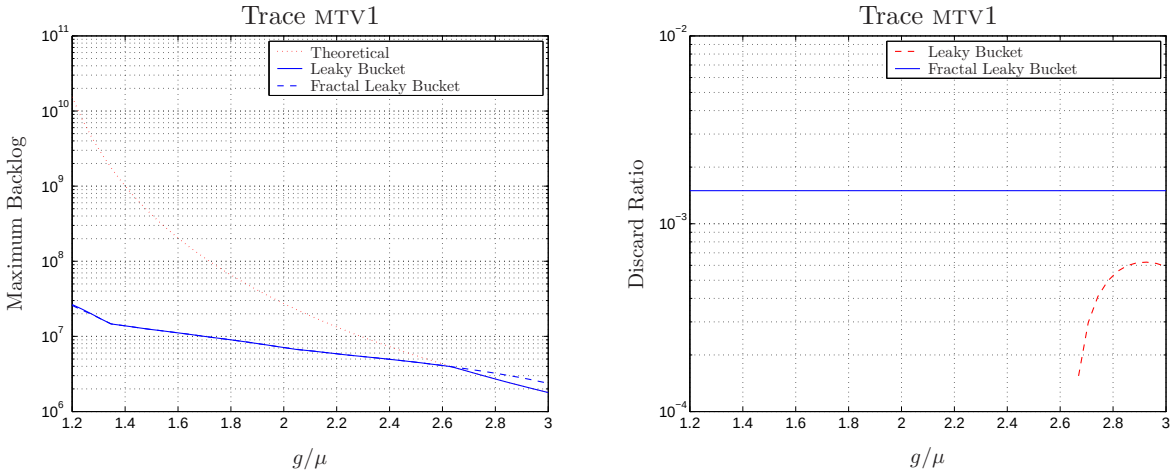

Figure 55: Comparison between the Leaky Bucket and the FLB algorithms for the trace MTV1.

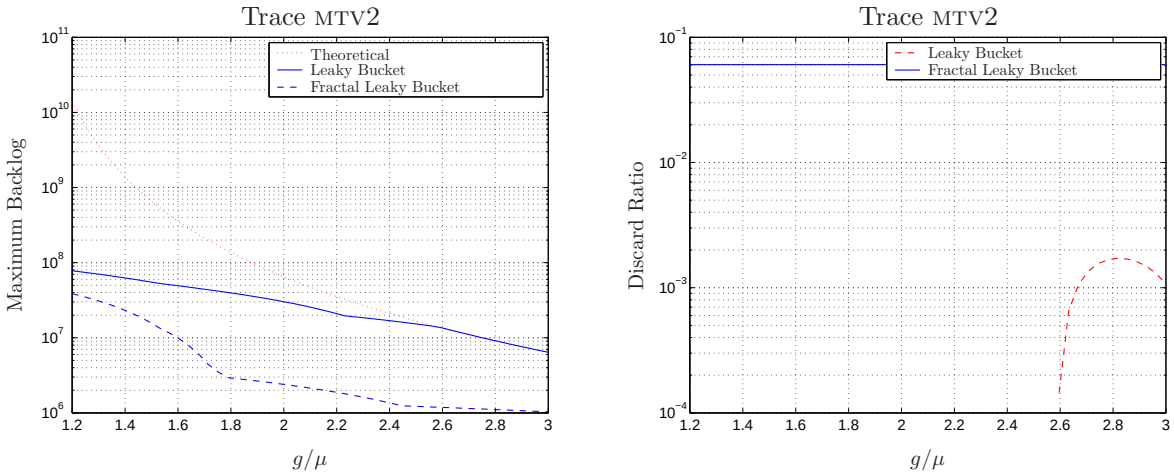

Figure 56: Comparison between the Leaky Bucket and the FLB algorithms for the trace MTV2.

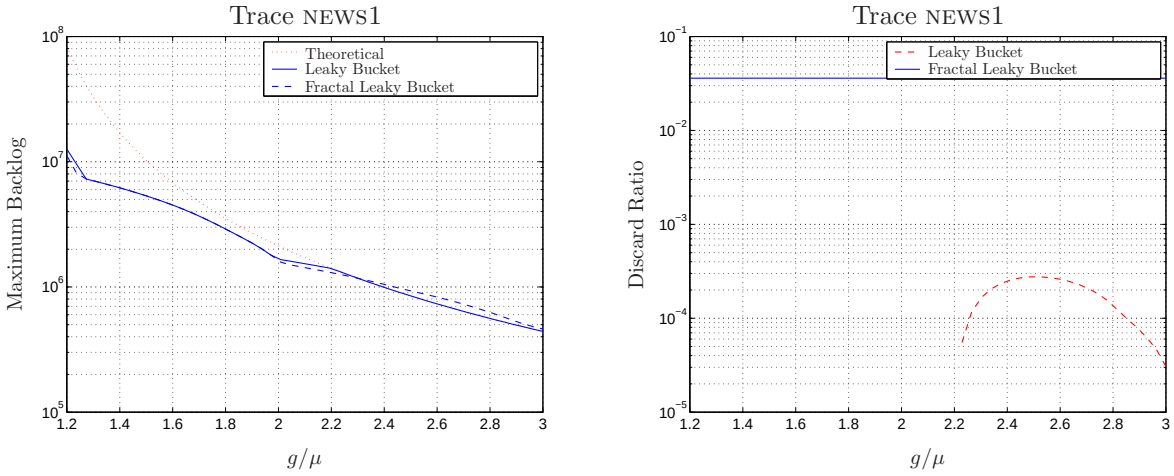

Figure 57: Comparison between the Leaky Bucket and the FLB algorithms for the trace NEWS1.

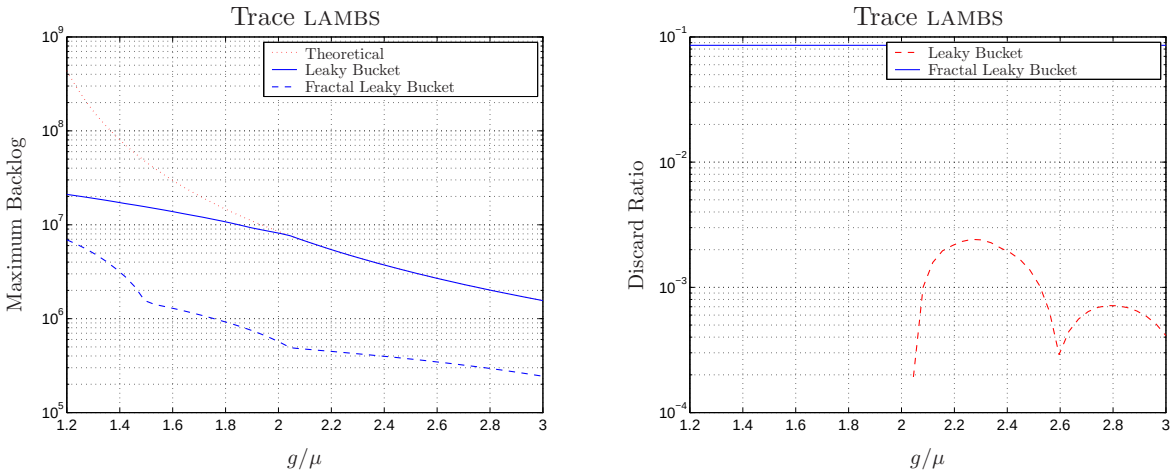

Figure 58: Comparison between the Leaky Bucket and the FLB algorithms for the trace LAMBS.

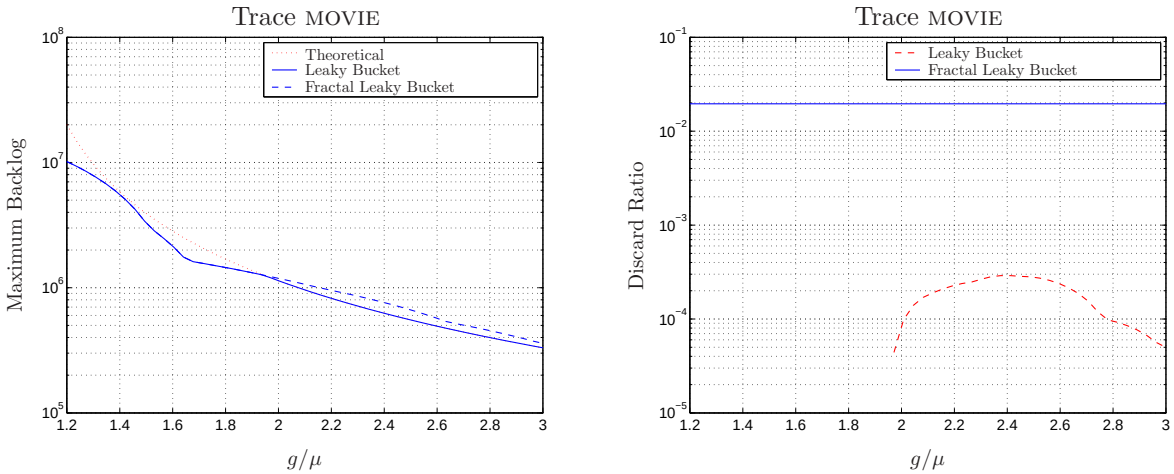

Figure 59: Comparison between the Leaky Bucket and the FLB algorithms for the trace MOVIE.

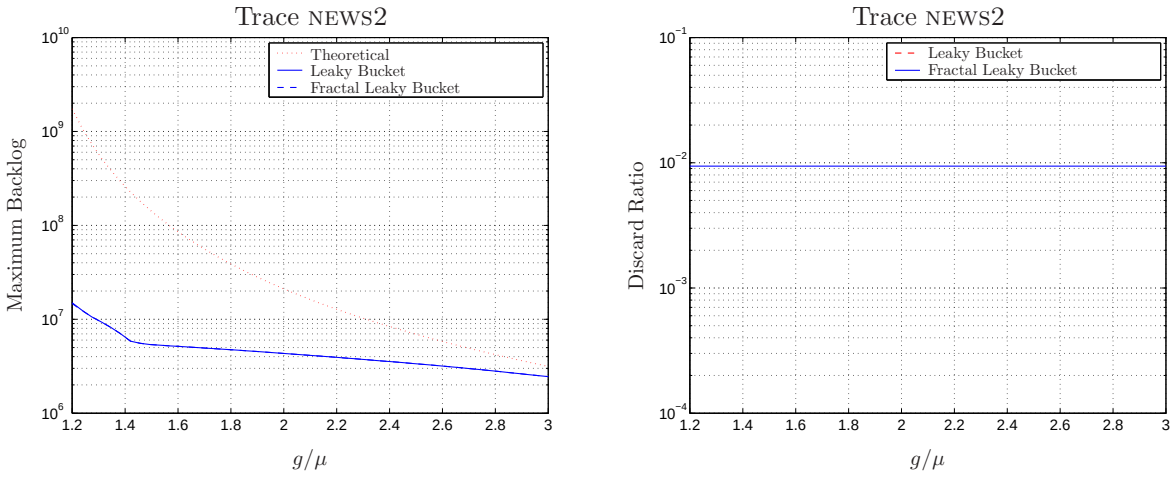

Figure 60: Comparison between the Leaky Bucket and the FLB algorithms for the trace NEWS2.

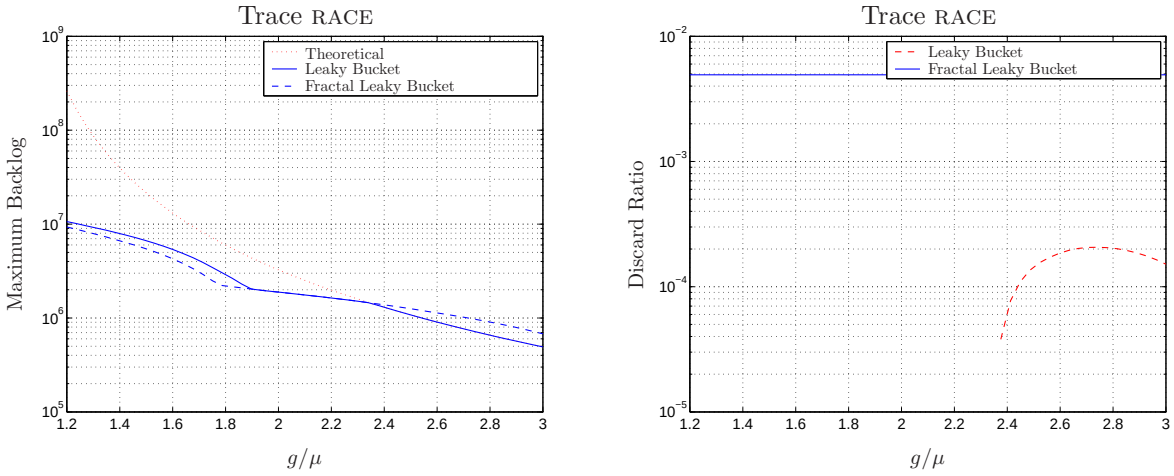

Figure 61: Comparison between the Leaky Bucket and the FLB algorithms for the trace RACE.

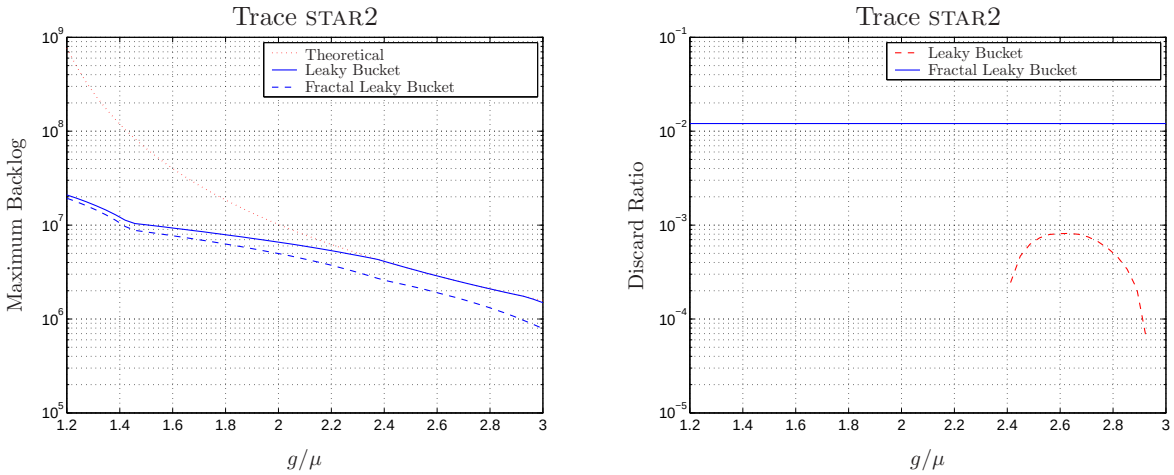

Figure 62: Comparison between the Leaky Bucket and the FLB algorithms for the trace STAR2.

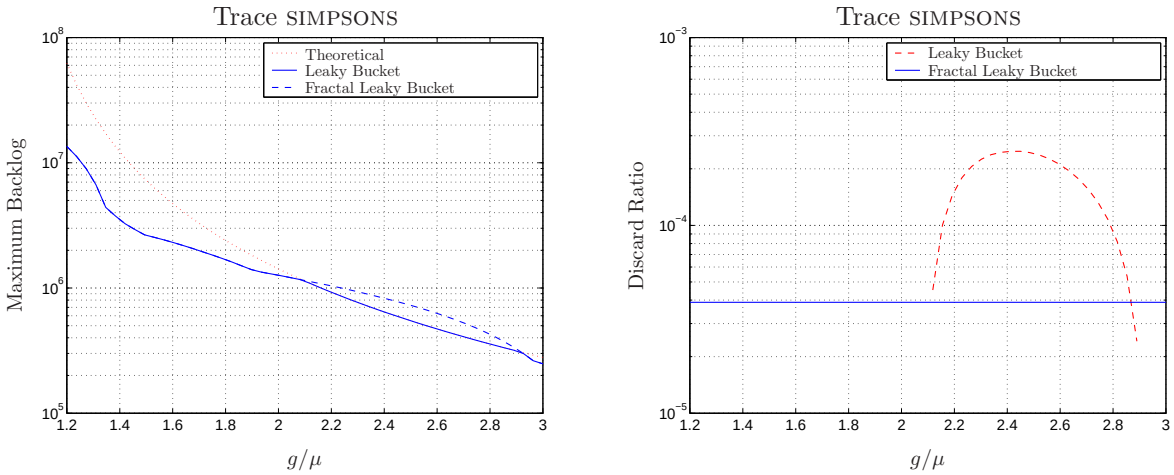

Figure 63: Comparison between the Leaky Bucket and the FLB algorithms for the trace SIMPSONS.

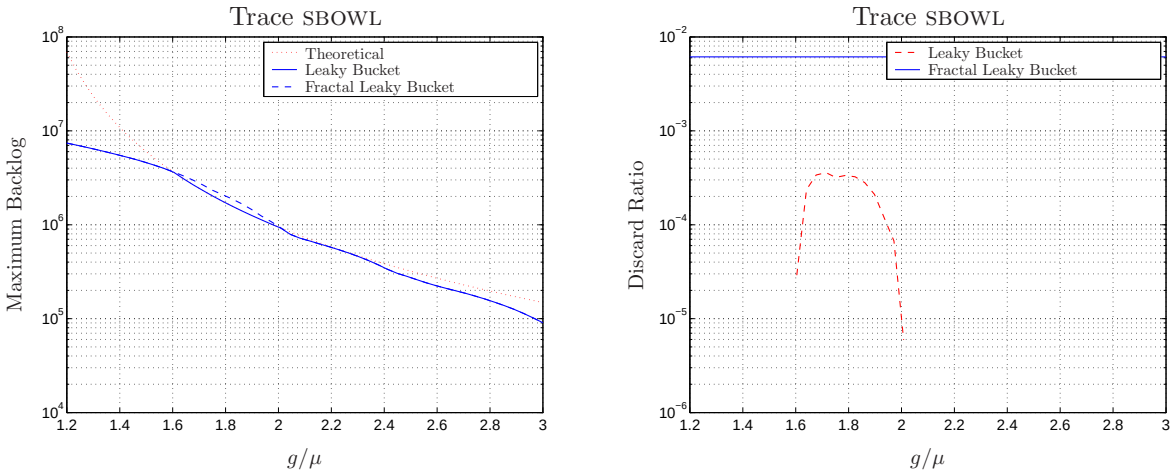

Figure 64: Comparison between the Leaky Bucket and the FLB algorithms for the trace SBOWL.

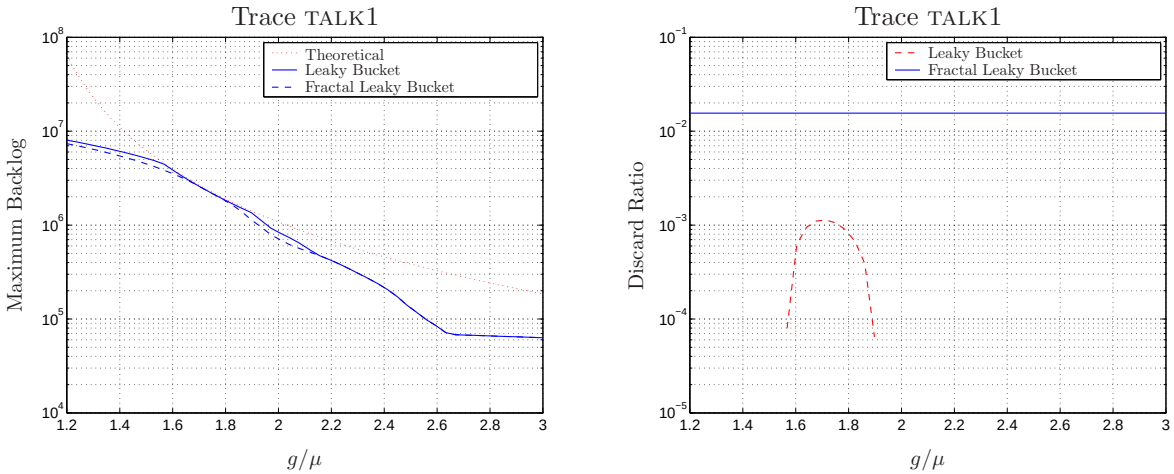

Figure 65: Comparison between the Leaky Bucket and the FLB algorithms for the trace TALK1.

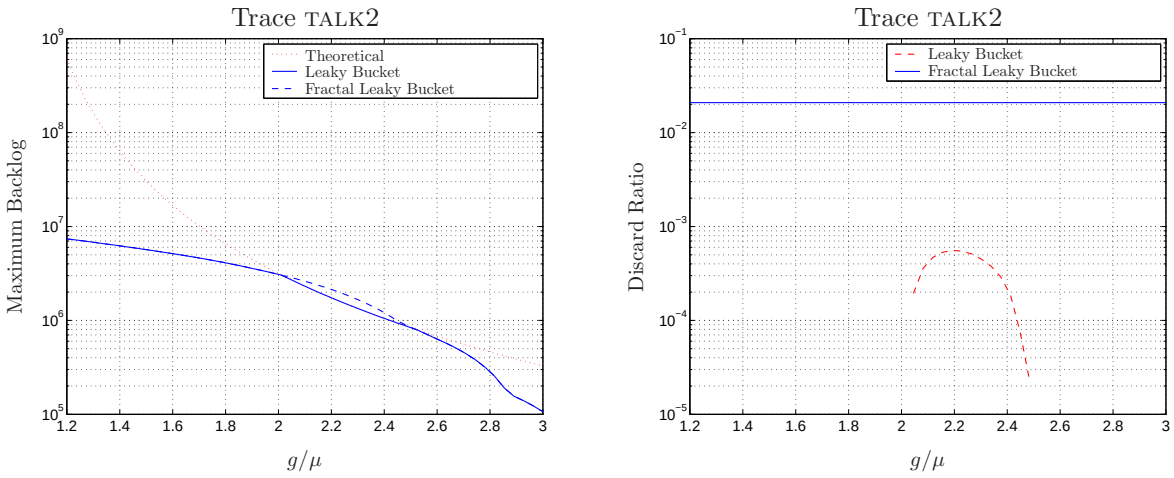

Figure 66: Comparison between the Leaky Bucket and the FLB algorithms for the trace TALK2.

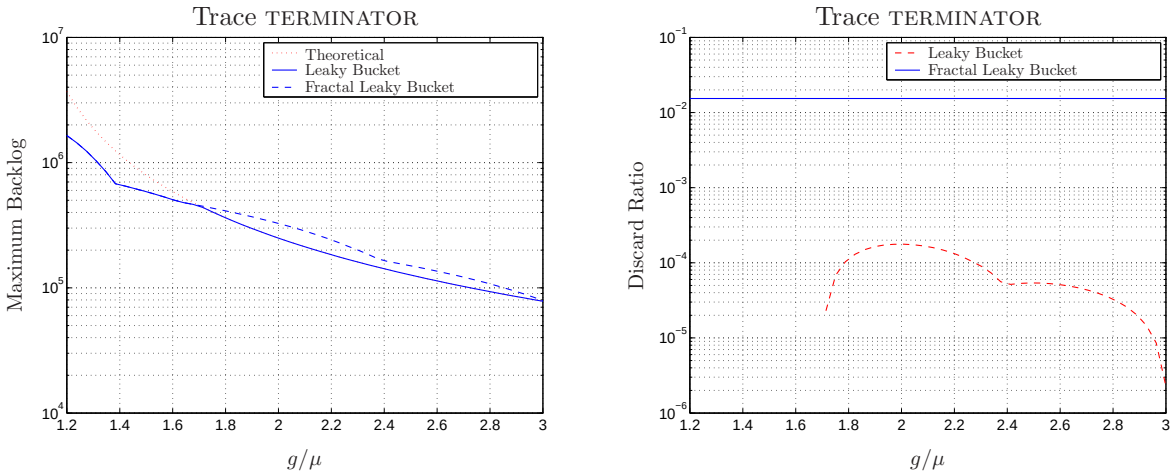

Figure 67: Comparison between the Leaky Bucket and the FLB for the trace TERMINATOR.

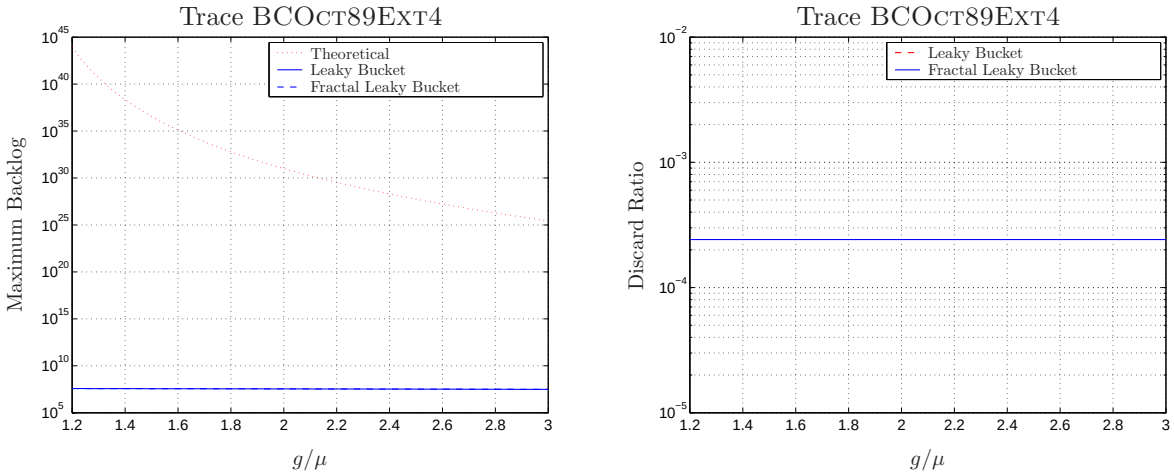

Figure 68: Comparison between the Leaky Bucket and the FLB for the trace BCOct89Ext4.

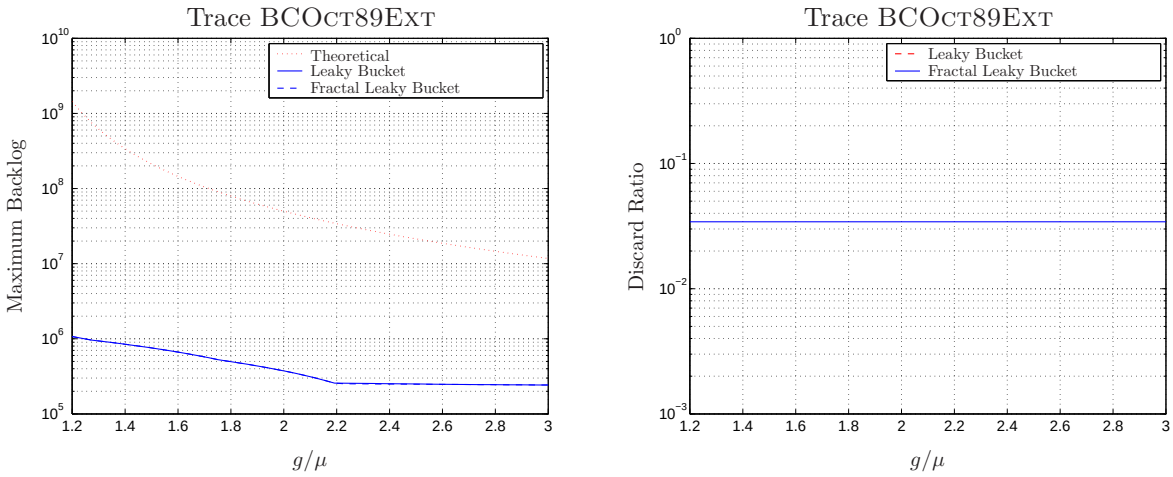

Figure 69: Comparison between the Leaky Bucket and the FLB for the trace BCOct89Ext.

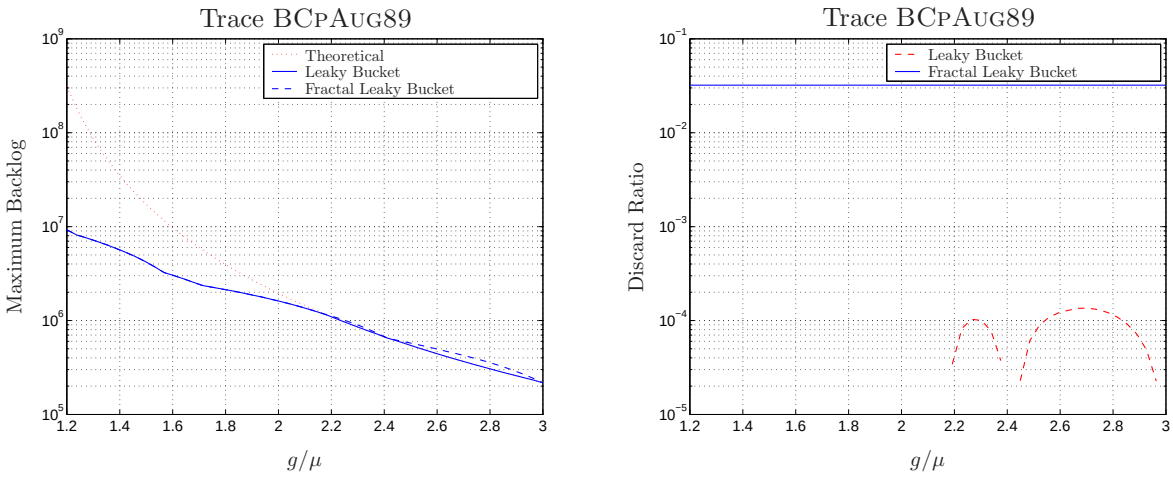

Figure 70: Comparison between the Leaky Bucket and the FLB for the trace BCpAug89.

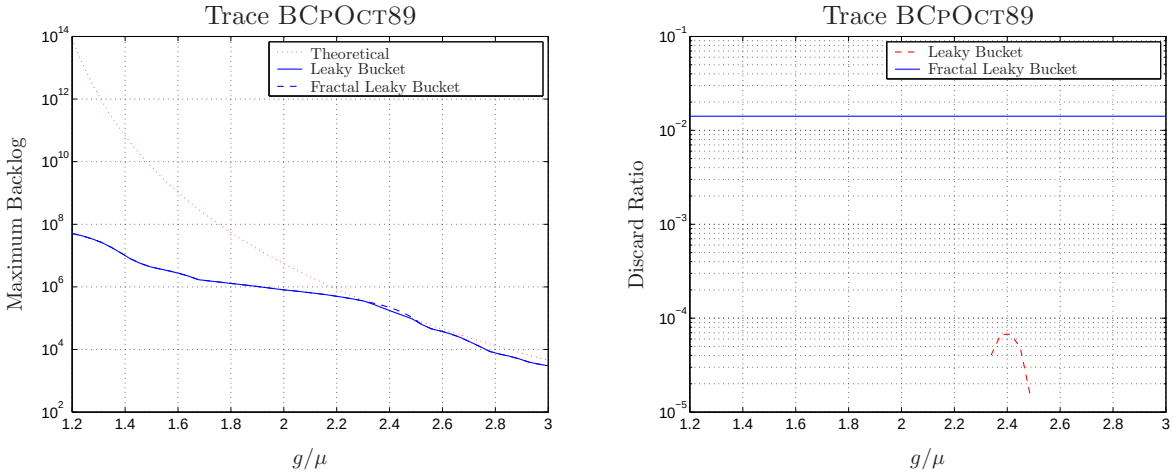

Figure 71: Comparison between the Leaky Bucket and the FLB algorithms for the trace BCpOct89.

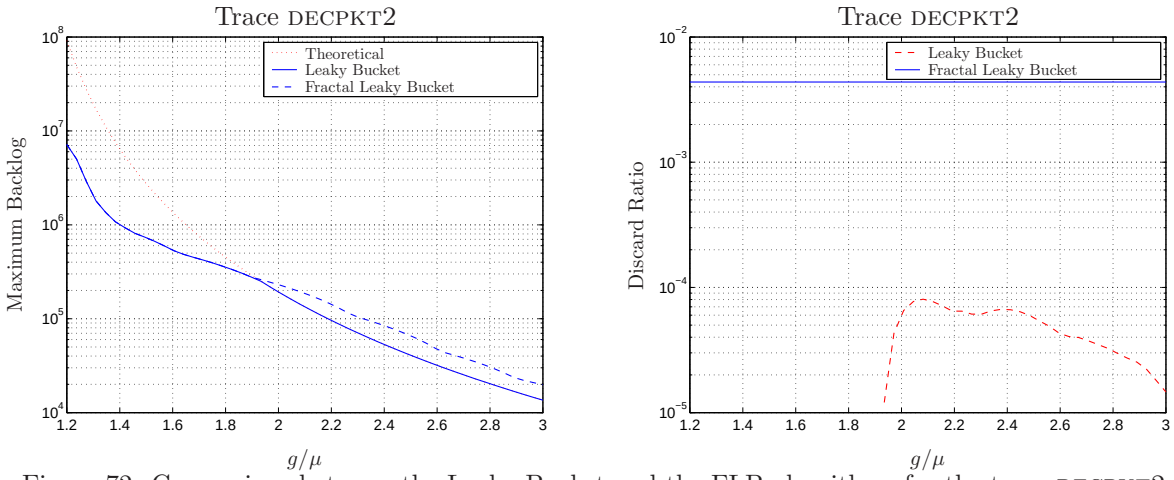

Figure 72: Comparison between the Leaky Bucket and the FLB algorithms for the trace DECPKT2.

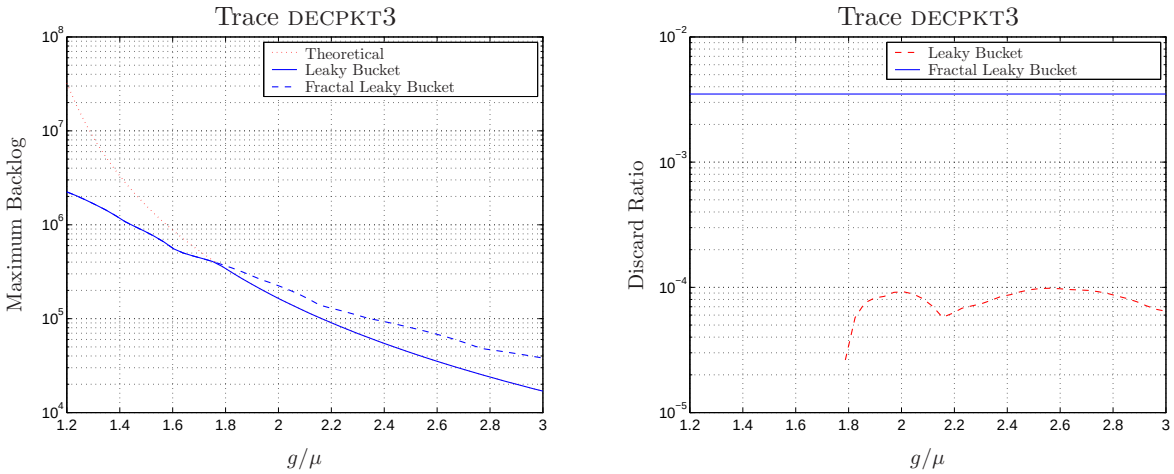

Figure 73: Comparison between the Leaky Bucket and the FLB algorithms for the trace DECPKT3.

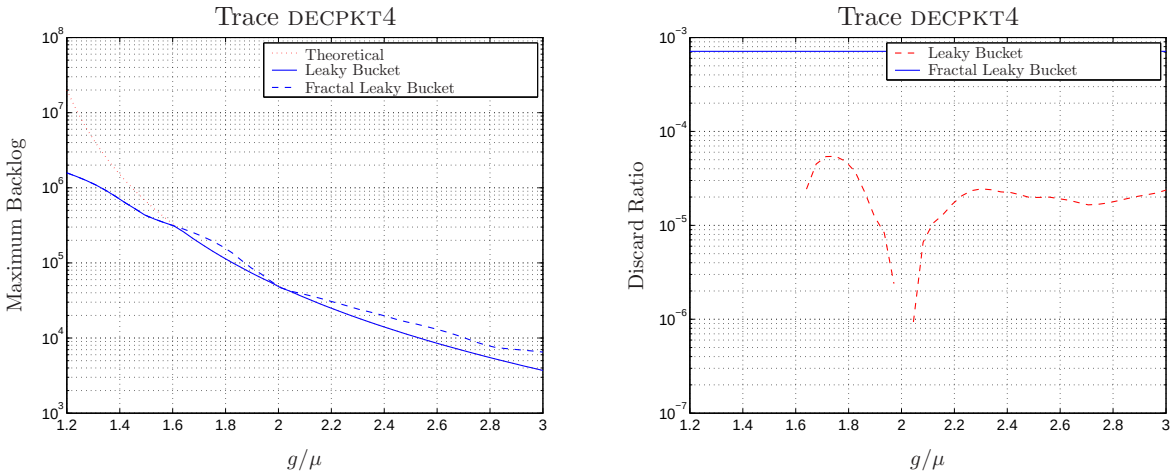

Figure 74: Comparison between the Leaky Bucket and the FLB algorithms for the trace DECPKT4.

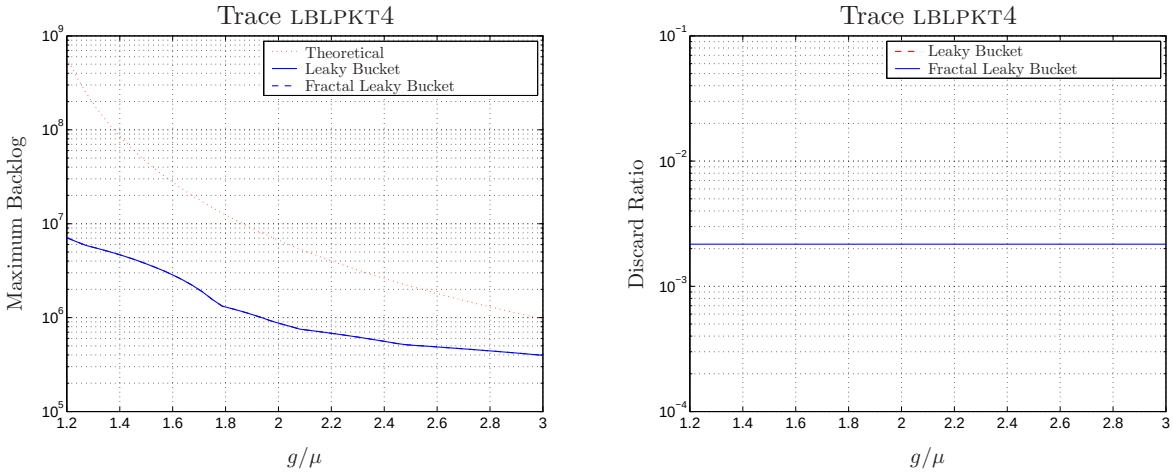

Figure 75: Comparison between the Leaky Bucket and the FLB algorithms for the trace LBLPKT4.

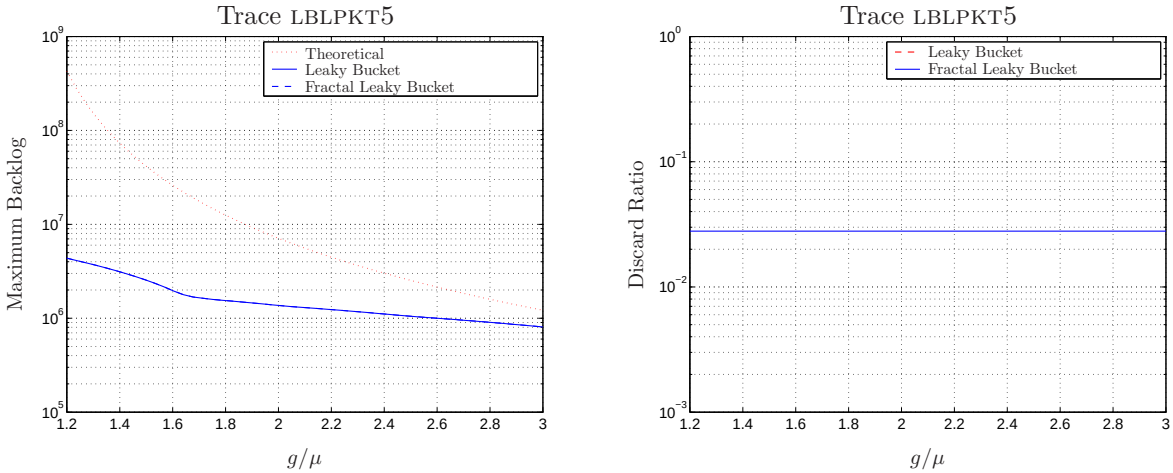

Figure 76: Comparison between the Leaky Bucket and the FLB algorithms for the trace LBLPKT5.

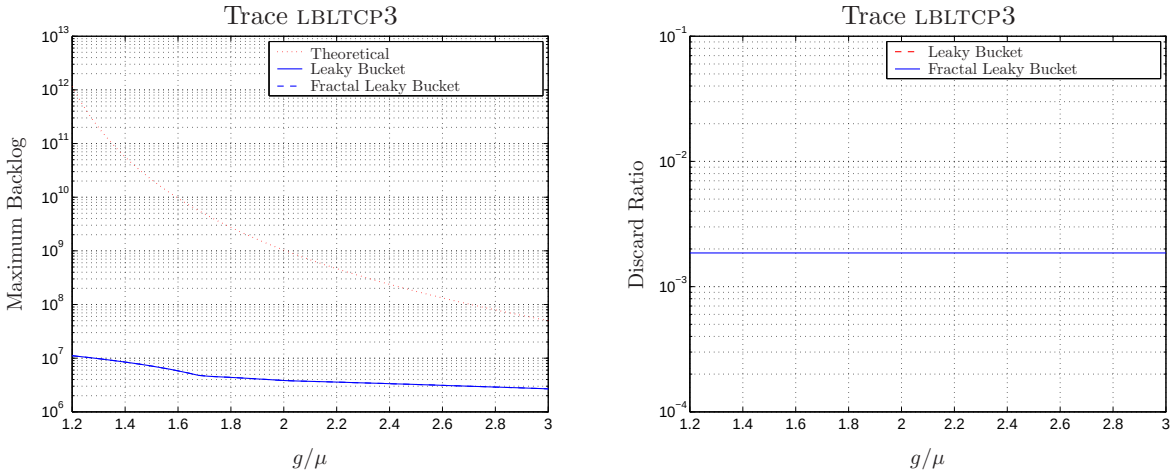

Figure 77: Comparison between the Leaky Bucket and the FLB algorithms for the trace LBLTCP3.

7 The Multifractal Bounded Arrival Process

In this section, a new envelope process which is capable of representing multifractal traffic is proposed. Let the cumulated traffic of a source be given by

$$\mathbf{A}(t) = \mu t + \gamma \mathbf{Z}(t),\ t \geq 0, \tag{37}$$

where $\mathbf{Z}(t)$ is a multifractal process with stationary increments, i.e., a process for which

$$\mathbf{Z}(mt) \stackrel{d}{=} \mathbf{C}(m)\mathbf{Z}(t),\ m, t \geq 0, \tag{38}$$

where $\mathbf{Z}(t)$ and $\mathbf{C}(m)$ are independent random functions. Let a generalized scaling index $\mathbf{H}(m)$ be defined as

$$\mathbf{H}(m) = \log_m \mathbf{C}(m)$$

From (38),

$$\mathbf{Z}(mt) \stackrel{d}{=} m^{\mathbf{H}(m)}\mathbf{Z}(t),\ m, t \geq 0.$$

Since the increments of $\mathbf{Z}(t)$ are assumed to be stationary,

$$\begin{aligned} \mathbf{Z}(t+\tau) - \mathbf{Z}(t) &\stackrel{d}{=} \mathbf{Z}(\tau) - \mathbf{Z}(0) \\ &\stackrel{d}{=} \tau^{\mathbf{H}(\tau)}\mathbf{Z}(1),\ \tau, t \geq 0. \end{aligned}$$

Therefore,

$$\mathbf{Z}(t+\tau) - \mathbf{Z}(t) \stackrel{d}{=} \tau^{\mathbf{H}(\tau)}\mathbf{Z}(1).$$

The increments of the process $\mathbf{A}(t)$ in the interval $[t; t+\tau]$ are then given by

$$\begin{aligned} \Delta\mathbf{A}(t; t+\tau) &= \mathbf{A}(t+\tau) - \mathbf{A}(t) \\ &= \mu\tau + \gamma\mathbf{Z}(\tau). \end{aligned} \tag{39}$$

Notice that the process $\Delta\mathbf{A}(t; t+\tau)$ is also stationary, and can thus be denoted by $\Delta\mathbf{A}(\tau)$ for the sake of simplicity. Assuming the realization of $\mathbf{H}(\tau)$ to be known and to be denoted by $H(\tau)$, an envelope process for $\Delta\mathbf{A}(\tau)$ can be defined as

$$\Delta\widehat{A}(\tau) = \mu\tau + k\gamma\tau^{H(\tau)},\ \forall\tau \geq 0. \tag{40}$$

The process $\Delta\widehat{A}(\tau)$ will be called a *Multifractal Bounded Arrival Process* (MFBAP), and can

be regarded as a generalization of the Fractional Bounded Arrival Process, which was introduced in Section 2.

The probability that the traffic increments exceed $\Delta\widehat{A}(\tau)$ is

$$\begin{aligned}\mathbb{P}\left\{\Delta\mathbf{A}(\tau) > \Delta\widehat{A}(\tau)\right\} &= \mathbb{P}\left\{\mu\tau + \gamma\mathbf{Z}(\tau) > \mu\tau + k\gamma\tau^{H(\tau)}\right\} \\ &= \mathbb{P}\left\{\mu\tau + \gamma\tau^{H(\tau)}\mathbf{Z}(1) > \mu\tau + k\gamma\tau^{H(\tau)}\right\} \\ &= \mathbb{P}\left\{\mathbf{Z}(1) > k\right\} \\ &= \overline{F}_{\mathbf{Z}}(k), \end{aligned} \tag{41}$$

where $\overline{F}_{\mathbf{Z}}$ denotes the residual distribution function of $\mathbf{Z}(1)$.

For a given traffic trace and a target probability of violation of ϵ, the MFBAP envelope can be obtained as follows. Let $\Delta\mathbf{A}_{(i)}[n]$ be the discrete increments of traffic, as defined in Section 2. The parameter μ is then given by

$$\mu = \frac{1}{T}\mathbb{E}\left\{\Delta A[n]\right\}. \tag{42}$$

Now, let θ_H denote the set of parameters of the generalized index function $H(\tau)$, and the empirical envelope for the process $\Delta\mathbf{A}_{(i)}[n]$ be defined as

$$\Delta A^{(e)}_{(i)}[n] = \mu T i + Z^{(e)}[i], \tag{43}$$

where

$$Z^{(e)}[i] = \left\{x \in \mathbb{R} : \mathbb{P}\left\{\Delta\mathbf{A}_{(i)}[n] - \mu T i > x\right\} = \varepsilon\right\}. \tag{44}$$

Assuming that a large number of samples of the time series $\Delta A_{(i)}[n]$ is available, the probability indicated in (16) can be approximated by the relative frequency of the event $\left\{\Delta\mathbf{A}_{(i)}[n] - \mu T i > x\right\}$, whenever the corresponding distribution is not known. The product $k\gamma$ and the set of parameters θ_H can be obtained by solving

$$\{k\gamma, \theta_H\} = \arg \min_{\substack{k\gamma \geq 0 \\ \theta_H \in \mathcal{D}(\theta_{\mathcal{H}})}} \left\{\frac{1}{2}\sum_{i=1}^{N}\left[f\left(\log Z^{(e)}[i] - \log k\gamma - \log i^{H[i]}\right)\right]^2\right\}, \tag{45}$$

where $\mathcal{D}(\theta_H)$ denotes de domain of the set θ_H, $H[i] = H(iT)$, and

$$f(x) = \begin{cases} x, & x < 0 \\ wx, & x \geq 0,\ w > 1. \end{cases}$$

The value of w must be chosen so that a feasible compromise between the violation of (43) and the looseness of the envelope is obtained.

The use of the MFBAP envelope for real traffic modeling is illustrated in Fig. 78–89. Traffic traces to be represented are those which were described in Section 10. For the MFBAP envelope, the function $H[i]$ is given by

$$H[i] = H_0 + \gamma_H \exp\left\{ -\frac{\left[\ln(iT) - m_H\right]^2}{2\sigma_H^2} \right\}$$

The choice of such a function is motivated by the fact that, for all the traces considered in the present paper, the scaling law of $Z^{(e)}[i]$ vanishes for sufficiently small and for sufficiently large time scales, as it is shown in Fig. 78–83.

The parameters of the resulting MFBAP envelopes are shown in Table 2. For each trace, the parameter μ is obtained by using (42), and the remaining parameters are obtained by using (45), for which $w = 10$, and a target probability of violation of 10^{-4} was arbitrarily chosen.

For each trace, the empirical envelope and the resulting MFBAP envelope are shown in Fig. 84–89. Notice that the MFBAP model is able to accurately represent the traffic, including the traces which could not be adequately represented by the FBAP envelope.

Trace	μ	$k\gamma$	H_0	γ_H	σ_H	μ_H
BCOct89Ext	7.49×10^2	5.30×10^4	0.378	0.406	4.05×10^{-2}	-1.569
BCOct89Ext4	6.22×10^3	1.86×10^5	0.489	0.486	2.32×10^{-3}	0.623
BCpAug89	1.38×10^5	4.52×10^5	0.393	0.374	3.70×10^{-3}	-0.428
BCpOct89	3.63×10^5	6.27×10^5	0.485	0.483	1.81×10^{-2}	-1.381
ASTERIX	5.59×10^5	1.58×10^6	0.480	0.347	1.86×10^{-2}	-1.555
ATP	5.47×10^5	1.14×10^6	0.468	0.557	1.10×10^{-1}	0.317
BOND	6.08×10^5	1.39×10^6	0.391	0.293	2.33×10^{-14}	0.000
DECPKT2	2.38×10^5	3.85×10^5	0.419	0.403	1.41×10^{-2}	-1.965
DECPKT3	1.81×10^5	3.10×10^5	0.401	0.403	1.30×10^{-2}	-1.630
DECPKT4	2.63×10^5	3.12×10^5	0.400	0.434	1.76×10^{-2}	-1.335
DINO	3.27×10^5	8.48×10^5	0.491	0.363	4.87×10^{-2}	0.324
FUSS1	6.78×10^5	1.61×10^6	0.405	0.304	1.65×10^{-2}	-0.745
FUSS2	6.28×10^5	1.81×10^6	0.491	0.455	8.00×10^{-2}	-0.605
LAMBS	1.83×10^5	7.15×10^5	0.385	0.497	3.23×10^{-2}	1.759
LBLPKT4	3.64×10^4	2.58×10^5	0.409	0.345	1.67×10^{-2}	-2.381
LBLPKT5	2.61×10^4	2.33×10^5	0.393	0.394	1.71×10^{-2}	-1.259
LBLTCP3	3.39×10^4	3.65×10^5	0.464	0.437	2.13×10^{-2}	-1.232
MOVIE	3.57×10^5	9.93×10^5	0.392	0.328	2.19×10^{-2}	0.473
MRBEAN	4.41×10^5	1.39×10^6	0.543	0.368	4.84×10^{-2}	0.313
MTV1	6.15×10^5	2.12×10^6	0.571	0.413	8.30×10^{-2}	-0.427
MTV2	4.95×10^5	2.51×10^6	0.460	0.372	1.58×10^{-2}	0.476
NEWS1	5.17×10^5	1.43×10^6	0.564	0.292	1.02×10^{-1}	-0.373
NEWS2	3.84×10^5	2.11×10^6	0.493	0.486	1.34×10^{-1}	-1.278
RACE	7.69×10^5	2.00×10^6	0.411	0.496	4.72×10^{-2}	-1.051
SBOWL	5.88×10^5	1.17×10^6	0.453	0.429	5.05×10^{-2}	-0.159
SIMPSONS	4.64×10^5	1.17×10^6	0.443	0.305	1.92×10^{-2}	-0.341
STAR2	2.33×10^5	7.51×10^5	0.462	0.465	5.09×10^{-2}	1.546
TALK1	3.63×10^5	6.50×10^5	0.369	0.608	5.28×10^{-2}	0.870
TALK2	4.48×10^5	1.03×10^6	0.470	0.571	8.59×10^{-2}	0.157
TERMINATOR	2.73×10^5	5.12×10^5	0.420	0.281	2.87×10^{-2}	-1.038

Table 2: Parameters of the MFBAP envelopes corresponding to the traces analyzed in this paper.

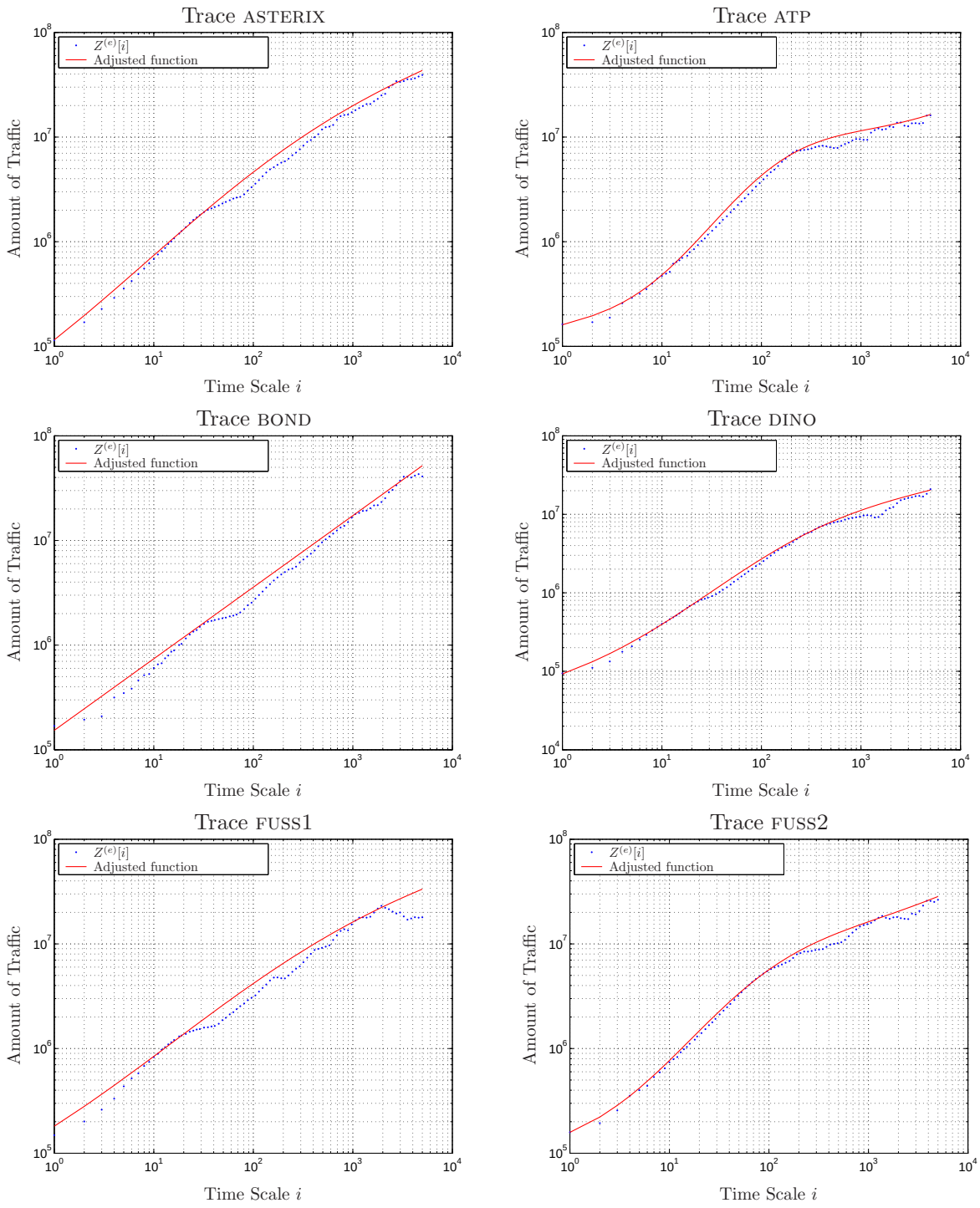

Figure 78: Representation of the traces ASTERIX,ATP,BOND,DINO,FUSS1 and FUSS2 by using the MFBAP envelope.

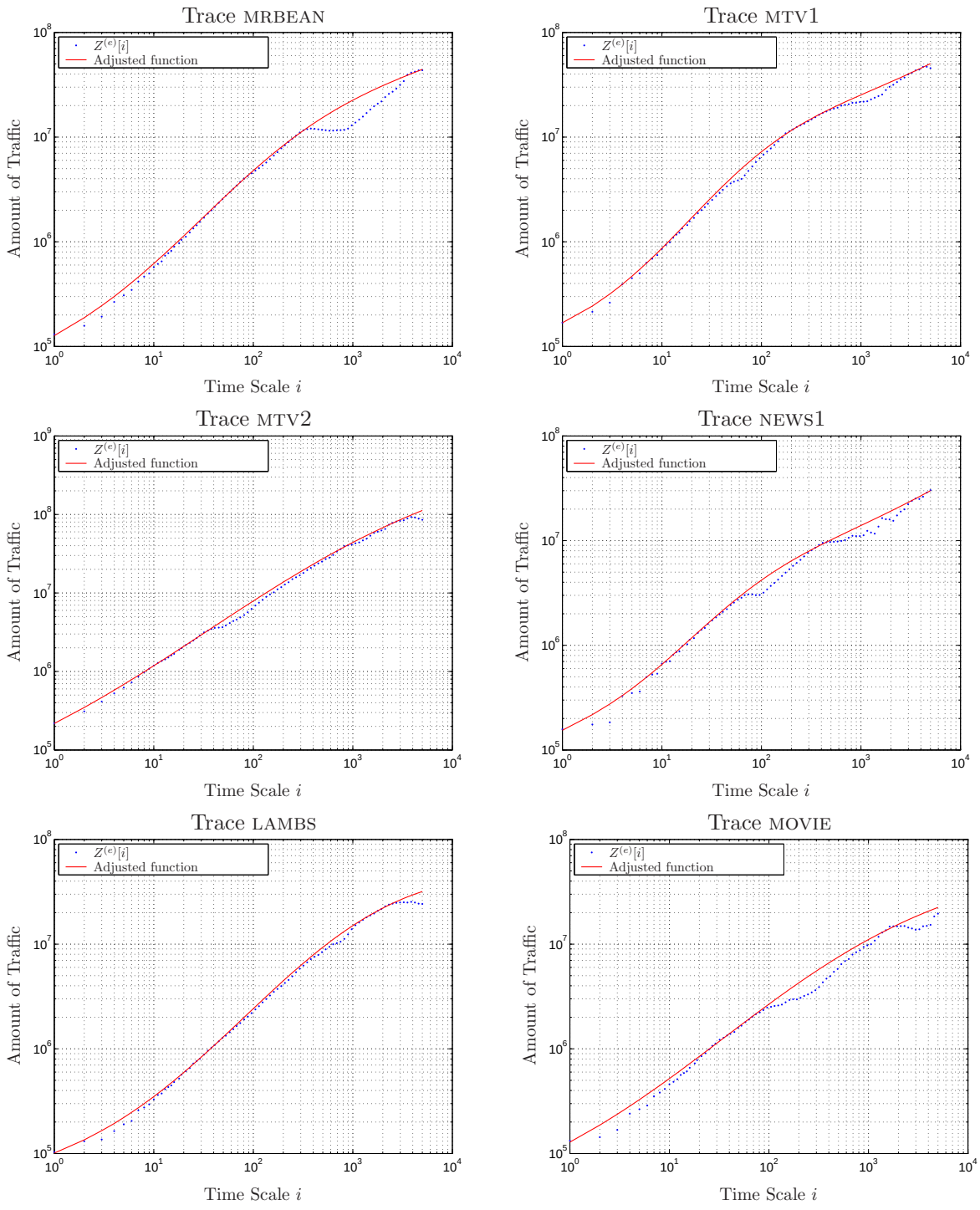

Figure 79: Representation of the traces MRBEAN,MTV1,MTV2,NEWS1, LAMBS and MOVIE by using the MFBAP envelope.

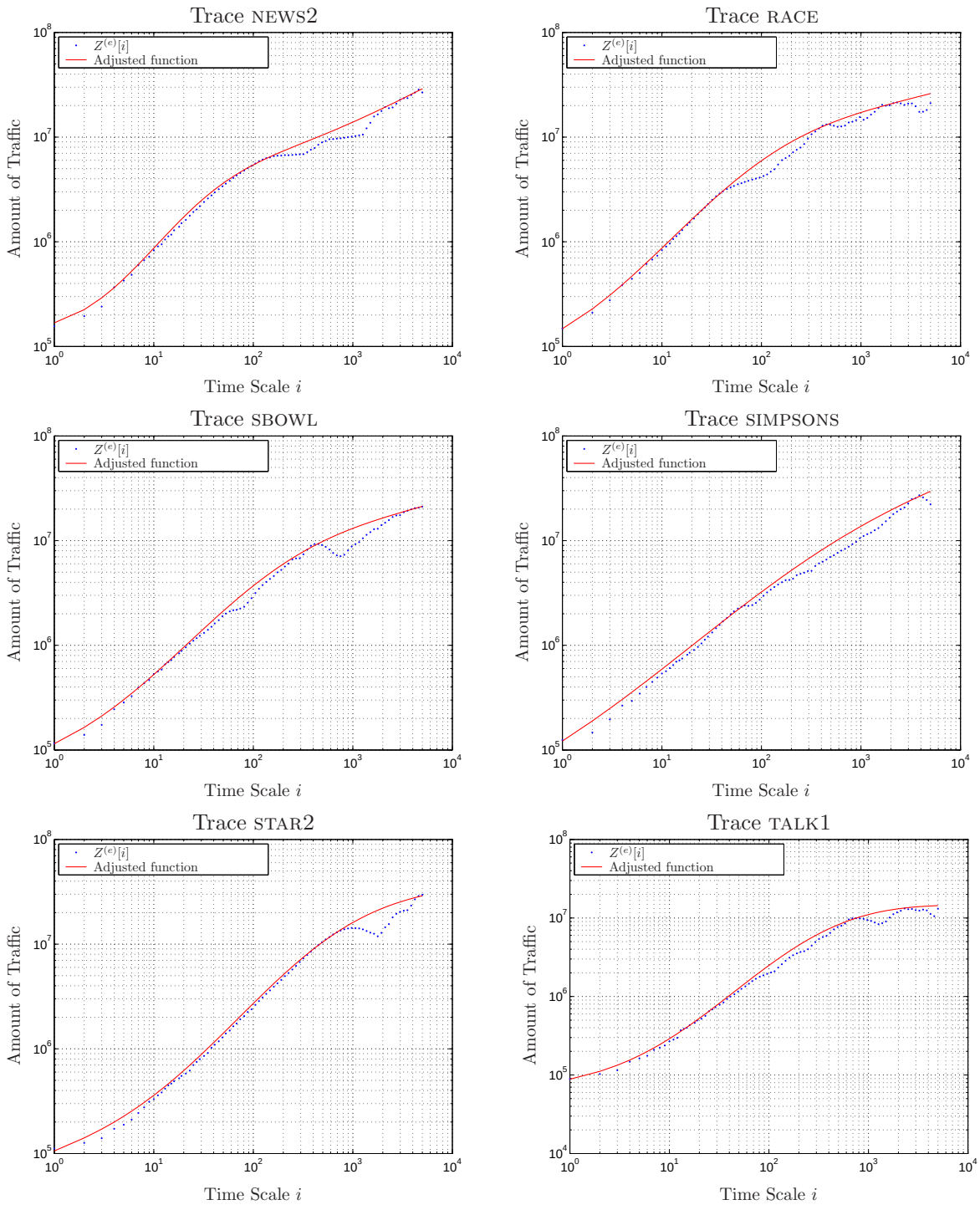

Figure 80: Representation of the traces NEWS2,RACE,SBOWL,SIMPSONS,STAR2 and TALK1 by using the MFBAP envelope.

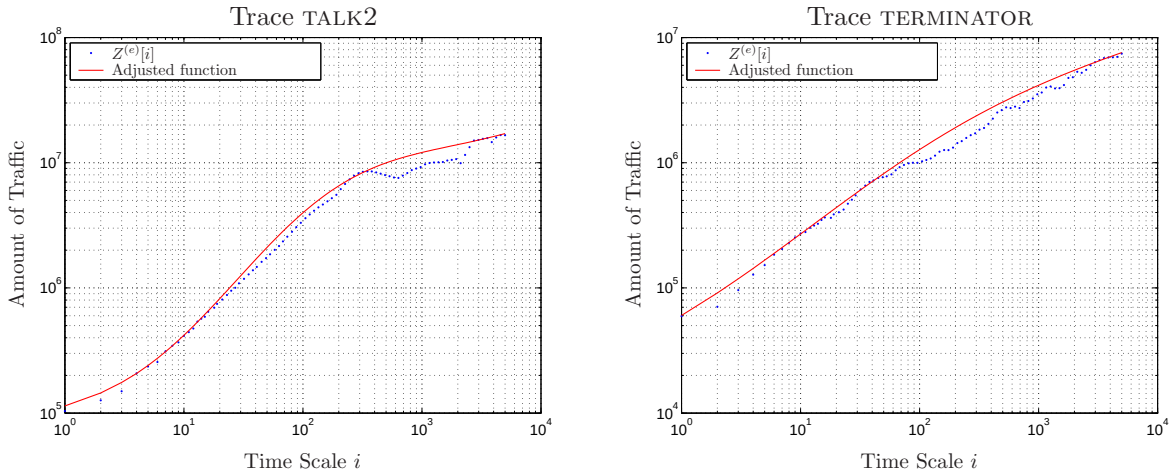

Figure 81: Representation of the traces TALK2 and TERMINATOR by using the MFBAP envelope.

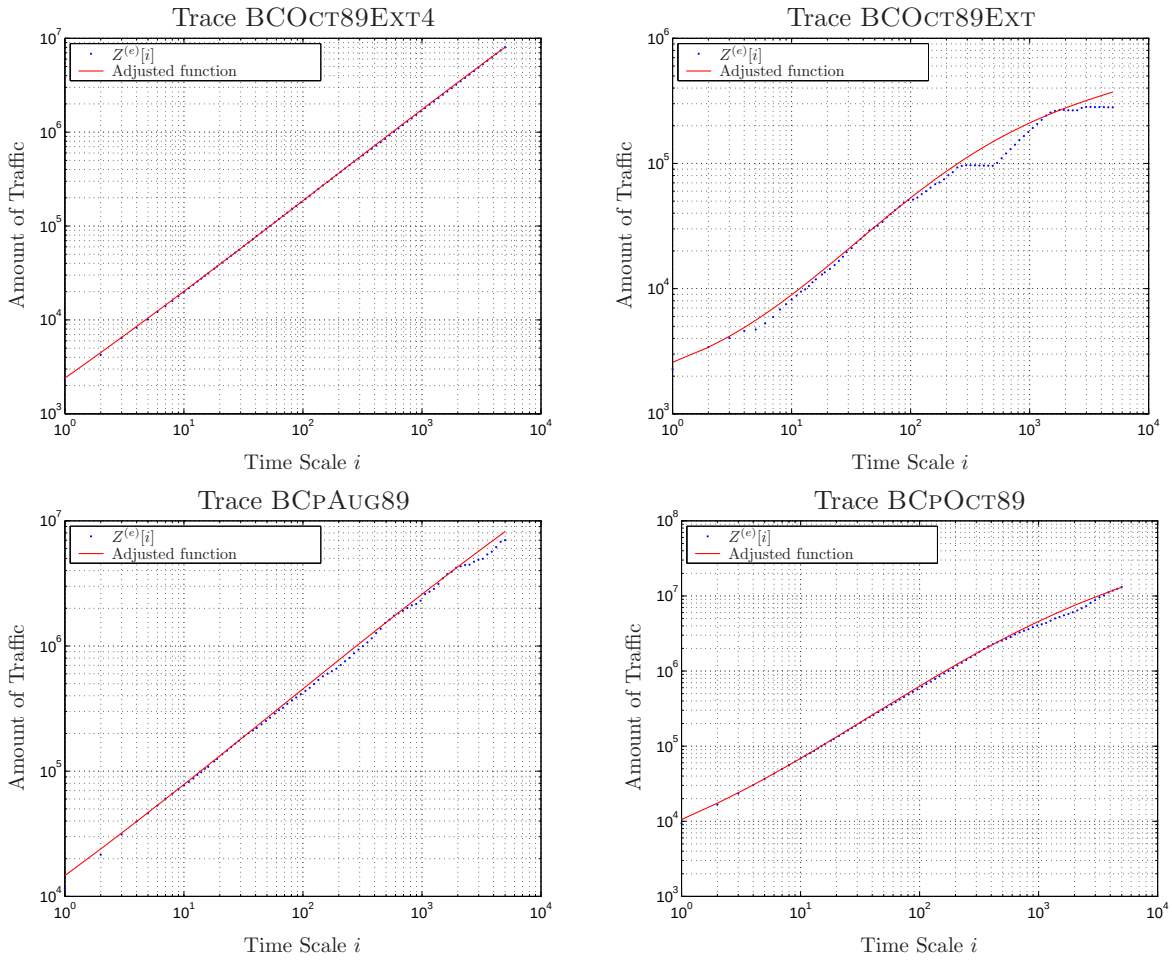

Figure 82: Representation of the traces BCOct89Ext4,BCOct89Ext,BCpAug89 and BCpOct89 by using the MFBAP envelope.

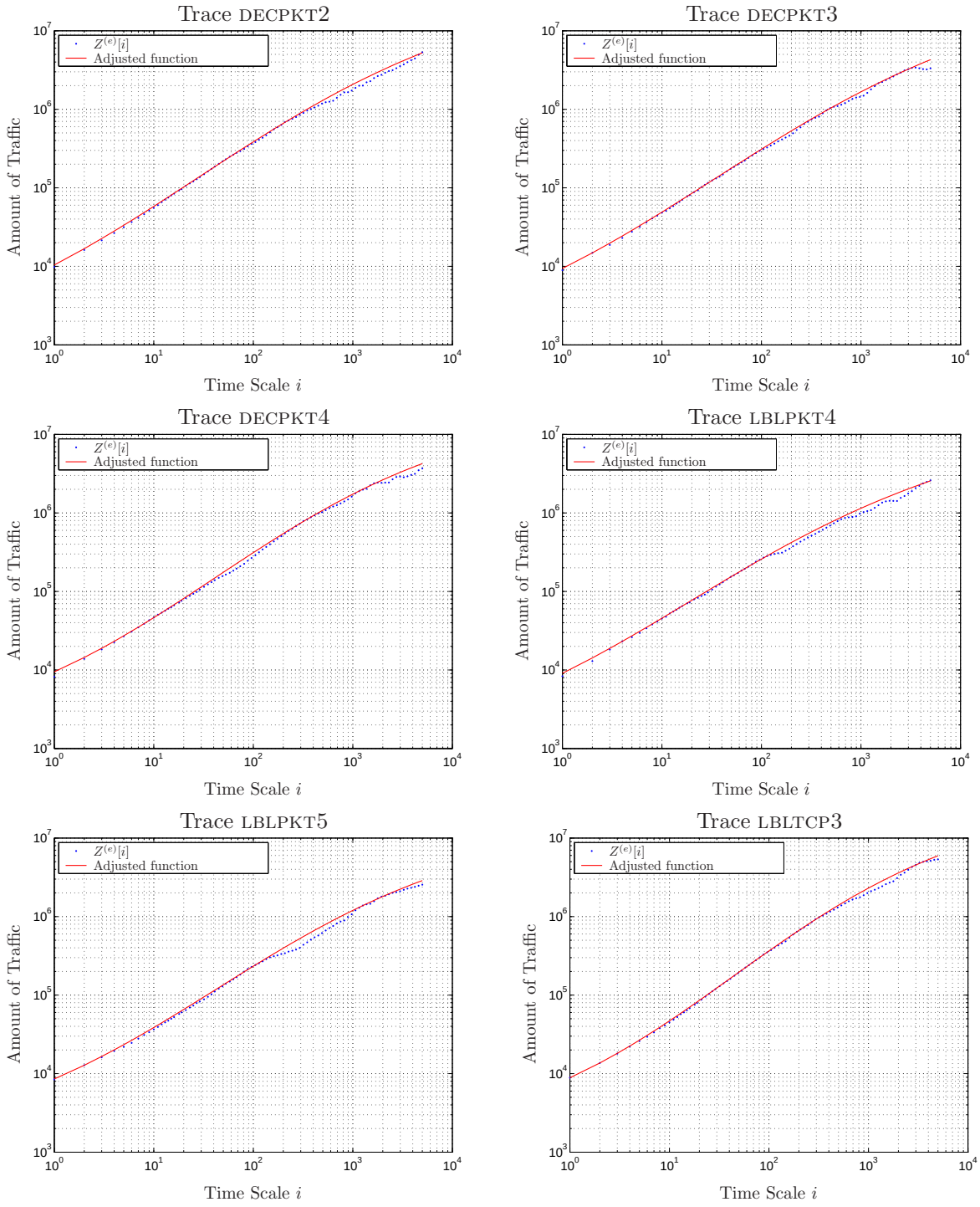

Figure 83: Representation of the traces DECPKT2, DECPKT3, DECPKT4, LBLPKT4, LBLPKT5 and LBLTCP3 by using the MFBAP envelope.

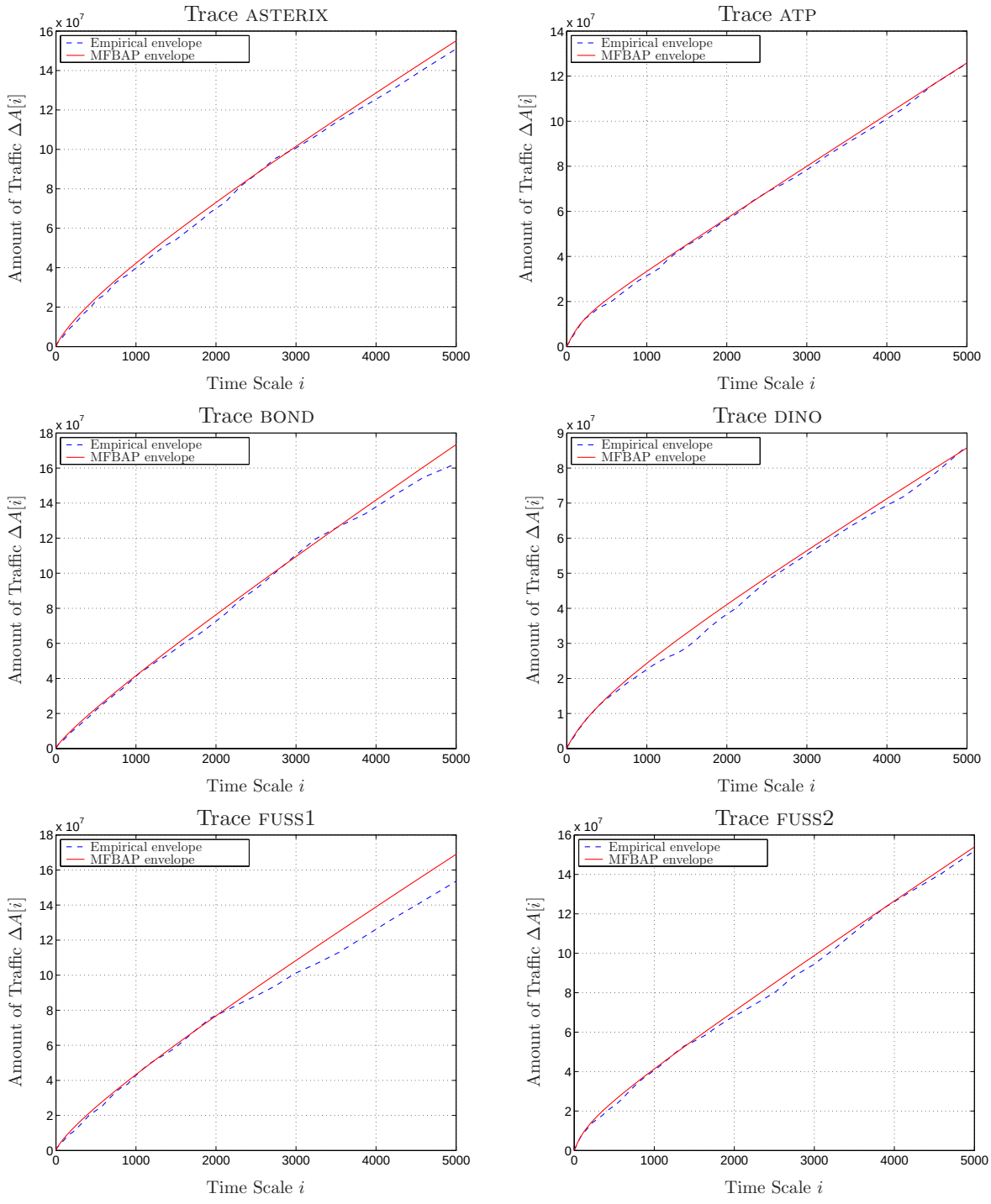

Figure 84: Representation of the traces ASTERIX, ATP, BOND, DINO, FUSS1 and FUSS2 by using the MFBAP envelope.

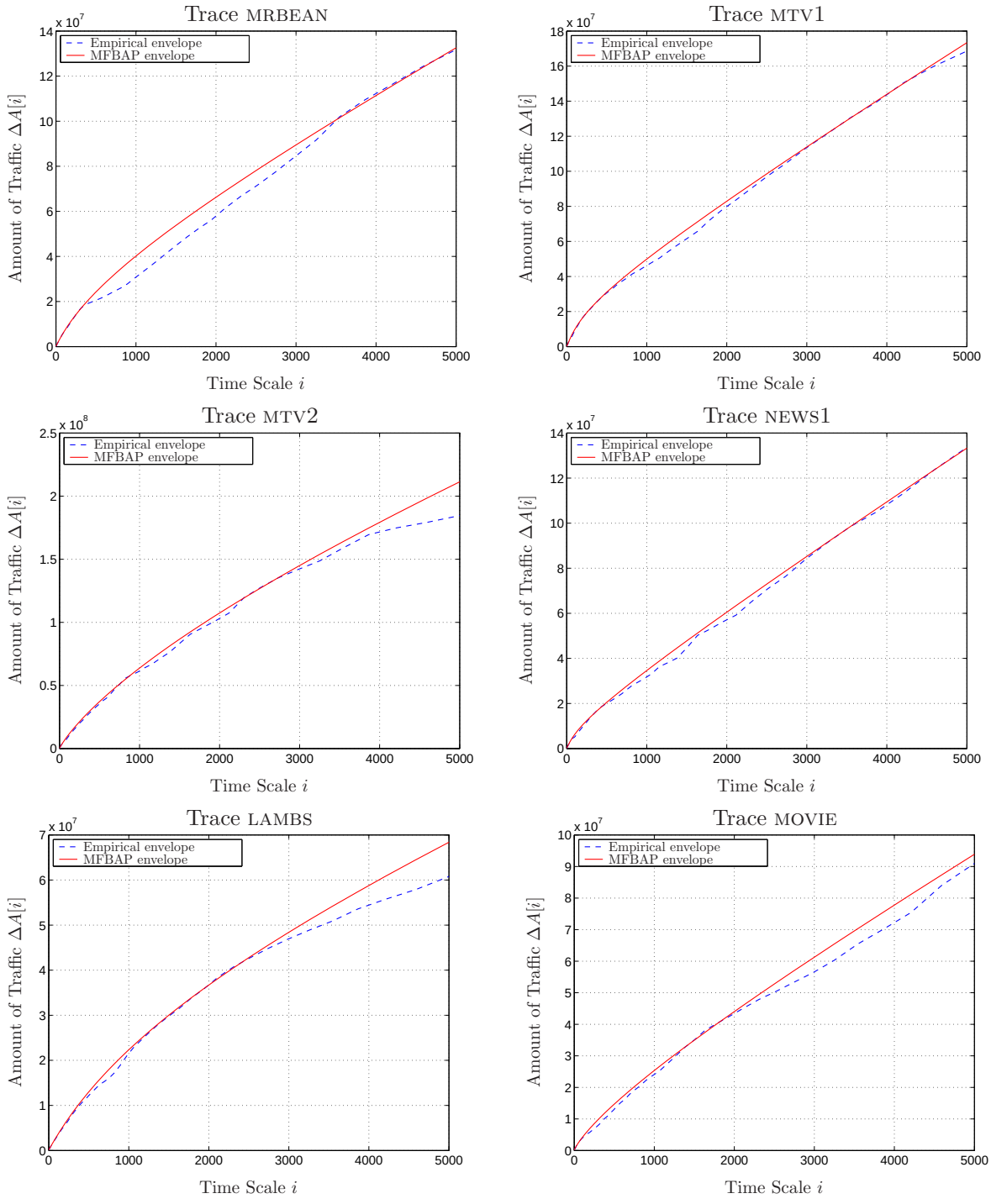

Figure 85: Representation of the traces MRBEAN, MTV1, MTV2, NEWS1, LAMBS and MOVIE by using the MFBAP envelope.

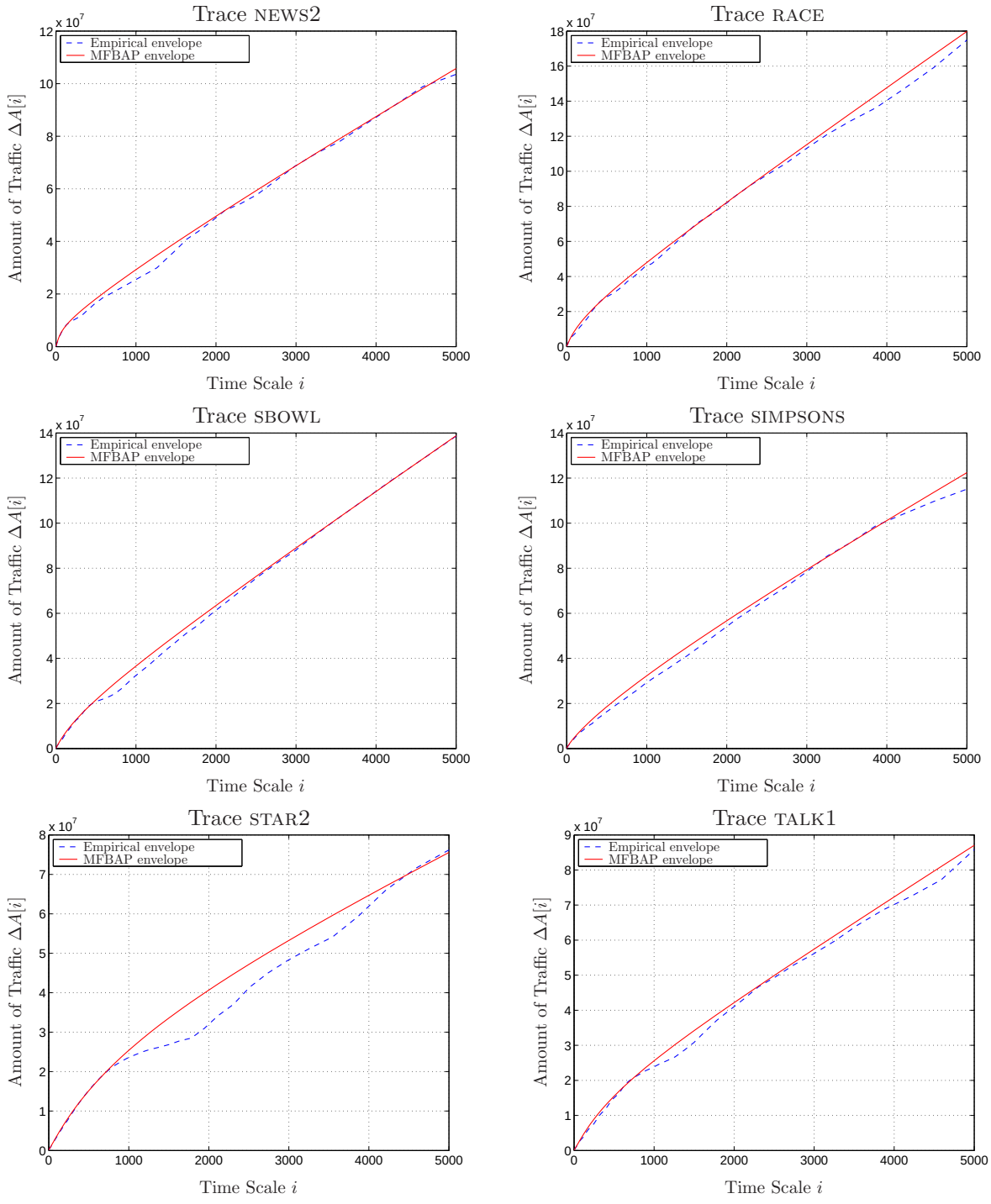

Figure 86: Representation of the traces NEWS2, RACE, SBOWL, SIMPSONS, STAR2 and TALK1 by using the MFBAP envelope.

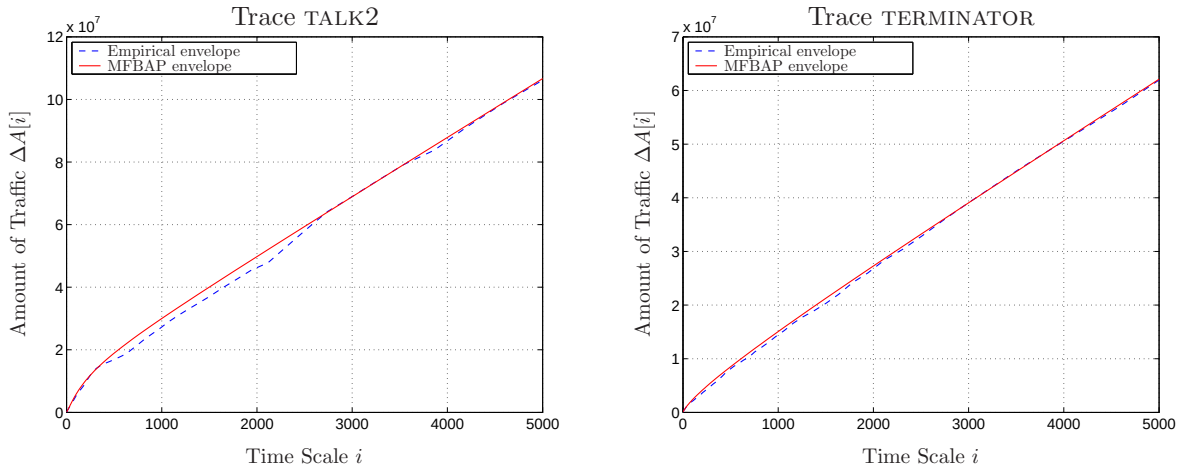

Figure 87: Representation of the traces TALK2 and TERMINATOR by using the MFBAP envelope.

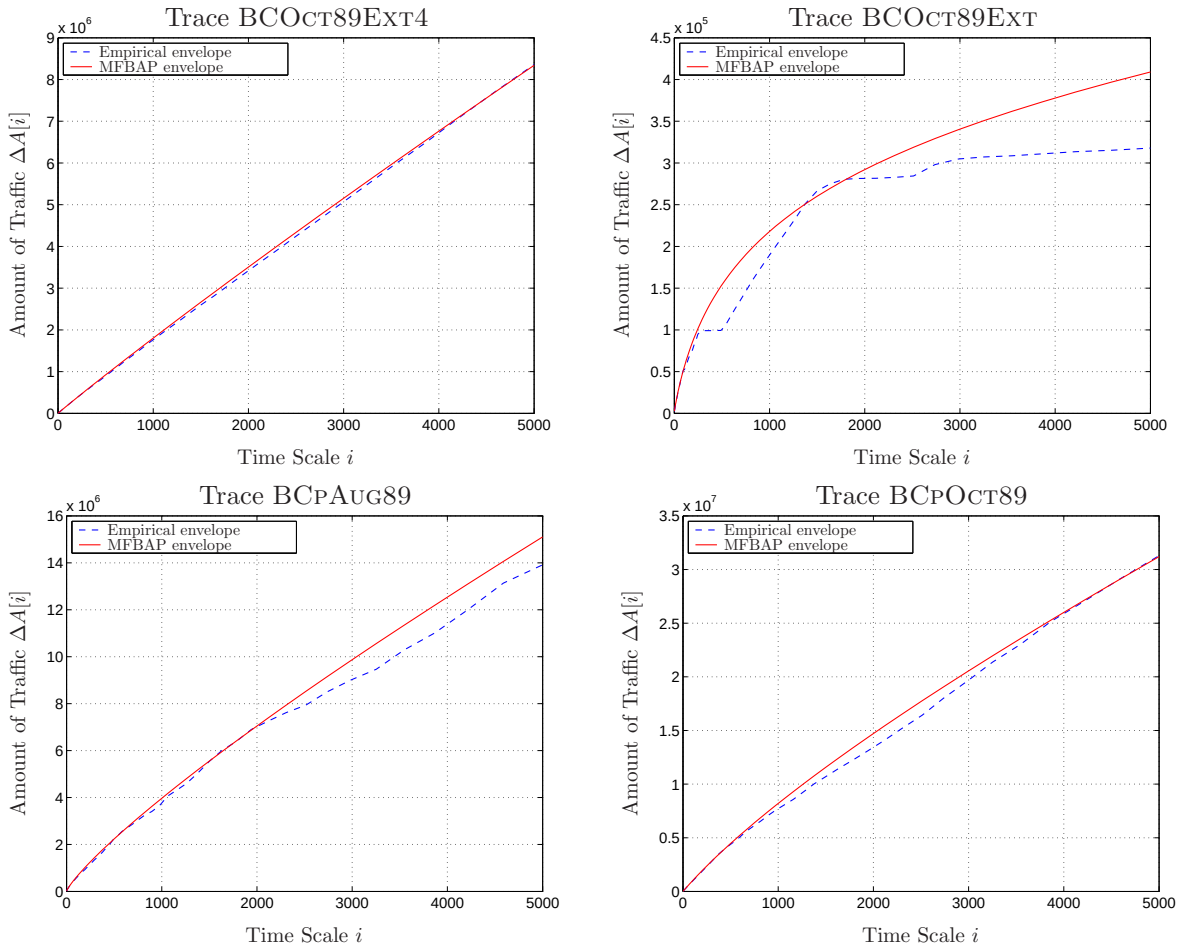

Figure 88: Representation of the traces BCOct89Ext4, BCOct89Ext, BCpAug89 andBCpOct89 by using the MFBAP envelope.

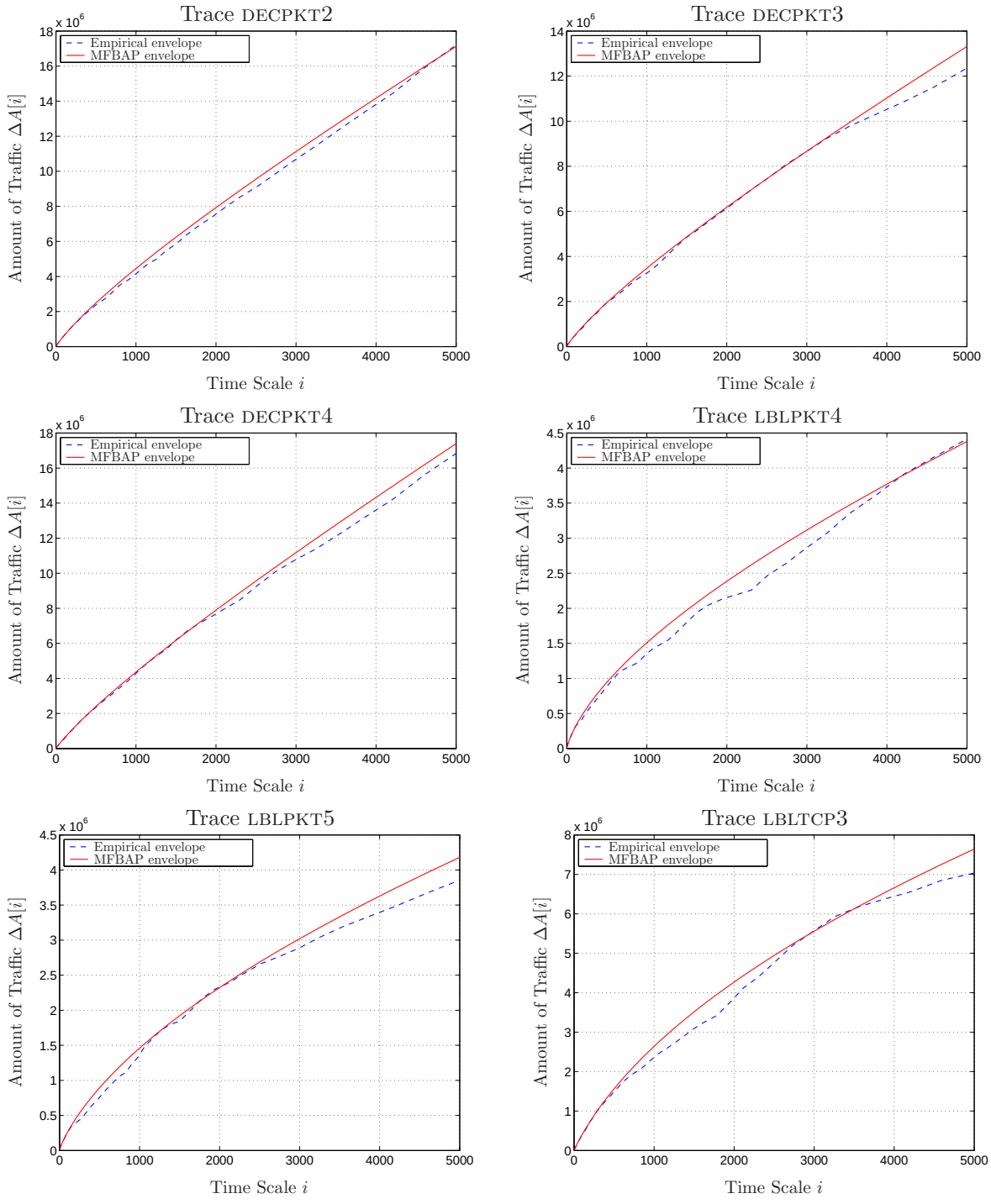

Figure 89: Representation of the traces DECPKT2, DECPKT3, DECPKT4, LBLPKT4, LBLPKT5 and LBLTCP3 by using the MFBAP envelope.

8 Comments on the queueing analysis and on the policing of MFBAP traffic

In Section 3, a queueing analysis for FBAP traffic was carried out. It is straightforward to extend such an analysis for the MFBAP traffic. Assuming the notation introduced in Section 3, the bounds for the backlog and for the delay are given by

$$\begin{aligned} Q^* &= \max_{\tau>0} \Delta\widehat{A}(\tau) - g\tau \\ &= \max_{\tau>0} (\mu - g)\tau + k\gamma\tau^{H(\tau)}, \end{aligned} \tag{46}$$

$$D^* = \frac{Q^*}{g}. \tag{47}$$

Notice that, from (46), it is also trivial to extend the analysis of the Leaky Bucket algorithm provided in Section 5 for the MFBAP traffic. Under such a type of traffic, the parameters r and s are related by

$$s = \max_{\tau>0} \Delta\widehat{A}(\tau) - r\tau.$$

It is worth noticing that, differently from the FBAP traffic case, analytical solutions for (46) and (47) cannot generally be obtained. Therefore, the computation of such bounds must rely on the use of numerical optimization methods. For the real traffic traces which were analyzed in the previous section, the backlog bounds and the corresponding probability of violation as functions of the service rate are shown in Fig.90–119. The probability of violation is approximated by the number of busy cycles for which a violation occurs, divided by the total number of busy cycles. For the sake of simplicity, simulation experiments were conducted assuming the traffic to be a fluid-type traffic. Moreover, the traffic is assumed to be uniformly distributed during the interval which corresponds to each sample of the traces.

From the results shown in Fig. 90–119 and the results obtained in Section 3, it is possible to conclude that the use of the MFBAP envelope leads to backlog bounds which are tighter than those obtained by using FBAP envelopes. In some cases, however, the bounds are still too conservative. The hypothesis for such a result is the fact that the function $H(\tau)$ which was assumed for the MFBAP model may lead to an envelope which is loose for the time scales at which the solution of (46) are obtained.

Finally, it is worth noticing that the use of numerical optimization methods may lead to bounds which do not correspond to global optimal solutions for for (46) and for (47). Thus, it would be desirable to establish convexity conditions for $\Delta\widehat{A}(\tau) - g\tau$ under which the uniqueness of the optimal solutions could be assured. However, such conditions are also hard to be established for the general case. Currently, the lack of analytical solutions and of convexity conditions constitute the major drawback for the use of the MFBAP model, or any similar multifractal envelope process, for queueing analysis purposes.

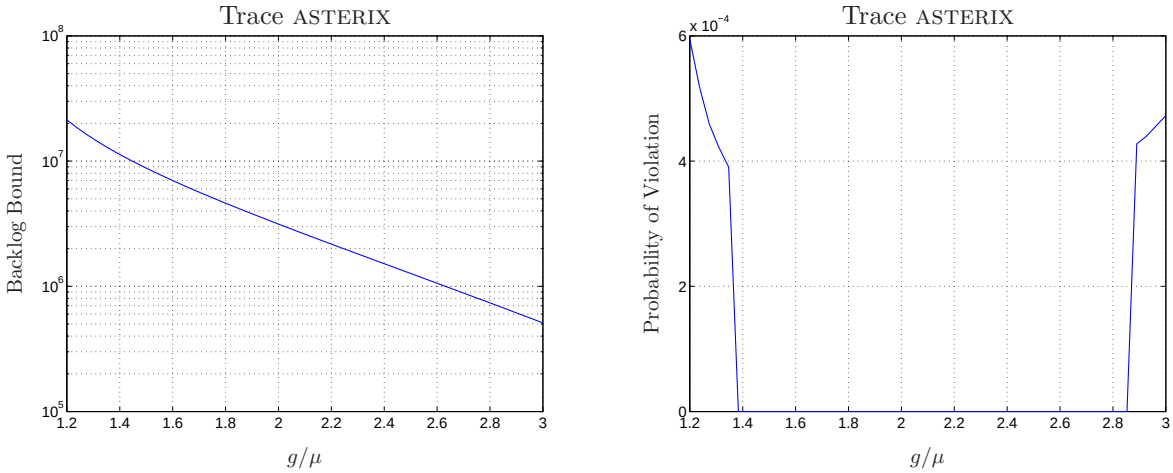

Figure 90: Backlog bound and probability of violation for the trace ASTERIX.

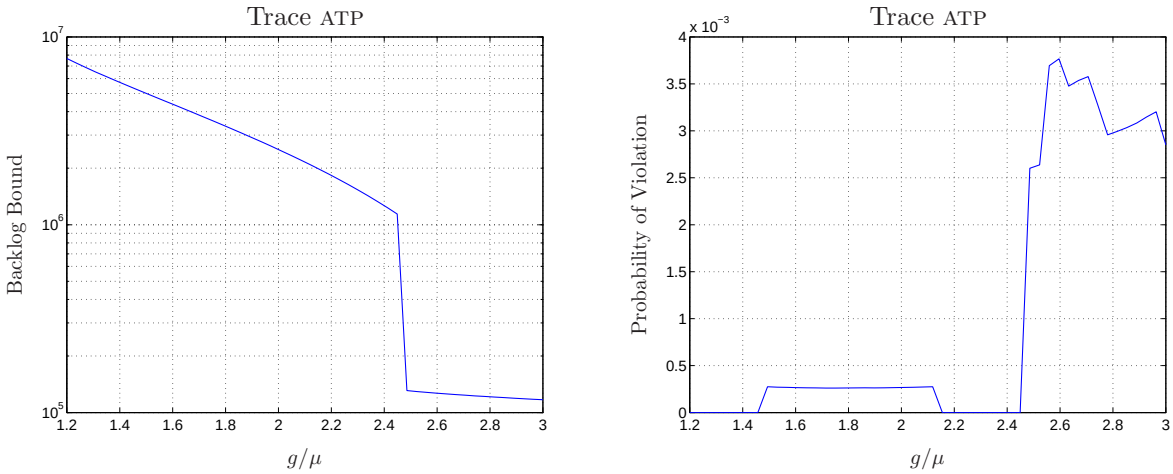

Figure 91: Backlog bound and probability of violation for the trace ATP.

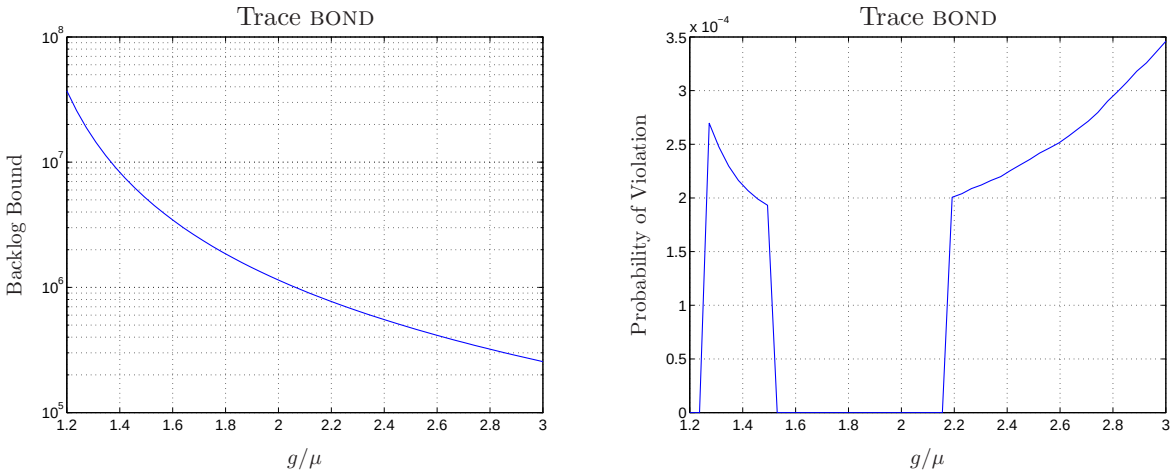

Figure 92: Backlog bound and probability of violation for the trace BOND.

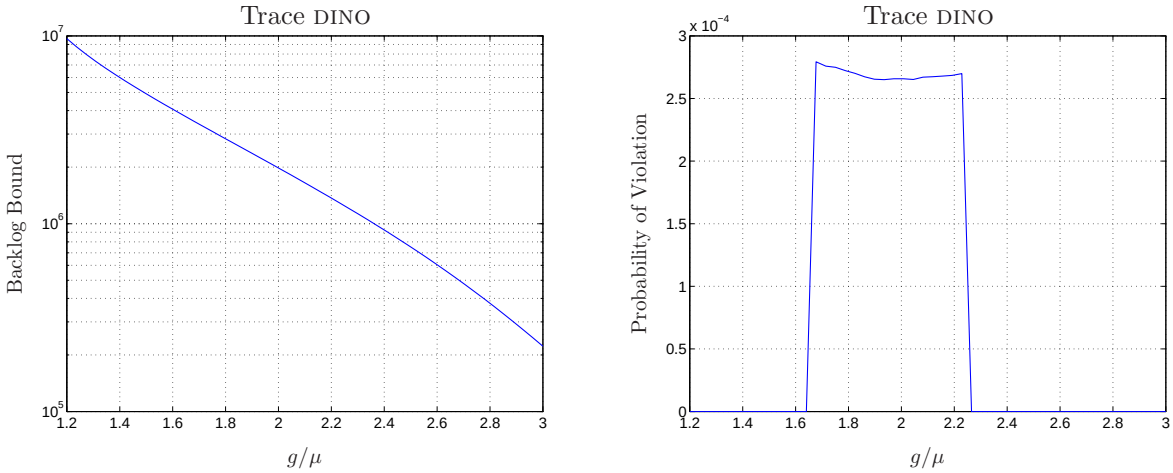

Figure 93: Backlog bound and probability of violation for the trace DINO.

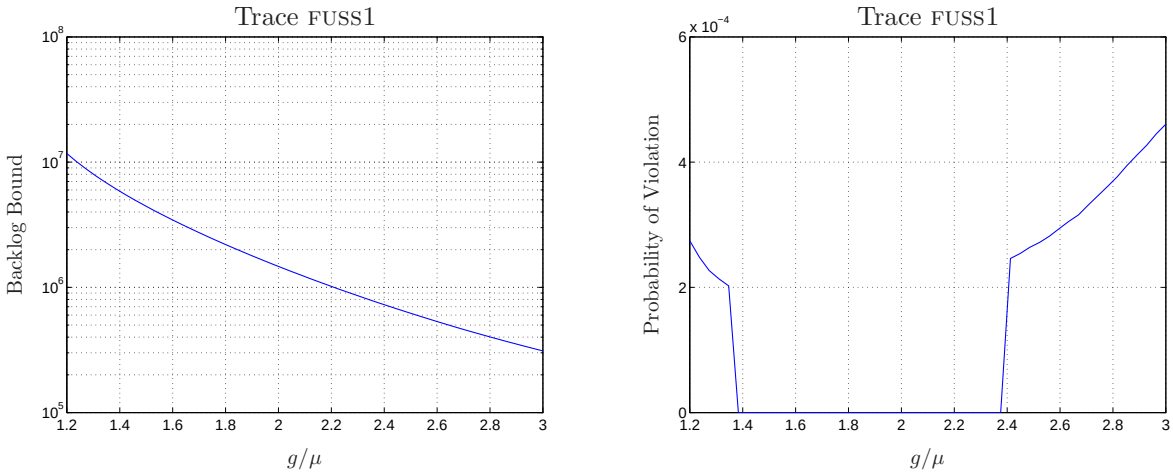

Figure 94: Backlog bound and probability of violation for the trace FUSS1.

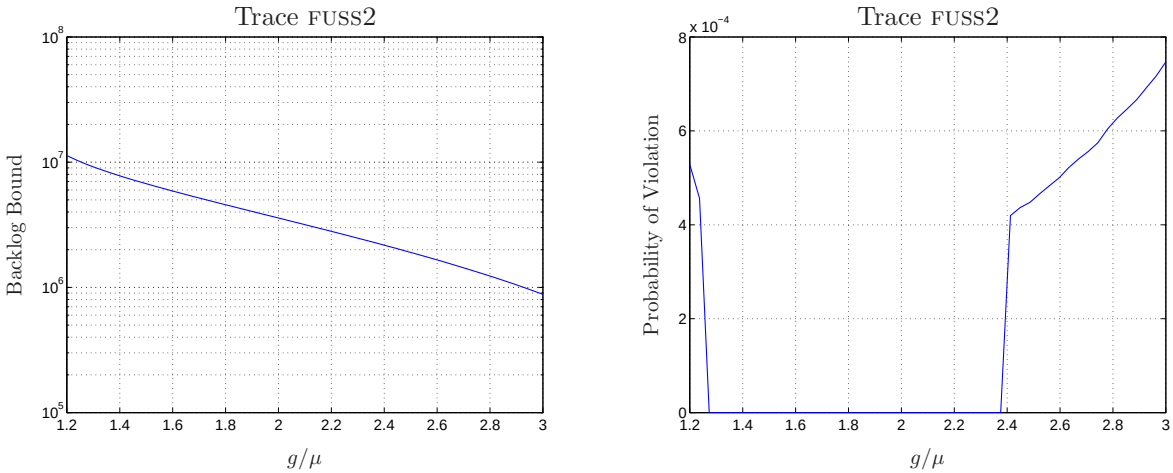

Figure 95: Backlog bound and probability of violation for the trace FUSS2.

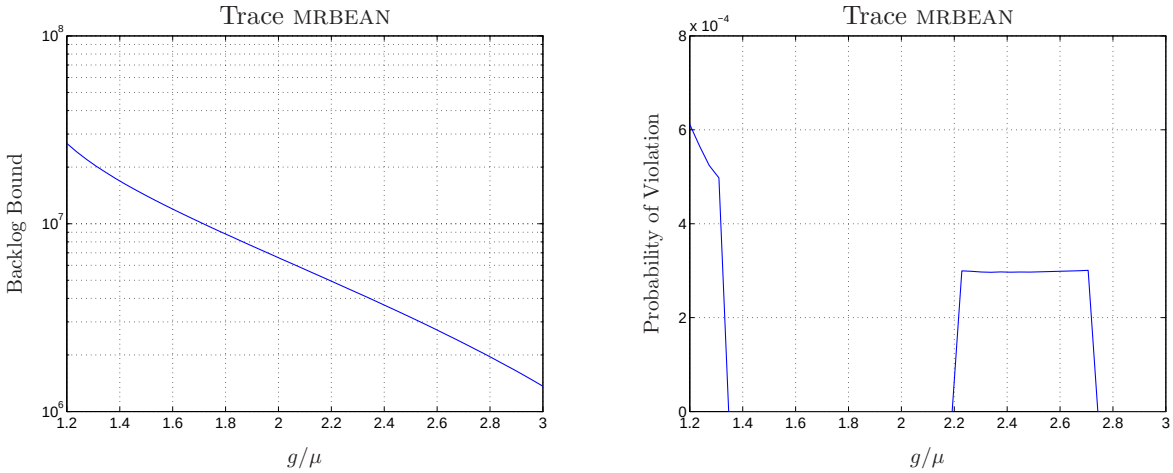

Figure 96: Backlog bound and probability of violation for the trace MRBEAN.

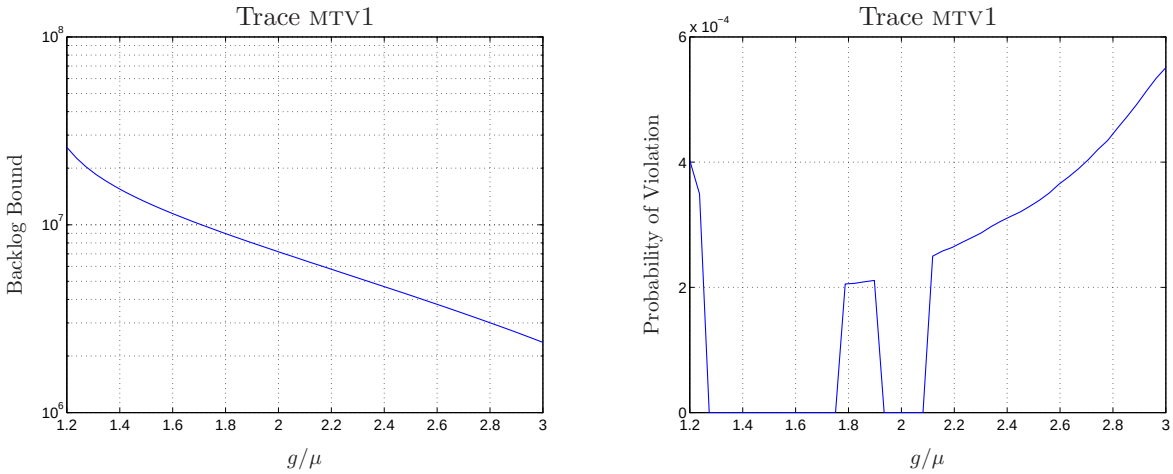

Figure 97: Backlog bound and probability of violation for the trace MTV1.

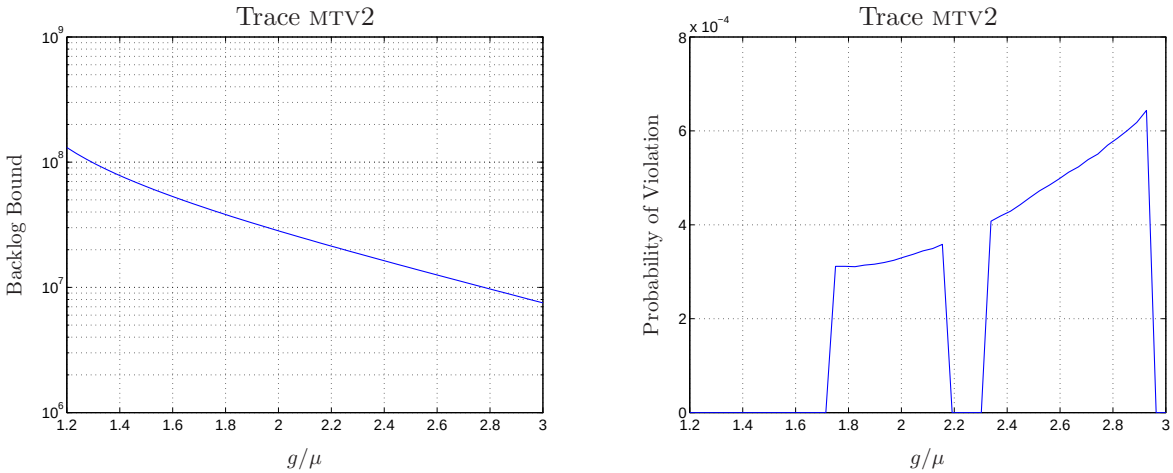

Figure 98: Backlog bound and probability of violation for the trace MTV2.

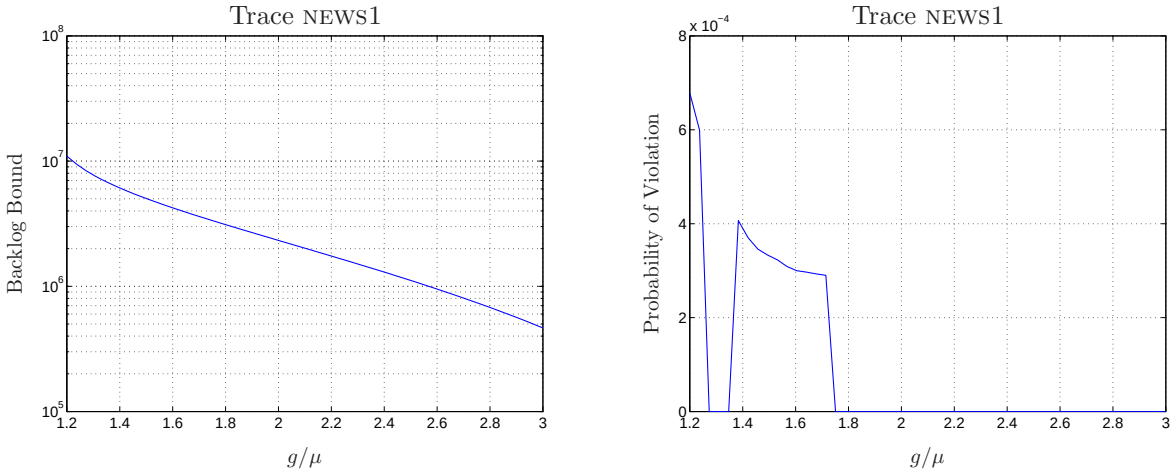

Figure 99: Backlog bound and probability of violation for the trace NEWS1.

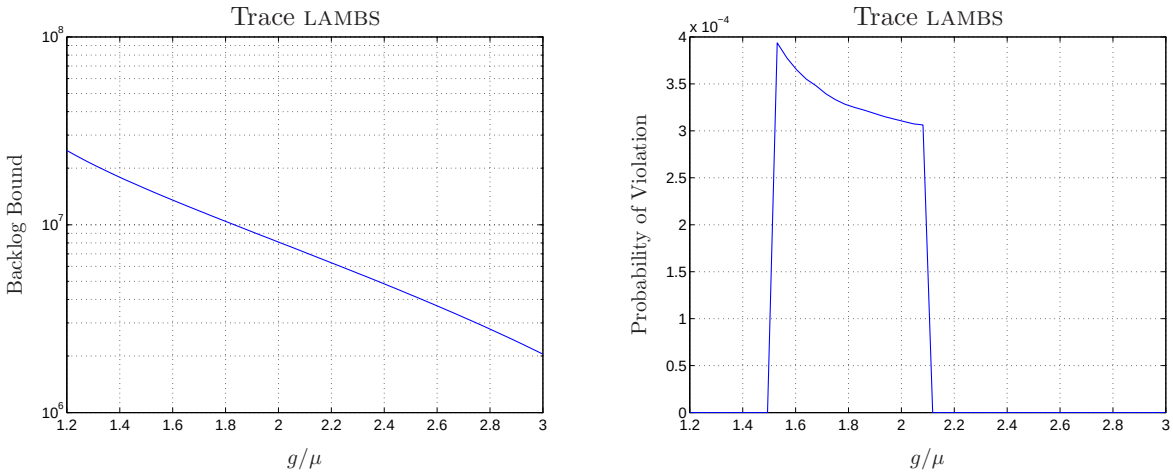

Figure 100: Backlog bound and probability of violation for the trace LAMBS.

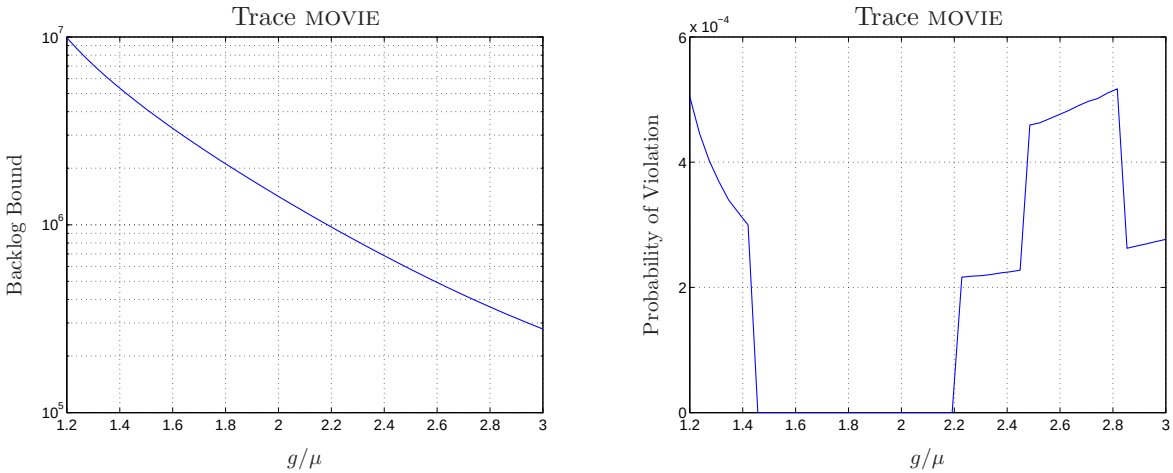

Figure 101: Backlog bound and probability of violation for the trace MOVIE.

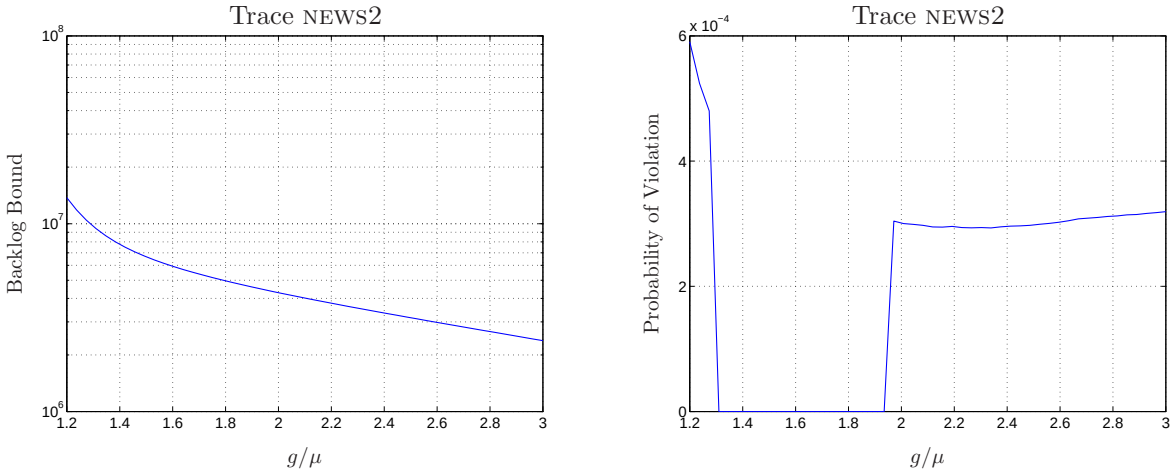

Figure 102: Backlog bound and probability of violation for the trace NEWS2.

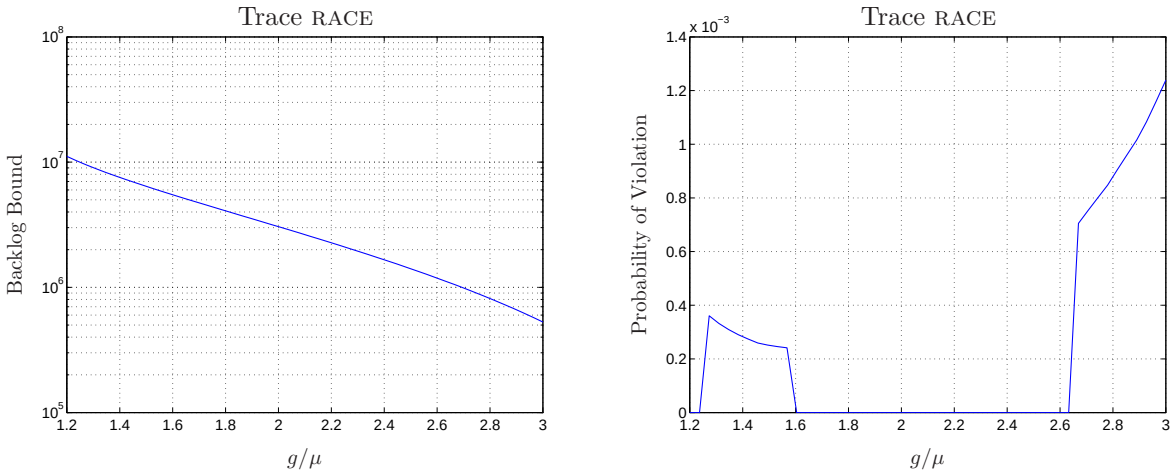

Figure 103: Backlog bound and probability of violation for the trace RACE.

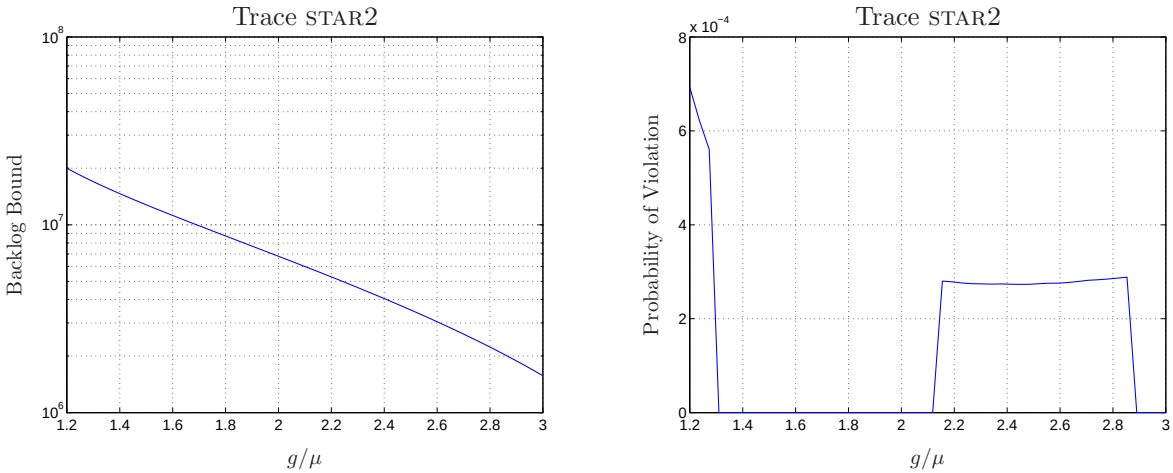

Figure 104: Backlog bound and probability of violation for the trace STAR2.

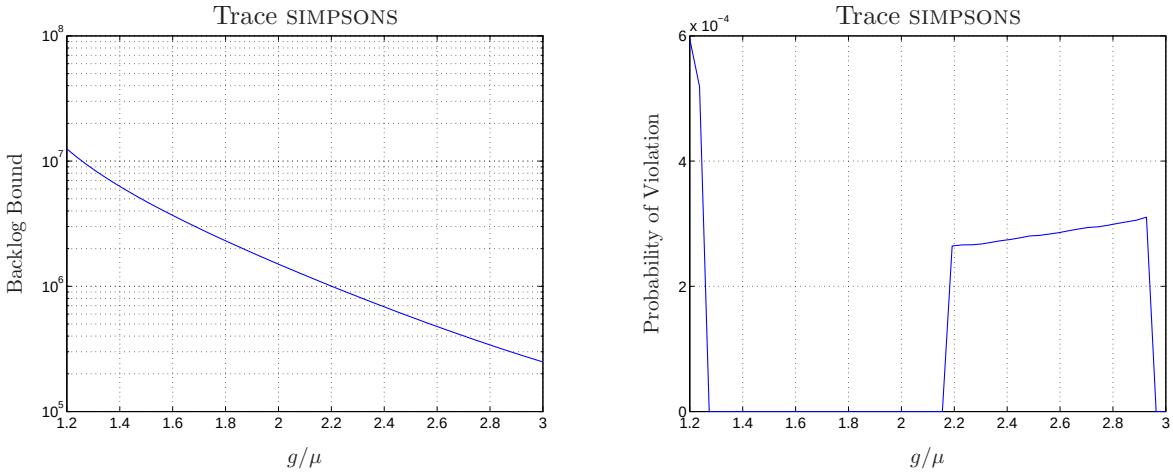

Figure 105: Backlog bound and probability of violation for the trace SIMPSONS.

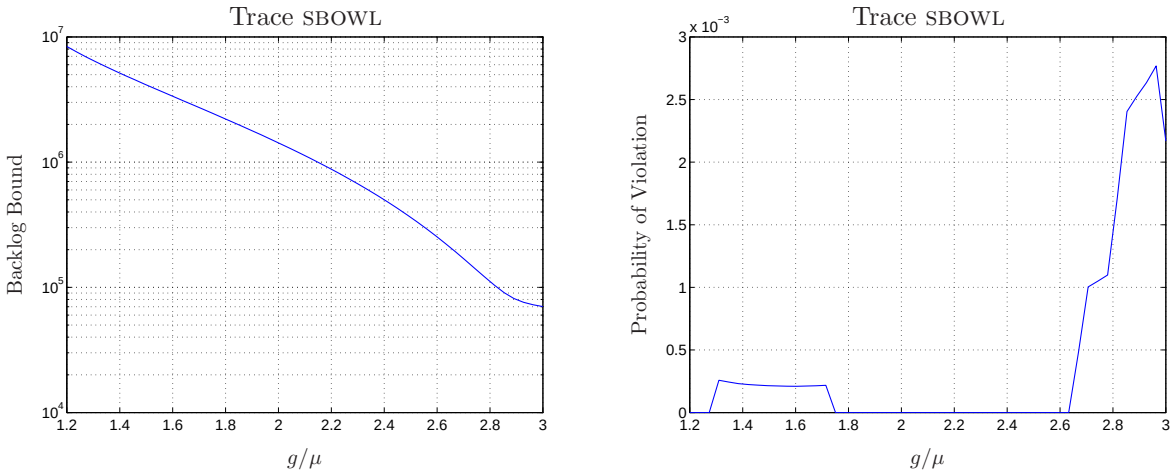

Figure 106: Backlog bound and probability of violation for the trace SBOWL.

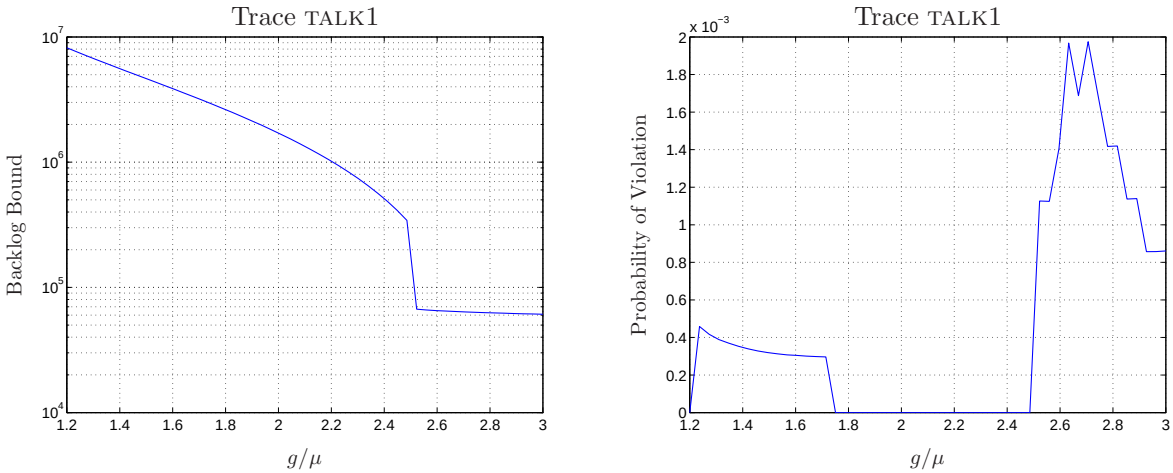

Figure 107: Backlog bound and probability of violation for the trace TALK1.

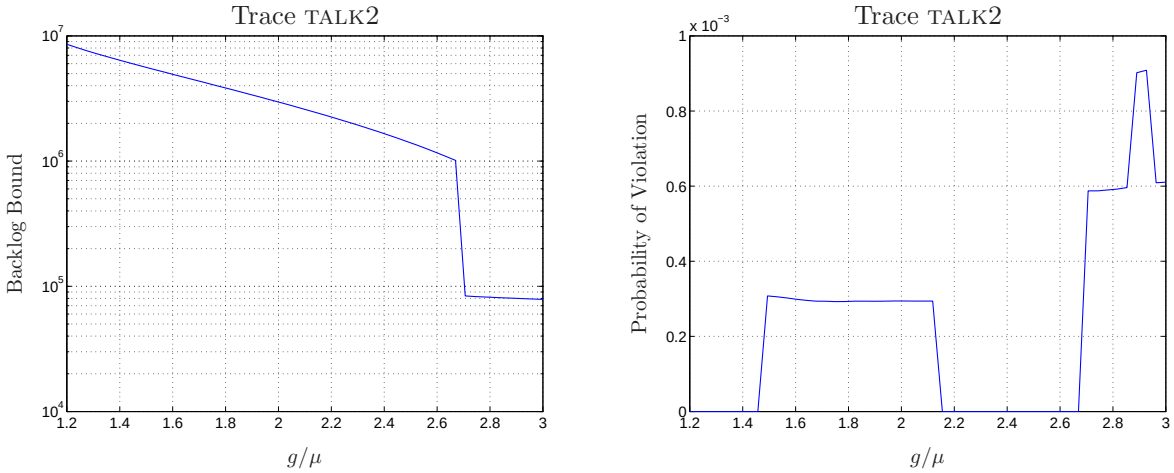

Figure 108: Backlog bound and probability of violation for the trace TALK2.

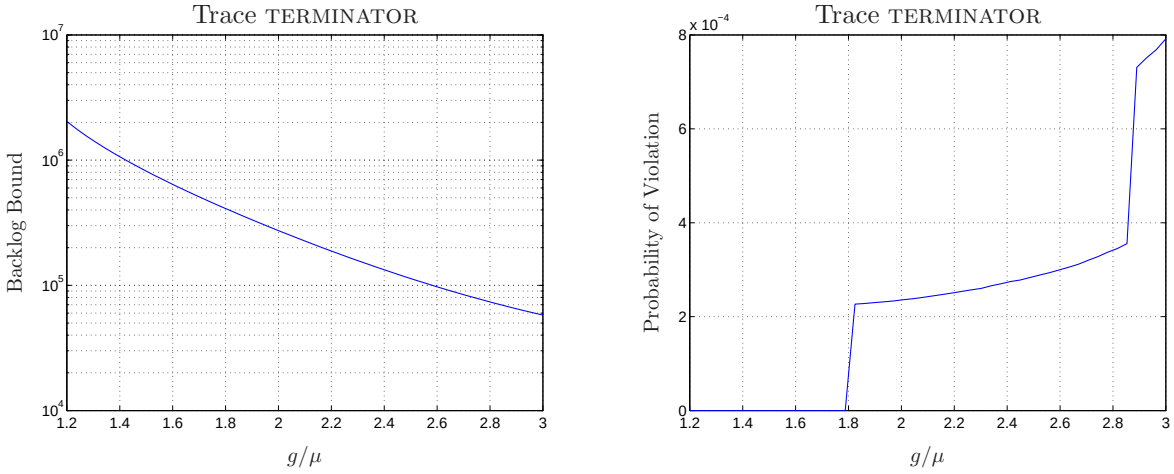

Figure 109: Backlog bound and probability of violation for the trace TERMINATOR.

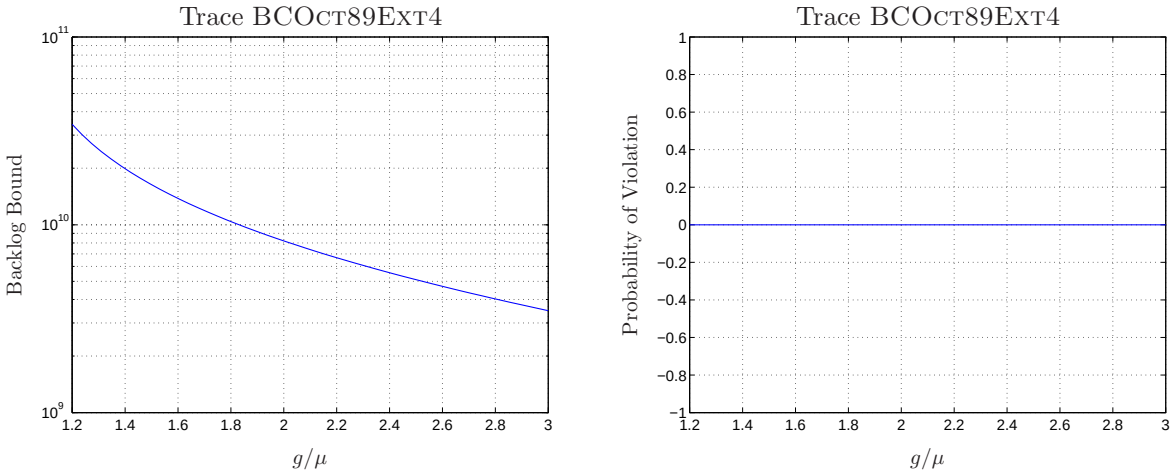

Figure 110: Backlog bound and probability of violation for the trace BCOct89Ext4.

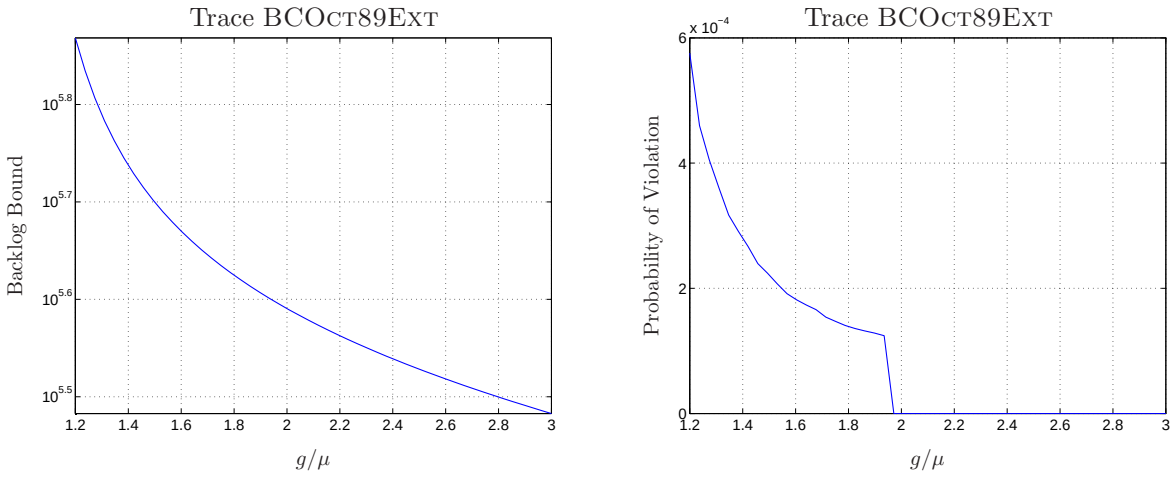

Figure 111: Backlog bound and probability of violation for the trace BCOct89Ext.

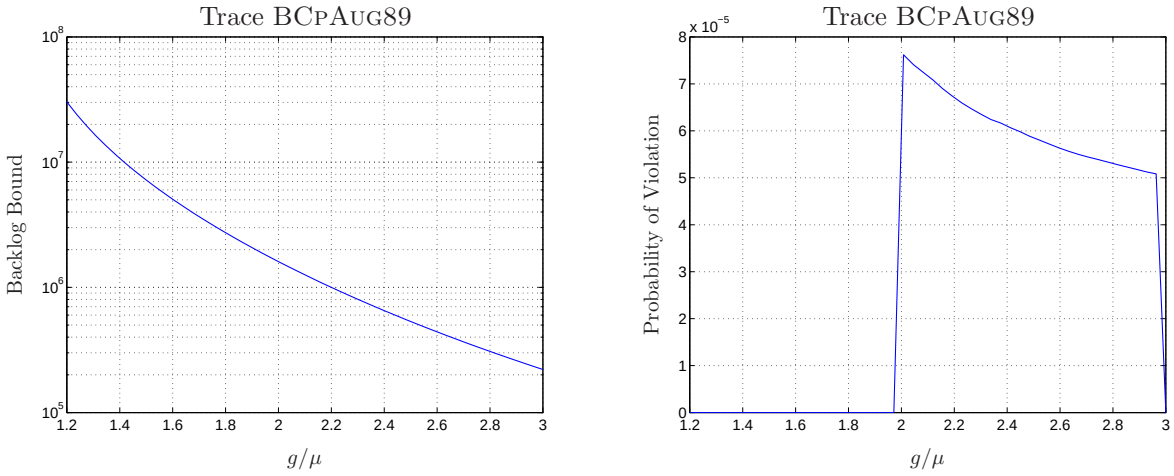

Figure 112: Backlog bound and probability of violation for the trace BCpAug89.

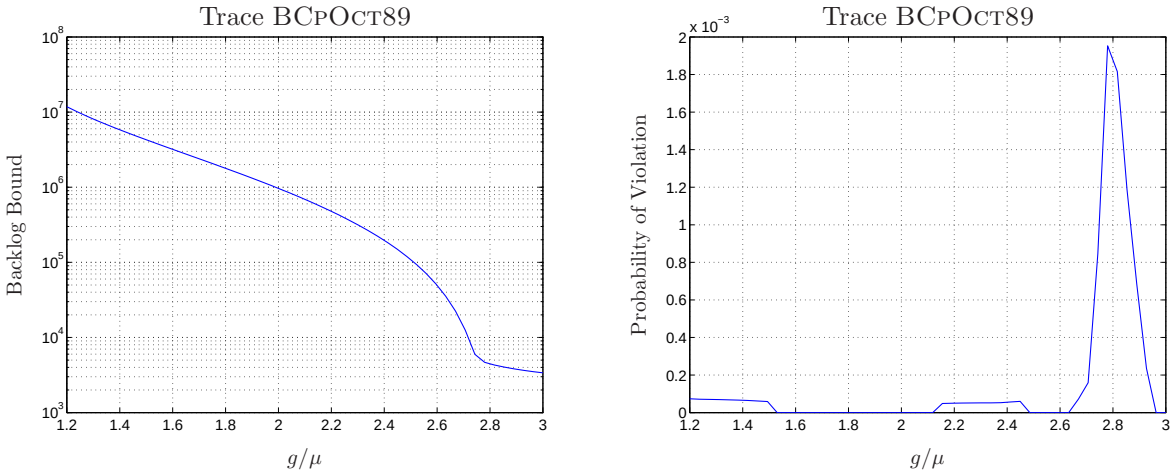

Figure 113: Backlog bound and probability of violation for the trace BCpAug89.

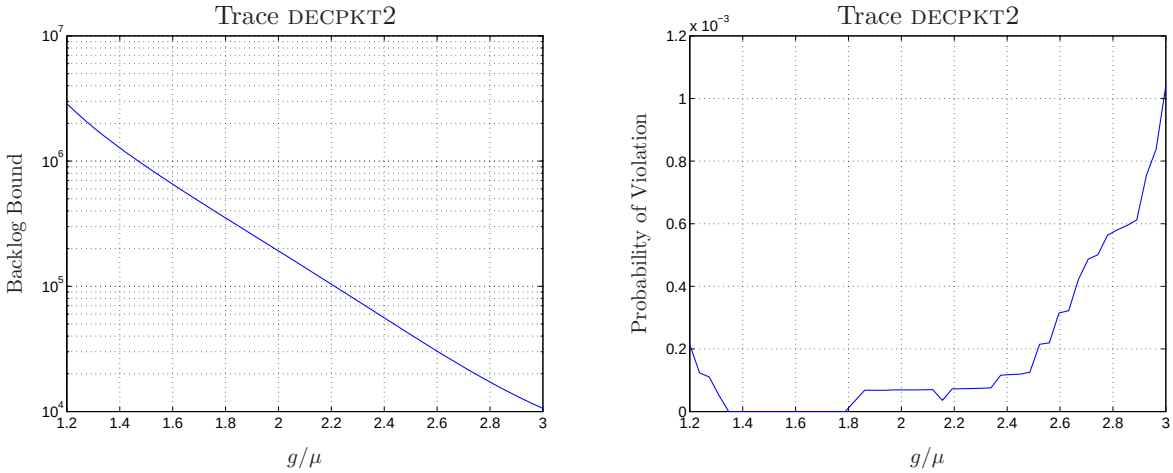

Figure 114: Backlog bound and probability of violation for the trace DECPKT2.

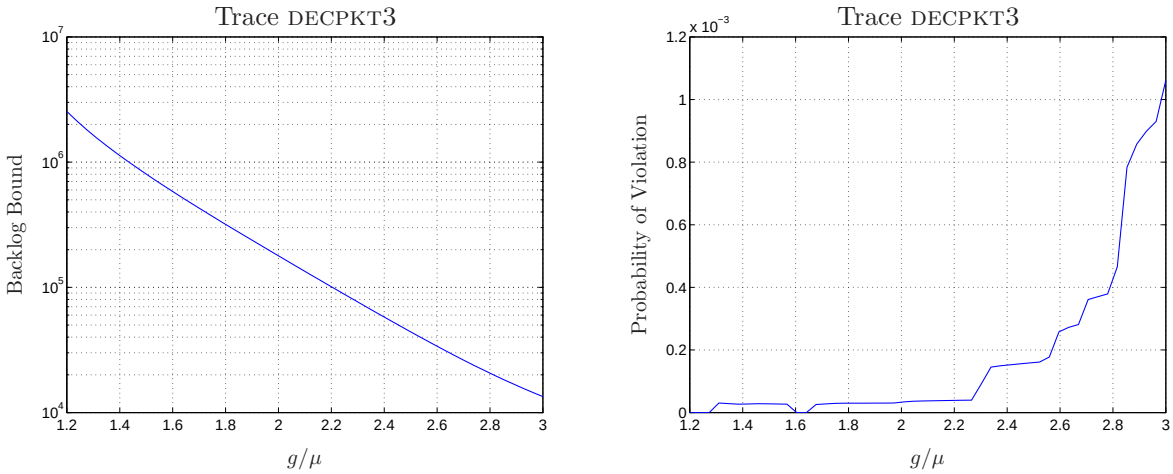

Figure 115: Backlog bound and probability of violation for the trace DECPKT3.

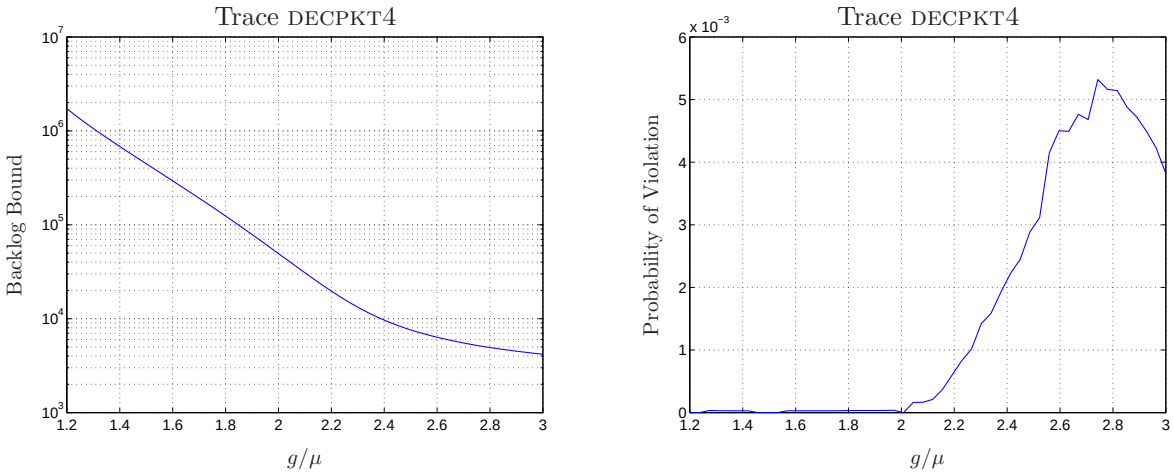

Figure 116: Backlog bound and probability of violation for the trace DECPKT4.

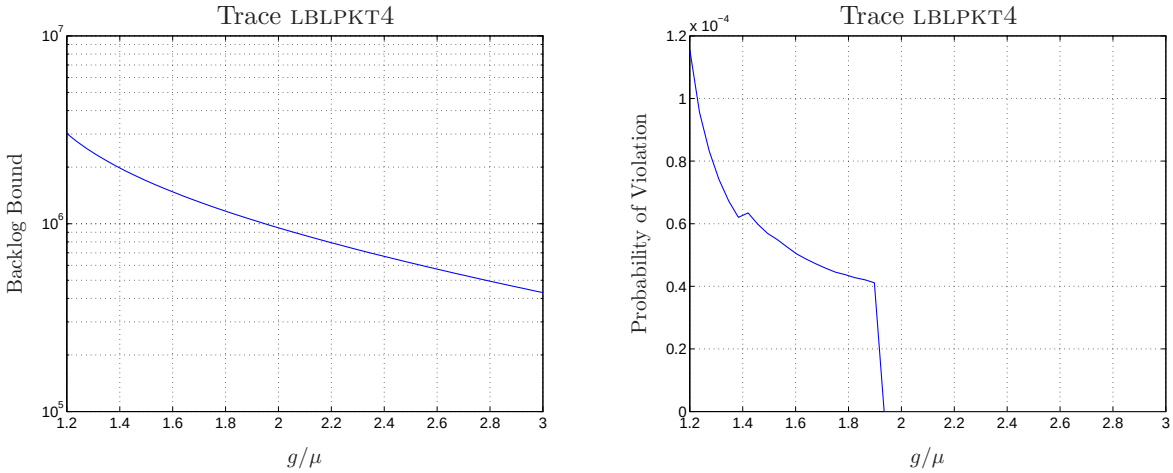

Figure 117: Backlog bound and probability of violation for the trace LBLPKT4.

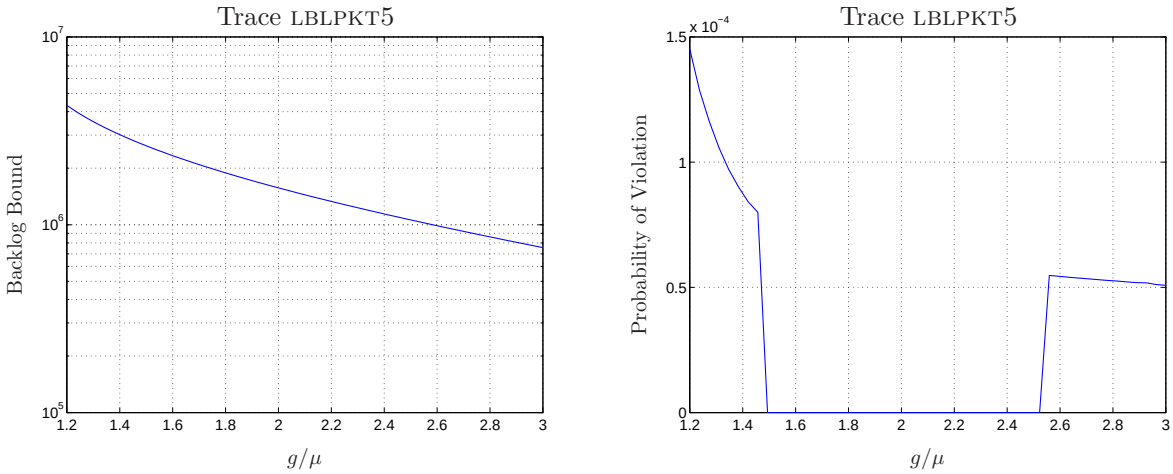

Figure 118: Backlog bound and probability of violation for the trace LBLPKT5.

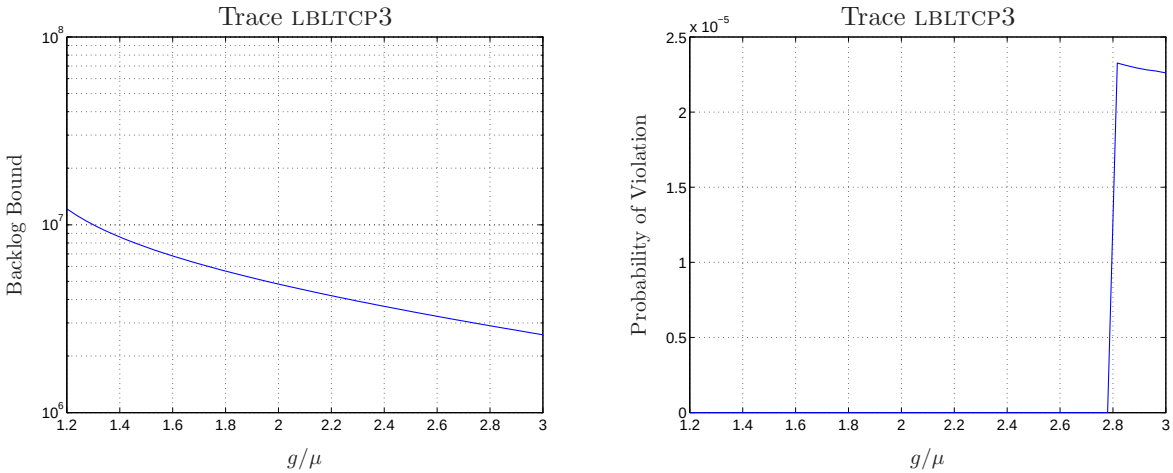

Figure 119: Backlog bound and probability of violation for the trace LBLTCP3.

9 Conclusions

In this paper, an envelope process called Fractional Bounded Arrival Process (FBAP) was proposed for self-similar traffic representation. The queueing analysis for the FBAP traffic was developed, and upper bounds for the maximum backlog, for the maximum delay and for the size of the busy cycles were obtained. Moreover, it was shown why similar results cannot be assured for cumulative envelope processes like the fBm envelope process and the Fractal Leaky Bucket constrained traffic.

The policing of FBAP traffic using the traditional Leaky Bucket algorithm was also investigated. Such approach is specially interesting for QoS-oriented networks, since several results on network performance analysis that assumes the traffic to be Leaky Bucket constrained are available. Moreover, the Leaky Bucket algorithm has been widely implemented in network switches, due to its robustness and low computational cost. The Leaky Bucket algorithm was proved to be more effective than the Fractal Leaky Bucket algorithm for policing FBAP traffic, specially in terms of capacity of supporting the provision of backlog and delay bounds.

Finally, results were extended for the multifractal traffic case. An envelope process called Multifractal Bounded Arrival Process was proposed for representing such a kind of traffic. Moreover, comments on a queueing analysis and on the policing of MFBAP traffic were outlined.

Acknowledgements

This work is sponsored by FAPESP (Process no. 01/14379-4), by CNPq (Process no. 300064/95-0) and by Ericsson Research (contract UNI-35/19-2000).

References

[1] W. E. Leland and D. V. Wilson, "High time-resolution measurement and analysis of LAN traffic: Implications for lan interconnection.," in *Proc. IEEE/INFOCOM*, pp. 1360–1366, apr 1991.

[2] W. E. Leland, W. Willinger, M. S. Taqqu, and D. Wilson, "On the self-similar nature of ethernet traffic (extended version)," *IEEE/ACM Trans. Networking*, vol. 2, pp. 1–15, Feb. 1994.

[3] M. W. Garret and W. Willinger, "Analysis, modeling and generation of self-similar VBR video traffic," in *Proc. ACM/SIGCOMM*, pp. 269–280, 1994.

[4] V. Paxson and S. Floyd, "Wide area traffic: The failure of poisson modeling," *IEEE Trans. Networking*, vol. 3, pp. 226–244, June 1995.

[5] O. Rose, "Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems," Tech. Rep. 101, Institute of Computer Sciences, University of Würzburg, 1995.

[6] M. Taqqu, V. Teverovsky, and W. Willinger, "Is network traffic self-similar or multifractal ?," *Fractals*, vol. 5, pp. 63–73, 1997.

[7] K. Park, G. Kim, and M. Crovella, "On the relation between file sizes, transport protocols and self-similar network traffic," in *Proc. IEEE Int'l. Conf. Network Protocols*, pp. 171–180, Oct. 1996.

[8] O. Rose, "Simple and efficient models for variable bit rate MPEG video traffic," *Perf. Eval.*, vol. 30, pp. 69–85, 1997.

[9] 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," *IEEE/ACM Trans. Networking*, vol. 5, pp. 71–86, Feb. 1997.

[10] M. E. Crovella, "Self-similarity in WWW traffic: evidence and possible causes," *IEEE Trans. Networking*, vol. 5, pp. 835–845, Dec. 1997.

[11] R. H. Riedi and J. L. Véhel, "TCP traffic is multifractal: a numerical study," Research Report 3129, INRIA, Mar. 1997.

[12] A. Feldmann, A. C. Gilbert, and W. Willinger, "Data networks as cascades: Investigating the multifractal nature of internet WAN traffic," in *Proc. SIGCOMM*, (Vancouver, Canada), pp. 42–55, Sept. 1998.

[13] K. Park and W. Willinger, eds., *Self-similar network traffic and performance evaluation.* New York: Wiley, 2000.

[14] I. Norros, "A storage model with self-similar input," *Queueing Syst.*, vol. 16, pp. 387–396, 1994.

[15] N. G. Duffield and N. O'Connell, "Large deviation and overflow probabilities for the general single-server queue, with applications," *Proc. Cambridge Philosophical Society*, no. 118, pp. 363–374, 1995.

[16] P. R. Morin, *The impact of self-similarity on network performance analysis.* Ph.d. dissertation, Carleton University, Dec. 1995.

[17] M. Montgomery and G. de Veciana, "On the relevance of time scales in performance oriented traffic characterizatons," in *Proc. IEEE/INFOCOM*, pp. 513–520, 1996.

[18] D. Heymand and T. Lakshmon, "What are the implications of long range dependence for VBR video traffic engineering ?," *IEEE/ACM Trans. Networking*, vol. 4, pp. 301–317, June 1996.

[19] B. Ryu and A. Elwalid, "The importance of long-range dependence of VBR video traffic in ATM traffic engineering: myths and realities," in *Proc. ACM/SIGCOMM*, 1996.

[20] G. S. Mayor and J. A. Silvester, "Time scale analysis of an ATM queueing system with long-range dependent traffic," in *Proc. IEEE/INFOCOM*, pp. 205–212, 1997.

[21] K. Park, G. Kim, and M. Crovella, "On the effect of traffic self-similarity on network performance," in *Proc. SPIE Int'l. Conf. Perf. and Control of Network Sys.*, pp. 296–310, 1997.

[22] G. Mayor and J. A. Silvester, "Providing QoS for long-range dependent traffic," in *The 7th IEEE Computer-Aided Modeling, Analysis and Design of Communications Link and Networks*, pp. 19–28, 1998.

[23] M. M. Krunz and A. M. Ranasamy, "The correlation structure for a class of scene-based video models and its impacts on the dimensioning of video buffers," *IEEE Trans. Multimedia*, vol. 2, pp. 27–36, Mar. 2000.

[24] S. Molnár, T. D. Dang, and I. Maricza, "On the queue tail asymptotics for general multifractal traffic," in *Proc. Networking*, (Pisa, Italy), 2002.

[25] A. Karasaridis and D. Hatzinakos, "A non-gaussian self-similar process for broadband heavy-traffic modeling," in *Proc. IEEE/GLOBECOM*, (Sydney, Australia), 1998.

[26] B. Mandelbrot, A. Fisher, and L. Calvet, "A multifractal model of asset returns," Cowles Foundation Discussion Paper 1164, Yale University, Sept. 1997.

[27] I. Norros, "On the use of fractional brownian motion in the theory of connectionless networks," *IEEE J. Select. Areas Commun.*, vol. 13, pp. 953–962, Aug. 1995.

[28] A. Karasaridis and D. Hatzinakos, "Broadband heavy-traffic modeling using stable self-similar processes," in *2nd Canadian Conference on Broadband Research*, (Ottawa, Canada), pp. 157–168, June 1998.

[29] F. Harmantzis, D. Hatzinakos, and I. Katzela, "Tail probabilities for the multiplexing of fractional α-stable broadband traffic," in *Proc. IEEE/ICC*, (Helsinki, Finland), 2001.

[30] T. Mikosch, S. Resnick, H. Rootzén, and A. Stegeman, "Is network traffic approximated by stable Lévy motion or fractional Brownian motion ?," Tech. Rep. 1247, Cornell University, 1999.

[31] N. L. S. Fonseca, G. S. Mayor, and C. A. Viana Neto, "On the equivalent bandwidth of self-similar sources," *ACM Trans. on Modeling and Computer Simulation*, vol. 10, pp. 104–124, Apr. 2000.

[32] R. Cruz, "A calculus for network delay, part I : Network elements in isolation," *IEEE Trans. Inform. Theory*, vol. 37, pp. 114–131, Jan. 1991.

[33] R. Cruz, "A calculus for network delay, part II : Network analysis," *IEEE Trans. Inform. Theory*, vol. 37, no. 1, pp. 132–141, 1991.

[34] M. Taqqu, V. Teverovsky, and W. Willinger, "Estimators for long-range dependence: an empirical study," *Fractals*, vol. 3, no. 4, pp. 785–798, 1995.

[35] P. Abry and D. Veitch, "Wavelet analysis of long-range dependent traffic," *IEEE Trans. Inform. Theory*, vol. 44, pp. 2–15, Jan. 1998.

[36] M. Chi, E. Neal, and G. Young, "Practical applications of fractional brownian motion and noise to synthetic hydrology," *Water Resources Research*, vol. 9, pp. 1523–1533, Dec. 1973.

[37] "The internet traffic archive." URL: http://ita.ee.lbl.gov/.

[38] V. Paxson, "Fast approximation of self-similar network traffic," Tech. Rep. LBL-36750, Lawrence Berkeley Laboratories, 1995.

[39] N. L. S. Fonseca, F. M. Pereira, and D. S. Arantes, "On the performance of Generalized Processor Sharing under long-range dependent traffic," in *Proc. IEEE/Globecom*, (Taipei, Taiwan, R.O.C.), Dec. 2002.

[40] F. M. Pereira, N. L. S. Fonseca, and D. S. Arantes, "On the performance of Generalized Processor Sharing servers under long-range dependent traffic," *Computer Networks*, no. 40, pp. 413–431, 2002.