THÈSE

en vue de l'obtention du grade de

Docteur de l'Université de Lyon, délivré par l'École Normale Supérieure de Lyon

Discipline: Informatique

Laboratoire de l'Informatique du Parallélisme

L'École Doctorale Informatique et Mathématiques (ED 512)

présentée et soutenue publiquement le juin 18, 2014

par M. Shubhabrata ROY

Titre: A Complete Framework for Modelling Workload Volatility of a VoD System: a Perspective to Probabilistic Management

---

Directeur de thèse: M. Paulo GONÇALVES
Co-Encadrant: M. Thomas BEGIN

Après avis de: M. Thierry TURLETTI
Mme. Sandrine VATON

Devant la commission d'examen formée de :
M. Thierry TURLETTI (Rapporteur)
Mme. Sandrine VATON (Rapporteur)
M. Fréderic DESPREZ (Examinateur)
M. Marcelo DIAS DE AMORIM (Examinateur)
Mme. Susana SARGENTO (Examinateur)
M. Paulo GONÇALVES (Directeur)
M. Thomas BEGIN (Co-Encadrant)

Acknowledgements

First of all I would like to thank my advisors for guiding me all these years and never loosing faith on me even when I was going through a tough period. I am also grateful to my wife, who has been a great support throughout that period, tolerating my tantrums. However, I believe she too some extent enjoyed the roller-coaster ride. Thanks to my parents for preparing by basics that helped me to reach this stage. A very special thank goes to my in-laws for motivating me and keep asking about my "research work". Finally I would like to dedicate this work to "Dadu" and both "Dadubhai-s" who had strong faith on the value of education.

Contents

1 Preamble **13**

2 State of the art **18**

2.1 A survey of existing Workload Models 18

2.2 A brief survey of Model Calibration 21

2.3 A survey of Resource Management (RM) approaches 24

2.3.1 Methods to facilitate resource management 26

2.3.1.1 Probabilistic Provisioning 26

2.3.1.2 Virtual Machine (VM) 27

2.3.1.3 Gossip 28

2.3.1.4 Auction 29

2.3.2 Issues to optimize for resource management 29

2.3.2.1 Utility Function 29

2.3.2.2 Hardware Resource Dependency 31

2.3.2.3 Service Level Agreements (SLA) 32

2.3.3 Conclusion 32

3 Model Description **34**

3.1 Introduction 34

3.2 A brief discussion of the epidemic models 35

3.2.1 SIR Model 35

3.2.2 SIS Model . . . 38
3.2.3 SEIR Model . . . 38
3.3 Workload Model description . . . 40
3.3.1 Differences between the proposed model and a standard epidemic model . . . 41
3.4 Generated VoD traces from the Model . . . 45
3.5 Addendum: . . . 55
3.5.1 Implementation of the VoD model on distributed environment . . . 55
3.5.2 Global Architecture of the Workload Generating System . . . 56
3.5.3 Implementation Issues . . . 57
3.6 Results from the distributed implementation . . . 58
3.7 Conclusion . . . 58

4 Estimation Framework **60**
4.1 Model Parameter estimation: a heuristic approach . . . 61
4.1.1 Introduction . . . 61
4.1.2 Heuristic procedure description . . . 63
4.1.2.1 Watching parameter γ estimation . . . 63
4.1.2.2 Memory parameter μ estimation . . . 65
4.1.2.3 Propagation parameters β and l estimation . . . 68
4.1.2.4 Transition rates a_1 and a_2 estimation . . . 71
4.1.3 Results . . . 72
4.2 Model Parameter estimation: an MCMC approach . . . 83
4.2.1 A brief introduction to Markov Chain Monte Carlo . . . 83
4.2.2 Calibration framework using MCMC . . . 85
4.2.2.1 Step I: Identification of buzz and buzz-free regime and estimation of a_1 and a_2 . . . 86

4.2.2.2 Step II: Estimation of $\beta_1, \beta_2, \mu, \gamma, l$ . . . 88
4.2.2.2.1 Substep II.1: Estimation of $\hat{\mu}$ and $\widehat{\mathbf{t_s}}$ . . . 89
4.2.2.2.2 **Substep II.2: Estimation of $\hat{\beta}_1$, $\hat{l}$** . . . 90
4.2.3 Results . . . 92
4.2.4 Discussion . . . 94
4.3 Data-model adequacy of the calibrated model . . . 102
4.3.1 Validation Against an Academic VoD Server: . . . 102
4.3.2 Validation Against World Cup 1998 workload . . . 106
4.4 Conclusion . . . 109

5 Resource Management **111**
5.1 Introduction . . . 111
5.2 Large Deviation Principle . . . 113
5.3 Probabilistic Provisioning Schemes . . . 117
5.3.1 Identification of the reactive time scale for reconfiguration . . . 119
5.3.2 Link capacity dimensioning . . . 121
5.4 Conclusion . . . 122

6 Conclusion **125**
6.1 Main contributions . . . 125
6.2 Challenges . . . 126
6.3 Originality and limitation of the work . . . 126
6.4 Future works . . . 127

A Proofs of Chapter. 4 **128**

B Algorithm of Chapter. 5 **131**

List of Figures

3.1 Flow diagram of the SIR epidemic model. 36
3.2 Evolution of S, I and R classes with time for a SIR epidemic. For the sake of generalization parameters of the three cases are not quantified. 37
3.3 Flow diagram of the SIS epidemic model. 38
3.4 Evolution of S and I classes with time for a SIS epidemic model. For the sake of generalization parameters of the three cases are not quantified. 39
3.5 Flow diagram of the SEIR epidemic model. 39
3.6 Markov Chain representing the possible transitions of the number of current (i) and past active (r) viewers. 43
3.7 Traces generated from the parameters, reported in Table. 3.1. The horizontal axis represents time (in hours) and the vertical axis represents VoD workload (i.e. the number of viewers). 47
3.8 Steady state distribution of the traces generated from the parameters, reported in Table. 3.1. 50
3.9 Empirical autocorrelation function of the traces generated from the parameters, reported in Table. 3.1. 52
3.10 Empirical Large Deviation Spectrum of the traces generated from the parameters, reported in Table. 3.1. 54
3.11 Topology of Grid'5000 55
3.12 Architecture and interactions between the nodes to replicate user behavior. . . . 56
3.13 Snap shot of the real-time server workload from the monitoring computer. . . . 59

4.1 Schematics showing the flow order in which the parameters are estimated from an input trace. 61

4.2 Influence of $\mathbf{t_a}$, $\mathbf{t_p}$ and $\mathbf{t_s}$ on the evolution of the current (I) and the past viewers (R). 62

4.3 The vertical axis represents the K-S distance and the horizontal axis represents the μ values. The red circles in the plots represent the estimated μ and the intersections of the curves and the green line represents the actual value of μ (which are 5×10^{-4}, 0.2, 0.01, 3.0 and 0.002 respectively). 67

4.4 Evolution of the number of past viewers (vertical axis) vs. time in hrs (horizontal axis). 68

4.5 Linear regression of $(\Omega(x))^{-1}$ against x to obtain β_1 and l. 70

4.6 An example to estimate β_2 using the ML estimator. 71

4.7 A sample box plot to interpret the descriptive statistics 72

4.8 Relative precision of estimation of the model parameters. Cases I to V correspond to the configurations reported in Chapter 3. Statistics are computed over 50 independent realizations of time series of length 2^{21} points. 75

4.9 Evolution of the Variance and Bias for β_1 *against* the data length N in a *log-log* plot for the 5 traces for the heuristic procedure. 76

4.10 Evolution of the Variance and Bias for β_2 *against* the data length N in a *log-log* plot for the 5 traces for the heuristic procedure. 77

4.11 Evolution of the Variance and Bias for γ *against* the data length N in a *log-log* plot for the 5 traces for the heuristic procedure. 78

4.12 Evolution of the Variance and Bias for μ *against* the data length N in a *log-log* plot for the 5 traces for the heuristic procedure. 79

4.13 Evolution of the Variance and Bias for l *against* the data length N in a *log-log* plot for the 5 traces for the heuristic procedure. 80

4.14 Evolution of the Variance and Bias for a_1 *against* the data length N in a *log-log* plot for the 5 traces for the heuristic procedure. 81

4.15 Evolution of the Variance and Bias for a_2 *against* the data length N in a *log-log* plot for the 5 traces for the heuristic procedure. 82

4.16 Flow chart of the overall estimation procedure 85

4.17 Flow chart describing step II of the estimation procedure 88

4.18 Relative precision of estimation of the model parameters for all 5 cases. Statistics are computed over 50 independent realizations of time series of length 2^{21} points 93

4.19 Evolution of the Variance and Bias for β_1 *against* the data length N in a *log-log* plot for the 5 traces for the MCMC procedure. 95

4.20 Evolution of the Variance and Bias for β_2 *against* the data length N in a *log-log* plot for the 5 traces for the MCMC procedure. 96

4.21 Evolution of the Variance and Bias for μ *against* the data length N in a *log-log* plot for the 5 traces for the MCMC procedure. 97

4.22 Evolution of the Variance and Bias for l *against* the data length N in a *log-log* plot for the 5 traces for the MCMC procedure. 98

4.23 Evolution of the Variance and Bias for a_1 *against* the data length N in a *log-log* plot for the 5 traces for the MCMC procedure. 99

4.24 Evolution of the Variance and Bias for a_2 *against* the data length N in a *log-log* plot for the 5 traces for the MCMC procedure. 100

4.25 Convergence plot of five sets of parameters in a semi-log scale. The horizontal axis represents # of iterations and the vertical axis represents the relative error 101

4.26 Modelled workload for Trace I (Left column) and Trace II (Right column). First row corresponds to the real traces; second row to the synthesised traces from the proposed model. Horizontal axes represent time (in hours) and vertical axes represent workload (number of active viewers). 103

4.27 Steady-state distribution of the real trace against the generated trace for GRNET. The horizontal axis represents workload (# of current viewers) 106
4.28 Auto-correlation of the real trace against the generated trace for GRNET. The horizontal axis represents time lag τ (*hours*) 106
4.29 Large Deviation Spectrum of the real trace against the generated trace for GRNET. 107
4.30 Trace of the world cup football (1998) final. Trace I is collected before the match started and Trace II covered the duration of the match. First row corresponds to the real traces; second row to the synthesised traces from the proposed model. Horizontal axes represent time (in hours) and vertical axes represent workload (number of active viewers). 108
4.31 Steady-state distribution of the real trace against the generated trace for WC98 server. The horizontal axis represents workload (# of current viewers) 109
4.32 Auto-correlation plot of the real trace against the generated trace for WC98 server. The horizontal axis represents time lag τ (*secs*) 109
4.33 Large deviation spectrum of the real trace against the generated traces for WC98 server. 110

5.1 Probability distribution of throughput for a homogeneous Poisson process (with rate equal to 1) for different time scales (τ). 113
5.2 Large Deviations spectra corresponding to two traces generated from the proposed model. (a) Theoretical spectra for the buzz free (blue) and for the buzz (red) scenarii. (b) & (c) Empirical estimations of $f(\alpha)$ at different scales from the buzz free and the buzz traces, respectively. 118
5.3 Probability density derived from the LD spectrum 119
5.4 Deviation threshold vs. probability of occurrence of overflow for different values of time scale (τ). 120

5.5 Dimensioning K, the number of hosted servers sharing a fixed capacity link C. The safety margin C_0 is determined according to the probabilistic loss rate negotiated in the *Service Level Agreement* between the infrastructure provider and the VoD service provider. 123

List of Tables

3.1 Parameter values, used to generate the traces plotted in Fig. 3.7 45

3.2 Comparison of $\mathbb{E}(i)$ and Emp. mean $\langle i \rangle$, from the traces plotted in Fig. 3.7 . . . 49

4.1 Estimated Parameters of the VoD model 102

4.2 Mean and standard deviation of real traces and the calibrated models. 102

4.3 Estimated Parameters from the World Cup Traffic Traces 107

Abstract

There are some new challenges in system administration and design to optimize the resource management for a cloud based application. Some applications demand stringent performance requirements (e.g. delay and jitter bounds), while some applications exhibit bursty (volatile) workloads. This thesis proposes an epidemic model inspired (and continuous time Markov Chain based) framework, which can reproduce workload volatility namely the "buzz effects" (when there is a sudden increase of a content popularity) of a Video on Demand (VoD) system. Two estimation procedures (heuristic and a Markov Chain Monte Carlo (MCMC) based approach) have also been proposed in this work to calibrate the model against workload traces. Obtained model parameters from the calibration procedures reveal some interesting property of the model. Based on numerical simulations, precisions of both procedures have been analyzed, which show that both of them perform reasonably. However, the MCMC procedure outperforms the heuristic approach. This thesis also compares the proposed model with other existing models examining the goodness-of-fit of some statistical properties of real workload traces. Finally this work suggests a probabilistic resource provisioning approach based on a Large Deviation Principle (LDP). LDP statistically characterizes the buzz effects that cause extreme workload volatility. This analysis exploits the information obtained using the LDP of the VoD system for defining resource management policies. These policies may be of some interest to all stakeholders in the emerging context of cloud networking.

Contribution of this thesis are the following:

- Dynamic Resource Management in Clouds: A Probabilistic Approach; P. Gonçalves, S. Roy, T. Begin, P. Loiseau, IEICE Transactions on Communications, 2012
- Demonstrating a versatile model for VoD buzz workload in a large scale distributed network; JB. Delavoix, S. Roy, T. Begin, P. Gonçalves, Cloud Networking (IEEE CLOUDNET), 2012
- Un modele de trafic adapté a la volatilité de charge d'un service de vidéo à la demande: Identification, validation et application à la gestion dynamique de ressources; S. Roy, T. Begin, P. Gonçalves, Inria research report, 2012
- A Complete Framework for Modelling and Generating Workload Volatility of a VoD System; S. Roy, T. Begin, P. Gonçalves, IEEE Int. Wireless Communications and Mobile Computing Conference, 2013
- An MCMC Based Procedure for Parameter Estimation of a VoD Workload Model; S. Roy, T. Begin, P. Gonçalves; GRETSI Symposium on Signal and Image Processing, 2013

Chapter 1

Preamble

Cloud Computing is defined as the applications delivered as services over the Internet along with the hardware and systems software at the data-centre providing those services. These services are termed as Software as a Service (SaaS) and the data-centre hardware and software are called as a Cloud. When a Cloud is made accessible in a pay-as-you-go manner to the users, designing a proper architecture for disseminating data depends on the following two facts:

- Information about the content of the data
- How do the users behave when they access the content

Such knowledge helps the system planners to have a deeper understanding of the system requirements while designing it. This includes the right choices of hardware and provisioning resources, such as file systems, storage, network, processing power and page caching algorithms. A well planned architecture can quickly respond to a large number of users and at the same time ensure cost-effectiveness of the system. Thus it avoids over-provisioning which ensures good services, but at the expense of higher deployment costs.

There are some new challenges in system management and design to optimize the resource utilization. Some applications demand stringent performance requirements (like delay and jitter bounds), some applications are customizable by its users (which means that request processing is more complicated being user specified), while some applications exhibit bursty workloads. This thesis concentrates on the last type of applications. Naturally, bursty workloads lead to highly volatile demand in resources. To better understand and to faithfully reproduce this demand volatility requires relevant workload models. By definition workload modeling is an attempt to develop a parsimonious model (with few parameters that can easily be calibrated from the observation) covering a large diversity in users practices. It can be used then to produce synthetic workloads easily, possibly with slight (but tightly controlled) modifications. A poor workload model does not facilitate proper performance evaluation of a system. For example it is not possible to model different types of applications as TCP- friendly, which requires each flow to adapt to a rate similar to the competing TCP flows. An application can generate traffic flows, where both inelastic and elastic flows might co-exist. Therefore, workload characterization and modeling is an important aspect while designing an architecture for large scale content distribution networks (CDN). Statistical knowledge of the user requests enables the system architect to dimension the system accordingly. This is specially important in hosting centres where several independent services share the resources. In shared systems, knowledge of access patterns of the hosted services can be useful to facilitate efficient resource usage while avoiding system overload because of aggregated traffic.

Another important aspect of workload modelling is benchmarking, which is defined as evaluating the performance of different systems under equivalent conditions, in particular with the same workload. Especially, networking research gets facilitated by the adoption of a set of workload traffic benchmarks to define network applications for empirical evaluations. This drives to select a set of workloads which are ported to different systems

and then used as a basis for comparison.

Past research on traffic workload modelling has yielded significant results for various types of applications such as Web, P2P or Video streaming [21] [78]. In all these cases, the developed traffic models have served as valuable inputs to assess the efficiency of adapted management techniques. This thesis considers a Video on Demand (VoD) system as a paradigm of applications subject to highly variable demand and elaborates a complete modelling framework able to reproduce similar bursty workload.

A VoD service delivers video contents to consumers on request. According to Internet usage trends, users are increasingly getting more involved in the VoD and this enthusiasm is likely to grow. According to [64] a popular VoD provider like Netflix [54] alone represents 28% of all and 33% of peak downstream Internet traffic on fixed access links in North America, with further rapid growth expected. IT giants like Apple, Adobe, Akamai and Microsoft are also emerging as competitive VoD providers in this challenging, yet lucrative market. Since VoD has stringent streaming rate requirements, each VoD provider needs to reserve a sufficient amount of server outgoing bandwidth to sustain continuous media delivery (not considering IP multicast). However, resource reservation is very challenging when a video becomes popular very quickly (i.e. buzz) and yields a *flood* of user requests on the VoD servers. To help the providers anticipating these situations, constructive models are sensible approaches to capture and to get a better insight into the mechanisms that underlie the applications. The goal of the model is then to reproduce, under controlled and reproducible conditions, the behaviour of real systems and to generate workloads. These workloads can eventually be used to evaluate the performance of management policies.

The first part of this thesis identifies the properties which describe user behaviors. Naturally, the collective behavior of the users govern the mechanism of the VoD workload generation. The user behavior is modelled such that it satisfies some mathematical properties. This model is inspired by disease propagation in epidemic systems where

Markovian approach is widely used and satisfy certain properties. This work deals with analysis of these properties to provide some insights on user behavior (based on the model parameters). The workloads generated by the model demonstrate its versatility to produce traffics with different profiles.

The second aspect of this thesis deals with calibration of the workload model. Naturally a model without a procedure to identify its parameters is difficult to exploit. In this work first a calibration procedure has been developed based on Ad-hoc (or heuristic) procedure. The procedure seems to perform satisfactory. However, it had been felt that a more systematic approach can benefit the model calibration and enhance its usability. Therefore, a Markov Chain Monte Carlo (MCMC) based procedure has been proposed. It has been found in the literature that there has been numerous instances of employing MCMC to identify model parameters. This chapter discusses pros and cons of both approaches and use them to verify data model adequacy of the model against real workload traces.

The third and the final part of the thesis deals with resource management. Resource Management is a core issue of the Cloud Computing regime, which utilizes large-scale virtualized systems to provision rapid and cost-effective services. To manage such large volume of resources, it heavily requires automation and dynamic resource management. But, there seems to be a need of vast research to manage virtualized resources in such an unprecedented scale. In this work a resource management framework has been proposed based on the workload model. As mentioned above during workload modeling, a model has been developed which satisfies some particular properties, known as the Large Deviation Principle (LDP). This work leverages these properties to derive a probabilistic assumption on the mean workload of the system at different time resolutions. This work proposes two possible and generic ways to exploit these information in the context of probabilistic resource provisioning. They can serve as the input of resource management functionalities of the Cloud environment. The proposed probabilistic approach is very

generic and can adapt to address any provisioning issues, provided the resource volatility can be resiliently represented by a stochastic process.

To sum up, the contribution of this thesis comprises:

(i) construction of an epidemic-inspired model adapted to VoD mechanisms

(ii) two estimation procedures to calibrate the model against a workload trace

(iii) resource management policies (exploiting the workload model) for better provisioning the system.

The thesis has been organized as follows.

Chapter 2 presents the state of the art regarding workload models, calibration procedure and the resource management. Chapter 3 introduces the model and provides an explicit description. Estimation procedures of the model parameters from a workload trace have been described in Chapter 4. This chapter contains three sub chapters. Chapter 4.1 describes parameter estimation in a heuristic approach, followed by the MCMC approach in Chapter 4.2. Next data model adequacy checking and comparison of two approaches, discussed in Chapter 4.1 and 4.2 are made in Chapter 4.3. Chapter 5 introduces and explains the Large Deviation Principle in context of the proposed workload model, followed by the resource management policies developed from it. Finally Chapter 6 concludes this work with several future research directions.

Chapter 2

State of the art

- State of the art of the relevant workload models
- Discussion of model calibration procedures
- State of the art of the resource management policies

2.1 A survey of existing Workload Models

Performance evaluation is a key element which is used to assess designs when building new systems, to calibrate parameter values of existing systems, and to evaluate capacity requirements when setting up systems for real world deployment. Lack of satisfactory performance evaluation can lead to poor decisions, which follows either not being able to accomplish mission objectives or inefficient usage of resources. A good evaluation study, on the contrary, can be instrumental in designing and implementing an efficient and useful system. It is imperative that the composition of the workload being processed is one of the main criterion in performance evaluation. Hence its quantitative description is a fundamental part of all performance evaluation studies. This section presents some of the existing workload models in context.

This study classifies workloads in two major categories

- Workload modelled for the centralized system
- Workload modelled for network based system

Workload modelling and characterization for centralized systems is a well researched topic since early seventies. One subcategory of this study includes batch and interactive systems. A popular method used for workload characterization of batch and interactive systems is the clustering. An early application of clustering can be found in [1] where this technique is used to construct a workload model of a dual processors system for using in a scientific environment.

In [23] the authors analyze interactive workloads in terms of a stochastic model. According to this analysis a user session is a sequence of jobs that consist of sequences of tasks. Each task can then be considered as a sequence of commands. At the next level there are the physical resources consumed by each command. Here the workload is the set of all the jobs. This is initially grouped by means of clustering techniques into seven sub-workloads. The parameters characterizing each job are functions of the software resources. At the task level, a Markov chain, whose states correspond to the software resources used by the job, is employed to describe users behavior.

Another sub-category of the centralized system is a database system. It can be seen as a part of a centralized system where the users interactively access the database from their terminals. The workload description in studies dealing with these types of system depends on the analysis of traces measured on real environments or synthesized according to some stochastic processes. VoD workload modelling closely relates to this category.

A study of the history of VoD modeling shows several changes of paradigms and platforms. An early work in this domain include [11] which studies a reference network architecture for video information retrieval. However, this wok focuses on the bursty workload generated by a VoD system, which has been an active area of research with

different approaches. Since the user behavior is directly related to workload generation in a system, the authors of [41], [51] and [31] develop a user activity model to reproduce the workload generated by an user. In a different vein researchers [58] [22] aim to model the aggregated workload generated by multiple users. This thesis discusses some basic as well as advanced models which address workload volatility of a VoD or similar systems in the next paragraphs.

Authors of [57] proposed a maximum likelihood method for fitting a Markov Arrival Process (MAP), a generalization of the Poisson process by having non-exponentially distributed (yet phase type) inter-arrival times, to the web traffic measurements. This approach is useful to describe the time-varying characteristics of workloads and seems to achieve reasonable accuracy in models to fit web server traces in terms of inter-arrival times and tail heaviness. However, the authors do not aim to model bursty workloads in this work. With a focus on buzz arrival modelling, the authors of [58] and [22] proposed a two-state $MMPP$ (a special case of MAP) based approach and a parameter estimation procedure using the index of dispersion. But Chapter 4 will demonstrate that the $MMPP$ model seems to include only very short memory and may not be suitable to represent a real VoD workload. Moreover, the model parameters of both MAP and $MMPP$ are not comprehensive to draw inference about the system dynamics. A parsimonious model like Lévy is also a tempting approach. Thanks to its inherent "α-stable" process, this process is suitable to model system volatilities. But it develops a long range memory (long-term correlation) which does not seem to match the dynamic feature of the real VoD traces.

In a distinct approach, impact of workload on server resources have been thoroughly studied in many works. Authors of [2], [53], [14], [13] and [17] provided detailed workload characterization study over web servers. Their works provide a statistical analysis of server workloads in context of usage pattern, caching, pre-fetching, or content distribution. They conclude that the lack of an efficient, supported and cache consistency

mechanism can be the main reason why web caches fail significantly to manage server workload during times of extreme user interest (buzz) in the contents on those servers. Clearly, workload modelling is not the basic objective of these authors.

Workload generators are also used to evaluate computer systems or Web sites. Some of the popular workload generators in this regards include [16], [24], [30] or [5]. However, none of them aims at reproducing satisfactory burstiness in the workload.

Since information propagation in a social network resembles infection propagation in a human network, some researchers develop workload models based on epidemiology. One such example is [39], where the authors propose a simple and intuitive epidemic model, requiring a single parameter to model information dissemination among users. But [10] shows that a simple epidemic model, as described in [39] can not reproduce main properties of real-world information spreading phenomena. It requires further enhancement. In chapter. 3 a workload model is proposed taking inspiration from a simple epidemic model. It embeds certain modification which make it capable to generate realistic VoD workload.

2.2 A brief survey of Model Calibration[1]

In the organization of this thesis, workload model description is succeeded by model calibration. Model calibration deals with estimation of parameters based on empirical data. It is an old and much researched field in statistics. Since model calibration demands a very special attention in this thesis, some of the procedures are discussed briefly. The followings are a few of some very well known estimation procedures:

- Maximum likelihood estimators (MLE)
- Markov chain Monte Carlo (MCMC) based estimator

[1] **DISCLAIMER:** This is a very brief survey of the model calibration procedures. It is no way an exhaustive one. Intention of including this section in this work is to briefly introduce some of the basic calibration procedures which have been used in this work or have relevance to the workload model.

- Method of moments estimators
- Minimum mean squared error (MMSE) based estimator
- Maximum a posteriori (MAP) estimator

This thesis develops two estimation frameworks based on the observed data. In that context this section discusses those procedures which have been used extensively in the both estimation frameworks.

The first procedure, that has been extensively used in model calibration (both frameworks) is the maximum likelihood estimator. This method selects the set of values of the model parameters that maximizes a previously defined likelihood function, for an observation dataset and underlying statistical model. Intuitively, this approach maximizes the "agreement" between the selected model and the observed data. For discrete random variables it maximizes the probability of the observed data under the resulting distribution. This work uses this estimator to estimate the model parameters, which depend only on the observable datasets. It is also worth mentioning that a least square fitting is also be an MLE, given it validates certain condition (Gauss-Markov assumptions hold true and estimation noise are normally distributed). This work uses this type of MLE to estimate propagation parameter of the model. Even though an MLE is an asymptotically efficient estimator it fails to provide a solution while dealing with unobservable or missing data. Moreover, an MLE solution of a complicated problem is extremely difficult to formulate.

The second framework uses an Markov chain Monte Carlo (MCMC) based estimator. It is a sophisticated method, based on the Law of Large Numbers for Markov chains. An MCMC estimator samples from probability distributions based on constructing a Markov chain. The state of the chain after a large number of steps is then used as a sample of the target distribution of interest. The quality of the sample enhances as a function of the number of steps. MCMC methods are commonly used to estimate parameters of a

given model when missing data needs to be inferred. Typically, the target distributions coincide with the posterior distributions of the parameters to be estimated. A more detailed description of the MCMC procedure is provided in Chapter 4.2.

Rest of the estimation procedures are not used in this thesis. But they are described briefly in this section.

The method of moments estimation is done by deriving equations that relates the population moments to the parameters of interest. Then a sample is drawn and the population moments are estimated from the sample. The equations are then solved for the parameters of interest. This results in estimates of those parameters. In some cases, with small samples, the estimates given by the method of moments fall outside of the parameter space. Therefore, it does not make sense to rely on them then. That problem never arises in the method of maximum likelihood.

A minimum mean square error (MMSE) estimator is an estimation procedure that minimizes the mean square error (MSE) of the fitted values of a dependent variable. The MSE is the second moment of the error. It incorporates both the variance of the estimator and its bias. Therefore it becomes the variance for an unbiased estimator. In statistical analysis the MSE represents the difference between the actual observations and the observation values predicted by the model. It is used to determine the extent to which the model fits the data and whether the removal of some explanatory variables is feasible to enhance model simplicity without significantly harming its predictive property. However, like variance, mean squared error has the disadvantage of heavily weighting outliers. It is due to the fact that it squares each term, which effectively weights larger errors more heavily than the smaller ones.

A MAP estimator is used to obtain a point estimate of an unobserved quantity on the basis of empirical data. It can be computed (a) analytically, when the mode(s) of the posterior distribution can be given in closed form or (b) by numerical optimization or (c) by a modification of an expectation-maximization (EM) algorithm. It is to be

stressed that the MAP estimates are point estimates, whereas Bayesian methods are characterized by the use of distributions to summarize data and draw inferences. There lies the weakness of MAP. One example exploiting the weakness of MAP is estimating the number of modes of a multi-modal mixture distribution. It becomes extremely difficult and sometimes not possible to manage a numerical optimization or a closed form to evaluate posterior distribution to find the number of modes using a MAP. It is comparatively simpler to rather employ an MCMC to simulate that distribution and infer the modes.

2.3 A survey of Resource Management (RM) approaches

In new computing paradigm steered by the cloud technology, optimization of resource management enjoys prime importance in the research community. This importance is growing as a foundation of most emerging networks with high dynamics such as peer to peer (P2P) overlays or ad-hoc networks. By definition a network can include many areas, namely the social network, computer network, population network etc. This survey, however restricts itself to the computer networks only in terms of the computing power, network bandwidth and memory management.

In cloud networks, resource management is all about integrating cloud provider activities to allocate and utilize the exact amount of resources within the limit of cloud network so that it can meet the needs of a cloud application. An optimal resource management strategy should try to avoid the following criteria:

- **Resource Contention:** multiple applications trying to access the same resource at the same instant
- **Over Provisioning:** application gets more resources than it has asked for
- **Under Provisioning:** application gets less resources than it has asked for - de-

grades of the Quality of Service (QoS)

It is possible to propose different resource management strategies based on application types. In [27] the authors designed resource management strategies for work-flow based applications, where resources are allocated based on the work-flow representation of applications. The advantage of work-flow based application is that the application logic can be interpreted and exploited to infer an execution schedule estimate. It enables the users to find the exact amount of resources that is needed to run his application. This is an example of an adaptive resource management strategy.

Real time applications pose a different challenge in terms of a deadline to complete a task. They have a light-weight web front end resource intensive back end. In order to dynamically allocate resources in the back end the authors in [28] implement a prototype system and evaluate it for both static and adaptive dynamic allocation on a test bed. This prototype functions by monitoring the CPU usage of each VMs and adaptively invoking additional VMs as required by the system.

Objective of a resource management strategy is to maximize the profits of both the customer and the provider in a large system by balancing the supply and demand in the market. Keeping this objective in mind this section classifies the resource management strategies in the following categories:

- Methods to facilitate resource management
  - Probabilistic Provisioning based RM
  - Virtual Machine Based RM
  - Gossip based RM
  - Auction based RM
- Issues to optimize for resource management
  - Utility Function

- hardware resource dependency
- Service level agreement (SLA)

2.3.1 Methods to facilitate resource management

2.3.1.1 Probabilistic Provisioning

Probabilistic approaches have the potential to be very efficient for resource management, since their use enables the exploitation of rich models of the studied systems that would be otherwise almost impossible to exploit. In [15] the authors proposed an user demand prediction-based resource management model that does Grid resource management through transaction management between resource users and resource suppliers. Badonnel et. al proposed a management approach in [4] which is centred on a distributed management self-organizing algorithm at the application layer for ad-hoc networks based on probabilistic guarantees. This work has been further enhanced by the authors at [12]. They proposed a probabilistic management paradigm for decentralized management in dynamic and unpredictable environments which can significantly reduce the effort and resources dedicated to management. In [60] Prieto et. al argued that adoption of a decentralized and probabilistic paradigm for network management can be crucial to meet the challenges of future networks, such as efficient resource usage, scalability, robustness, and adaptability.

A number of machine learning approaches have also been suggested to address the issues related to network resource management. Decision trees have been used to achieve proactive network management by processing the data obtained from the Simple Network Management Protocol (SNMP) Management Information base objects [35]. In [65] the authors applied Fuzzy logic to facilitate the task of designing a bandwidth broker. Classical Recursive Least Squares (RLS) based learning and prediction was proposed in [36] for achieving proactive and efficient IP resource management. [26] used reinforcement

learning to provide efficient bandwidth provisioning for per hop behaviour aggregates in diffserv networks. Authors of [9] in a different approach proposed a proactive system to enhance the network performance management functions by use of machine learning technique called Bayesian Belief Networks (BBN) that exploits the predictive and diagnostic reasoning features of BBN to make accurate decisions for effective management.

2.3.1.2 Virtual Machine (VM)

In [63] the authors suggest a system, which can automatically scale up or down its infrastructure resources. With the power of VMs, this system manages live migration across multiple domains in a network. This virtual environment, by using dynamic availability of infrastructure resources, automatically relocates itself across the infrastructure and consequently scales its resources. However, the authors do not consider preemptable scheduling policy (where some running jobs, called the preemptees can be interrupted by other running jobs, called the preemptors) in their work.

Other authors [37] proposed some resource management policies which consider preemptable scheduling and suitable for real time resource management. They formulate the RM problem as a constrained optimization problem and propose a polynomial-time solution to allocate resources efficiently. However, their approach sacrifices scalability to facilitate an economical allocation. Recent works from [61], [68] use a Service-Oriented Cloud Computing Architecture (SOCCA) so that clouds can interoperate with each other to enable real-time tasks on VMs. The work by [37] proposed an approach to allocate the resources based on speed and cost of different VMs in the Infrastructure as a Service (IaaS). Uniqueness of this approach stems from its user friendly approach. It allows the users to tune the resources according to the workload of his application and pay appropriately. It is implemented by enabling the user to dynamically add or remove one or more instances of the resources based on the VM load and conditions given by the user. This resource management on IaaS is different from the approach on Software

as a Service (SaaS) in cloud (SaaS delivers only the application to the cloud user over the internet).

In [34] authors discussed frameworks to allocate virtualized resources among selfish VMs in a non-cooperative cloud environment. In this environment one VM does not consider the benefits of other. The authors used a stochastic approximation technique to model and analyze QoS performance under different virtual resource allocations. Their results show the resource management technique oblige the VMs to report their types (such as the parameters defining a valuation function quantifying its preference on a specific resource allocation outcome) truthfully. Therefore the virtual resources can be allocated efficiently. However, this method is very complex and not validated against a real workload on a real-life virtualized cloud system.

2.3.1.3 Gossip

In [75] the authors proposed a gossip-based protocol for resource management in a large scale distributed system. In this work, the system is modelled as a dynamic set of nodes. Each node represents a machine in cloud environment. These nodes have a specific CPU and memory capacity. The gossip-based protocol implements a distributed scheme which allocates cloud resources to a set of applications. These applications have time independent storage demands. Thus it maximises the utilization globally. Experimental results show that the protocol provides optimal allocation when the storage demands is less than available storage in the cloud. However, their work needs additional functionalities to make the resource management robust to machine failure, that might span multiple clusters.

The authors in [49] provided with an different approach, where the cloud resources are being allocated by getting resources from remote nodes when there is a change in user demand. They developed a model of an “elastic site” that efficiently adapts services provided within a site, such as batch schedulers, storage archives, or web services to take

advantage of elastically provisioned resources. Keahey et al. in their research on sky computing (an emerging pattern of cloud computing) [32] focus on bridging multiple cloud providers using the resources as a single entity, which would allow elastic site to leverage resources from multiple cloud providers.

In [55] a gossip-based co-operative VM management has been introduced. This method enables the organizations to cooperate to share the available resources and thereby reduce the cost. They consider both public and private clouds in their work. They adopt network game approach for the cooperative formation of the organizations. However, they do not consider a dynamic co-operation formation of the organization.

2.3.1.4 Auction

Auction mechanism is also used by the researchers in a bid to provide efficient resource management techniques. In [42] the authors proposed a mechanism based on scale-bid auction. The cloud service provider collects user bids and then determine a price. Then the resource is distributed to the first k^{th} highest bidders under the price of the $(k+1)^{th}$ highest bid. This approach changes the management of resource allocation into ordering problem. However, this approach does not guarantee maximized profit, since it does not consider truth telling property as a constraint.

The authors in [77] achieve the maximization of the profits of both the customer and the provider in a large system by using a market based resource allocation strategy in which equilibrium theory is introduced. Market Economy Mechanism thus obtained is responsible for balancing the resource supply and market demand system.

2.3.2 Issues to optimize for resource management

2.3.2.1 Utility Function

Several approaches have been proposed by the researchers to dynamically manage the VMs and the IaaS by optimizing some objective functions such as minimizing the cost

function, meeting the QoS objectives etc. These objective functions are termed as the utility function and based on the parameters such as response time, QoS criteria, profit etc.

In [76] the authors proposed an approach to dynamically allocate the CPU resources to meet the QoS criteria. They allocate requests to high priority applications primarily to attain their objectives. Authors of [52] devised an utility (profit) based resource management for VMs that uses live VM migration as a resource allocation mechanism. Their work mainly focus on scaling CPU resources in the IaaS. Authors of [73] also use a live migration strategy for resource management. They use a policy based heuristic algorithm to facilitate live migration of VMs.

For cloud computing systems with heterogeneous servers, resource management tries to minimize the response time as measure of utility function [19]. The authors characterize the servers based on their processor, memory and bandwidth. Requests of the applications are distributed among some of the available servers. Each client request is sent to the server following the principles of queueing theory and the system meets the requirements of the service level agreements (SLA) based on its response time. However, their approach also follows a heuristic called "force-directed" search based resource management for resource consolidation.

Execution time is another another parameter in the utility functions. In [40] exact task execution time and preemptable scheduling are used for resource management, since it reduces resource contention and enhance resource utilization. However, estimation of execution time seems to be non-trivial and error prone [48]. In [50] the authors proposed a novel matchmaking strategy, i.e. assigning appropriate resource to an advance reservation request. On contrary to other matchmaking strategies, which use a priori knowledge of the local scheduling policy used at a resource, this one does not use a detailed knowledge of the scheduling policies. This strategy is based on "Any-Schedulability" [47] criterion that determines whether or not a set of advanced reservations can meet their

deadlines on a resource for which the details of the scheduling policy is unknown. This management strategy mostly depends on the user estimated job execution time of an application.

2.3.2.2 Hardware Resource Dependency

Resource management largely depends on the physical resources and an improved hardware utilization can greatly influence effectiveness of a resource management approach. In [25] the authors proposed a multiple job optimization scheduler. This scheduler classifies the jobs by hardware-dependency such as CPU/ Network I/O/ Memory bound and allocate the resources accordingly.

Open source frameworks like the Eucalyptus, Nimbus and Open Nebula facilitate resource management by virtualization [45]. All these frameworks allocate virtual resources based on the available physical resources. They form a virtualization resource pool decoupled with physical infrastructure. But, the virtualization technique is complex enough to bar these framework from supporting all types of applications. Authors of [45] proposed a system termed as " Vega Ling Cloud" to support heterogeneous applications on shared infrastructure.

Numerous works address the resource management on different cloud environments. Authors of [72] discussed an adaptive resource management approach based on CPU consumption. First they find a co-allocation scheme by considering the CPU consumption by each physical machine. Then they determine whether to allocate applications on the physical machine by using simulated annealing (that tries to perturb the configuration solution by randomly changing the elements). Finally the exact CPU share for each VM is calculated and optimized. Evidently this work mostly focuses on CPU and memory resources for co-allocation. However, they do not consider dynamic property of the resource request.

2.3.2.3 Service Level Agreements (SLA)

In cloud, SLA related resource management approach by SaaS providers have lots of spaces to develop further. With the development of SaaS, applications have started to tilt towards web delivered-hosted services. In [59] the authors consider QoS parameters (price and the offered load) on the provider's side as the SLA. Authors in [38] address the issue of profit driven service request scheduling by considering the objectives of both customers and service providers. However, the authors of [74] have considered a resource management approach by focusing on SLA driven user based QoS parameters to maximize the profit for the SaaS providers. They also propose mapping the customer requests to infrastructure level policies and parameters which minimize the cost by optimizing the resource allocation within a VM.

In [46] the authors propose a framework for resource management for the SaaS providers to efficiently control the service levels of their users. This framework can scale SaaS provider application under dynamic user arrival or departure. This technique mostly focuses on SaaS provider's benefit and significantly reduce resource wastage and SLA violation.

2.3.3 Conclusion

The above discussed resource management strategies show that resource management demands a significant importance in the emerging popularity of cloud computing. This study categorized the resource management strategies depending on the subjects to optimize and the methods to facilitate resource management. Furthermore this study observed that they are not mutually exclusive; on the contrary they complement each other. But, none of the above mentioned strategies include time scale in their framework. However, It is evident that it is not possible to define elasticity, which is a prime feature in cloud computing (or any application in general) without the notion of a time scale.

With this motivation some resource management strategies have been proposed in this work. They have been introduced with a detailed discussion in Chapter 5.

Chapter 3

Model Description

- Introduction
- A brief discussion of the epidemic models
- Workload Model description
- Implementation of the model on distributed environment
- Synthetic traces, generated from the model

3.1 Introduction

This chapter features a description of the VoD workload generator. Here workload is the amount of processing (allocation/ release of resources) that a system needs to perform at a given time. In the context of this work the workload relates to the applications running in the system and the number of users connected to and interacting with the system.

If a client-server architecture of a VoD system is considered then the workloads observed at the client side and the server side are different. A client only interacts with a limited number of servers, but the overall population as a whole interacts with many more servers and generate workloads with different profiles. Servers, again, interact

with many clients at a given instant, so they observe a global workload more than the workload of a single client. This inspires us to model a Video on Demand (VoD) system workload as a population model. In this system the users generate workload based on collective behavior. Following the trails of related works which have been described in the literature survey, the proposed model is *inspired* by epidemic models to represent the way information diffuses among the viewers (gossip-like phenomenon) of a VoD system.

3.2 A brief discussion of the epidemic models[1]

Epidemic spreading models commonly subdivide a population into several compartments: susceptible (noted **S**) to designate the persons who can get infected, and contagious (noted **C**) for the persons who have contracted the disease. This contagious class can further be categorized into two parts: the infected subclass (**I**) corresponding to the persons who are currently suffering from the disease and can spread it, and the recovered class (**R**) for those who got cured and do not spread the disease anymore [8]. In these models $S(t)_{t\geq 0}$, $I(t)_{t\geq 0}$ and $R(t)_{t\geq 0}$ are stochastic processes representing the time evolution of the number of susceptible, infected and recovered populations respectively. This section discusses some of the very popular epidemic models derived from compartmental model.

3.2.1 SIR Model [2]

In a SIR model each member of the population progresses from susceptible to infectious to recovered. This is shown as a flow diagram in Fig. 3.1 in which the boxes represent the different compartments and the arrows show the transitions between compartments. In this epidemic model $\beta > 0$, is the rate of infection per unit time. Then, the probability

[1]The list of the epidemic models, presented in this study is no way an exhaustive one. But, they relate closely to the proposed model which has been described in the next section.

[2]Both SIS and SIR models are developed following a continuous time Markov chain.

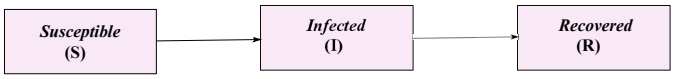


Figure 3.1: Flow diagram of the SIR epidemic model.

for a susceptible individual to get infected during a period dt is:

$$\mathbb{P}_{S\to I} = \frac{I(t)\,\beta\,\mathrm{d}t}{N} \tag{3.1}$$

Here, N designates the total population size. Therefore assuming that $\mathbb{P}_{S\to I} \ll 1$, the probability that at time t, k persons become infected during the same interval of time dt, is given by the binomial law:

$$\mathbb{P}\{I(t+\mathrm{d}t) - I(t) = k\} \quad = \quad \begin{pmatrix} S(t) \\ k \end{pmatrix} \mathbb{P}^{k}_{S\to I}(1-\mathbb{P}_{S\to I})^{S(t)-k} \tag{3.2}$$

For $k \approx S(t)\cdot p_{S\to I}$, it is known that the binomial expression of equation (3.2) can be accurately approximated with the following Poisson distribution

$$\mathbb{P}\{I(t+\mathrm{d}t)-I(t)=k\} \quad \approx \quad \exp\left\{-\frac{S(t)I(t)\beta\,\mathrm{d}t}{N}\right\}\frac{\left(\frac{S(t)I(t)\beta\,\mathrm{d}t}{N}\right)^{k}}{k!} \tag{3.3}$$

with rate

$$\lambda_{I(t)} = \frac{S(t)I(t)\beta}{N}. \tag{3.4}$$

Similarly the transition rate between I and R is γ. It is called the recovery rate. It is to be noted that the recovery rate has a constant value unlike the arrival rate that depends on number of infected people ($I(t)$) at a given instant.

In a compartmental epidemic model like the SIR there is a threshold quantity which determines whether an epidemic occurs or the disease simply dies out. This quantity is called the basic reproduction number, and is defined as $\beta N/\gamma$. This value quantifies the

transmission potential of a disease in the three following categories:

- $\beta N/\gamma < 1$; None the infective may not pass the infection on during the infectious period. Therefore, the infection dies out.
- $\beta N/\gamma > 1$; There is an epidemic in the population. It means infection propagates among the population.
- $\beta N/\gamma = 1$; The disease becomes endemic. It means the disease remains in the population at a consistent rate.

Fig. 3.2 shows how the population in three classes evolve with time for the three categories of transmission potential of a stochastic SIR model.

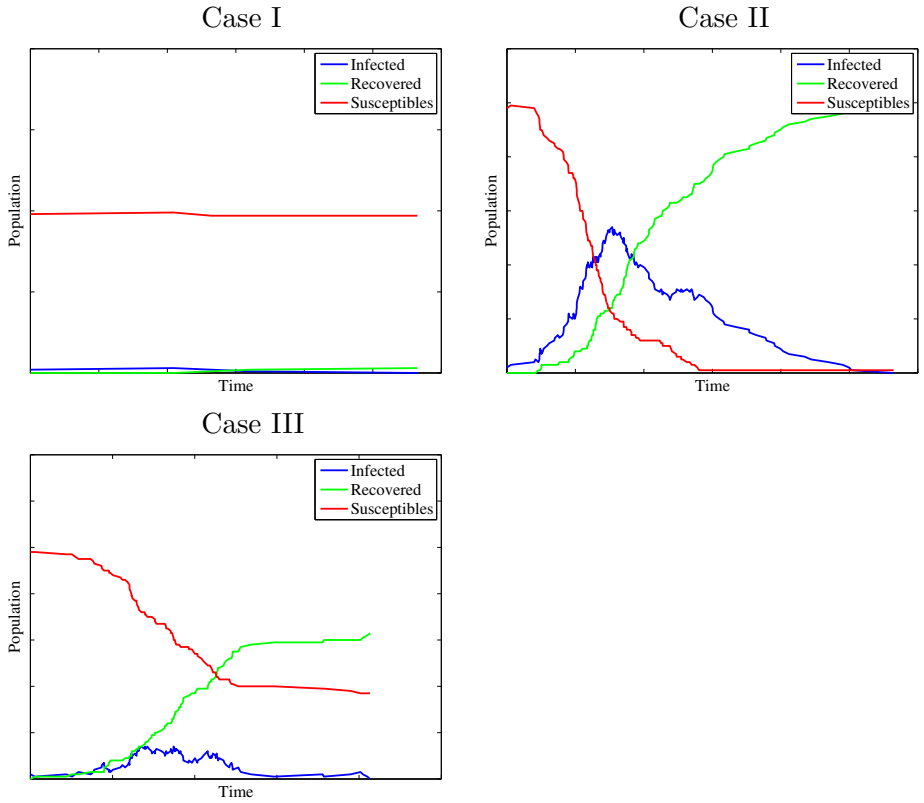


Figure 3.2: Evolution of S, I and R classes with time for a SIR epidemic. For the sake of generalization parameters of the three cases are not quantified.

In case I of the Fig. 3.2 the basic reproduction is less than one. It is observed that the infection dies out without spreading the disease among the population. Case II shows the situation when the basic reproduction is greater than one. The plot shows the evolution of infection in this case. In case III the basic reproduction number is equal to one. Naturally, the disease stays in the population with a rate which is almost constant.

3.2.2 SIS Model

In an SIS epidemic model, a susceptible individual, after a successful contact with an infectious individual becomes infected and the infectious person does not develop any immunity to the disease, i.e he becomes susceptible again.

Similar to the SIR model the transition rate between S and I for this model is also βI and takes into the account the probability of getting the disease in a contact between a susceptible and an infected person. The transition rate between I and S, is γ. However, there is no threshold phenomenon as the previous case, since the evolution of infected people in an SIS model does not end and the susceptible and the infected reaches stability eventually. The flow diagram in Fig. 3.3 shows the transitions between the S and I compartments.



Figure 3.3: Flow diagram of the SIS epidemic model.

Fig. 3.4 shows evolution of S and I populations for a stochastic SIS model, which attains stability after a certain amount of time. Here the basic reproduction number is greater than one.

3.2.3 SEIR Model

The SEIR epidemic model is originated from the SIR model. Here another extra compartment is considered. For some types of infections there can be a significant period

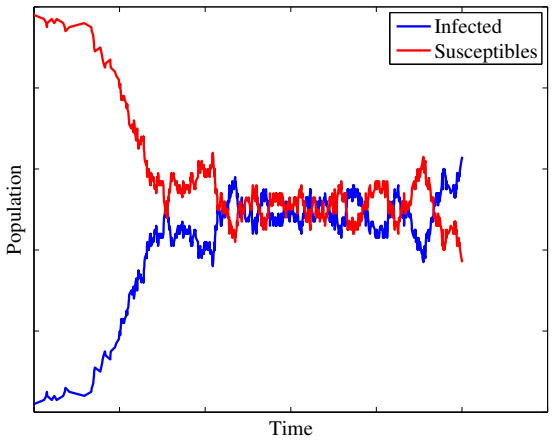


Figure 3.4: Evolution of S and I classes with time for a SIS epidemic model. For the sake of generalization parameters of the three cases are not quantified.

of time during which the individual gets infected but they do not become contagious. This period is called the latent period and represented by the compartment E (exposed). The flow diagram in Fig. 3.5 shows the transitions between different compartments in a SEIR model.

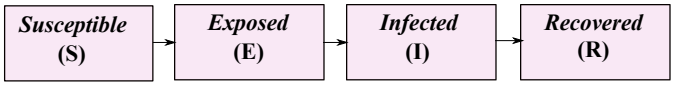


Figure 3.5: Flow diagram of the SEIR epidemic model.

The three epidemic models which are described in this section can be either deterministic (can be represented by differential equations) or stochastic (can be represented by Markov models). The work of this thesis is based on stochastic epidemic models, which incorporates the real world uncertainty in the system. However, being a Markov process a stochastic epidemic model is a memoryless system. But it might not be appropriate to consider a social system (where people gossip over an issue) without memory. Therefore, further modifications of the epidemic models are required, as illustrated in the following section.

3.3 Workload Model description

This work considers the total number of current viewers as workload (current aggregated video requests from the users) and contextualize the epidemic models for a VoD system. In case of a VoD provider the number of potential subscribers (**S**) can go very high. Therefore it is assumed that **S** is *infinite* and not considered in the model. Infected **I** refers to the people who are currently watching the video and can pass the information along. This work considers the workload as the total number of *current viewers*, but it can also refer to total bandwidth requested at the moment. The class **R** refers to the *past viewers*, who can still disseminate information about a video before leaving the system (i.e loosing interest on a video gradually, thereby stopping to talk about it).

In contrast to the classical compartmental epidemic this model does not show a threshold phenomenon, i.e if the initial infected population exceeds a critical threshold (which quantifies the transmission potential of the disease), then the epidemic spreads, otherwise it dies out. There is no such situation in the proposed model since there is always some spontaneous arrival of users in this system. This feature differentiates it from a classical epidemic model. Another major distinction of this approach arises from introducing a memory effect in the model. It is assumed that the **R** compartment can still propagate the gossip during a certain random latency period. This assumption is considered necessary from normal social behavior where people keep discussing about a video even after watching it. Then after a certain duration they stop doing it.

The process defining the evolution of the population is modeled as a continuous time Markov process whose states correspond to each possible value i and r of $I(t)$ and $R(t)$. In each state there are a number of possible events that can cause a transition. The event that causes a transition from one state to another takes place after an exponential amount of time. As a result, in this model transitions take place at random points in time. For a discrete time Markov chain the transitions occur only at discrete intervals

(equally spaced). These intervals are kept small so that more than one transition does not happen within a single interval. It is also possible to discretize the proposed model and realize it using ordinary differential equations which considers state changes at equally spaced instants.

3.3.1 Differences between the proposed model and a standard epidemic model

Even though the proposed model draws inspiration from standard epidemic models (namely SIR) its mechanism differs from a standard one. The transition probability of a susceptible to turn infectious in a SIR model follows Eq. (3.1). Following Eq. (3.1) the corresponding transition probability for the proposed model within a small time interval $\mathrm{d}t$ reads:

$$\mathbb{P}_{S\to I} = (l + (I(t) + R(t))\,\beta)\mathrm{d}t + o(\mathrm{d}t)$$

Here, $\beta > 0$ is the rate of information dissemination per unit of time and $l > 0$ represents the ingress rate of spontaneous viewers. Therefore, at time t, the instantaneous rate of newly active viewers in the system reads:

$$\lambda(t) = l + (I(t) + R(t))\beta \tag{3.5}$$

This rate corresponds to a non-homogeneous (state-dependant) Poisson process which varies linearly with $I(t)$ and $R(t)$.

When $\beta \gg l$, the arrival process induced by peer-to-peer contamination dominates the workload increase, whereas it is the spontaneous viewers arrival process that prevails when $l \gg \beta$. Moreover, l abstains the system to reach its absorbing state when both $I(t) = i$ and $R(t) = r$ becomes zero.

Regarding the sojourn time in the (**I**) compartment, it is assumed that the watch time of a video is an exponentially distributed random variable with mean value γ^{-1}

(reasons to choose an exponential distribution has been discussed later on). It means that the viewers leave for the (**R**) class at rate γ. As already mentioned, it also seems reasonable to consider that a past viewer will not keep propagating the information of a video for an indefinite period of time. Rather, they stay active only for a random latency period. This period is also assumed to be exponentially distributed with mean value μ^{-1}. After this period the past viewers leave the system (at rate μ). From a general perspective and without loosing generality can be stated that the watching time (γ^{-1}) of a video is much smaller compared to the memory (μ^{-1}) persistence. Therefore, $\mu \ll \gamma$.

Another novelty of the approach is modelling buzz event with a Hidden Markov Model (HMM). *Buzz* is defined by a sudden increase of propagation rate β (or gossip) due to the popularity of a given content. This work considers a two states HMM. The state with the dissemination rate $\beta = \beta_1$ corresponds to the buzz-free or the normal case as described above. In this state the number of viewers attain a stationary state and stays there until there is a buzz. The other hidden state corresponds to the buzz situation, where the value of β increases significantly and takes on a value $\beta_2 \gg \beta_1$. The buzz state can be either stationary or non-stationary. For stationary case the workload increases and the system can attain steady-state if it remains in buzz for long durations. For non-stationary case the workload keeps increasing and the system never attains a steady-state. If it stays in non-stationary buzz for long time the overall system can become unstable. Jump from buzz-free to buzz state triggers a sudden and steep increase of the current demand.

Transitions between these two hidden and memoryless Markov states occur with rates a_1 and a_2 respectively and characterize the buzz in terms of frequency and duration. In the VoD context it is evident that $a_1 \ll a_2$, i.e. buzz periods are less frequent and shorter in duration than normal periods. Theoretically, it is possible to generalize the model to include many hidden states. But Chapter 4 demonstrates that only two states suffice to

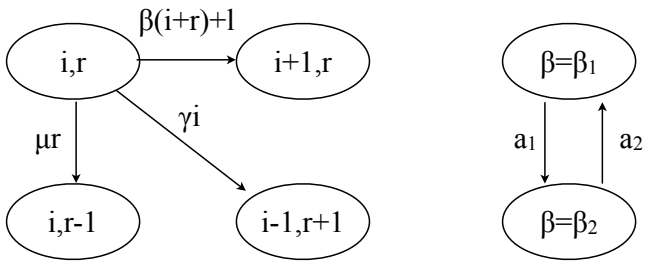


Figure 3.6: Markov Chain representing the possible transitions of the number of current (i) and past active (r) viewers.

reproduce different types of buzz with peaks and troughs at many scales. With these assumptions, and posing (i, r) as the current state, Fig. 3.6 shows the state-transition diagram of the model.

The proposed model is based on exponential properties of distribution for the following reasons:

- It is common practice to use it in epidemiology,
- It is simpler to analyze due to its memoryless property,
- Obtained results shows that (Chapter 4.3) the proposed model with exponential distribution succeeds to yield realistic results,
- Author of [43] demonstrates that the use of other probability distributions results less stable behaviour in an epidemic model,

A more detailed study of the effect of other distributions for the proposed model is postponed to a further step of this work.

Another, appealing aspect of this model is that it verifies a Large Deviation Principle (LDP) which statistically characterizes extreme rare events, such as the ones produced by "buzz/flash crowd effects" and result workload overflow in the VoD context. This chapter introduces the LDP briefly. An elaborate description of this concept is provided in Chapter 5.

A deeper study of the VoD model suggests that a closed-form expression for the steady-state distribution of the workload (i) of this model might not to be trivial to derive. Instead, it is possible to express the analytic mean workload of the system equating the incoming and outgoing flow rates in steady regime (fundamentally the incoming and outgoing flow rates are equal in a steady system). For convenience, it is reasonable to start with $\beta = \beta_1 = \beta_2$ and then generalize the result to $\beta_1 \neq \beta_2$. It is known that the average rate of incoming flow in I is $\beta\big(\mathbb{E}(i)+\mathbb{E}(r)\big)+l$ and in R is $\gamma\mathbb{E}(i)$. Similarly the average rate of outgoing from I is $\gamma\mathbb{E}(i)$ and from R is $\mu\mathbb{E}(r)$ (naturally $\mathbb{E}(i), \mathbb{E}(r)$ denotes mean i and r). Equating the rates at the steady state and replacing $\mathbb{E}(r)$ by $\mathbb{E}(i) \cdot \gamma/\mu$ it is possible to obtain:

$$\mathbb{E}(i) = \mu l/(\mu\gamma - \mu\beta - \gamma\beta) \tag{3.6}$$

Naturally, $\mathbb{E}(i)$ is to be a positive and finite quantity. Therefore the denominator of Eq. (3.6) should be greater than zero which yields the stability criterion in buzz-free regime:

$$\beta^{-1} > \mu^{-1} + \gamma^{-1}. \tag{3.7}$$

Next these results are extended to the case where the model may exhibit a buzz activity. β alternates between the hidden states $\beta = \beta_1$ and $\beta = \beta_2$, with respective state probabilities $a_2/(a_1 + a_2)$ and $a_1/(a_1 + a_2)$. Therefore the mean workload in this situation reads:

$$\mathbb{E}(i) = a_2/(a_1 + a_2) \cdot \mathbb{E}_{\beta_1}(i) + a_1/(a_1 + a_2) \cdot \mathbb{E}_{\beta_2}(i), \tag{3.8}$$

Eq. (3.8) is validated in section 3.4. Clearly for Eq. (3.8) to hold true the model has to be stable at both buzz and buzz-free regimes, i.e both ${\beta_1}^{-1} > \mu^{-1} + \gamma^{-1}$ and ${\beta_2}^{-1} > \mu^{-1} + \gamma^{-1}$ must hold true. It is the stationary buzz condition as described earlier.

However, as an approximation it is also possible to consider the mean rate of infor-

mation propagation for the overall system and that can be expressed as:

$$\beta_{mean} = a_2/(a_1 + a_2) \cdot \beta_1 + a_1/(a_1 + a_2) \cdot \beta_2 \tag{3.9}$$

Stability criteria for this case is ${\beta_{mean}}^{-1} > \mu^{-1} + \gamma^{-1}$. This stability criteria less stringent than the previous one. It includes the possibility to have a non-stationary buzz in this system where ${\beta_2}^{-1} < \mu^{-1} + \gamma^{-1}$. Therefore the system can be globally stable even though it might yield local instability in the buzz regime. The stability criteria in this case depends on the values of a_1 and a_2 beside β_1 and β_2. In order to attain the global stability, when ${\beta_2}^{-1} < \mu^{-1} + \gamma^{-1}$ a high value of a_2/a_1 is necessary, so that the system spends less time in the buzz regime, leading to overall stability.

3.4 Generated VoD traces from the Model

For illustrating the versatility of the proposed workload model and validating Eq. (3.8), synthetic traces have been generated corresponding to five different sets of parameters. Table 3.1 reports them. Particular realizations of these processes generated over $N = 2^{21}$ points are displayed in Fig. 3.7. The plots in Fig. 3.7 display both current and the past viewers to have a better understanding of the process. These five sets of parameters

Table 3.1: Parameter values, used to generate the traces plotted in Fig. 3.7

	β_1	β_2	γ	μ	l	a_1	a_2
Case I	$2.7600.10^{-4}$	$4.7380.10^{-4}$	0.0111	$5.0000.10^{-4}$	$1.0000.10^{-4}$	$1.00.10^{-7}$	0.0667
Case II	0.0100	0.1200	0.6000	0.2000	1.0000	0.0060	0.0100
Case III	0.0082	0.0083	0.0500	0.0100	0.0100	$1.00.10^{-4}$	0.0100
Case IV	1.9989	1.9999	6.0000	3.0000	0.1000	$1.00.10^{-3}$	0.0667
Case V	$1.3700.10^{-4}$	0.0014	0.0050	0.0020	0.1808	$1.00.10^{-5}$	$2.00.10^{-5}$

have been chosen in an ad-hoc manner (keeping stability condition in mind) such that they lead to five distinct types of workload starting from lightly loaded system in Case I to heavily loaded system in Case V. The parameters have been considered such that the system stays stable in both buzz and buzz-free regimes (imposing the more stringent criteria on system stability). In Case III and Case IV we considered the value of β_1

and β_2 very close to demonstrate that even close values of the these two parameters can cause considerable buzz in the system, owing to the rest of the parameters. In these two cases the model is very close to the stability (i.e. $1/\beta_1 - 1/\mu - 1/\gamma \approx 0$). In this case even a slight change of arrival rates (when β_1 becomes β_2) cause enough perturbation in the system leading to a considerable buzz.

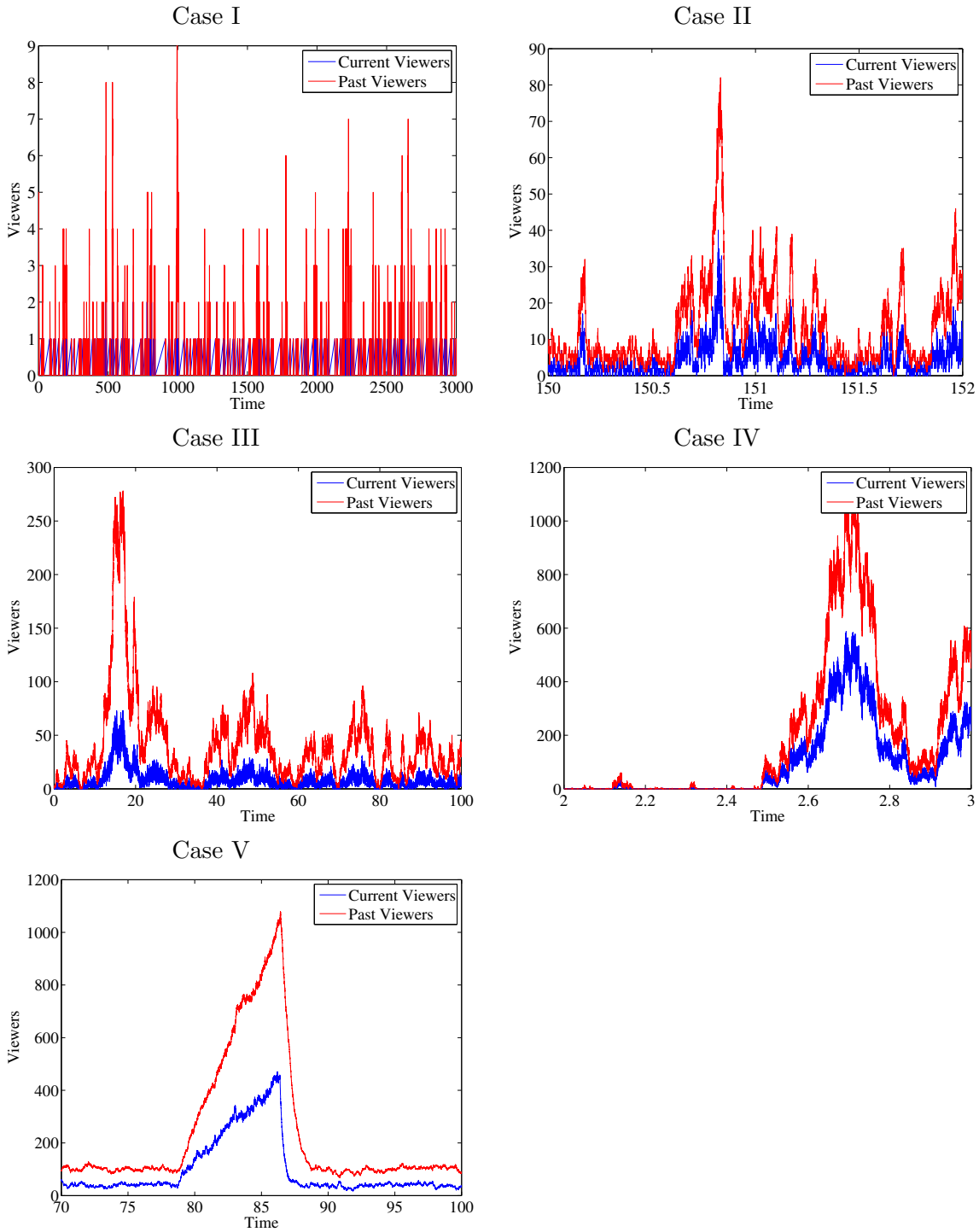


Figure 3.7: Traces generated from the parameters, reported in Table. 3.1. The horizontal axis represents time (in hours) and the vertical axis represents VoD workload (i.e. the number of viewers).

A closer look at Table. 3.1 shows that the buzz duration is the lowest for Case I among all. Moreover the duration of buzz-free period is considerably higher than the buzz period making this system very lightly loaded. For Case II the buzz duration increases and is much closer to the buzz-free duration (compared to Case I). The possibility of staying in a buzz state increases in this case and makes the workload higher. Moreover a higher value of l also contributes to the more spontaneous arrival of people in the system. In Case III it is observed that the mean workload increases further even though the buzz duration stays the same and β_1 and β_2 have lower values than Case II. This is due to the fact that the values of μ and γ are considerably less than the previous cases and the viewers stay longer in the system. In Case IV the values of β_1 and β_2 increases considerably to increase the overall workload of the system. Finally for Case V the buzz duration is the highest among all and β_2 is much higher than β_1. Combined effect of these two makes the system heavily loaded with long-duration buzz for Case V.

Plots of the steady state distribution of current viewers for the five cases in Fig. 3.8 illustrate the process volatility. For the first three cases the steady state distribution is computed numerically from the rate matrix (which can be obtained from the knowledge of model parameters and maximum number of current and the past viewers) using the GTH (Grassmann-Taksar-Heyman) algorithm. A detailed description of the approach has been provided in [67]. This method provides an accurate solution at the expense of very high computational cost.This method has been used for the first three cases ranging from very lightly loaded system to the moderately loaded system. For higher workloads (represented by last two traces) the steady state distribution is calculated empirically from the workload trace. Moreover, for all five configurations, the empirical means estimated from the 2^{21} samples of the traces are in good agreement with the expected values of Eq. (3.8). Table 3.2 reports this finding. It is observed that the highest error occurs for Case III which is around 15%. Moreover, Fig. 3.8 shows that Case I is the least volatile one since the ratio of a_2 to a_1 is the highest, implying the process stays

Table 3.2: Comparison of $\mathbb{E}(i)$ and Emp. mean $\langle i \rangle$, from the traces plotted in Fig. 3.7

	$\mathbb{E}(i)$	Emp. mean $\langle i \rangle$	% Error
Case I	0.0213	0.0214	0.4212
Case II	4.2411	3.8957	8.1446
Case III	12.8713	11.8254	8.1258
Case IV	36.9424	37.6880	2.0184
Case V	320.6829	296.2218	7.6278

in one state (the buzz-free state in this case) most of time. This ratio is the lowest for Case II and Case V (causing frequent alternation between buzz-free and buzz states). But the β_2/β_1 value is higher for the Case V than Case II, leading to higher change in arrival rates for the later case when the process changes states (buzz to buzz-free or vice versa). Therefore Case V is the most volatile among all five cases.

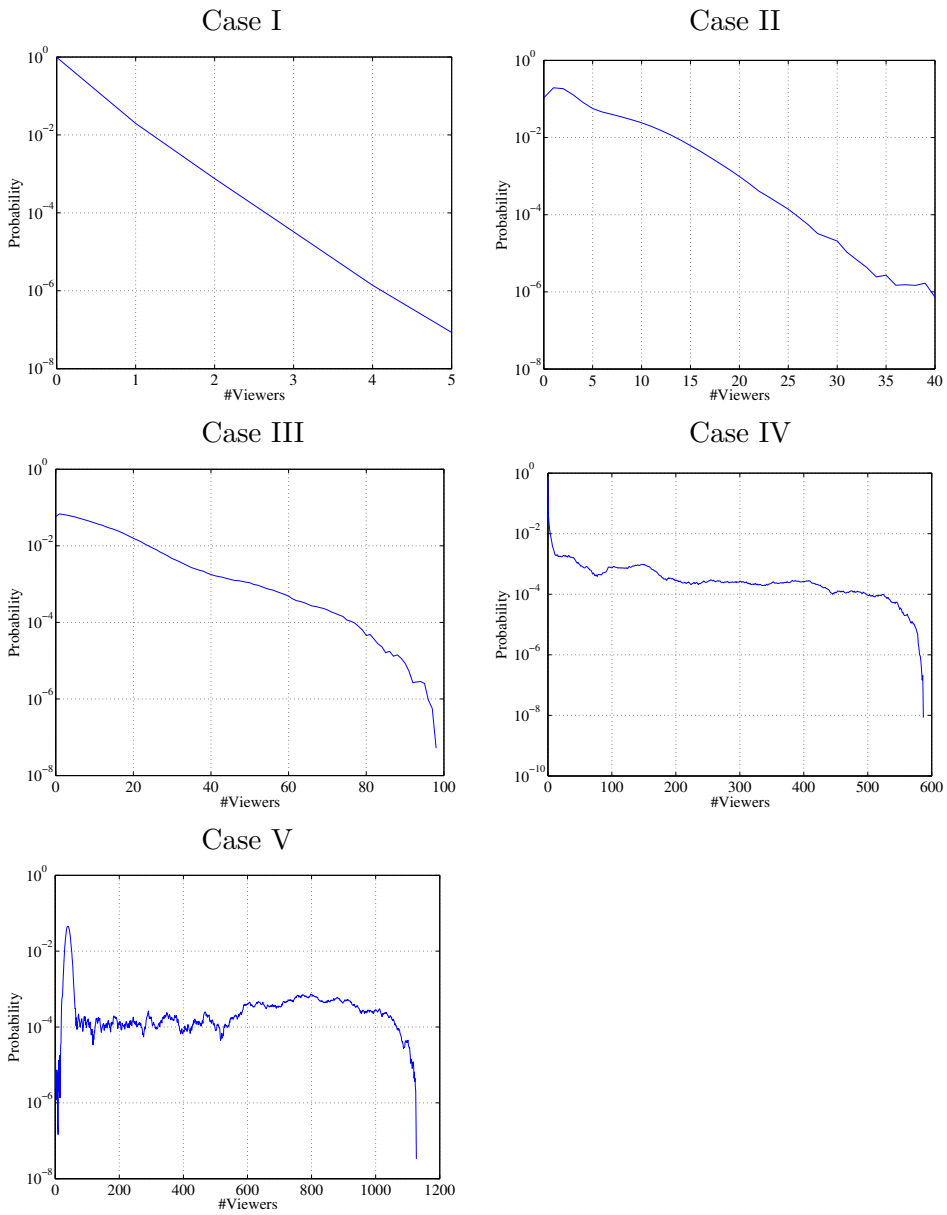


Figure 3.8: Steady state distribution of the traces generated from the parameters, reported in Table. 3.1.

Fig. 3.9 illustrates the memory effect (controlled by the parameter μ) which has been injected in the proposed model by autocorrelation plots. Autocorrelation measures the statistical dependency $R_I(\tau) = \mathbb{E}\{I(t)\, I^*(t+\tau)\}$ between two samples of a (stationary) process I, distant of a time lag τ: the larger $R_I(\tau)$, the smoother the path of I at scale τ. Case I shows the longest memory among all (around 2.25 hours), since the μ value is the minimum for this case. Similarly Case IV shows an insignificant memory effect (around 4 seconds) on the system (having the highest μ).

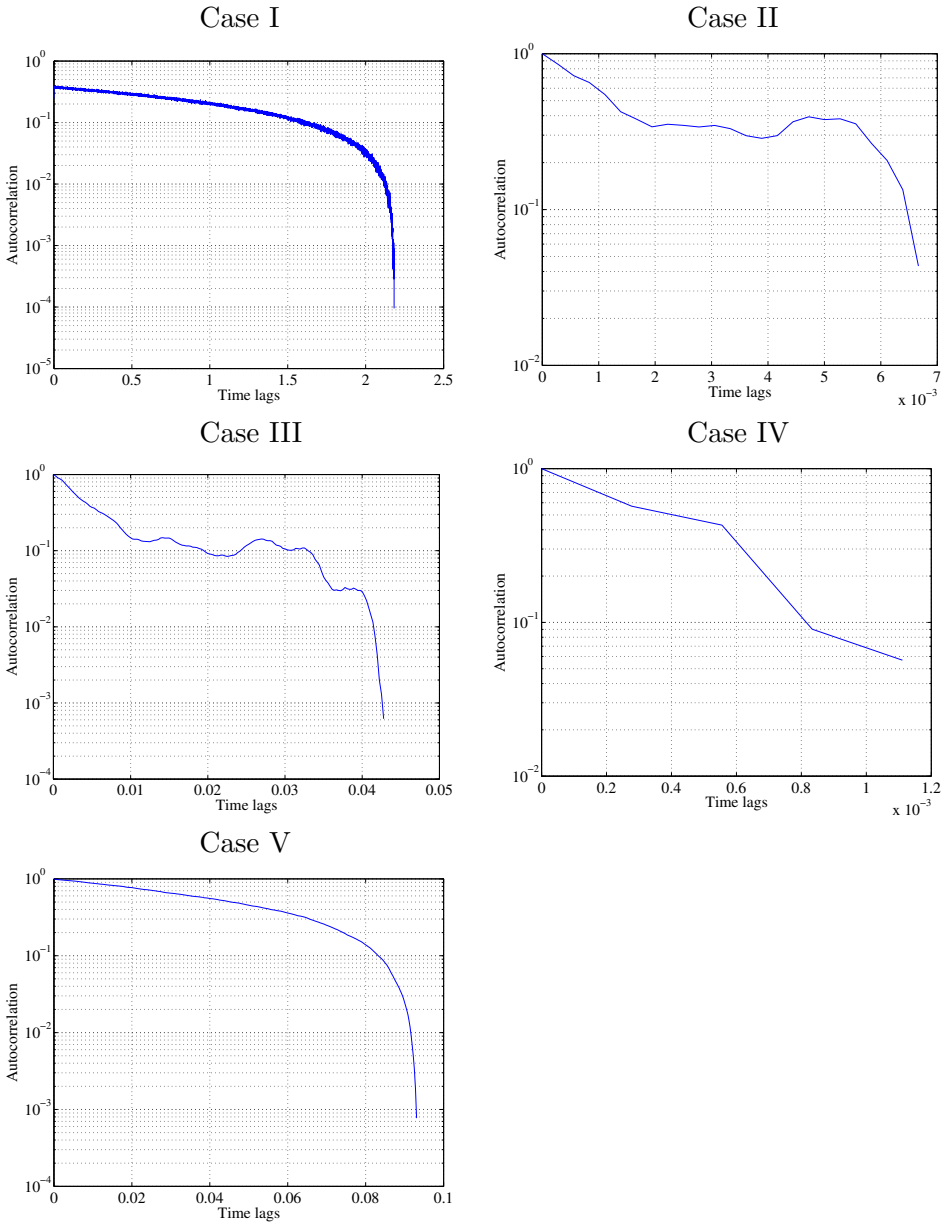


Figure 3.9: Empirical autocorrelation function of the traces generated from the parameters, reported in Table. 3.1.

Fig. 3.9 illustrates the memory effect (controlled by the parameter μ) which has been injected in the proposed model by autocorrelation plots. Autocorrelation measures the statistical dependency $R_I(\tau) = \mathbb{E}\{I(t)\, I^*(t+\tau)\}$ between two samples of a (stationary) process I, distant of a time lag τ: the larger $R_I(\tau)$, the smoother the path of I at scale τ. Case I shows the longest memory among all (around 2.25 hours), since the μ value is the minimum for this case. Similarly Case IV shows an insignificant memory effect (around 4 seconds) on the system (having the highest μ).

Fig. 3.10 shows empirical estimation of the large deviation spectrum of the five traces (sampling time scale is one thousandth of the total time duration). In the plots $\alpha_\tau = \langle i \rangle_\tau$ corresponds to the mean number of users i observable over a period of time of length τ and $f(\alpha)$ relates to the probability of its occurrence as follows:

$$\mathbb{P}\{\langle i \rangle_\tau \approx \alpha\} \sim e^{\tau \cdot f(\alpha)}. \tag{3.10}$$

A detailed description of the large deviation computation has been provided in Chapter 5. The purpose of briefly introducing this concept here is to illustrate another useful property of the model which would be exploited later on (in Chapter 5). This apex of the spectrum(s) of Fig. 3.10 is called the almost sure value. As the name suggests almost sure workload corresponds to the mean value that is almost surely observed on the trace for the time scale τ. More interestingly, the LD spectrum corresponding to the more prominent buzz case (from Case I to Case V), spans over a larger interval of observable mean workloads. This remarkable support widening of the theoretical spectrum shows that LDP can accurately quantify the occurrence of extreme, yet rare events.

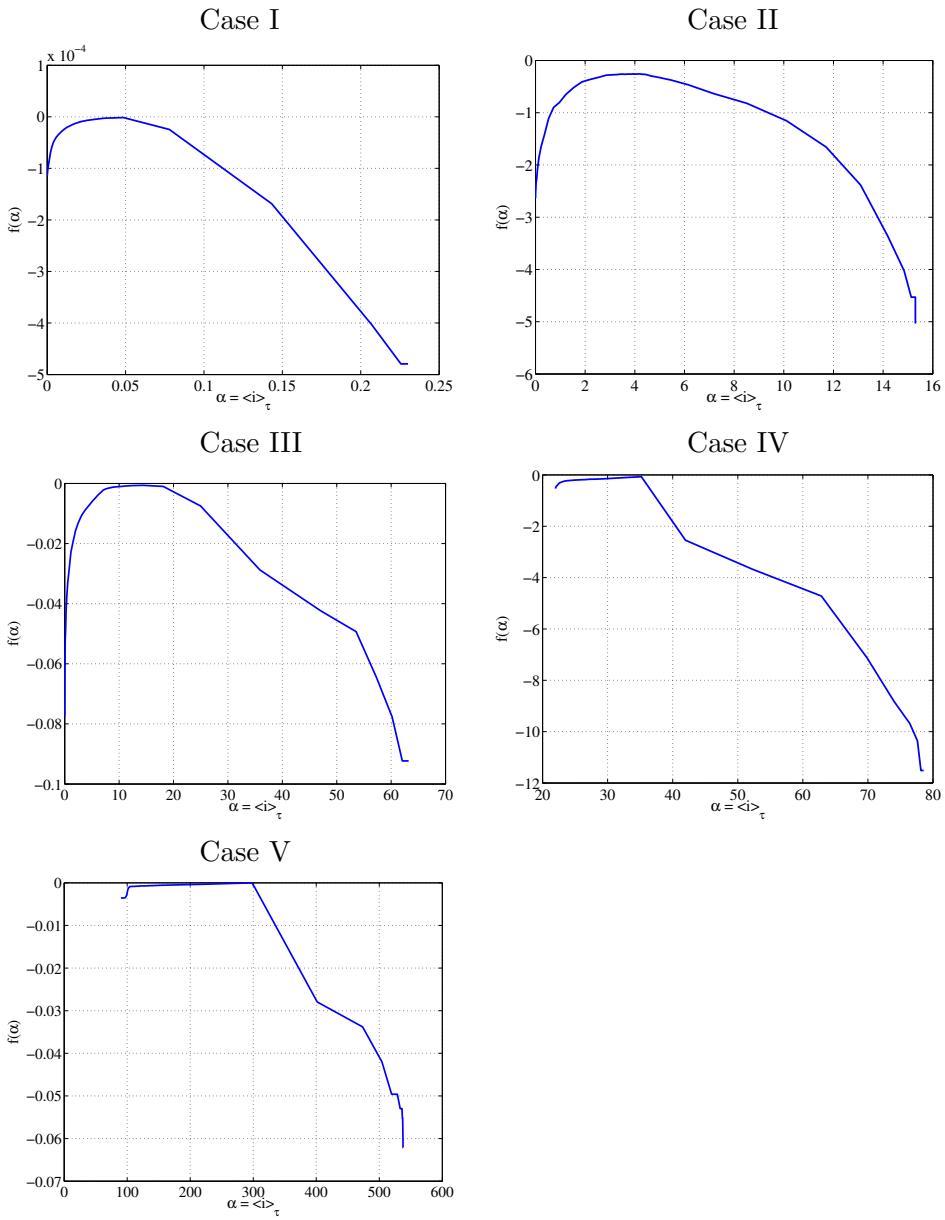


Figure 3.10: Empirical Large Deviation Spectrum of the traces generated from the parameters, reported in Table. 3.1.

3.5 Addendum:

3.5.1 Implementation of the VoD model on distributed environment

With a future objective to exploit the traces generated out of this model to frame resource management policies in a cloud network, the model has been deployed on Grid 5000. Grid 5000 is a 5000-CPU nationwide grid infrastructure for research in grid computing, providing a scientific tool for computer scientists similar to the large-scale instruments used by physicists, astronomers, and biologists. It is a research tool featuring deep reconfiguration, control, and monitoring capabilities designed for studying large-scale distributed systems and for complementing theoretical models and simulators. As much as 17 French laboratories are involved, and nine sites host one or more clusters of about 500 cores each. A dedicated private optical networking infrastructure interconnects the Grid'5000 sites. In the Grid'5000 platform, the network backbone is composed of private 10-Gb/s Ethernet links. Figure 3.11 shows the Grid'5000 topology.

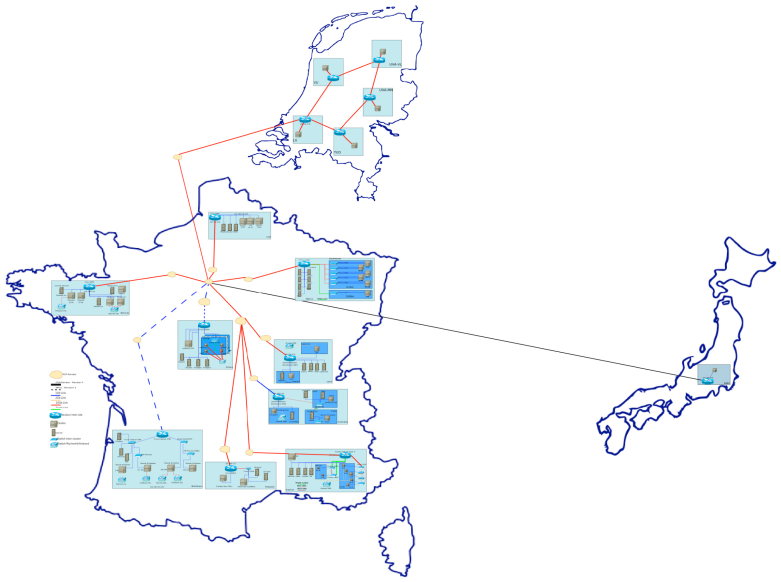


Figure 3.11: Topology of Grid'5000

Grid'5000 enables researchers to run successive experiments reproducing the exact

experimental conditions several times, a task that is almost impossible with shared and uncontrolled networks.

3.5.2 Global Architecture of the Workload Generating System

In order to generate a certain workload on a VoD server, each node in the Grid'5000 is considered as an user entity. Figure 3.12 shows how the nodes interact among themselves to emulate user behavior. During implementation, all nodes (users) are considered to be

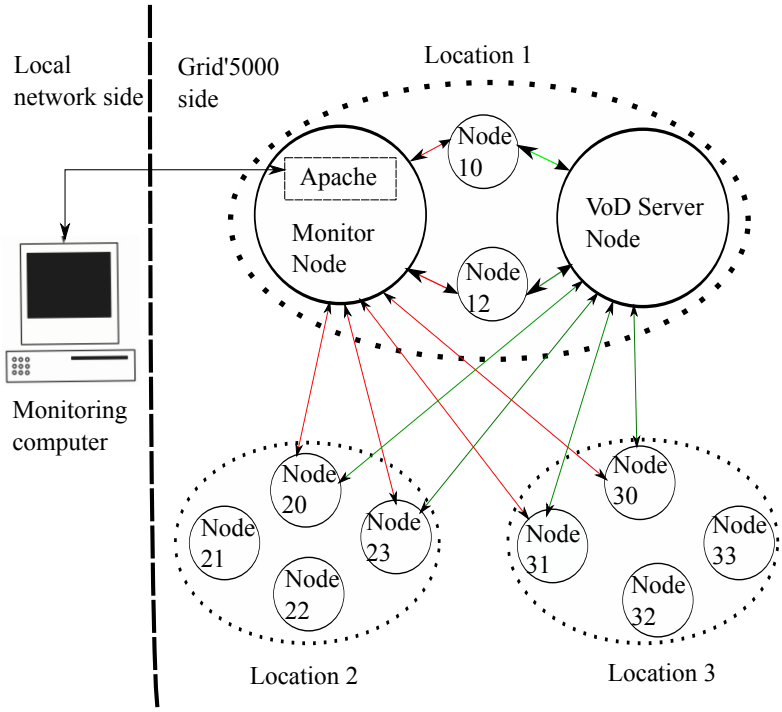


Figure 3.12: Architecture and interactions between the nodes to replicate user behavior.

independent. Following a centralized architecture a monitor node has been fixed, which controls the state of all nodes as-well-as allows and controls communications among them. Each node is connected to the monitor node (red links) and to the VoD server (green link) as schematized in Fig. 3.12.

Since an user can be in any three S, I or R states and only solicits the VoD server when it is in infected state. Each time an user changes its state, it sends a message

to the monitor to update its status. Moreover, when an user wants to infect another user, it first requests the monitor to choose randomly among susceptible users. Then the monitor node sends back a message to the chosen node to turn it infected. The implementation considers that an user goes back to the S state after leaving the R state with the possibility of getting contacted by the server again. An apache web server has also been implemented on the monitor node to visualize evolution of the workload in real-time (Fig. 3.12).

3.5.3 Implementation Issues

The first issue which has been encountered, is to generate independent random variables, as required by the Markov Model. It is known that the classical approach to generate random variables is to define an unique seed to ensure the independence of variables. However, since all nodes are launched at the same time, it is not possible to use the current time to define the seed. This is however a well known problem and managed by summing up the IP address to the current time for generating the seeds. The last operation is done to facilitate independent realizations in case it is required to repeat the same experiments on the same nodes.

The second issue is to implement an efficient server to manage the significant amount of communication among hundreds of nodes. A multi-threaded server has been used to handle this. Each time a node wants to communicate with the server, a new thread is created to process the request while the original thread holds ready to listen to any new communication. This process has been used in the experiments having 300 nodes at the most without any performance issue. However, this approach might not be scalable enough to handle communication among higher number of nodes and falls within the scope of further development of the model in a distributed environment. Using the threads to handle communication raises another problem regarding protection of shared variables from multiple accesses. To prevent this, mutual exclusion has been used by

defining specific variables to manage access to these shared variables between threads.

3.6 Results from the distributed implementation

This implementation shows the effectiveness of the proposed workload generator to emulate several realistic VoD traffic traces (having different workload profile) with different sets of parameters on a distributed system. The main asset of the approach lies in the combination of a versatile, plausible theoretical model with a fully controllable large-scale test-bed involving heterogeneous equipment and an advanced networking infrastructure. Figure. 3.13 shows a snapshot from the monitoring computer displaying a typical real-time server workload. The top plot of Figure. 3.13 shows the evolution of customers (it shows both current and past viewers and represents in the software as N_i and N_r respectively. $N_i + N_r + N_s$ implies the total number of nodes. The VoD: Buzz and VoD: Buzz-free identifies the buzz and buzz-free parts of the plots respectively). The bottom plots shows the steady state distribution of the current (N_i), past (N_r) and the susceptible (N_s) viewers respectively. To conclude this section, it is to be mentioned that the implementation differs from the theoretical model on the aspect of the number of the susceptible viewers. It is infinite for the theoretical scenario whereas it is limited to the number of available nodes in Grid'5000.

3.7 Conclusion

This chapter introduces an epidemic inspired model and presents an in-depth analysis of the model. Obtained results confirm that the model succeeds to yield different workloads with diverse volatility based on its parameters. Some basic as well as advanced statistical properties of the model had also been presented here which would exploited in Chapter V for framing resource management polices. Finally, this model has been implemented in Grid 5000 test-bed. Even though the implemented model has the scope of further

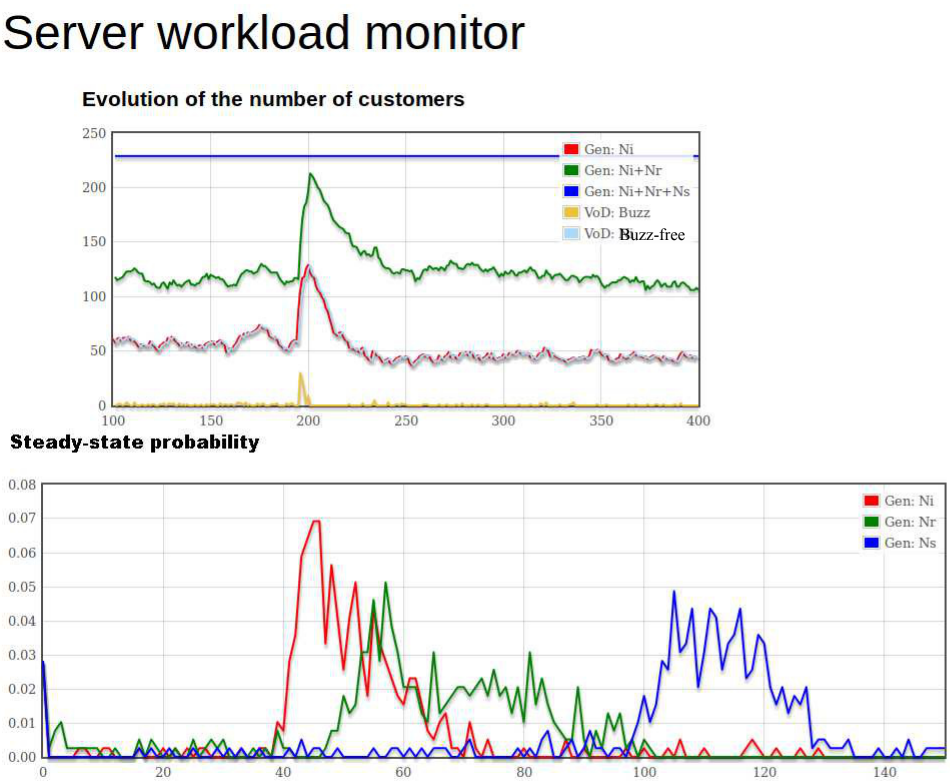


Figure 3.13: Snap shot of the real-time server workload from the monitoring computer.

development, it can be still utilized by the VoD service providers to simulate workload in a distributed environment.

Chapter 4

Estimation Framework

- Description and experimental validation of heuristic approach to estimate model parameters
- A short introduction to Markov Chain Monte Carlo (MCMC)
- Description and experimental of MCMC approach to estimate model parameters
- Validation of the framework on real traces
- Merits and demerits of both approaches

This chapter follows two different approaches to estimate the model parameters and describe them in three consecutive sub-chapters. The introductory sub-chapter (4.1) describes the first approach based on a heuristic procedure. A heuristic procedure can deliver approximate results within a reasonable time duration. This procedure leverage the fact that the model is built from a constructive approach. Naturally, estimation of the parameters become the "inverse problem", where given a workload trace it is required to estimate the parameters from the knowledge of the model mechanism.

The following sub-chapter (4.2) describes the second approach, based on a Markov Chain Monte Carlo (MCMC) framework. In the final sub-chapter (4.3) the data model adequacy on real VoD traces is shown.

This work admits the fact that an MCMC approach is a more standard procedure to

estimate model parameters than a heuristic one. But, the rationale of using the heuristic approach comes from the fact that it embeds the model mechanism in the procedure. The results show that, this approach provides an intuitive solution with reasonable accuracy.

4.1 Model Parameter estimation: a heuristic approach

4.1.1 Introduction

This section is starts by showing a simple schematic (Fig. 4.1) for estimating the parameters $\beta_1, \beta_2, \gamma, \mu, l, a_1$ and a_2 from workload trace(s) using the heuristic procedure.

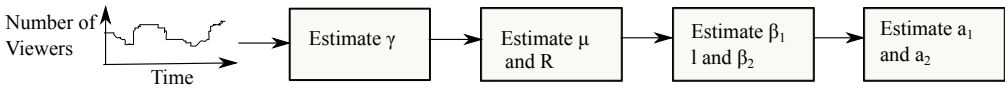


Figure 4.1: Schematics showing the flow order in which the parameters are estimated from an input trace.

It is important to briefly recall the model and its parameters to be estimated. In the model I refers to the process describing how the number of current viewers (system workload) evolves with time. R defines time evolution of the number of past viewers. $\beta > 0$ is the rate of information dissemination per unit of time, $l > 0$ fixes the rate of spontaneous viewers, γ^{-1} is the mean watch time of a video. μ^{-1} denotes the mean active period after which an user stops propagating information. In this framework β can assume two values depending on its state; $\beta = \beta_1$ in the buzz-free state and $\beta = \beta_2 \gg \beta_1$ in buzz state. Transition between these two states occur with rates a_1 and a_2. Since, $I(t)$ and $R(t)$ are point processes it is reasonable to define the time vectors corresponding to the processes. This framework considers $\mathbf{t_a}$ to be the time vector related to arrivals of new viewers. $\mathbf{t_p}$ relates to the times viewers stop watching a video and start to disseminate information. $\mathbf{t_s}$ signifies the time when past viewers stop to spread information. Fig. 4.2 shows how $\mathbf{t_a}$, $\mathbf{t_p}$ and $\mathbf{t_s}$ influence evolution of

current viewers ($I(t)$) and past viewers ($R(t)$) with time. In the upper plot of Fig. 4.2 it is observed that there are arrivals of new viewers at $t_a \approx 375s$ and $t_a \approx 394s$. The upper plot of Fig. 4.2 shows that at $t_p \approx 395s$ one current viewer leaves the system. The corresponding lower plot shows that one past viewer increases at the same instant. It is also observed that one past viewer leaves the system at around $t_s \approx 393s$.

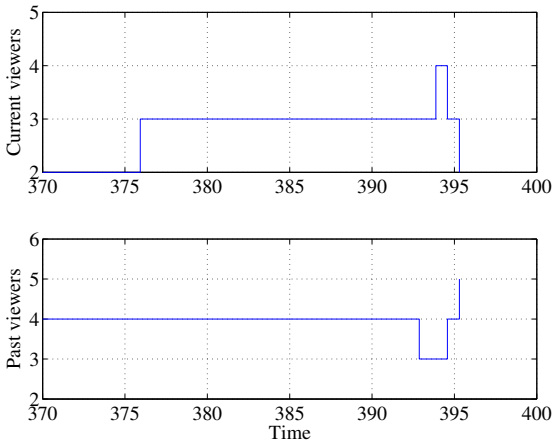


Figure 4.2: Influence of $\mathbf{t_a}$, $\mathbf{t_p}$ and $\mathbf{t_s}$ on the evolution of the current (I) and the past viewers (R).

In the model the observables are $I(t), \mathbf{t_a}$ and $\mathbf{t_p}$. But, it is not possible to observe either $\mathbf{t_s}$ or $R(t)$, since the VoD server can not know how long does a viewer talk about a video. The hidden states (either $\beta = \beta_1$ or $\beta = \beta_2$) are also unobservable.

This framework constructs a set of empirical estimators for each parameter and then numerically evaluate their performance on synthetic traces. This chapter extensively uses likelihood function in the estimation framework. A likelihood function (sometimes refereed to as the likelihood) of a set of parameter values, θ, given outcomes $\mathbf{x}$, equals to the probability of the observed outcomes given the parameter values. Formally $\mathcal{L}(\theta|\mathbf{x}) = p(\mathbf{x}|\theta)$

4.1.2 Heuristic procedure description

It is possible to directly estimate γ from the workload trace, since it solely depends on I, i.e. number of current viewers. Rest of the parameters also depend on the unobservable variable R, i.e. the number of past viewers. Naturally it is not possible to estimate them directly. The following discussion starts by explaining how is γ estimated followed by rest of the parameters.

4.1.2.1 Watching parameter γ estimation

This section begins by deriving the probability density of t_a, t_p, t_s for a buzz-free case (i.e. $\beta = \beta_1$). It is to be noted that the three possible events in a buzz-free regime are 1) arrival of an active viewer, 2) departure of an active viewer (or arrival of a past viewer), 3) departure of a past viewer. The corresponding rates are $\beta_1(I(t) + R(t)) + l, \gamma I(t)$ and $\mu R(t)$ with $I(t)$ and $R(t)$ being the number of active and past viewers at time instant t respectively. The overall rate of the system is thus given by $\Lambda(t|\beta_1, \gamma, \mu, l) = \beta_1(I(t) + R(t)) + l + \gamma I(t) + \mu R(t)$. Now the density of t_a, t_p, t_s is formalised given β_1, l, γ, μ. Likelihood of any type of event is $\Lambda(t|\beta_1, \gamma, \mu, l)p(T_{event} \geq t|\beta_1, \gamma, \mu, l)$. It is known from the model description that the rates, corresponding to the three events follow an exponential distribution. Since the minimum of three exponentials is also an exponential with rate corresponding to the sum of the three:

$$p(T_{event} \geq t|\beta_1, \gamma, \mu, l) = \exp\Big(- \int_0^t \Lambda(x|\beta_1, \gamma, \mu, l)\mathrm{d}x\Big) \tag{4.1}$$

The proof of Eq. (4.1) has been demonstrated in Appendix (A).

It is known that the three events of the system are independent of each other. If there are n_1 first type of events, n_2 second type of events and n_3 third type of events

then the overall likelihood is computed within the time span as:

$$\begin{aligned}
p(\mathbf{t_a}, \mathbf{t_p}, \mathbf{t_s} \mid \mathbf{\Theta}) \;\propto\; & \Big[\prod_{j=1}^{n_1}[\beta_1(I(t_{a_j}^-) + R(t_{a_j}^-)) + l] \exp\Big(-\int_{t_{a_{j-1}}}^{t_{a_j}} \Lambda(t|\beta_1,\gamma,\mu,l)\mathrm{d}t\Big)\Big] \\
\times\; & \Big[\prod_{j=1}^{n_2} \gamma I(t_{p_j}^-) \exp\Big(-\int_{t_{p_{j-1}}}^{t_{p_j}} \Lambda(t|\beta_1,\gamma,\mu,l)\mathrm{d}t\Big)\Big] \\
\times\; & \Big[\prod_{j=1}^{n_3} \mu R(t_{s_j}^-) \exp\Big(-\int_{t_{s_{j-1}}}^{t_{s_j}} \Lambda(t|\beta_1,\gamma,\mu,l)\mathrm{d}t\Big)\Big] \\
=\; & \prod_{j=1}^{n_1}[\beta_1(I(t_{a_j}^-) + R(t_{a_j}^-)) + l] \times \prod_{j=1}^{n_2} \gamma I(t_{p_j}^-) \times \prod_{j=1}^{n_3} \mu R(t_{s_j}^-) \\
\times\; & \Big[\prod_{j=1}^{n_1} \exp\Big(-\textstyle\int_{t_{a_{j-1}}}^{t_{a_j}} \Lambda(t|\beta_1,\gamma,\mu,l)\mathrm{d}t\Big)\Big] \times \Big[\prod_{j=1}^{n_2} \exp\Big(-\textstyle\int_{t_{s_{j-1}}}^{t_{s_j}} \Lambda(t|\beta_1,\gamma,\mu,l)\mathrm{d}t \\
\times\; & \Big[\prod_{j=1}^{n_3} \exp\Big(-\textstyle\int_{t_{s_{j-1}}}^{t_{s_j}} \Lambda(t|\beta_1,\gamma,\mu,l)\mathrm{d}t\Big)\Big] \\
=\; & \prod_{j=1}^{n_1}[\beta_1(I(t_{a_j}^-) + R(t_{a_j}^-)) + l] \times \prod_{j=1}^{n_2} \gamma I(t_{p_j}^-) \times \prod_{j=1}^{n_3} \mu R(t_{s_j}^-) \\
\times\; & \Big[\exp\Big(-\sum_{j=1}^{n_1}\big(\int_{t_{a_{j-1}}}^{t_{a_j}} \Lambda(t|\beta_1,\gamma,\mu,l)\mathrm{d}t\big) - \sum_{j=1}^{n_2}\big(\int_{t_{p_{j-1}}}^{t_{p_j}} \Lambda(t|\beta_1,\gamma,\mu,l)\mathrm{d}t\big) - \\
& \sum_{j=1}^{n_3}\big(\int_{t_{s_{j-1}}}^{t_{s_j}} \Lambda(t|\beta_1,\gamma,\mu,l)\mathrm{d}t\big)\Big)\Big] \qquad (4.
\end{aligned}$$

Here, t^- stands for the time just before t and $\mathbf{\Theta}$ stands for all the parameters to be estimated. Owing to the memoryless property of the exponential distribution and since the events are consecutive in time the sums can be simplified into a single integral. Therefore,

$$\begin{aligned}
p(\mathbf{t_a}, \mathbf{t_p}, \mathbf{t_s} \mid \mathbf{\Theta}) \;\propto\; & \prod_{j=1}^{n_1}[\beta_1(I(t_{a_j}^-) + R(t_{a_j}^-)) + l] \\
& \times \prod_{j=1}^{n_2} \gamma I(t_{p_j}^-) \times \prod_{j=1}^{n_3} \mu R(t_{s_j}^-) \times \\
& \exp\big(-\int_0^T [\beta_1(I(t) + R(t)) + l + \gamma I(t) + \mu R(t)]\mathrm{d}t\big) \qquad (4.3)
\end{aligned}$$

In the model, $(I(t), t \in [0, T])$ is the only observation that can be accessed to calibrate the proposed model. From this, it is possible to readily identify the instants $\{t_{a_n}\}_{n=1,\ldots n_1}$ and $\{t_{p_n}\}_{n=1\ldots n_2}$ at which individuals enter and leave the state I, respectively. As the

exponential parameter γ of the watching time only depends on the sojourn time in I, it can then straightforwardly be estimated with a maximum likelihood procedures described here. Eq. (4.3) is differentiated with respect to γ and solved for 0 to obtain:

$$\hat{\gamma}_{\text{MLE}} = \frac{n_2}{\int_0^T I(t)\,\mathrm{d}t} \tag{4.4}$$

In contrast to γ though, all other parameters of the model rely on the unobserved time series $(R(t), t \in [0, T])$, or both $I(t)$ and $R(t)$. More precisely many parameters depend on the unknown departure instants from state R, that are denoted as $\{t_{s_n}\}_{n=1,\ldots,n_3}$. With this incomplete dataset, it is not possible to employ a maximum likelihood estimate in the form of (4.4) to estimate rest of the parameters.

4.1.2.2 Memory parameter μ estimation

μ defines the rate at which past viewers stop propagating the information about a video. It relates to the decrement density of the non-observed process $R(t)$. It is thus impossible to simply apply the Maximum Likelihood estimator as previously done in Eq. (4.4) unless a substitute $\widehat{R}(t)$ is constructed first to the missing data from the observable data set $I(t)$. It can be recalled that in the model, all current viewers turn and remain contagious for a mean period of time $\gamma^{-1} + \mu^{-1}$. Then, in first approximation, it can be considered that $R(t)$ derives from the finite memory cumulative process:

$$\widehat{R}(t) = \int_{t-(\gamma^{-1}+\mu^{-1})}^{t} I(u)\,\mathrm{d}u, \tag{4.5}$$

Evidently this approach depends on the parameter to be estimated, μ.

This framework proposes an estimation procedure based on the inherent exponential property of the model. From the Poisson assumption, the inter-arrival time $\mathbf{w}$ between the consecutive arrivals of two new viewers is an exponentially distributed random vari-

able such that $\mathbb{E}\left(\mathbf{w}|\, I(t) + R(t) = x\right) = (\beta\, x + l)^{-1}$. It means that, for fixed x, the normalized random variable $\widetilde{\mathbf{w}} = \mathbf{w}/\mathbb{E}(\mathbf{w}|x)$ is exponentially distributed with unitary parameter and becomes independent of x.

Therefore, for each value of $R(t) + I(t) = x$, all the sub-series $\mathrm{w}_x = \{w_n \ : \ R(t_n) + I(t_n) = x\}$, after normalization by their own empirical mean, yield independent and identically distributed (iid) realizations of a unitary exponential random variable. In practice, since $R(t)$ is not observable, this unitary exponential i.i.d. assumption would not be valid unless $\widehat{R}(t)$ is accurately estimated. Based on this property, this work proposes the following sequence of steps:

- Consider different values of μ spanning a *reasonable* interval based on γ,
- Compute $\widehat{R}_\mu(t)$ using Eq. (4.5) with the assumed value of μ,
- Build the normalized series $\widetilde{\mathbf{w}} = \widetilde{\mathbf{w}}_\mu$ for each value of μ using previously computed $\widehat{R}_\mu(t)$,
- Apply the Kolmogorov-Smirnov (K-S) statistical test (described in this section) on each $\widetilde{\mathbf{w}}_\mu$ to assess the exponential iid hypothesis,
- Select the value of μ that yields the best score.

The statistical K-S based test is derived in [29], which compares two probability distributions. The K-S test statistic quantifies a distance between the two empirically estimated cumulative distribution functions. The heuristic approach applies this test on $\widetilde{\mathbf{w}}_\mu = (\widetilde{w}_n)_{n=1,\ldots,N}$ and calculates the *normalized spacings* $\mathbf{v}_\mu = \left(v_{(n)} = (N-n+1)(\widetilde{w}_{(n)} - \widetilde{w}_{(n-1)})\right)_{n=}$ where $\left(\widetilde{w}_{(n)}\right)_{n=1,\ldots,N}$ stands for $\widetilde{\mathbf{w}}_\mu$ rearranged in ascending order. If F and G denote the cumulative distribution functions of $\widetilde{\mathbf{w}}_\mu$ and $\mathbf{v}_\mu$ respectively, then the classical Kolmogorov-Smirnov distance is defined as follows:

$$T_\mu = \tfrac{1}{\sqrt{N}} \sup_{1\le k\le N} |F(k) - G(k)|. \tag{4.6}$$

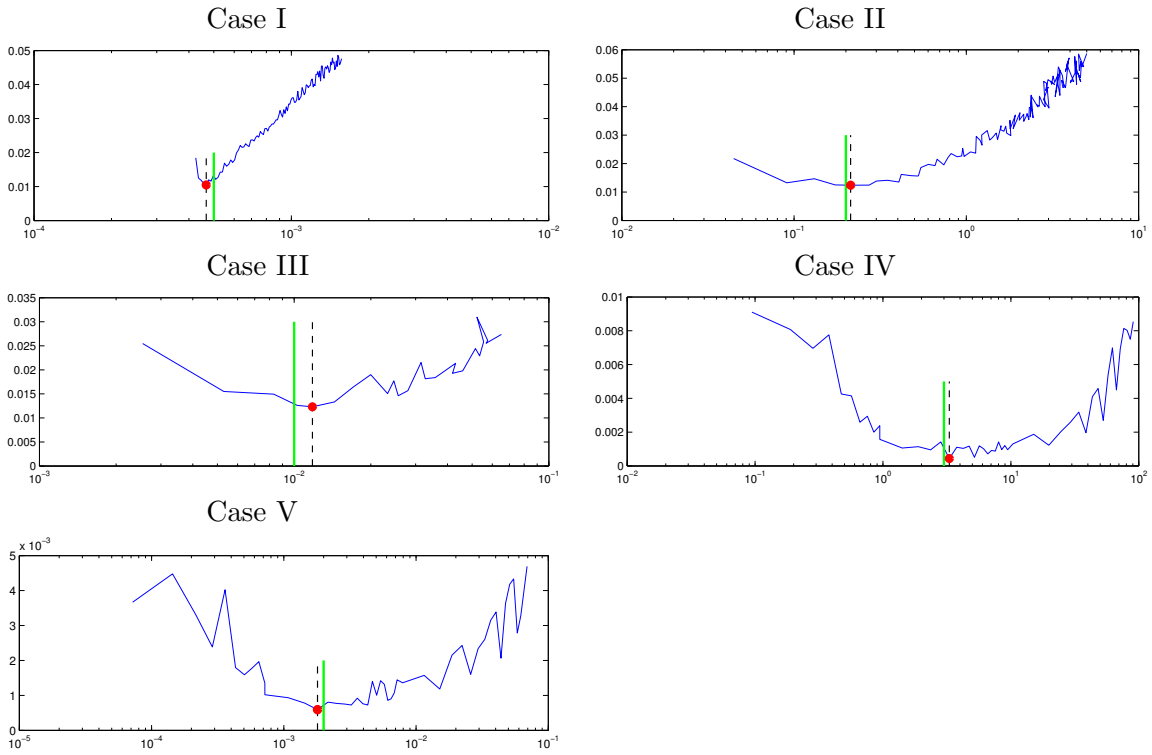


Figure 4.3: The vertical axis represents the K-S distance and the horizontal axis represents the μ values. The red circles in the plots represent the estimated μ and the intersections of the curves and the green line represents the actual value of μ (which are 5×10^{-4}, 0.2, 0.01, 3.0 and 0.002 respectively).

Since, F and G are identical for an exponentially i.i.d. random series, T_μ is expected to reach its minimum for the value of μ that gives the best estimate $\widehat{R}_\mu(t)$ of $R(t)$:

$$\widehat{\mu} = \text{argmin}_\mu \left(T_\mu \right) \text{ and } \widehat{R} = \widehat{R}_{\widehat{\mu}}. \tag{4.7}$$

Plots of Fig. 4.3 show the evolution of the Kolmogorov-Smirnov distance for different values of μ. Five different traces have been used, which are mentioned in previous chapter (Model Description). In all cases, T_μ clearly attains its minimum bound for a $\widehat{\mu}$ (represented by the red circle and the black dotted line in Fig. 4.3) which is close to its true value (represented by the green line in Fig. 4.3).

The corresponding estimated process $\widehat{R}(t)$, derived while estimating μ for one case

has been displayed in Fig. 4.4. Other cases follow the same trend and not included to avoid redundancy. Evidently $\widehat{R}(t)$ and $R(t)$ match fairly well and validates the proposed approach to reconstruct $\widehat{R}(t)$ from $I(t)$ and μ. Plot II of Fig. 4.4 zooms on a particular period of Plot I and compares the actual and the reconstructed process at a smaller scale.

From Eq. (4.5) it can be observed that estimation of R depends on $\hat{\mu}$. For larger values of $\hat{\mu}$ this accuracy decreases (the reconstruction process misses many events when there is a decrease in R). This problem has been resolved in the second approach using the MCMC (to be discussed in the next sub-chapter).

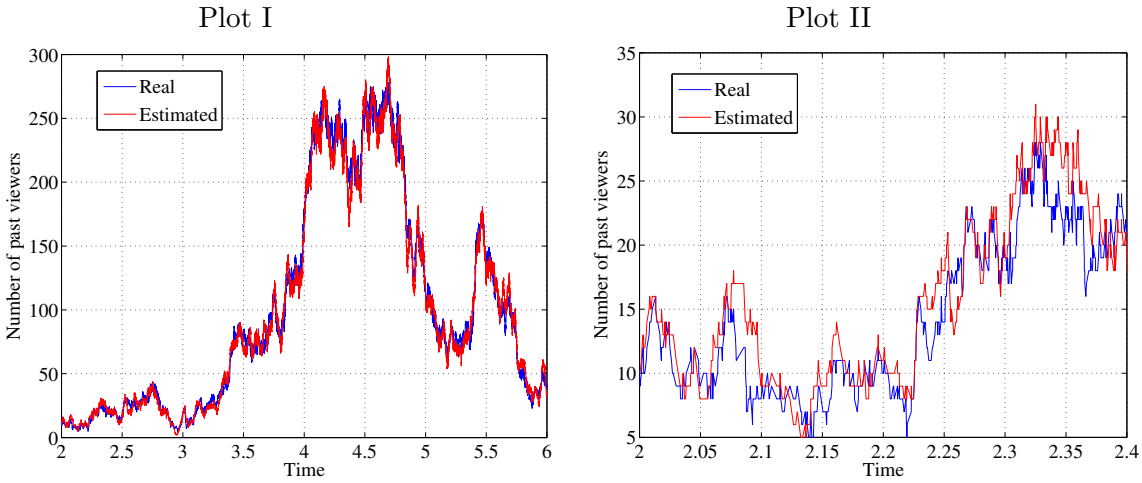


Figure 4.4: Evolution of the number of past viewers (vertical axis) vs. time in hrs (horizontal axis).

4.1.2.3 Propagation parameters β and l estimation

According to the proposed model, the arrival rate of new viewers $\lambda(t)$ is $\beta(I(t)+R(t))+l$. $\lambda(t)$ linearly depends on the current number of active and past viewers. Therefore, from the observation $I(t)$ and the reconstructed process $\widehat{R}(t)$, it is possible to formally apply the maximum likelihood (as done previously for γ) to estimate β. However, in practice, it is required to keep in mind the following facts:

- the arrival process of rate $\lambda(t)$ comprises a spontaneous viewers ingress that is governed by parameter l. It is independent of the current state of the system,
- depending on the current hidden state of the model (buzz-free *versus* buzz state), it is alternately $\beta = \beta_1$ and $\beta = \beta_2$ that determines the arrival rates of the new viewers.

In order to address these two issues this work proposes an estimation procedure based on a weighted linear regression. This approach can be broken down in the following two steps:

First, this approach considers only the buzz-free state, where $\beta = \beta_1$. As discussed in the estimation of μ, the inter-arrival time $\mathbf{w}$ between the consecutive arrivals of two new viewers is an exponentially distributed random variable such that $\mathbb{E}(\mathbf{w}|\, I(t) + R(t) = x) = (\beta\, x + l)^{-1}$. Concretely then, for different values of the sum $I(t) + \widehat{R}(t)$, it is possible to calculate the conditional empirical mean:

$$\Omega(x) = \tfrac{1}{|\mathrm{w}_x|} \sum_{t_n \in \mathrm{w}_x} w_n \;:\; \mathrm{w}_x = \{w_n : I(t_n) + \widehat{R}(t_n) = x\} \tag{4.8}$$

The linear regression of $(\Omega(x))^{-1}$ against x yields simultaneous estimation of both parameters $\widehat{\beta}$ (slope) and $\widehat{l}$ (intercept) (see Fig. 4.5).

In the buzz-free case, $\beta = \beta_1$ corresponds to a normal workload activity, meaning that the sum $I(t) + \widehat{R}(t)$ takes on rather moderate values. Conversely, when the system undergoes a buzz (i.e. $\beta = \beta_2$) then the population $I(t) + \widehat{R}(t)$ suddenly increases to attain significantly larger values. But, in both cases, the quantity $(\Omega(x))^{-1}$ remains linear with x, but with two different regimes (slopes) depending on the amplitude of $I(t) + \widehat{R}(t) = x$.

Clearly, β_2 adds a bias to the estimation β_1. In order to reduce that a weighted linear regression of Ω^{-1} *vs* x has been used where the weights $p(x)$ are proportional to the cardinal of the indicator sets w_x. Indeed, $|\mathrm{w}_x|$ should be smaller for larger values of

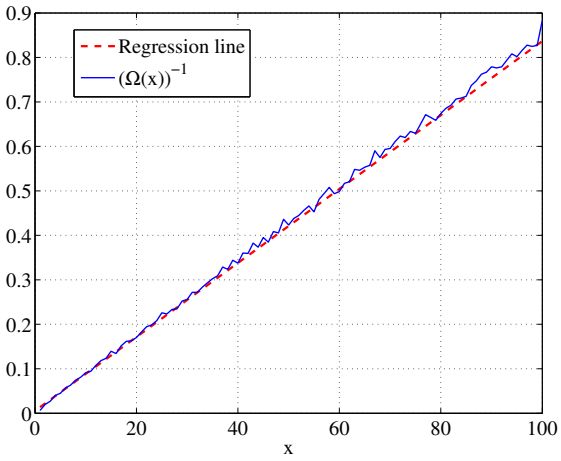


Figure 4.5: Linear regression of $(\Omega(x))^{-1}$ against x to obtain β_1 and l.

x because buzz episodes are expected to be less frequent than nominal activity periods.

It is possible to apply the exact same procedure to estimate β_2, but considering opposite weights to favor the large values of x's. However, due to the large fluctuations of $(\Omega(x))^{-1}$ in the corresponding region, the slope $\widehat{\beta_2}$ is subject to a very poor estimation variance. Instead, this work proposes to apply the ML estimator like it did for estimating γ on the restriction of $I(t)$ to the buzz periods only. Strictly speaking, it is reasonable to consider $\widehat{R}(t)$ as well, but since a buzz event normally occurs on very small interval of time, it is assumed that $\widehat{R}(t)$ remains constant in the meanwhile (flash crowd viewers will enter in R compartment only after the visualization time).

Understandably the buzz regime is mostly dominated by information propagation (β_2) among viewers, rather than spontaneous arrival of new viewers due to l. Therefore, this approach manages to use the ML estimator directly as discussed before. In practice, to automatically identify the buzz periods, it is required to put a reasonably high threshold value of $I(t)$ based on the observation of the trace and consider only the persistent increasing parts that remain above the threshold. This is clearly a limitation of this approach since the shareholding is done arbitrarily according to the experience of the practitioner.

Fig. 4.6 shows one example of how β_2 can be estimated from a sample trace. However, it is an arbitrary example and therefore no numbers have been provided regarding the threshold value or the β_2 value here. In this case a reasonably high threshold has been selected, therefore only the time between T_1 and T_2 have been considered to estimate β_2. If there are n_4 instances of arrivals of a new viewer within this period, it is possible to formulate the MLE of β_2 as follows:

$$\hat{\beta}_{2_{\mathrm{MLE}}} = \frac{n_4}{\int_{T_1}^{T_2} (I(t) + \widehat{R}(t))\,\mathrm{d}t} \tag{4.9}$$

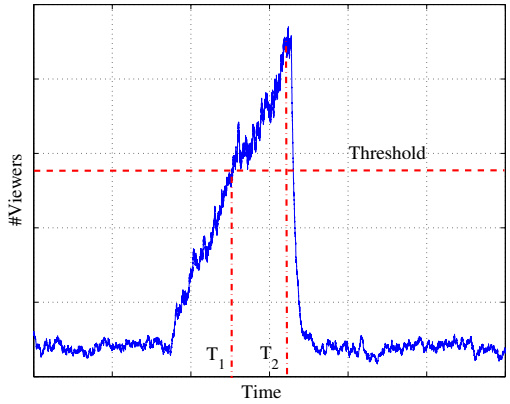


Figure 4.6: An example to estimate β_2 using the ML estimator.

4.1.2.4 Transition rates a_1 and a_2 estimation

Now the estimation of a_1 and a_2 is discussed knowing the rest of the parameter values. Evidently, at time t, the inter-arrival time $\mathbf{w}$ separating two new incomers is a random variable drawn from an exponential law of parameter $\lambda = \beta(i + r) + l$, where β is either equal to β_1 or to β_2. $f_1(\mathbf{w})$ and $f_2(\mathbf{w})$ are denoted as the corresponding densities built upon the reconstructed process $\widehat{R}(t)$ and the estimated parameters $(\widehat{\beta_1}, \widehat{l})$ and $(\widehat{\beta_2}, \widehat{l})$ respectively. For a given inter-arrival time $\mathbf{w} = w_n$ observed at time t_n, the likelihood

ratio $f_2(w_n)/f_1(w_n)$ is formed to determine whether the system is in buzz or in buzz-free state. Moreover, in order to avoid non-significant state transitions this approach resort to a restoration method inspired by the Viterbi algorithm [33]. The Viterbi algorithm is used to find the most likely sequence of hidden states, known as the Viterbi path, which results in a sequence of observed events. This algorithm is extensively used for the Markov information sources and hidden Markov models. Once the hidden states of the process are identified, it becomes trivial to estimate the transitions rates $\widehat{a_1}$ and $\widehat{a_2}$ from the average times spent in each state.

4.1.3 Results

This estimation procedure has been validated against the synthetic traces with five different workloads (ref:Fig. 3.7) as shown in the previous chapter. The estimation procedure is applied on each of the traces. For each parameter a so called "descriptive statistics" have been obtained. It describes the smallest observation (sample minimum), lower quartile, median, upper quartile, and largest observation (sample maximum) from the box-and-whisker plot. Fig. 4.7 shows a sample box plot.

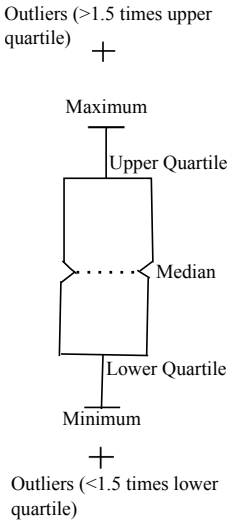


Figure 4.7: A sample box plot to interpret the descriptive statistics

The box plots of Fig. 4.8 indicate for each estimated parameter (centred and normalized by the corresponding actual value) the descriptive statistics obtained from an arbitrarily chosen and reasonably long time series of length 2^{21} points. As expected (from maximum likelihood), estimation of γ shows to be the most accurate, both in terms of bias and variance. Surprisingly, although the estimation $\widehat{\beta_1}$ derives from a heuristic procedure that itself depends on the raw approximation $\widehat{R}(t)$ of Eq. (4.5), the resulting performance is remarkably good: bias is always negligible (less than 5% in the worst cases I, III and IV) and the variance always confines to less than 10% interval. Notice also that the estimation of β_1 goes from a slight underestimation in case I to a slight overestimation in case V, as the buzz effect. Moreover the corresponding workload grows from traces from case I to case V. Compared to $\widehat{\beta_1}$, the estimation of β_2 behaves more poorly and proves to be the hardest parameter to estimate. This is consistent with the fact that this latter is only based on buzz periods which represent only a small fraction of the entire time series. Regarding the parameter μ, its estimation remains within a 15% inter-quartile range but all cases show a systematic bias (median hits the lower or upper quartile bound). Remind that the procedure, to determine $\widehat{\mu}$ selects within some discretized interval, the value of μ that yields the best T_μ score. It is then very likely that the true value does not coincide with any sampled point of the interval and therefore, the procedure picks the closest one that systematically lies beneath or above. However, it is possible to recursively refine this estimator by focusing the interval around the estimated value, which yields the best score. But it comes with a heavier computational cost. Finally, estimation of the transition parameters a_1 and a_2 between the two hidden states relies on all other parameters estimation and therefore gets impacted by all. Nonetheless and despite a systematic underestimating trend, precision remains within a very acceptable confidence interval.

Convergence rate of the empirical estimators is another important feature that binds the estimate precision to the amount of available data. Variance and bias of each esti-

mated parameters have been plotted against the length N of the observable time series (From Fig. 4.9 to Fig. 4.15) . Since the purpose is to stress the rate of convergence of these quantities towards zero, to ease the comparison, variance and bias of each parameter have been normalized by its particular value at maximum data length (i.e 2^{21} points here). Then, the estimator's rate of convergence α_θ corresponds to the decaying slope of the variance with respect to N in a *log-log* plot, i.e. variance$(\widehat{\theta}) \sim O(N^{-\alpha_\theta})$.

Fig. 4.9 shows that for β_1 the convergence rate for variance varies between -0.6 (Case II) to -0.9 (Case III and IV). Convergence rate of variance for β_2 (Fig. 4.10) is maximum for Case III and Case IV, which is around -0.8. Being an optimal estimator (maximum likelihood estimator) this rate of γ (Fig. 4.11) is almost -1.0 for all five cases. Naturally this rate is lower for μ and l (Fig. 4.12 and Fig. 4.13 respectively) and is around -0.6. Surprisingly convergence rate of a_1 (Fig. 4.14) is considerably high for some cases (Case I and Case V), whereas it is expectedly low for a_2 (Fig. 4.15) in all cases.

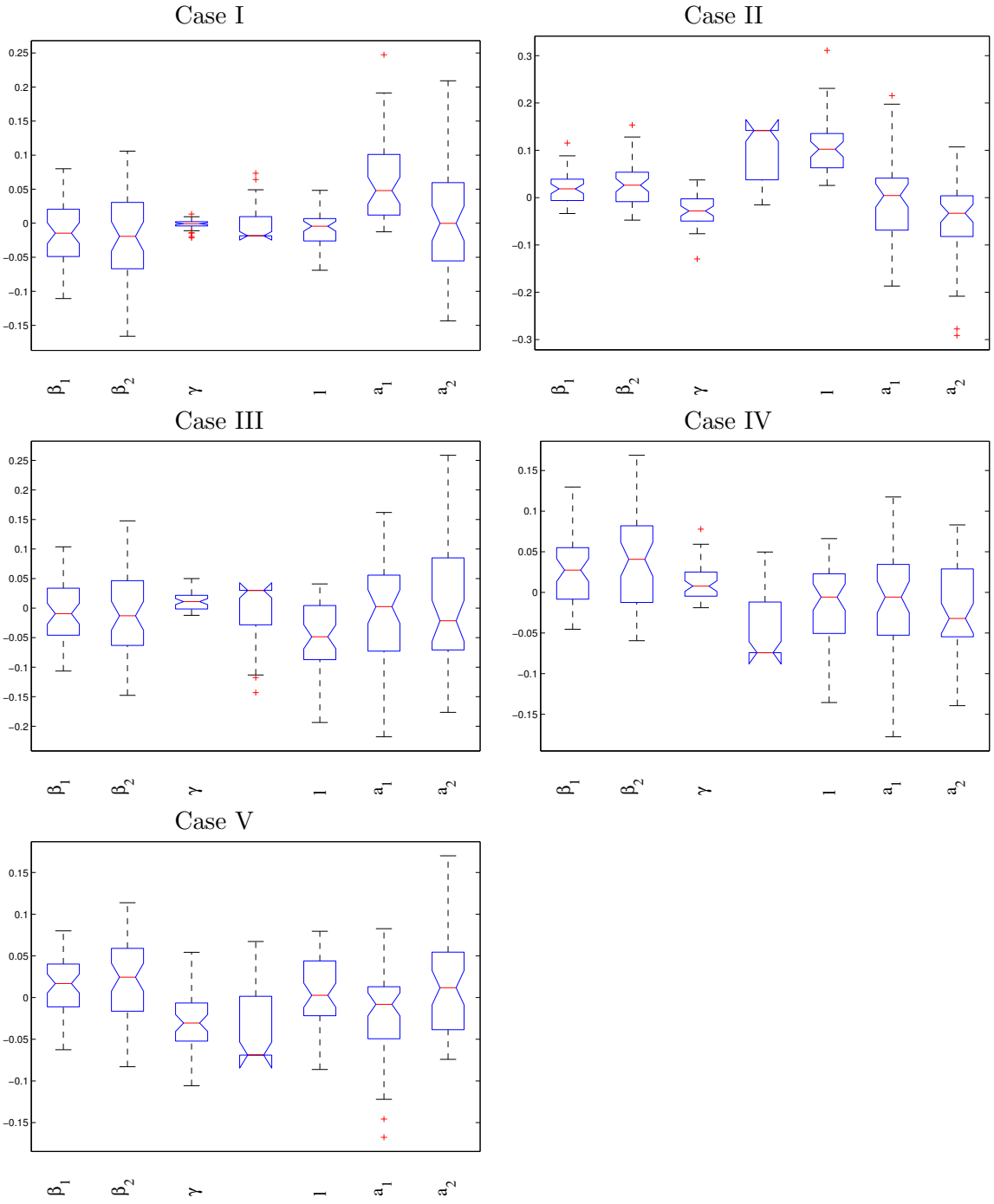


Figure 4.8: Relative precision of estimation of the model parameters. Cases I to V correspond to the configurations reported in Chapter 3. Statistics are computed over 50 independent realizations of time series of length 2^{21} points.

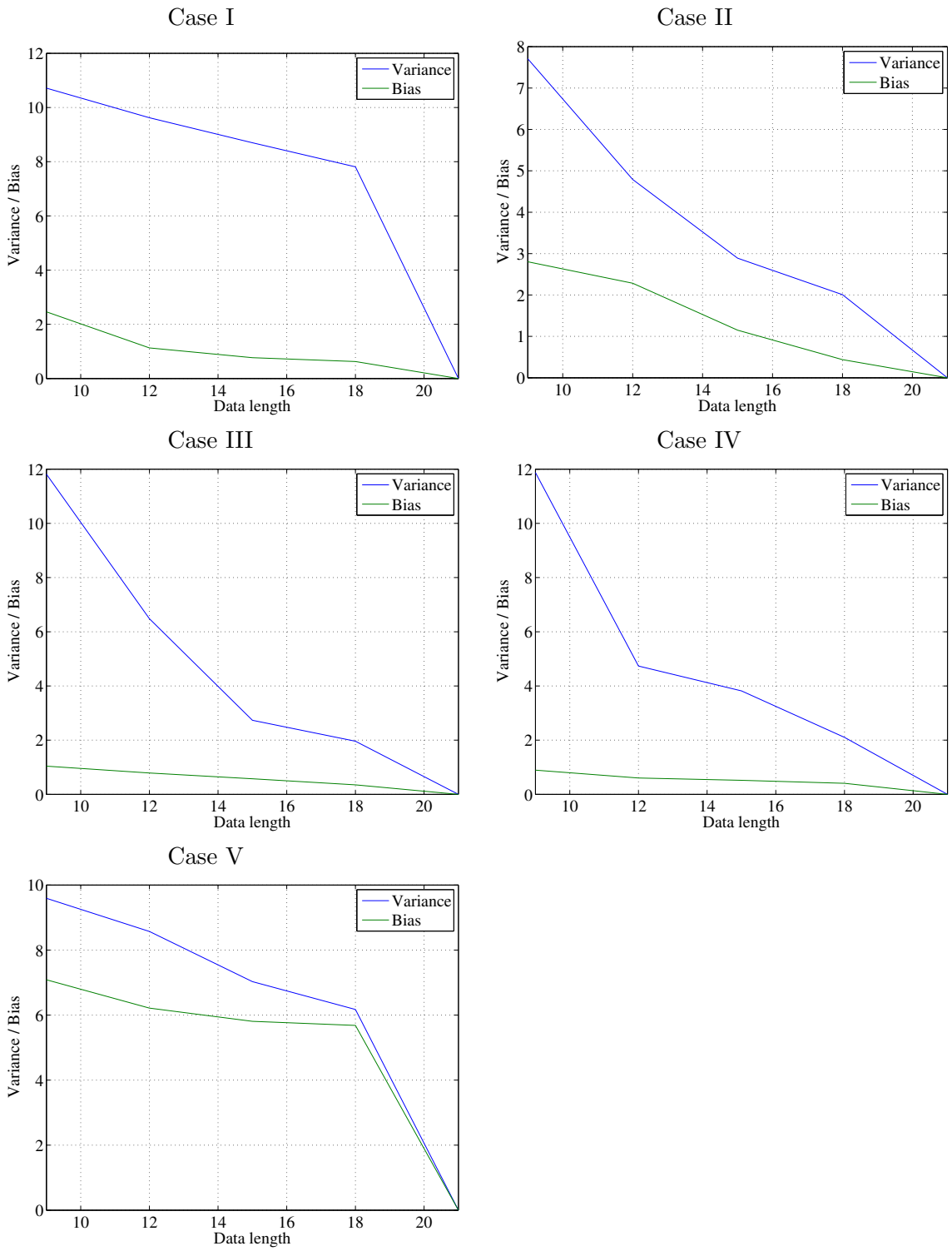


Figure 4.9: Evolution of the Variance and Bias for β_1 *against* the data length N in a *log-log* plot for the 5 traces for the heuristic procedure.

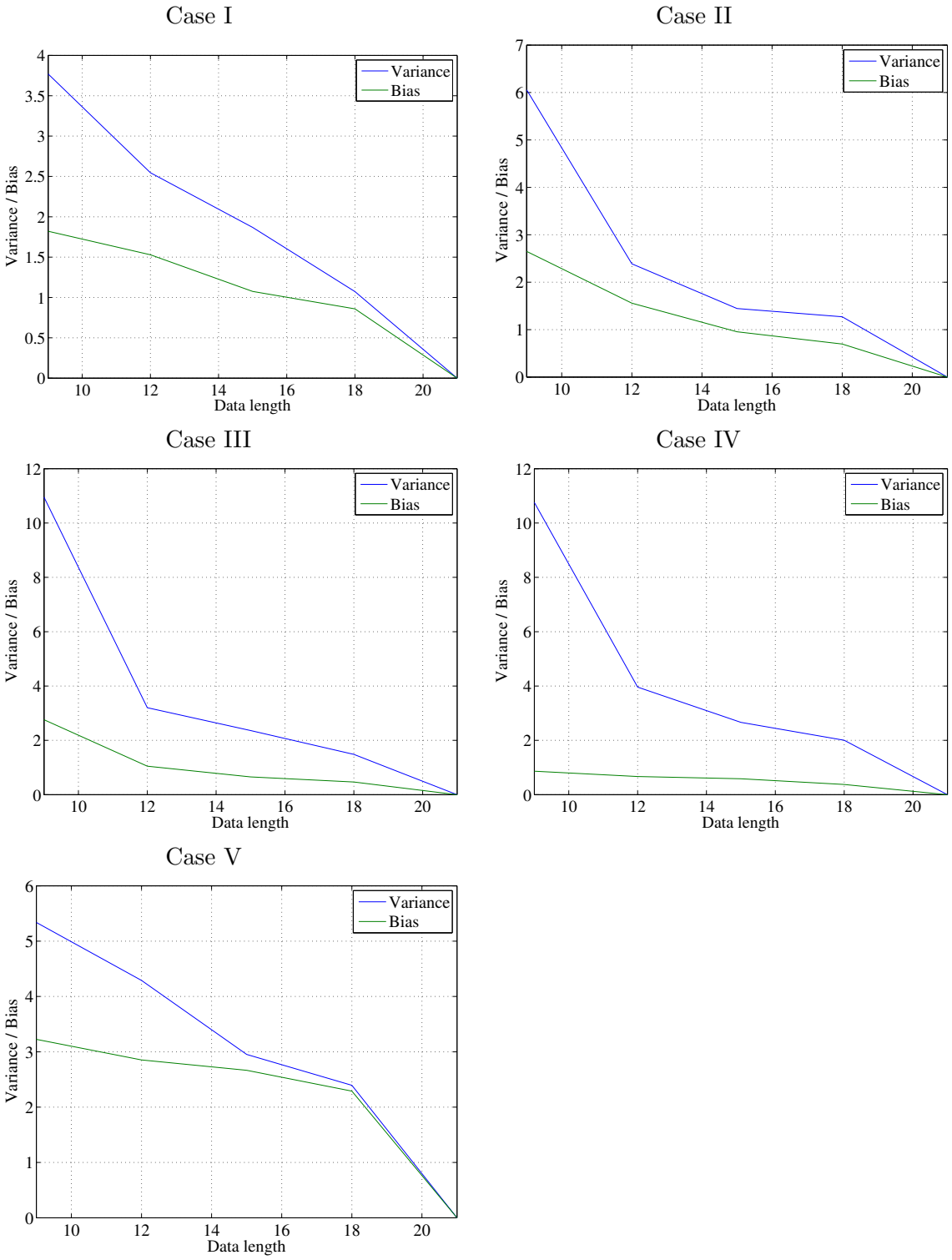


Figure 4.10: Evolution of the Variance and Bias for β_2 *against* the data length N in a *log-log* plot for the 5 traces for the heuristic procedure.

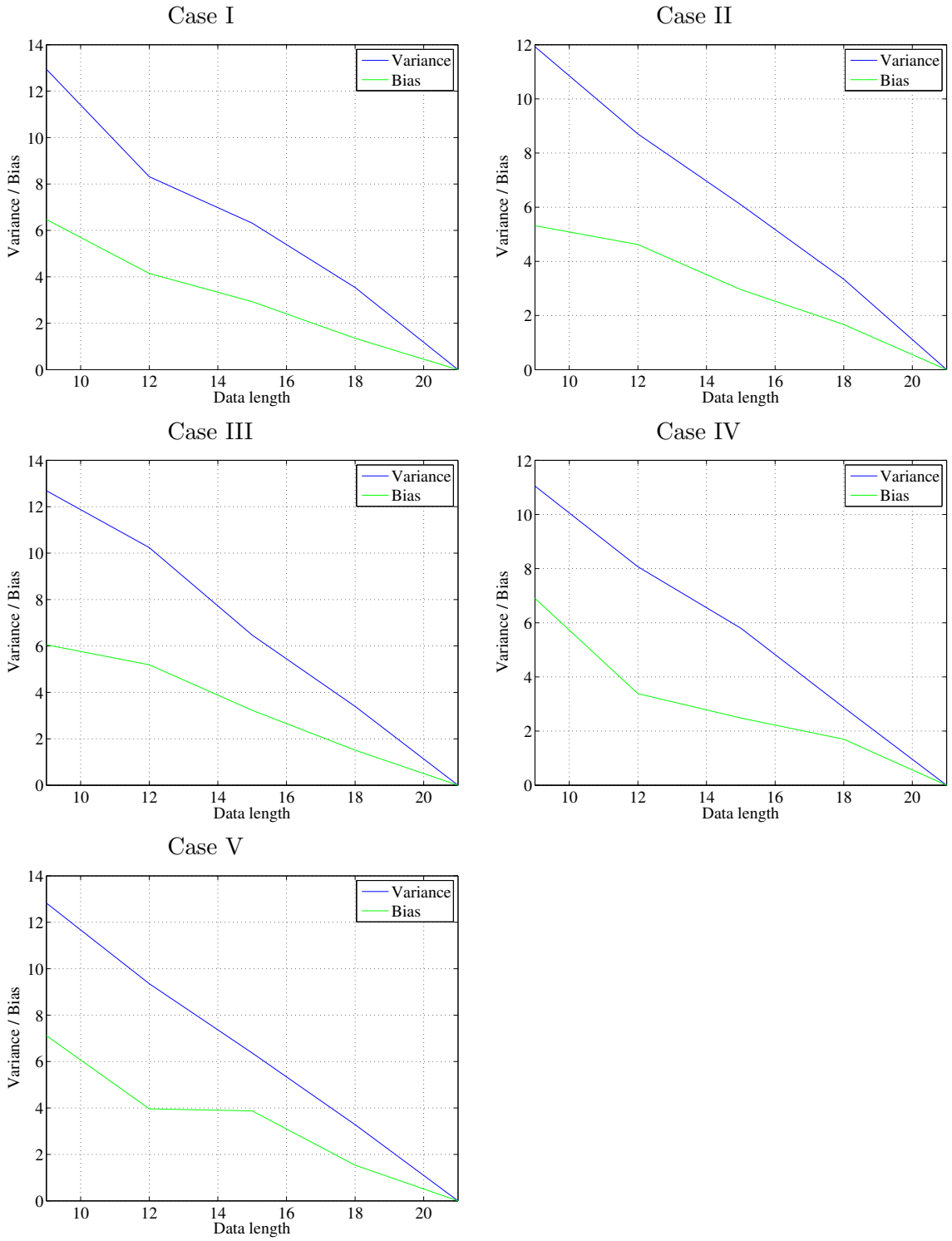


Figure 4.11: Evolution of the Variance and Bias for γ *against* the data length N in a *log-log* plot for the 5 traces for the heuristic procedure.

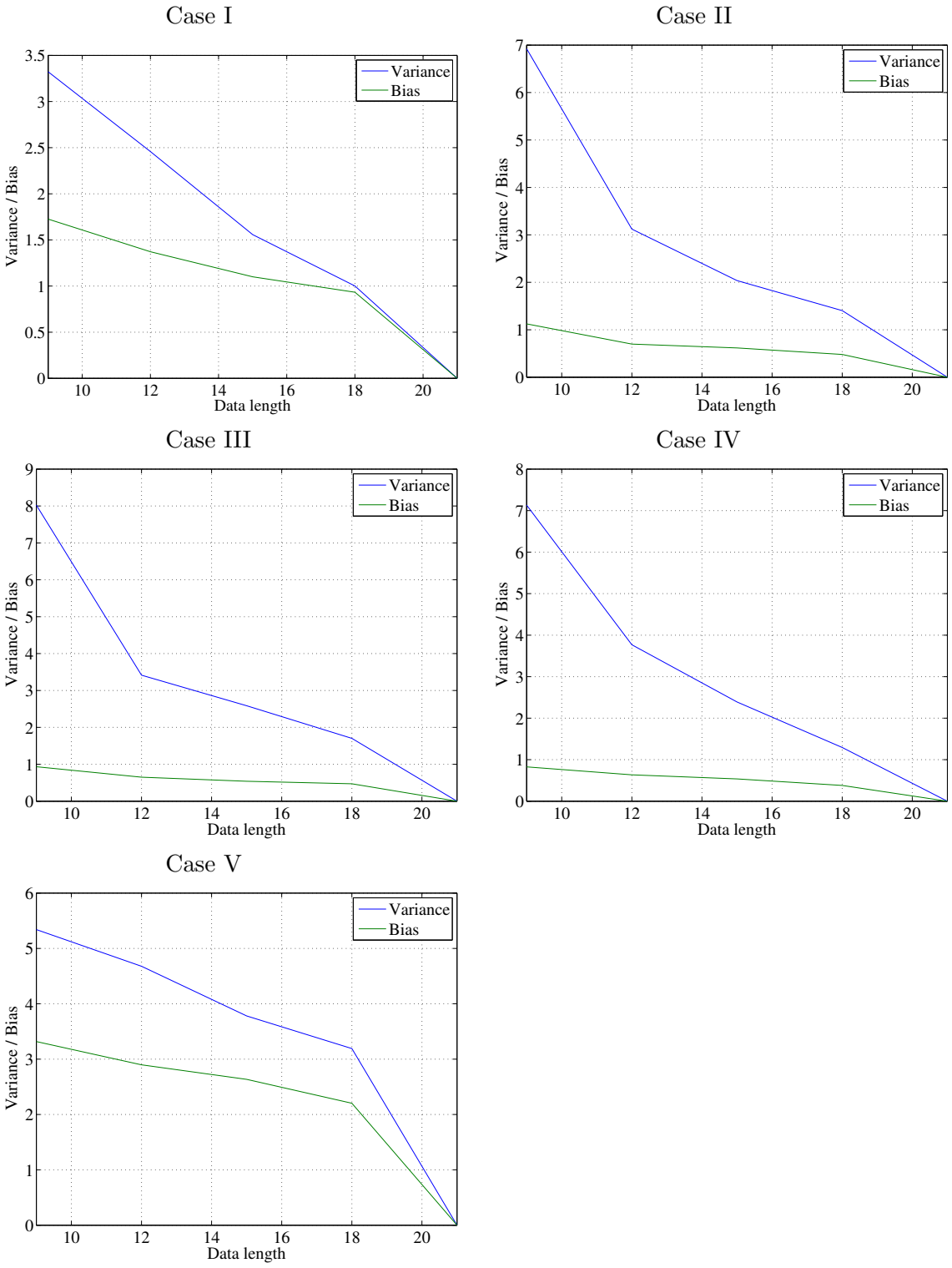


Figure 4.12: Evolution of the Variance and Bias for μ *against* the data length N in a *log-log* plot for the 5 traces for the heuristic procedure.

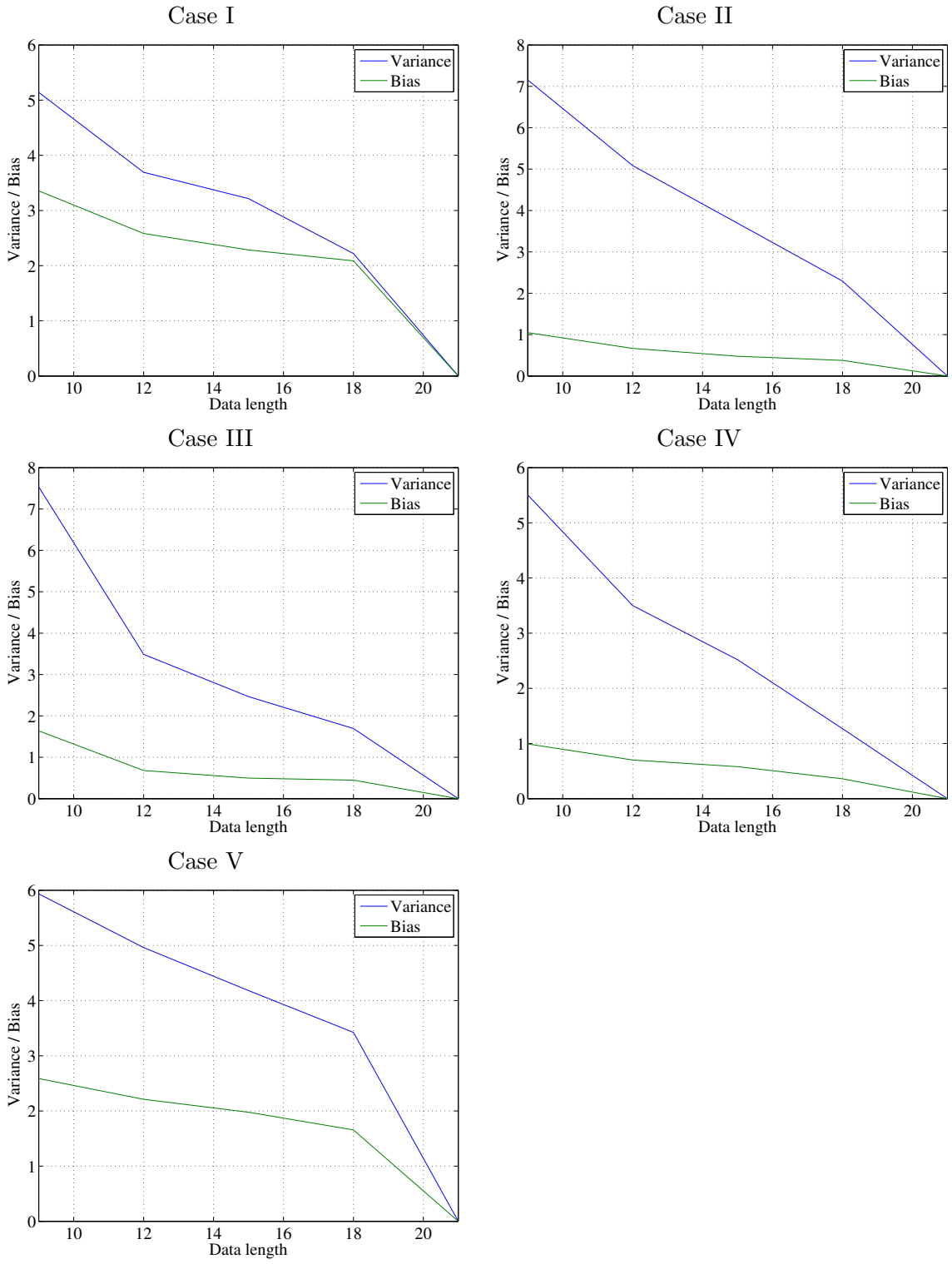


Figure 4.13: Evolution of the Variance and Bias for l *against* the data length N in a *log-log* plot for the 5 traces for the heuristic procedure.

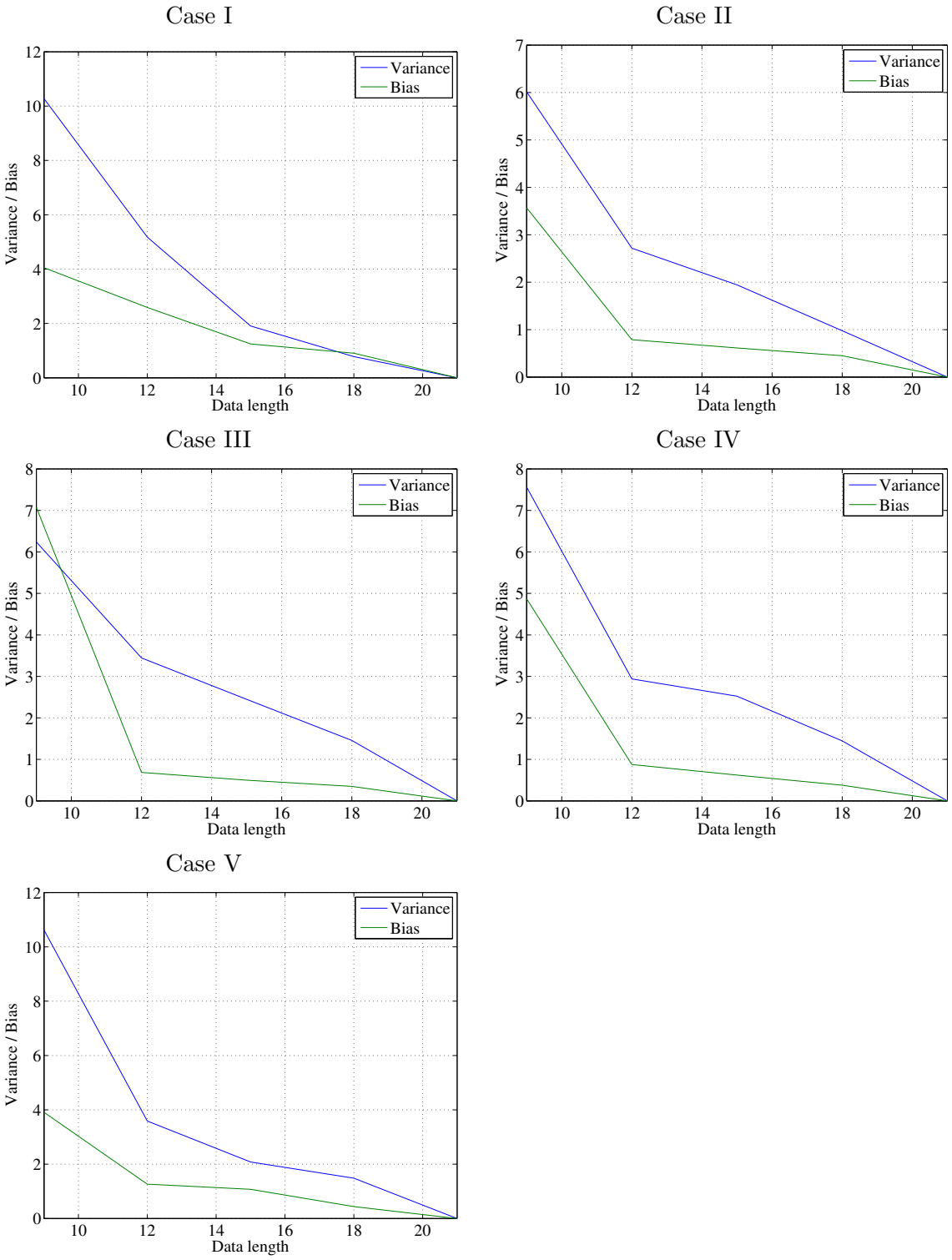


Figure 4.14: Evolution of the Variance and Bias for a_1 *against* the data length N in a *log-log* plot for the 5 traces for the heuristic procedure.

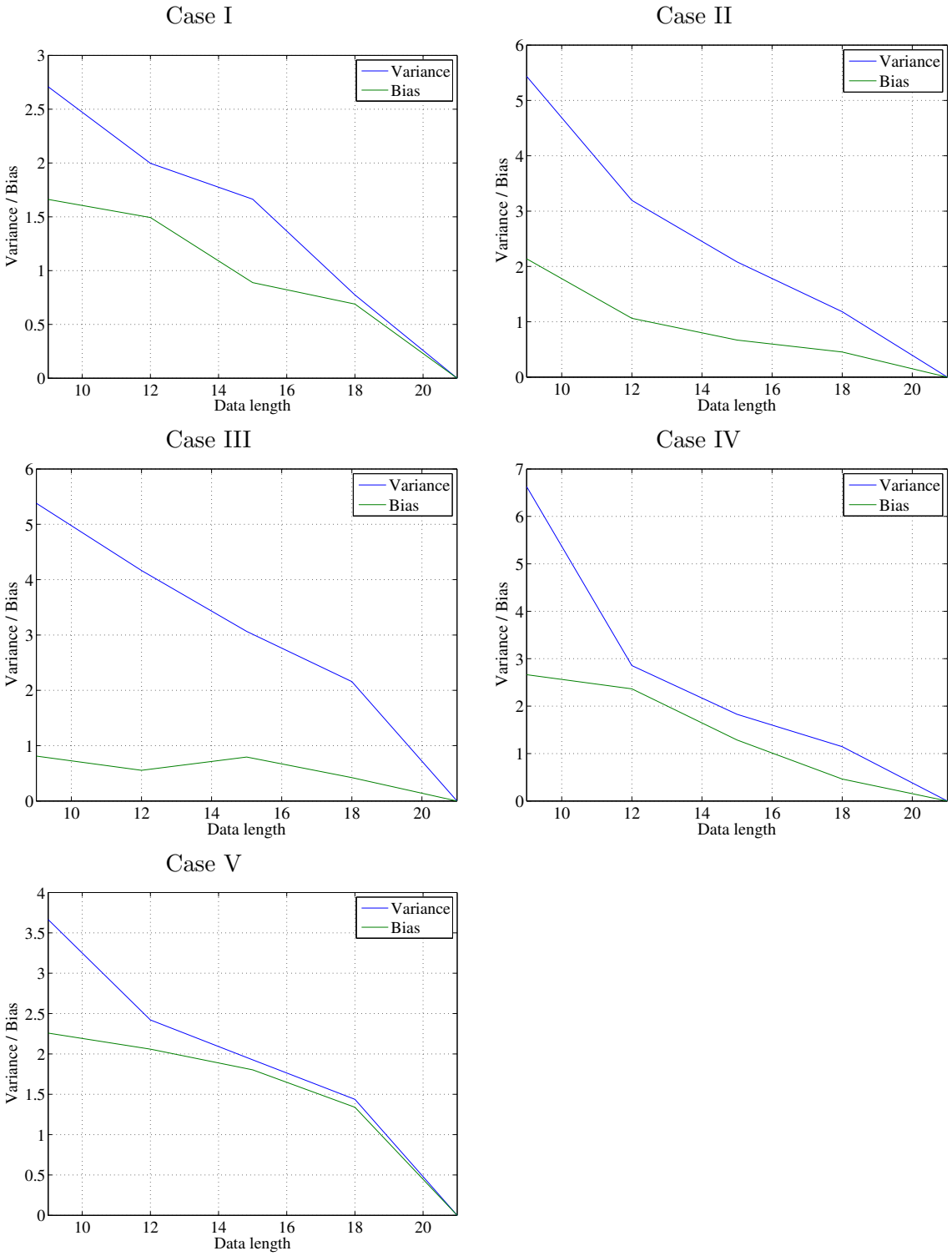


Figure 4.15: Evolution of the Variance and Bias for a_2 *against* the data length N in a *log-log* plot for the 5 traces for the heuristic procedure.

4.2 Model Parameter estimation: an MCMC approach

4.2.1 A brief introduction to Markov Chain Monte Carlo

A Markov Chain Monte Carlo (MCMC) is a sophisticated method, based on the Law of Large Numbers for Markov chains. It can be explained with the following example:

Suppose it has been desired to approximate $\mathbb{E}(Y)$ and there is an algorithm that generates successive states $X_1, X_2, \ldots$ of a Markov chain on a state space χ with stationary distribution π. If there is a real valued function $f : \chi \to \mathbb{R}$ such that:

$$\sum_{x \in \chi} f(x)\pi(x) = \mathbb{E}(Y)$$

then the sample average

$$\frac{1}{n}\sum_{j=1}^{n} f(X_j)$$

may be used as an estimator of $\mathbb{E}(Y)$.

Briefly an MCMC method can be summarized as an algorithm to generate samples from a *target* distribution of interest. MCMC methods are commonly used to estimate parameters of a given model when missing data needs to be inferred. Typically, the target distributions coincide with the posterior distributions of the parameters to be estimated. If I is the observable data and it is required to estimate the model parameters $\mathbf{\Theta}$, the posterior distribution of $\mathbf{\Theta}$ derives from the Bayes rule:

$$p(\mathbf{\Theta} \mid I) \propto p(\mathbf{\Theta}) \cdot p(I \mid \mathbf{\Theta}). \tag{4.10}$$

Here, $p(\mathbf{\Theta})$ is the pdf of the prior distribution of $\mathbf{\Theta}$ and $p(I \mid \mathbf{\Theta})$ is the likelihood of $\mathbf{\Theta}$. As in general, $p(\mathbf{\Theta})$ is unknown, a standard practice [56] [71] in MCMC algorithms is to choose adequate conjugate priors that multiplied with the likelihood yield computationally convenient posterior distributions.

In context of this thesis $\boldsymbol{\Theta}$ denotes the model parameters which are considered as random variable to be estimated.

There are several algorithms in MCMC family, the Metropolis and the Gibbs algorithms being definitely the most widely used in practice.

When the full conditionals for each parameter cannot be obtained easily, the Metropolis algorithm is used for sampling from the posterior distribution. This algorithm produces a Markov chain whose values approximate a sample from the posterior distribution. For this algorithm, a function is required describing the posterior $p(\boldsymbol{\Theta}|\ I)$ for $\boldsymbol{\Theta}$, the parameter(s) of interest. A proposal (or instrumental) distribution q is also needed which is easy to sample from. To closely match to the actual posterior distribution, at each step, the new sample is accepted (otherwise the previous draw is kept) with a probability given by the Metropolis ratio α.

This algorithm can be summarized as follows:

- Specify an initial value for $\boldsymbol{\Theta}$, say $\boldsymbol{\Theta}^{(0)}$
- After iteration k, suppose the most recently drawn value is $\boldsymbol{\Theta}^{(k)}$
- Sample a candidate value $\boldsymbol{\Theta}^*$ from the instrumental distribution
- $(k+1)^{th}$ value in the chain would be

$$\boldsymbol{\Theta}^{(k+1)} = \begin{cases} \boldsymbol{\Theta}^* & \text{with probability } \alpha = \min\left\{1, \frac{p(\boldsymbol{\Theta}^*|I)}{p(\boldsymbol{\Theta}^{(k)}|I)} \cdot \frac{q(\boldsymbol{\Theta}^{(k)}|\boldsymbol{\Theta}^*)}{q(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}^{(k)})}\right\} \\ \boldsymbol{\Theta}^{(k)} & \text{with probability 1 - } \alpha \end{cases}$$

- Continue until convergence

When the instrumental distribution is symmetric, i.e. $q(\boldsymbol{\Theta}^{(k)}|\boldsymbol{\Theta}^*) = q(\boldsymbol{\Theta}^*|\boldsymbol{\Theta}^{(k)})$ then the metropolis ratio is $\alpha = \min\{1, \frac{p(\boldsymbol{\Theta}^*|I)}{p(\boldsymbol{\Theta}^{(k)}|I)}\}$.

As for the Gibbs sampler, it is mainly used when the $\mathbf{\Theta}$ parameter of the model is multi-dimensional. Suppose $\mathbf{\Theta} = \{\theta_1, \theta_2, \ldots\}$. If someone wants to sample from $p(\mathbf{\Theta})$, he/she can use Gibbs sampler to sample from the joint distribution, provided he/she knows the **full conditional** distributions of each parameters. For each parameter, the **full conditional** distribution is the distribution of the parameter conditional on the known information and all other parameters: $p(\theta_j|\theta_{-j}, I)$. Whenever these conditional posteriors are hard to sample, instrumental laws can be used, leading to the so-called **Metropolis within Gibbs sampler**. Metropolis with Gibbs sampler has been extensively used in this chapter.

4.2.2 Calibration framework using MCMC

As discussed previously in (4.1), it is known that the observables in the proposed model are $I(t), \mathbf{t_a}$ and $\mathbf{t_p}$. But, $\mathbf{t_s}$ and $R(t)$ can not be observed [1]. The hidden states (i.e. $\beta = \beta_1$ or $\beta = \beta_2$) are also unobservable.

This procedure first infers the hidden states (step I) to separate the buzz-free and buzz regimes and then estimates rest of the parameters (step II). Flowchart of Fig. 4.16 provides a high level description of the overall estimation procedure.

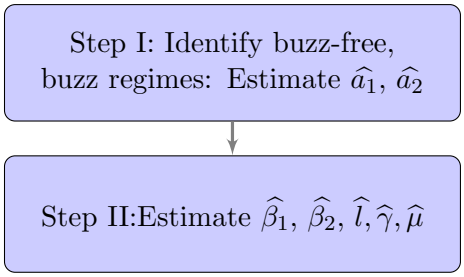


Figure 4.16: Flow chart of the overall estimation procedure

[1] All the variables bear the same meaning as they do in (4.1).

4.2.2.1 Step I: Identification of buzz and buzz-free regime and estimation of a_1 and a_2

This step identifies the buzz-free and the buzz regimes following the approach of [33] which determines whether an inter-arrival time belongs to a buzz-free or buzz state. Let there be n arrivals of new viewers within time T and the corresponding inter-arrival times are $\mathbf{x} = (x_1, x_2, ...x_n)$. This approach uses Bayesian methods to determine the conditional probability of a state sequence $\mathbf{q} = (q_1, q_2, ..)$. The probability density of the inter-arrival times $\mathbf{x}$ given the sequence $\mathbf{q}$ can be written as $P(\mathbf{x}|\mathbf{q}) = \prod_{j=1}^{n} P(x_j|q_j)$. If there are b occasions when there is a state transition, i.e. $q_j \neq q_{j+1}$ then the prior probability of $\mathbf{q}$ reads $(\prod_{j \neq j+1} p)(\prod_{j=j+1} 1-p) = p^b(1-p)^{n-b} = (\frac{p}{1-p})^b(1-p)^n$, where p is the probability of changing state. Then,

$$\begin{aligned} P(\mathbf{q}|\mathbf{x}) &= \frac{P(\mathbf{q})P(\mathbf{x}|\mathbf{q})}{\sum_{\mathbf{q}'} P(\mathbf{q}')P(\mathbf{x}'|\mathbf{q}')} \\ &= \frac{1}{Z}(\frac{p}{1-p})^b(1-p)^n \prod_{j=1}^{n} P(x_j|q_j) \end{aligned} \tag{4.11}$$

In Eq. (4.11) Z is the normalizing constant denoted by $\sum_{\mathbf{q}'} P(\mathbf{q}')P(\mathbf{x}'|\mathbf{q}')$. The objective of this approach is to estimate a sequence $\hat{\mathbf{q}}$ such that, $\hat{\mathbf{q}} = \arg\max_{\mathbf{q}} P(\mathbf{q}|\mathbf{x})$. This is equivalent to find $\hat{\mathbf{q}}$ that minimizes

$$-\ln P(\mathbf{q}|\mathbf{x}) = b \cdot \ln(\frac{1-p}{p}) + \sum_{j=1}^{n} -\ln P(x_j|q_j) - n \cdot \ln(1-p) + \ln Z \tag{4.12}$$

The last two terms of Eq. (4.12) are independent of $\mathbf{q}$. Therefore, it is required to find a state sequence $\hat{\mathbf{q}}$ which minimizes the following function:

$$c(\mathbf{q}|\mathbf{x}) = b \cdot \ln(\frac{1-p}{p}) + \sum_{j=1}^{n} -\ln P(x_j|q_j) \tag{4.13}$$

It is called the cost function.

In the proposed model the inter-arrival time of new viewers are exponentially distributed. Since, the frequency of buzz occurrence is very low it is possible to safely assume that the rate at which the inter-arrival times are distributed for a buzz-free trace is $\alpha_0 = \frac{n}{T}$. For the buzz state this rate is α_1. The mean of s minimum inter-arrival times has been computed to obtain α_1. After several experimentations this work considers s to be $\frac{n}{3}$ for all cases. Clearly deciding the value of s depends on the experience of the practitioner. With this setting $P(x_j|q_j) = \alpha_0 \exp(-\alpha_0 x_j)$ for buzz free case and $P(x_j|q_j) = \alpha_1 \exp(-\alpha_1 x_j)$ otherwise.

This approach also assigns costs for transitions between states to control the frequency of such transitions, prevent shorter buzz and make the identification of long buzz easier despite transient changes in the rate of the arrivals of new viewers. This cost is proportional to the number of arrivals when there is a transition from buzz-free to buzz state. For a transition from buzz state to buzz-free state this is considered to be 0. The procedure of state identification can be briefly described as follows:

Let j be the index of arrival of a new viewer, $\tau(l, m)$ be the state transition cost from state $l \in (0, 1)$ to state $m \in (0, 1)$. $C_0(j)$ be the cost related to the inter-arrival time of the viewer being in buzz-free state and $C_1(j)$ be the cost related to the inter-arrival time of the viewer being in buzz state.

1. For initial state $j = 0$, define $C_0(j) = 0$ and $C_1(j) = \infty$

2. j = j+1

3. Calculate the cost $C_0(j)$ and $C_1(j)$.
   $C_0(j) = -\ln(\alpha_0 e^{-\alpha_0 x_j}) + \min((C_0(j-1) + \tau(0,0)), (C_1(j-1) + \tau(1,0)))$
   $C_1(j) = -\ln(\alpha_1 e^{-\alpha_1 x_j}) + \min((C_0(j-1) + \tau(0,1)), (C_1(j-1) + \tau(1,1)))$
   Since, only transition costs from buzz-free to buzz state have been considered $\tau(0,0) = \tau(1,0) = \tau(1,1) = 0$. $\tau(0,1) = \Gamma \cdot \ln(n)$. The larger the value of Γ the more likely it is possible to avoid shorter buzz and consider the prominent ones.

In all experiments Γ has been considered to be 2.

4. Repeat steps 2 and 3 for all arrivals

5. Select the sequence of states that has the minimum cost

After identifying the buzz-free and buzz states corresponding to the inter-arrival times of a new viewer, this framework computes the time spent in each of the states thereby leading to computing the value of a_1 and a_2. This algorithm has been verified on 5 different sets of parameters reported in Chapter. 3. It correctly classifies a state for an inter-arrival time (of new viewer) for over 90% of cases.

4.2.2.2 Step II: Estimation of $\beta_1, \beta_2, \mu, \gamma, l$

This step estimates $\beta_1, \beta_2, \mu, \gamma$ and l. Flowchart of Fig. 4.17 gives a high level overview of the estimation procedure in this step.

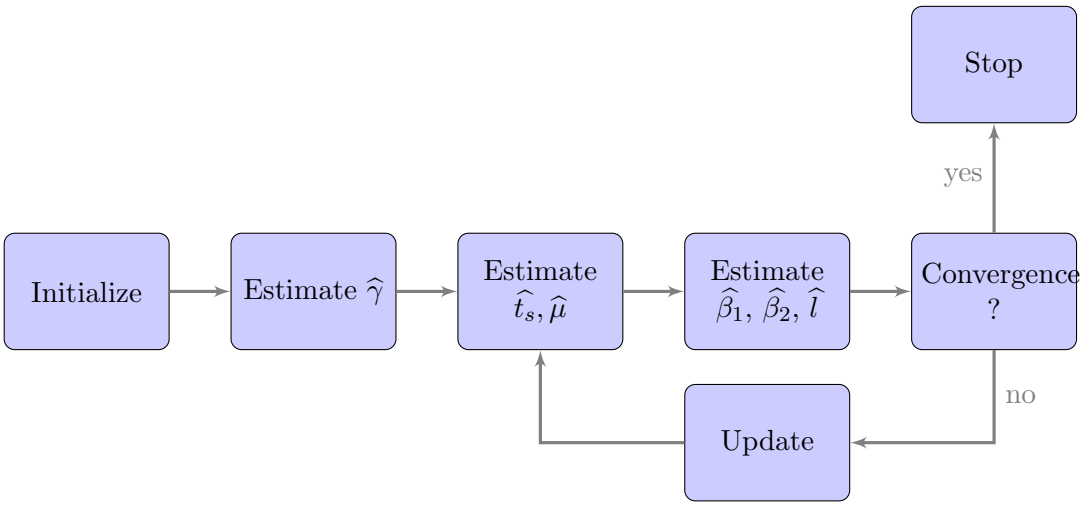


Figure 4.17: Flow chart describing step II of the estimation procedure

This approach derives the probability density of t_a, t_p, t_s for a buzz-free case as described in (4.1) and straightforwardly estimate γ using a maximum likelihood procedures. Naturally, in contrast to γ though, rest of the model parameters depend on the unobserved time series $(R(t), t \in [0, T])$, which is associated to the unknown departure instants

from state R. Like (4.1) this framework denotes it as $\{t_{s_n}\}_{n=1,\dots,n_3}$. With this incomplete dataset, a maximum likelihood estimate is precluded to estimate the propagation parameter μ.

Instead this framework resorts to a Metropolis-Hastings within Gibbs [18] procedure to estimate simultaneously and iteratively $\widehat{\mathbf{t_s}}$ and $\hat{\mu}$, assuming at each iteration step k, known values for all the other parameters. This step is described below in Algorithm. 1, which defines the sub-step. I of the complete estimation procedure. Now, regarding the current estimates of the remaining parameters $(\hat{\beta}_1, \hat{l},)$ at step k, they also need to be updated according to the ongoing values of $\hat{\mu}^{(k)}$ and $\widehat{\mathbf{t_s}}^{(k)}$.

4.2.2.2.1 Substep II.1: Estimation of $\hat{\mu}$ and $\widehat{\mathbf{t_s}}$

As discussed in section 4.2.2.1 the buzz and the buzz-free inter-arrival times for the trace have been already identified. This approach considers the longest sequence of buzz-free inter-arrival times for estimating β_1, μ, l and use the likelihood function of the sought parameters $\mathbf{\Theta} = (\mu,\ \beta_1,\ l)$ described in Eq. (4.3). The longest buzz-free period is considered simply due to the fact that if the entire sequence is considered a weighted linear regressor would have been necessary to separate buzz and buzz-free arrival rates. Experimental results show that even a good regressor generates enough bias that can introduce error in estimating β_1 and l. This equation plays a central role throughout this procedure, since it would be used directly as the target distribution of $\mathbf{t_s}$.

Given the current values of $\hat{\beta}_1$, $\hat{l}$ the values of $\hat{\mu}$ and $\widehat{\mathbf{t_s}}$ are updated as follows. First, the procedure uses a Gamma distribution parametrized by (λ_μ, ν_μ) as the prior distribution for μ. This latter multiplied by the likelihood of Eq. (4.3), leads to the posterior distribution of $\hat{\mu}$:

$$p(\hat{\mu}|\mathbf{t_a}, \mathbf{t_p}, \widehat{\mathbf{t_s}}) \propto \Gamma\big(\lambda_\mu + n_3 - 1, \nu_\mu + \int_0^T \widehat{R(t)}\mathrm{d}t\big), \tag{4.14}$$

from which it is possible to draw an updated value for $\hat{\mu}$. Note that the posterior

distribution for $\hat{\mu}$ does not depend directly on $\hat{\beta}_1, \hat{l}$. Second, the procedure updates $\widehat{\mathbf{t_s}}$ by modifying an arbitrary percentage (15%) of its component. This arbitrary percentage determines how fast the algorithm converges. But it is reasonable not to take a very high value for this arbitrary percentage, since it leads to higher rejection in Metropolis test. It is to be noted that the acceptance of this new $\widehat{\mathbf{t_s}}$ is not systematic and depends on the outcome of the Metropolis ratio. Considering the updated time series $\widehat{\mathbf{t_s}}$, the procedure refreshes the current values of $\hat{\beta}_1, \hat{l}$ applying the approach described in Substep II.2. The procedure iterates these three steps until $\hat{\mu}$ converges to a stable estimate. Algorithm 1 summarises the details of Substep II.1.

Algorithm 1

Assume $n_3 \leftarrow n_2$ (all past viewers stop gossiping eventually)
Set arbitrary initial guess $\hat{\mu}^{(0)} \leftarrow \hat{\gamma}_{\text{MLE}}$
Draw $\Delta\mathbf{t_s}^{(0)} = \{\Delta {t_s}_1^{(0)}, \Delta {t_s}_2^{(0)}, ...\}$ from exponential distribution with rate $\hat{\mu}^{(0)}$
$\widehat{\mathbf{t_s}}^{(0)} \leftarrow \{t_{p_1} + \Delta {t_s}_1^{(0)}, t_{p_2} + \Delta {t_s}_2^{(0)}, ...\}$
repeat for k = 1, 2, ...

1. Construct $\widehat{R}^{(k)}$ from $\mathbf{t_p}$ and $\widehat{\mathbf{t_s}}^{(k-1)}$
2. Estimate $\hat{\beta}_1^{(k)}$ and $\hat{l}^{(k)}$ using method described in section 4.2.2.2.2.
3. Draw $\hat{\mu}^{(k)}$ according to the posterior distribution described in Eq. (4.14)
4. Generate a new candidate for $\widehat{\mathbf{t_s}}^{(k)}$ by modifying the c^{th} component of $\widehat{\mathbf{t_s}}^{(k-1)}$ with a new value uniformly sampled in $[0, T]$; $c \in [1, n_3]$. Note that selecting t_s randomly within an interval is equivalent to consider that the rate at which a viewer stops gossiping follows an exponential distribution (Claim. 1).
5. Accept the latter candidate as the new current estimate of $\widehat{\mathbf{t_s}}$ according to the following Metropolis ratio: $\alpha = \min\{1, p(\widehat{\mathbf{t_s}}^{(k+1)}|\hat{\Theta}^{(k)})/p(\widehat{\mathbf{t_s}}^{(k)}|\hat{\Theta}^{(k)})\}$
6. Otherwise, $\widehat{\mathbf{t_s}}^{(k)} \leftarrow \widehat{\mathbf{t_s}}^{(k-1)}$
7. Repeat Steps 4, 5 and 6 sequentially $0.15 \cdot n_3$ time, thus changing 15% of $\widehat{\mathbf{t_s}}$ for each iteration k

until convergence

4.2.2.2.2 Substep II.2: Estimation of $\hat{\beta}_1$, $\hat{l}$

As discussed in the previous sub-chapter (4.1) the arrival rate $\lambda(t)$ of new viewers linearly

depends on the current number of active and past viewers. So, from the observation $I(t)$ and the reconstructed process $\widehat{R}(t)$, it is possible to formally apply the maximum likelihood to estimate β_1. In practice however, it is to be kept in mind that: *(i)* the arrival process of rate $\lambda(t)$ comprises a spontaneous viewers ingress that is governed by parameter l and which is independent of the current state of the system; *(ii)* depending on the current hidden state of the model (buzz-free *versus* buzz state), it is alternately $\beta = \beta_1$ and $\beta = \beta_2$ that fix the propagation rate. Since the procedure has already identified the longest sequence of buzz-free inter-arrival times it can separately estimate β_1 and β_2. As discussed in (4.1) the inter-arrival time $\mathbf{w}$ between the consecutive arrivals of two new viewers is an exponentially distributed random variable such that $\mathbb{E}\left(\mathbf{w}|\, I(t)+ R(t) = x\right) = (\beta\, x + l)^{-1}$. This property leads us to design Algorithm 2 which describes the details of Substep II.2.

Algorithm 2

Calculate the conditional empirical mean: $\Omega(x) = \frac{1}{|\mathrm{w}_x|} \sum_{t_n \in \mathrm{w}_x} w_n \;:\; \mathrm{w}_x = \{w_n : I(t_n) + \widehat{R}(t_n) = x\}$ for different values of the sum $I(t) + \widehat{R}(t)$
Perform linear regression of $(\Omega(x))^{-1}$ against x
Slope of the regression line equals $\widehat{\beta_1}$ and vertical-axis intercept indicates $\widehat{l}$

The only parameter left to be inferred remains β_2. From Section 4.2.2.1 it is possible to identify the longest sequence of buzz states. Moreover, during the buzz state $\beta_2(i + r) \gg l$. Therefore the maximum likelihood for β_2 can be computed by modifying Eq. (4.3) as follows:

$$p(\mathbf{t_a}, \mathbf{t_p}, \mathbf{t_s} \mid \mathbf{\Theta}) \;\propto\; \prod_{j=1}^{n_4} [\beta_2 (I(t_{a_j}^-) + R(t_{a_j}^-))] \times \prod_{j=1}^{n_5} \gamma I(t_{p_j}^-) \times \prod_{j=1}^{n_6} \mu R(t_{s_j}^-) \times \exp\left(-\int_0^T [\beta_2 (I(t) + R(t)) + \gamma I(t) + \mu R(t)] \mathrm{d}t\right) \tag{4.15}$$

Eq. (4.3) is differentiated with respect to β_2 and solved for 0. It yields:

$$\hat{\beta}_{2_{\mathrm{MLE}}} = n_4 \cdot (\int_{T_1}^{T_2} (I(t) + R(t))\,\mathrm{d}t)^{-1}. \tag{4.16}$$

n_4 is the number of arrivals during buzz regime of the trace under consideration and $(T_2 - T_1)$ is the longest buzz duration. However, the procedure could have also followed the same approach which it followed for estimating β_2. But lack of sufficiently long sequence of buzz inter-arrival times render the linear regression procedure reasonably inaccurate.

4.2.3 Results

The estimation procedure, detailed in Section 4.2.2, is validated against the synthetic traces corresponding to 5 different sets of parameters, used in the previous chapter.

The box plots in Fig. 4.18 indicate for each estimated parameter (centred and normalized by the corresponding actual value). Compared to the box-plot obtained from the heuristic procedure (illustrated in the previous section) it is observed that the MCMC framework estimates the parameters with higher precision (lower bias and variance). However, this work does not report the parameter γ in these box plots, since same Maximum Likelihood procedure has been used in both heuristic and the MCMC approach. Nevertheless it is to be noted that its estimation is the most accurate, both in terms of the bias and variance.

As this work did for the heuristic procedure, it plots the variance and bias of each estimated parameters against the length N of the observable time series for the MCMC procedure as well (Fig. 4.19 to Fig. 4.24). For easily comparing the rate of convergence for different observations, variance and bias of each parameter have been normalized by its particular value at maximum data length (i.e 2^{21} points here).

Fig. 4.19 shows that for β_1 the convergence rate of variance ranges between -0.5

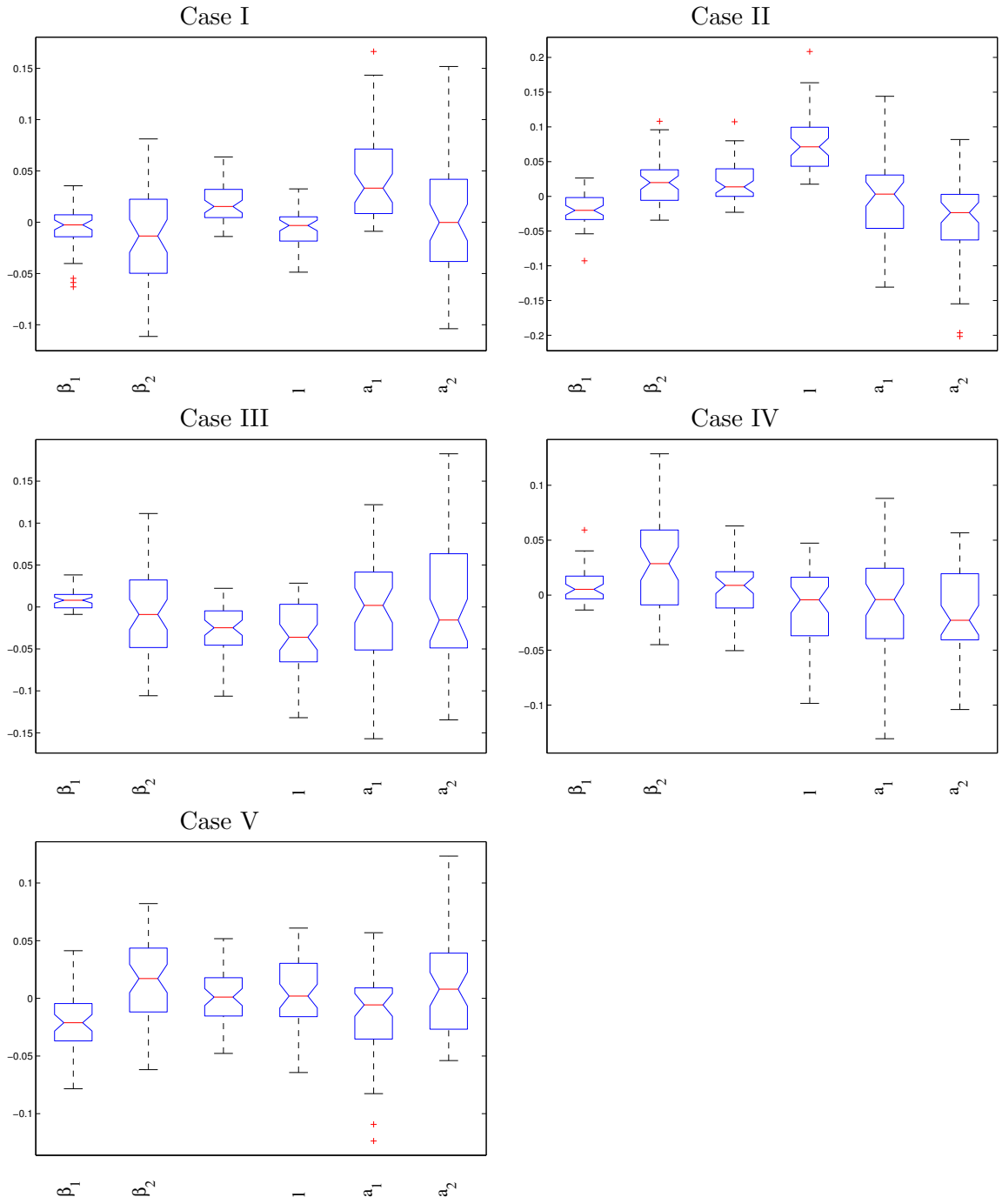


Figure 4.18: Relative precision of estimation of the model parameters for all 5 cases. Statistics are computed over 50 independent realizations of time series of length 2^{21} points

(Case II) to -0.8 (Case III). Convergence rate for variance for β_2 (Fig. 4.20) is maximum for Case III and Case IV, which is around -0.6. For μ the maximum rate is around -0.9 Fig. 4.21. It is not surprising that the μ estimator for MCMC performs better that the one for heuristic procedure. The previous procedure used a naive estimator based on discrete intervals to estimate the μ value. Convergence rate of a_1 (Fig. 4.23) is considerably high for some cases (Case I and Case II), whereas it is low for a_2 (Fig. 4.24) in all cases (lack of points to estimate the buzz-period).

Fig. 4.25 illustrates convergence of the relative estimation error (not in %) for some key parameters of the model. Obtained results show that for all 5 cases the estimated values stay within 10% vicinity of the true parameter values after a couple of hundreds of iterations. In order to focus on the region of interest the horizontal axis has been plotted in log scale. Since, γ, a_1 and a_2 are not estimated using MCMC they are naturally omitted from the convergence plots.

4.2.4 Discussion

In the previous two sections, two estimation procedures have been discussed. Their performances are compared based on five different workloads. Clearly the MCMC procedure performs better than the heuristic procedure in terms of bias and variance. However, each of this procedure has their own merits. The heuristic procedure is based on the model mechanism and provides an intuitive solution. MCMC on the other hand seems to be a natural choice for this problem, since it deals effectively with hidden states and missing values (R in this case). But, the experiments in this work loosely suggest that it is possibly computationally heavier (in terms of computational time for same traces) than the heuristic procedure. However, this study has not performed any complexity analysis of both procedures. Therefore it refrains from affirming this claim. Nevertheless, both procedures perform reasonably well on the given workloads with different profiles.

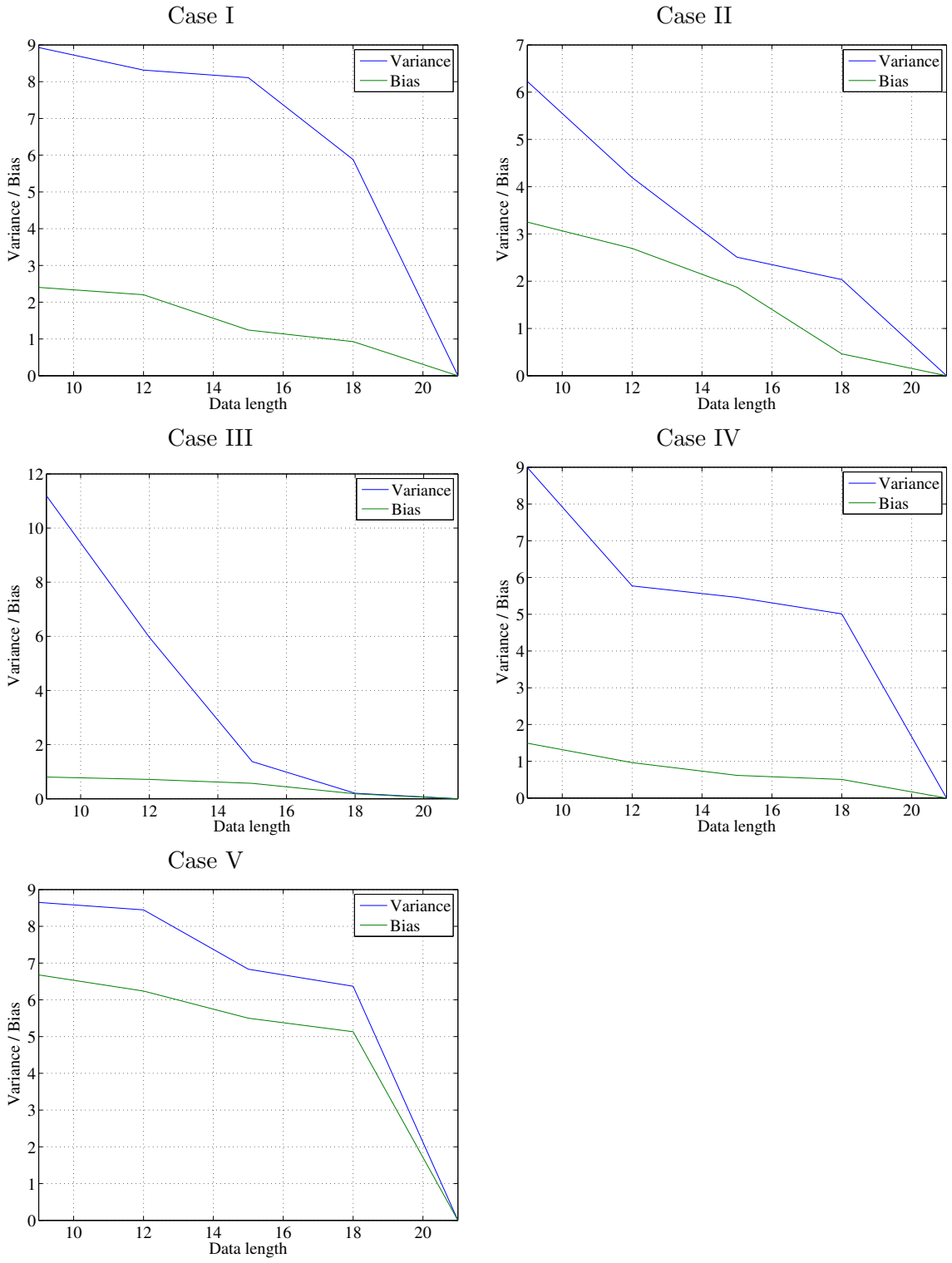


Figure 4.19: Evolution of the Variance and Bias for β_1 *against* the data length N in a *log-log* plot for the 5 traces for the MCMC procedure.

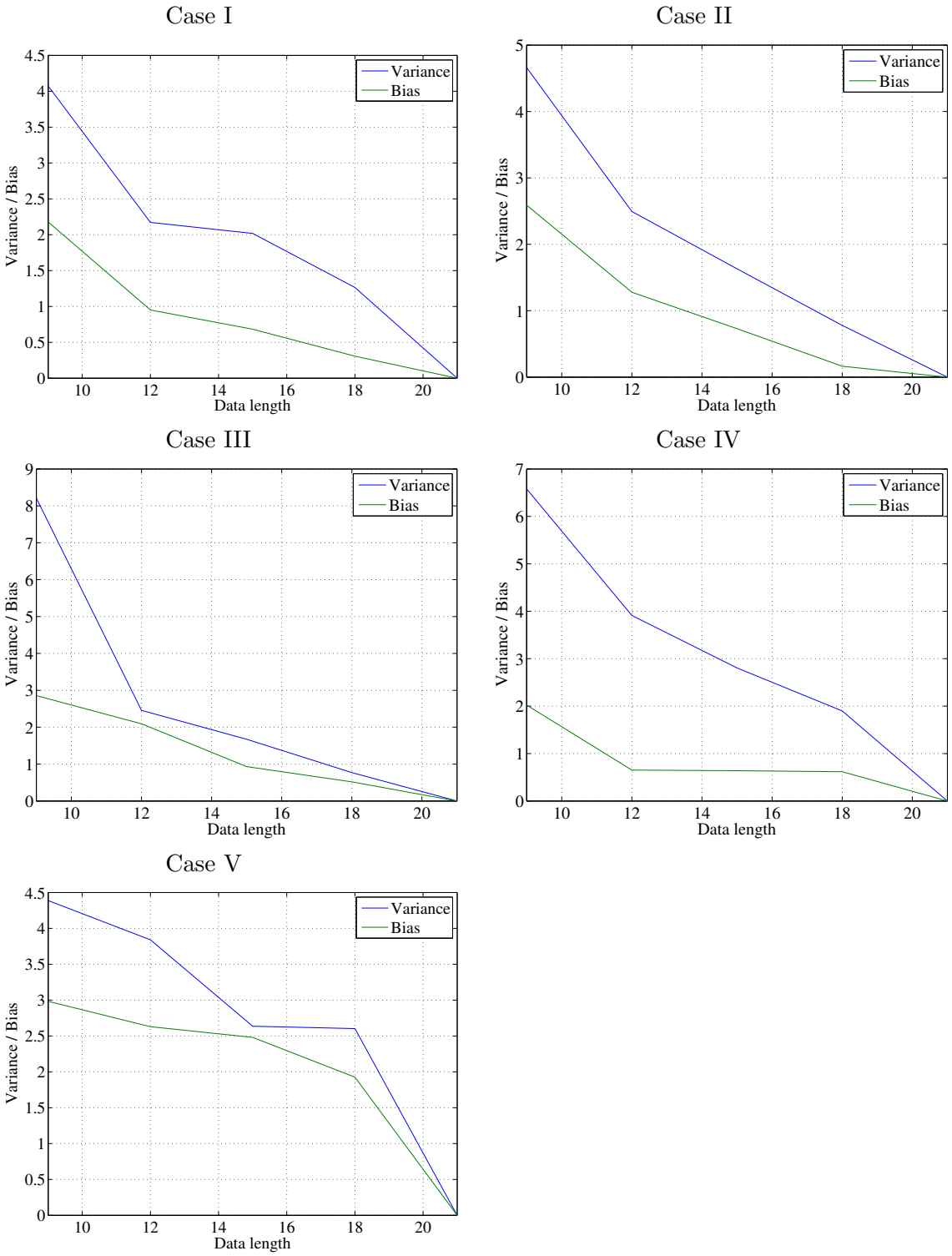


Figure 4.20: Evolution of the Variance and Bias for β_2 *against* the data length N in a *log-log* plot for the 5 traces for the MCMC procedure.

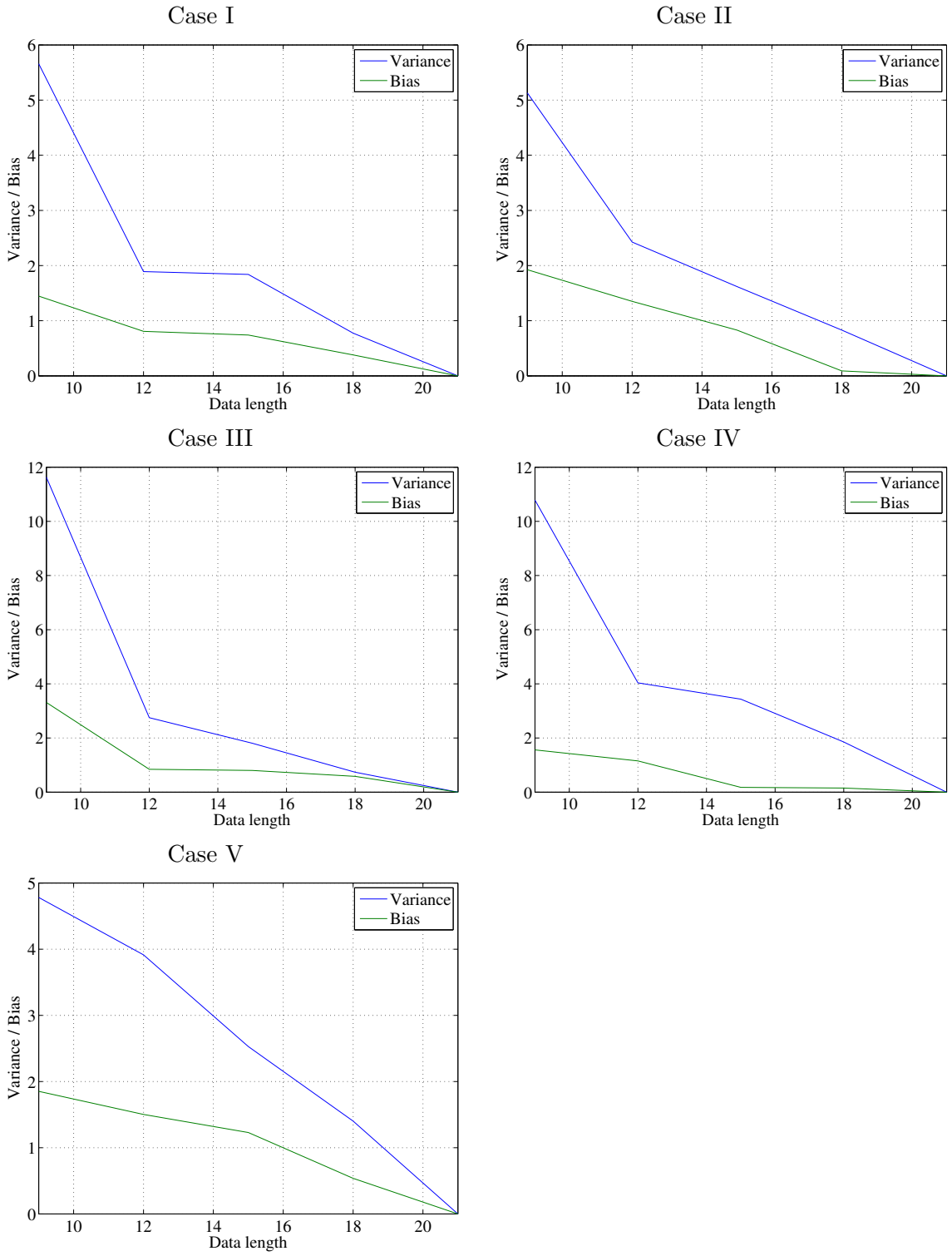


Figure 4.21: Evolution of the Variance and Bias for μ *against* the data length N in a *log-log* plot for the 5 traces for the MCMC procedure.

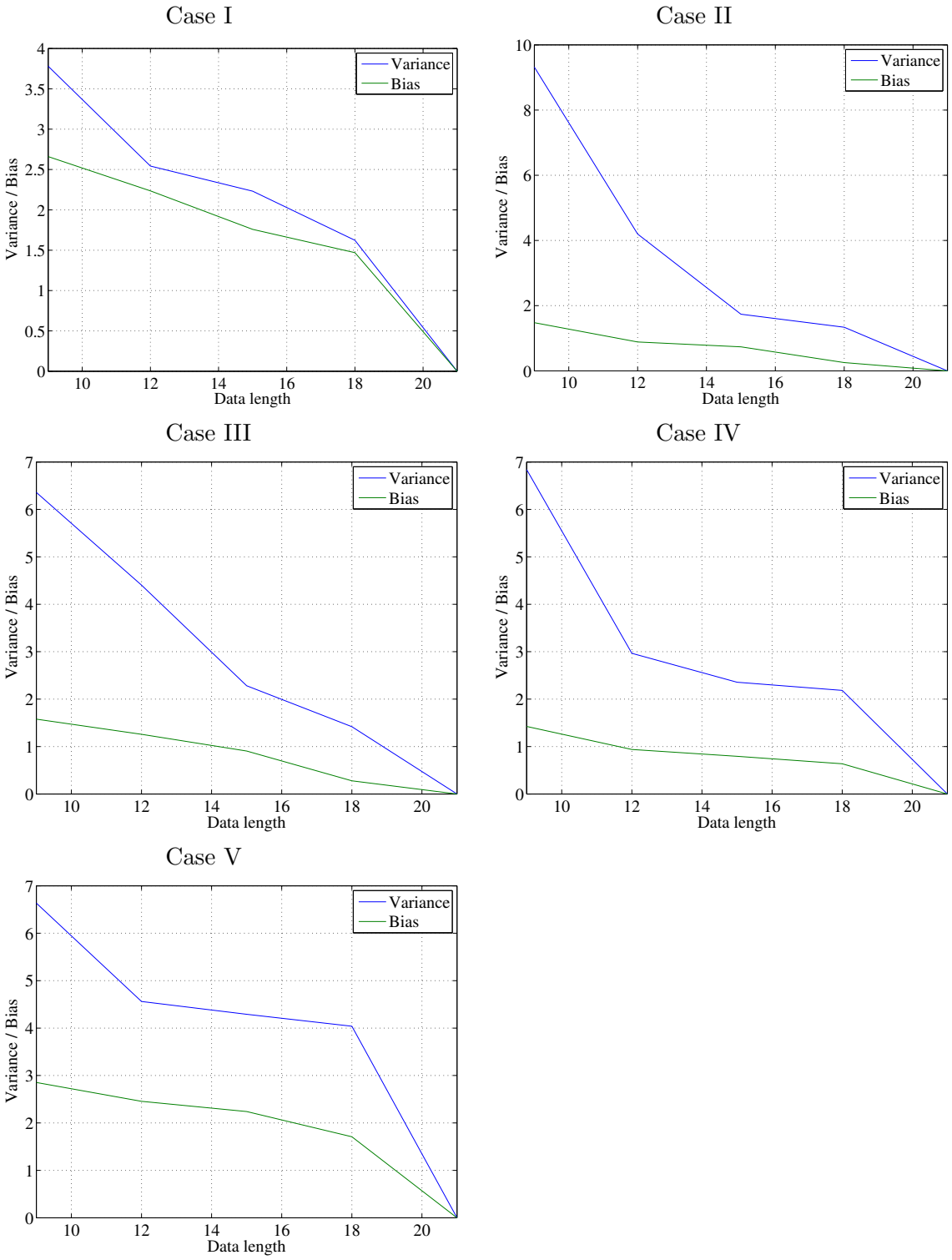


Figure 4.22: Evolution of the Variance and Bias for l *against* the data length N in a *log-log* plot for the 5 traces for the MCMC procedure.

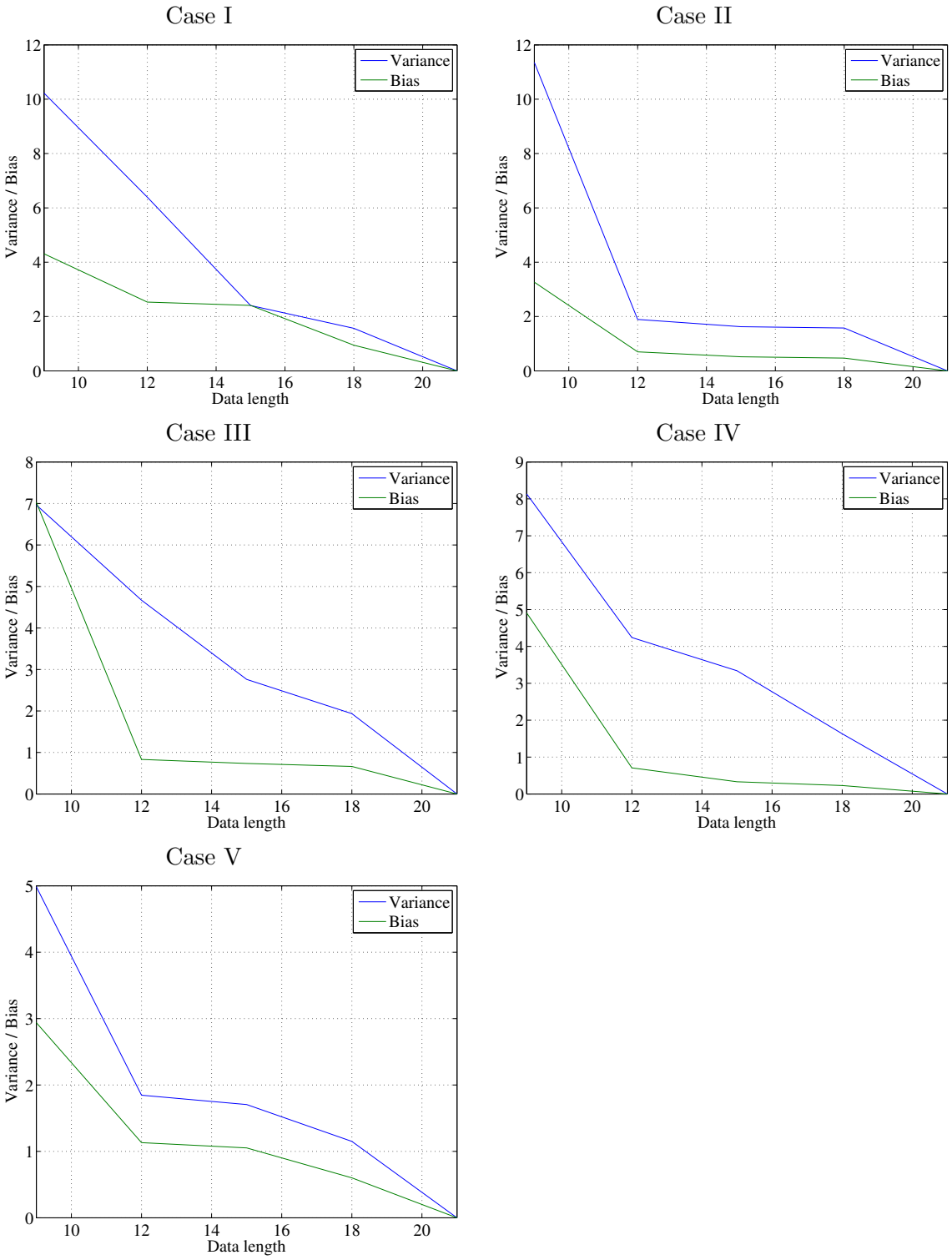


Figure 4.23: Evolution of the Variance and Bias for a_1 *against* the data length N in a *log-log* plot for the 5 traces for the MCMC procedure.

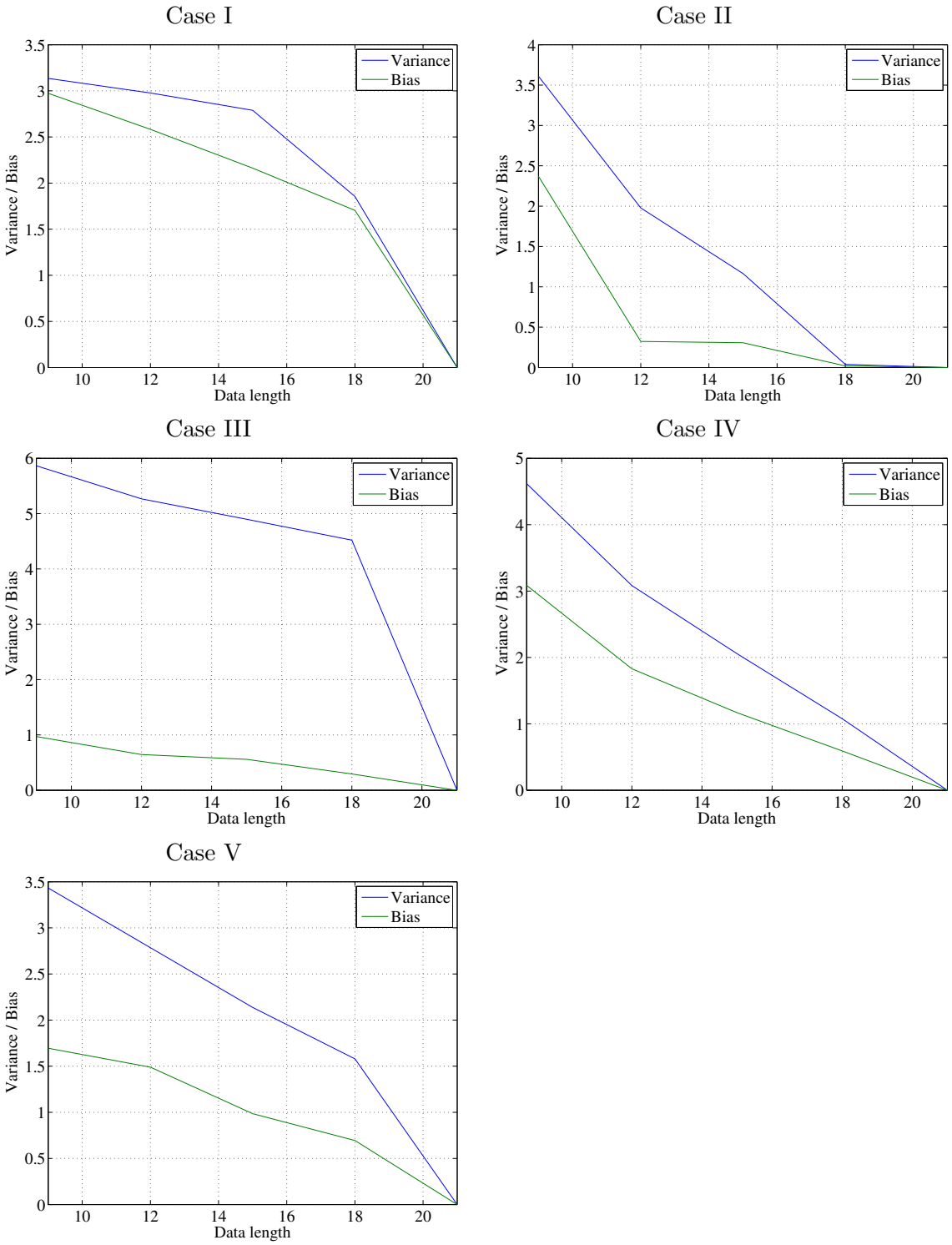


Figure 4.24: Evolution of the Variance and Bias for a_2 *against* the data length N in a *log-log* plot for the 5 traces for the MCMC procedure.

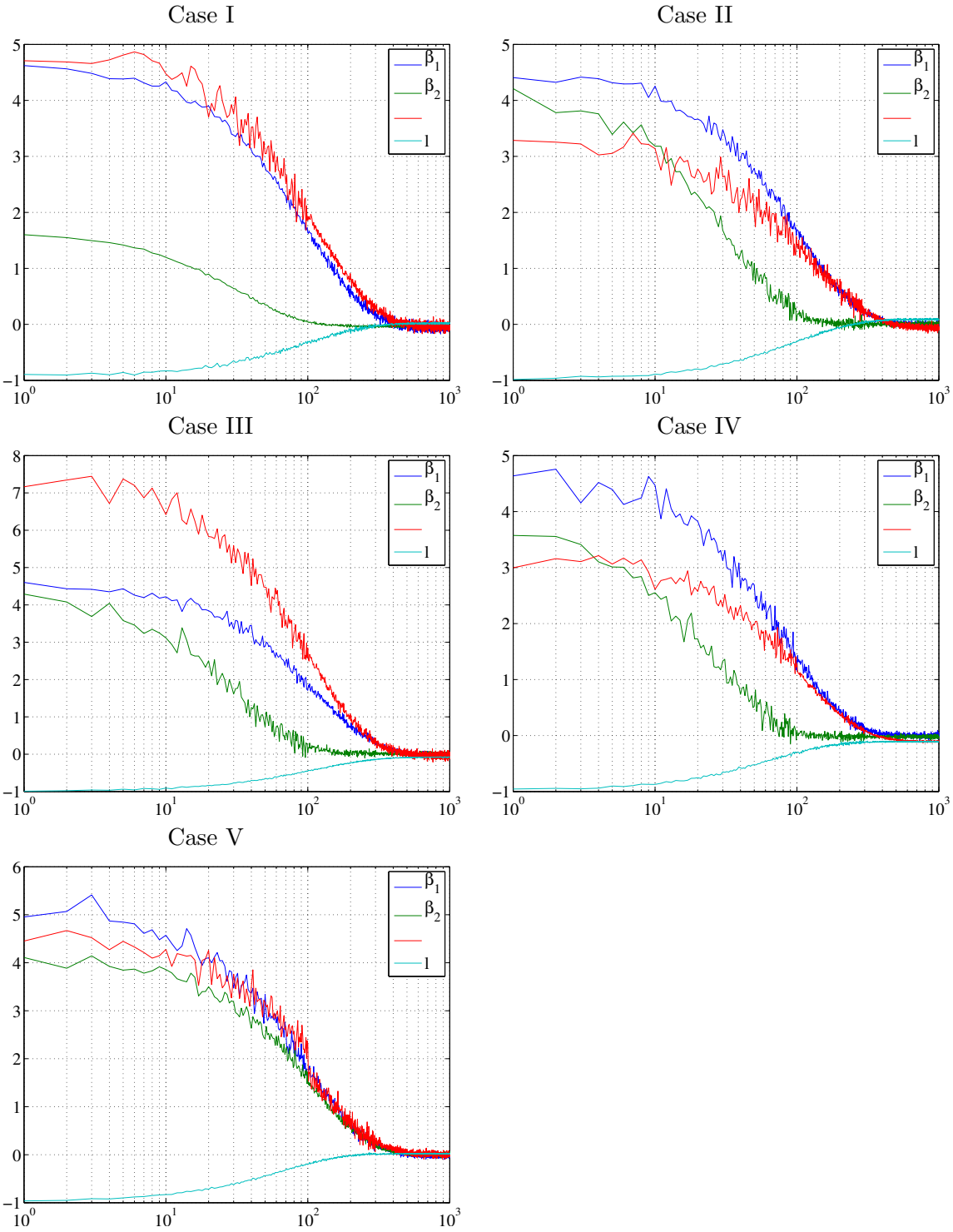


Figure 4.25: Convergence plot of five sets of parameters in a semi-log scale. The horizontal axis represents # of iterations and the vertical axis represents the relative error

4.3 Data-model adequacy of the calibrated model

4.3.1 Validation Against an Academic VoD Server:

After assessing the accuracy of the estimator on the synthetic traces it is required to verify the adequacy of the proposed model at reproducing real workload traces. Since the MCMC procedure performs better than the heuristic approach this study uses the former on the two VoD traces, recorded in January 2011 by the Greek Research and Technolohy Network (GRNET) [20]. They are denoted as Trace I ($\sim$ 200 hours long) and Trace II ($\sim$ 150 hours long) and plotted in Fig. 4.26-(a) and -(b), respectively. For both cases, this study checks the two sets of estimated parameters reported in Table. 4.1 to verify the stability condition derived in the section Model Description (Chapter III). Then these calibrated models are used to generate corresponding realisations of synthetic workloads (plots (c)-(d) of Fig. 4.26) for some statistical comparison, described later.

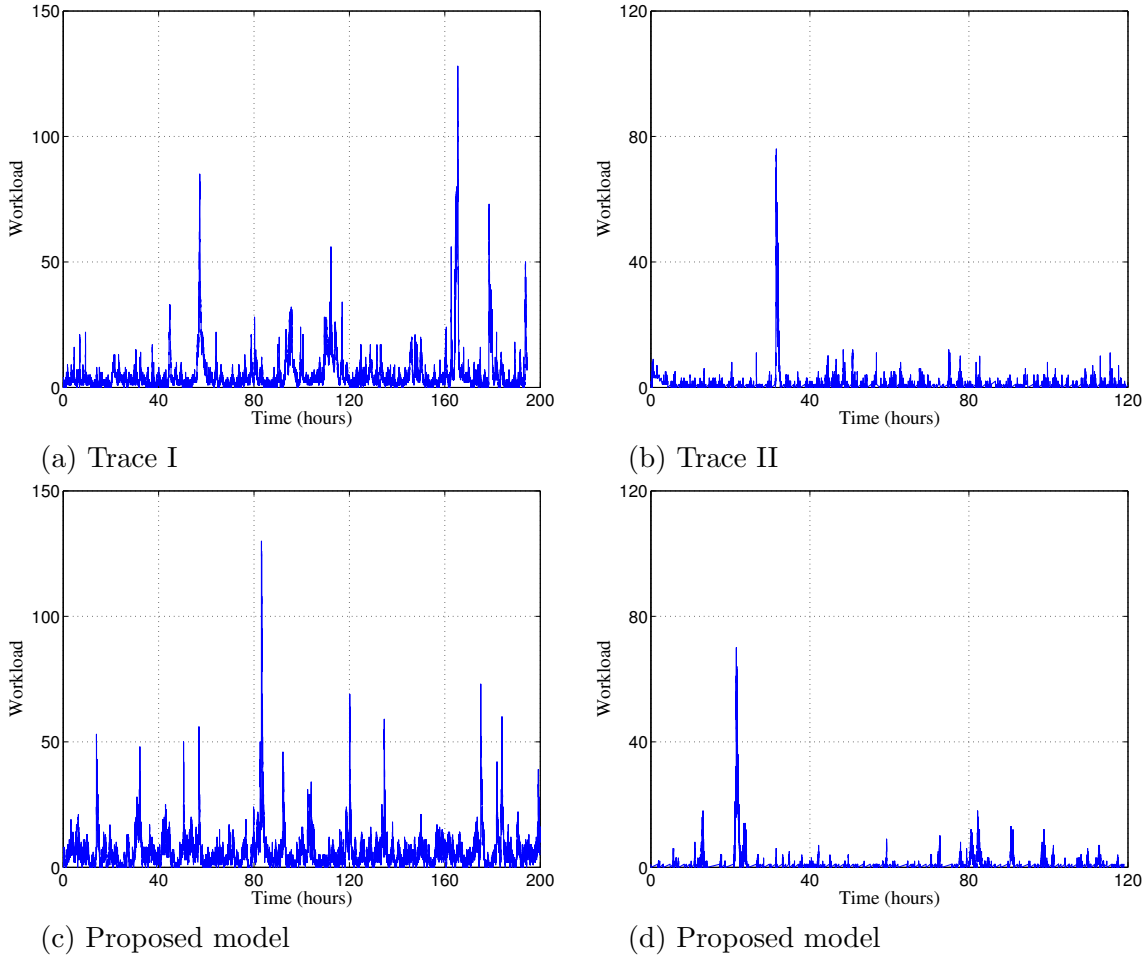
Table 4.1: Estimated Parameters of the VoD model

	$\widehat{\beta_1}$	$\widehat{\beta_2}$	$\widehat{\gamma}$	$\widehat{\mu}$	$\widehat{l}$	$\widehat{a_1}$	$\widehat{a_2}$
Trace I	$1.3.10^{-3}$	$8.4.10^{-3}$	$3.9.10^{-3}$	$2.8.10^{-3}$	$3.2.10^{-3}$	$3.1.10^{-4}$	$2.2.10^{-2}$
Trace II	$4.9.10^{-3}$	$1.8.10^{-2}$	$1.2.10^{-2}$	$9.5.10^{-3}$	$4.8.10^{-4}$	$1.3.10^{-5}$	$4.1.10^{-2}$

Table 4.2: Mean and standard deviation of real traces and the calibrated models.

		Real	Proposed Model	Simple Markov	$MMPP/M/1$
Trace I	Mean	4.99	5.59	12.68	6.45
	Std. Dev	18.26	17.87	17.15	20.02
Trace II	Mean	0.71	0.62	1.23	0.94
	Std. Dev	16.82	15.99	15.85	17.95

Comparing the means and the standard deviations of both real and synthetic traces (Table 4.2), it is clear that the proposed model successfully reproduces the average number of active viewers but also its variability along time. The observed difference (about 10% for the mean values) is not as striking as it was with the synthetic traces of Section 4.3.1. But the readers must bear in mind that first, *ab initio* nothing guarantees



(a) Trace I (b) Trace II

(c) Proposed model (d) Proposed model

Figure 4.26: Modelled workload for Trace I (Left column) and Trace II (Right column). First row corresponds to the real traces; second row to the synthesised traces from the proposed model. Horizontal axes represent time (in hours) and vertical axes represent workload (number of active viewers).

that the underlying system matches the model dynamics and, second, Traces I and II can possibly encompass short scale non-stationary periods (e.g. day *versus* night activity) which are not accounted for in the proposed model.

Nonetheless, for the sake of a fair analysis, it is a must to compare the performance of the proposed approach with that of simpler, yet sensible models and with that of more elaborated models that were proposed in the literature for similar purposes. This study starts with a simple Markov model where the transition rates are derived from all

possible changes of states, observed in real time series. Calibrated on Traces I and II, this model produces synthetic evolutions of active viewers, whose mean can significantly differ from real values (see Table 4.2). However the discrepancy is not that pronounced for the standard deviations (relative error remains below 10%), which tends to prove that a naive model like a Markov chain succeeds to catch the inherent variability of a VoD workload process!

Next this study considers a more refine $MMPP/M/1$ queue model proposed in [62]. This queueing system assumes an arrival process that alternates between two Poisson processes according to a two hidden state Markov chain, an exponentially distributed service time and a single server to serve the viewers. In the author's own words, this Modulated Markov Poisson Process is particularly adapted for modelling correlated arrival streams and bursty workload behaviour. As previously then, this model is also calibrated with Traces I and II. Comparing the means and the standard deviations between the real and the modelled traces, the fitting performance of the $MMPP/M/1$ model are fairly comparable to that of the proposed model (see values in Table 4.2).

Beyond its mean and standard deviation, the steady state distribution of a (stationary) stochastic process is a more complete indicator of the process volatility. In particular, the way it decreases towards zero defines the frequency of large values and therefore directly reflects the burstiness of the process. Plots of Fig. 4.27 represent the estimated steady state distributions corresponding to the real workloads of Traces I and II, respectively and superimposed, the three traces from each calibrated model. Despite having comparable means and variances (Table 4.2), these curve show that not all the synthetic traces do reproduce accurately the statistical distribution of the number of active viewers. In particular, it is clear from the plots that the occurrence of large amplitudes are overvalued by the simple Markov model and also by the $MMPP/M/1$ queue. In contrast, the good fit of the proposed model proves its capacity to reproduce the occurrence and the amplitude range of buzz events (i.e. bursts in the evolution of

active viewers).

Another very important feature that characterises the volatility of a process is the local regularity of its path. In particular, the rapidity of the amplitude variations at small scales fixes the dynamics of the bursts, and can subtly be formalised via the auto-correlation function of the process. This latter measures the statistical dependency $R_I(\tau) = \mathbb{E}\{I(t)\, I^*(t+\tau)\}$ between two samples of a (stationary) process I, distant of a time lag τ: the larger $R_I(\tau)$, the smoother the path of I at scale τ. So, for the real and all the generated traces, this study estimates their auto-correlation functions and plots in Fig. 4.28. It is striking then, how the proposed model is able to reproduce the long-term correlative structure of the real traces, whereas both simple Markov and $MMPP/M/1$ models fail at imposing a statistical continuity beyond a 30 minutes time scale for Trace I, and only 3 minutes for Trace II!

This study stresses that, the reproduced dynamics is a direct consequence of the memory effect (controlled by the parameter μ) that has been injected in the proposed model. However, this work does not intend with this mechanism, to originate a Long Range Dependence (LRD) property (in the strict sense of a power law decay of the autocorrelation function), as such behaviour in real data has not been observed .

The last and the final feature which this study presents here is the large deviation (LD) spectrum which intrinsically, embeds a time scale notion into the statistical description of the aggregated observable at different time resolutions. A detailed description of this feature is presented in the next chapter along with its theoretical properties. Fig. 4.29 shows the empirical LD spectrum (sampling time scale is one thousandth of the total time duration) of all traces. It shows that for both cases the modelled traces resemble the LD spectrum of the real ones. The simple Markov and $MMPP/M/1$ seems to have a much wider spectrum than the actual one. Moreover, the almost sure value (peak of the spectrum) of the real traces match with the proposed one.

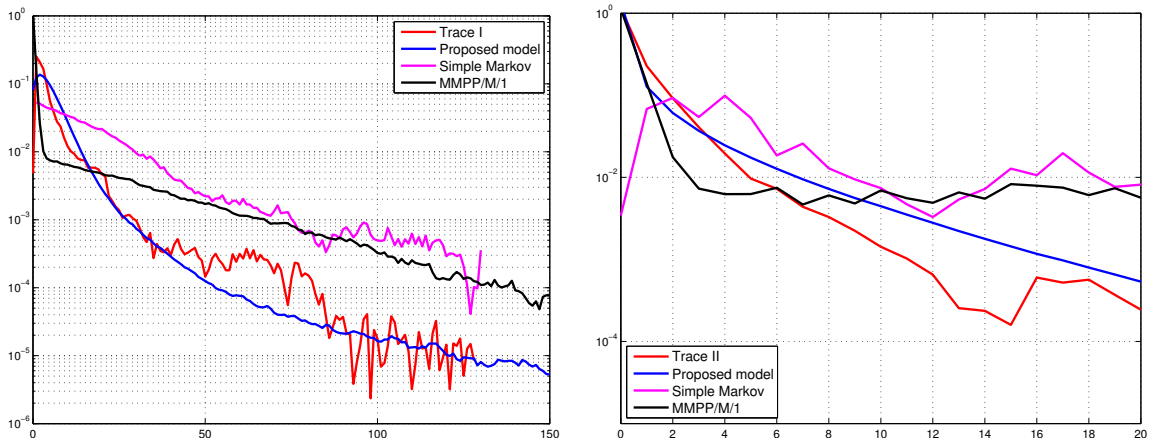


Figure 4.27: Steady-state distribution of the real trace against the generated trace for GRNET. The horizontal axis represents workload (# of current viewers)

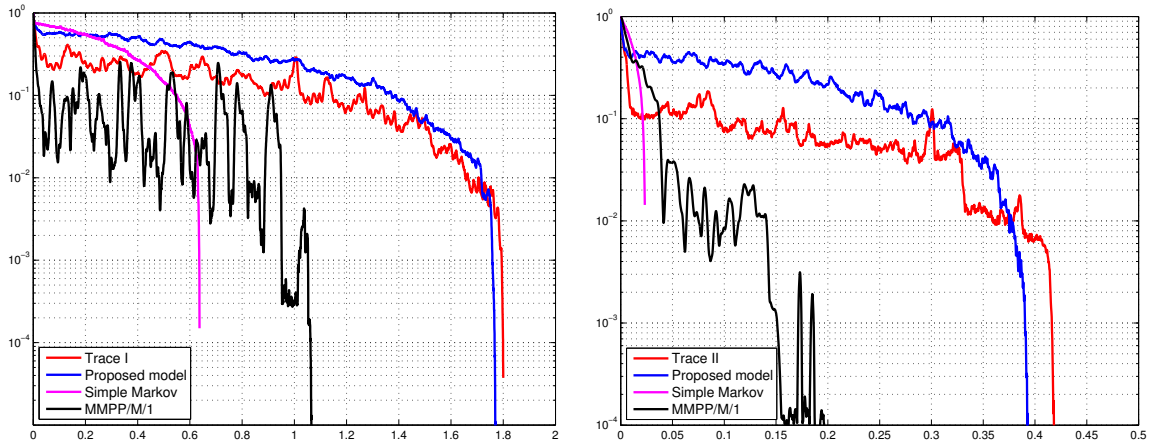


Figure 4.28: Auto-correlation of the real trace against the generated trace for GRNET. The horizontal axis represents time lag τ (*hours*)

4.3.2 Validation Against World Cup 1998 workload

Next this work analyses two traces obtained from the viewers log of the world cup football final (12th July 1998) (source [3])). From visual observation the traces shows no distinct regimes (buzz and buzz free) unlike the previous two traces. The algorithm, which was developed in the MCMC approach to identify a_1 and a_2 has been applied on these traces. The obtained results confirm that there is no sudden increase of interest

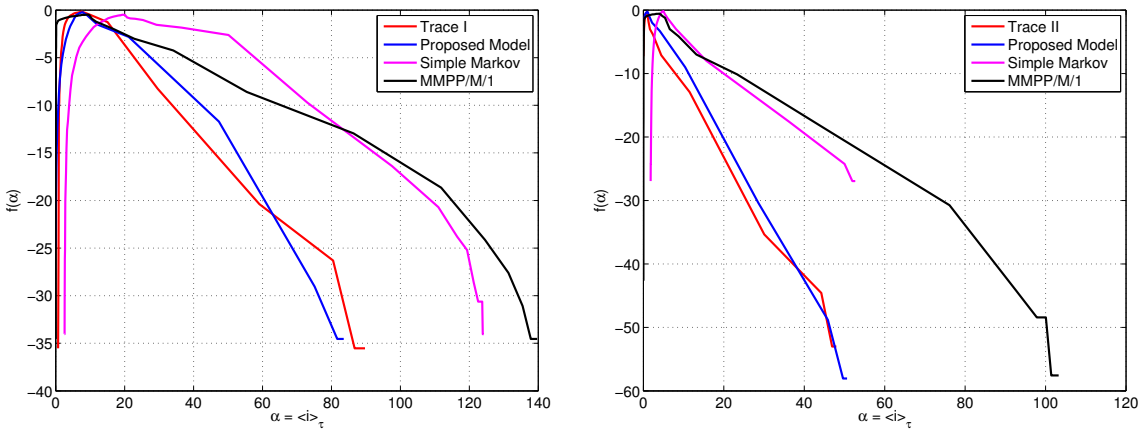


Figure 4.29: Large Deviation Spectrum of the real trace against the generated trace for GRNET.

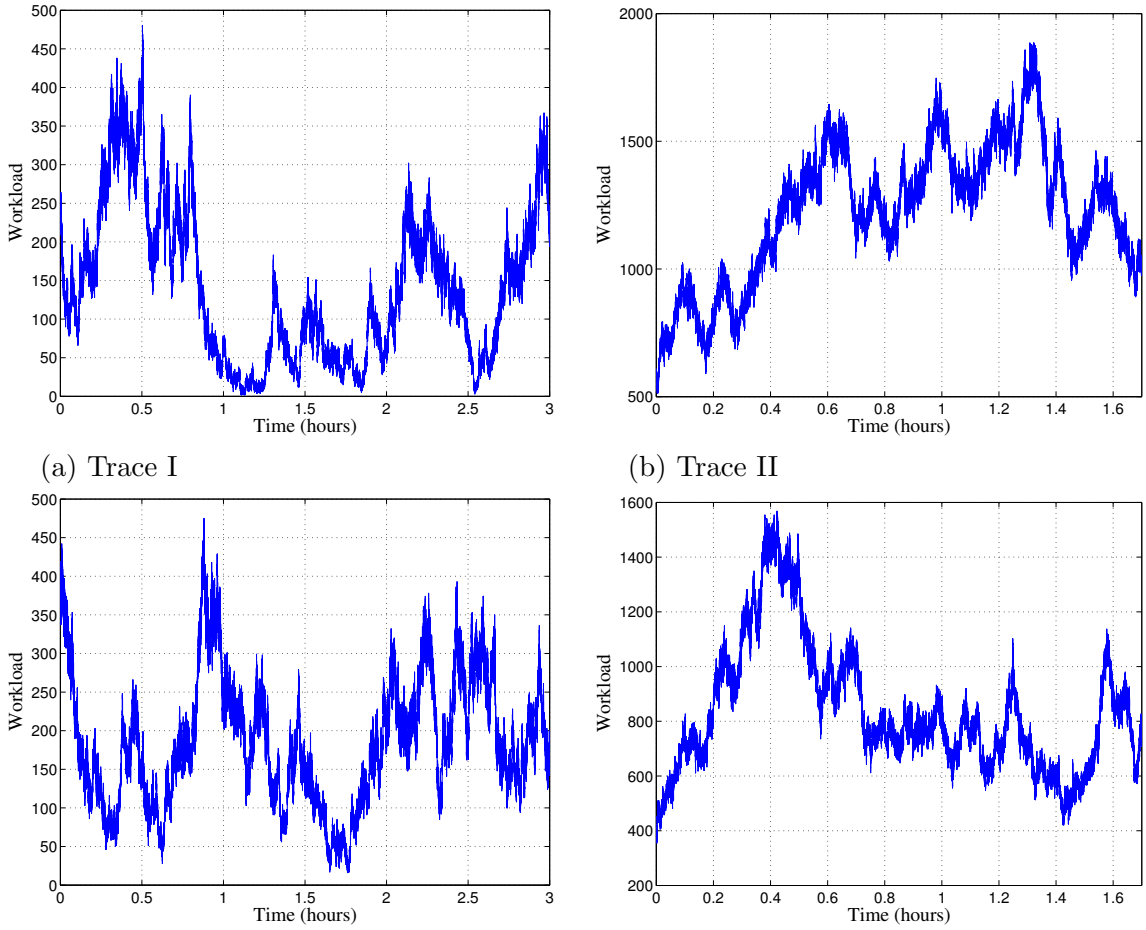
over an event. This work infers this fact to the much less usage and popularity of internet during that period of time. This study, therefore, considers these traces to be buzz free and contemplate the proposed model without the hidden Markov chain (i.e. only one value of β).

Using the estimation procedure the following parameters can be found for the two traces:

Table 4.3: Estimated Parameters from the World Cup Traffic Traces

	$\widehat{\beta}$	$\widehat{\gamma}$	$\widehat{\mu}$	$\widehat{l}$
Trace I	0.1259	0.2939	0.2223	0.3043
Trace II	0.1288	0.3532	0.2031	0.7043

From the trace it is observed that the workload gets higher (Trace II) when a match starts and gradually diminishes once it ends. Another major difference between these two different regimes are the rate of spontaneous arrivals into the system. It has been observed that the effect of gossip spread is less for this workload (β is less than l), since the world cup final is a well known event with viewers having advanced plans to stay tuned with the event (through television or website of the event). This is represented by the spontaneous arrival rates which is higher during the match than before the start of



(a) Trace I

(b) Trace II

(c) Proposed model

(d) Proposed model

Figure 4.30: Trace of the world cup football (1998) final. Trace I is collected before the match started and Trace II covered the duration of the match. First row corresponds to the real traces; second row to the synthesised traces from the proposed model. Horizontal axes represent time (in hours) and vertical axes represent workload (number of active viewers).

the match. Moreover, this study reveals that the memory effect is not very dominating for both traces.

Like the previous Greek VoD traces this study also compares the steady state distribution, auto-correlation and the LD spectrum of the real traces and the traces generated from the calibrated (proposed) model and a simple Markov model. A $MMPP/M/1$ has not been considered here since the real traces contain a single regime for both cases. Fig. 4.31, Fig. 4.32 and Fig. 4.33 shows that the real traces and the generated traces

from the proposed model have comparable statistical distribution and time coherence. As expected the simple Markov process again fails to capture the variability and time coherence of a real process.

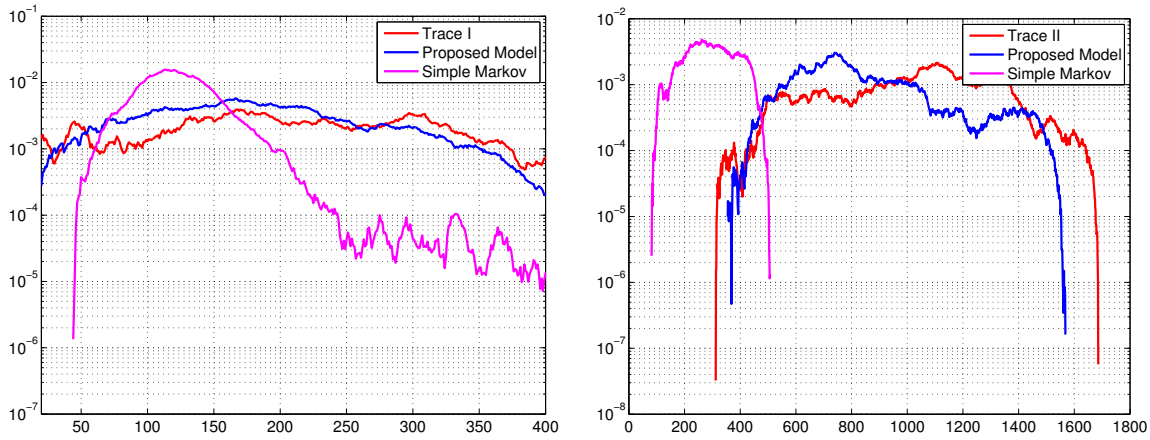


Figure 4.31: Steady-state distribution of the real trace against the generated trace for WC98 server. The horizontal axis represents workload (# of current viewers)

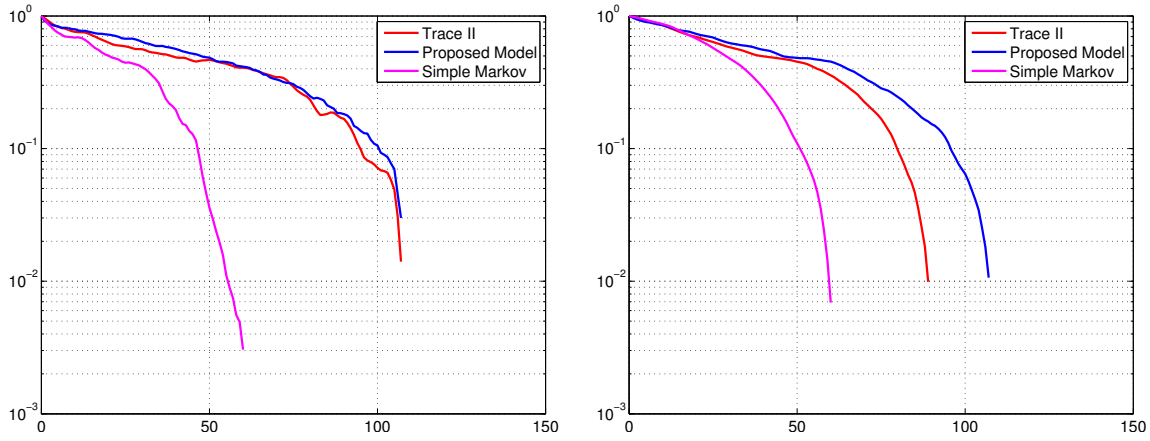


Figure 4.32: Auto-correlation plot of the real trace against the generated trace for WC98 server. The horizontal axis represents time lag τ (*secs*)

4.4 Conclusion

This chapter presents two estimation procedures. Both of them performs reasonably well on synthetic traces. MCMC, being the superior approach is applied on four real traces

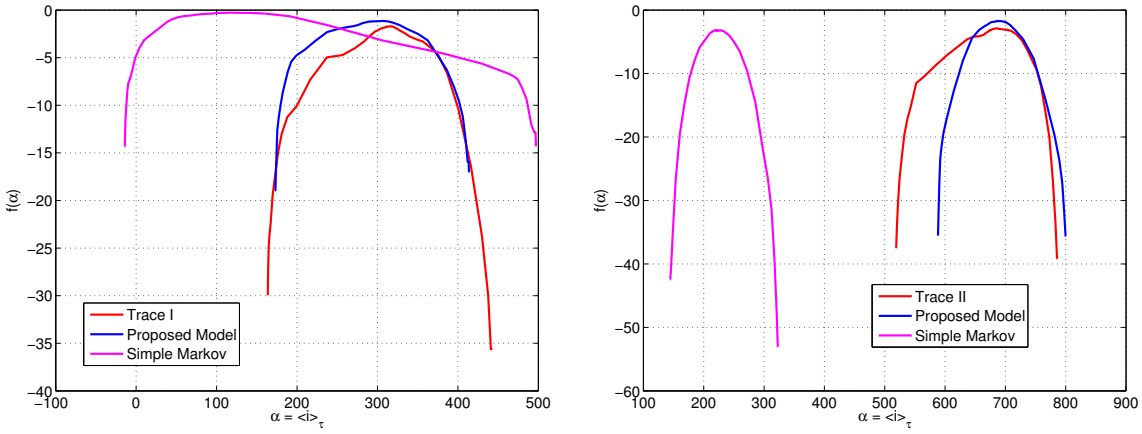


Figure 4.33: Large deviation spectrum of the real trace against the generated traces for WC98 server.

from two different sources. Obtained results demonstrate that the proposed model, if properly calibrated produces comparable statistical distribution and time coherence. Owing to the constructive nature of the model, the estimated values of the parameters provide valuable insight on the application that is difficult to infer readily from the raw traces. The captured information may answer questions of practical interest to cloud oriented VoD service providers. Finally, a key-point of this model is that it permits to reproduce the workload time series with a Markovian process, which is known to verify a Large Deviation Principle (LDP). This particularly interesting property yields a large deviation spectrum whose interpretation enriches the information conveyed by the standard steady state distribution: For a given observation (workload trace), LDP allows to infer (theoretically and empirically) the probability that the time average workload, calculated at an arbitrary aggregation scale, deviates from its nominal value (i.e. almost sure value). The following chapter describes the LDP elaborately and shows how the service providers can leverage it from practical aspects.

Chapter 5

Resource Management

- Illustration of Large Deviation Principle
- Discussion on two possible Probabilistic Provisioning Schemes

5.1 Introduction

Internet applications undergo dynamically varying workloads that contain long-term variations such as time-of-day effects as well as short-term fluctuations due to flash crowds. Predicting the peak workload of an Internet application and capacity provisioning based on these rare event estimates is notoriously difficult.

Underestimating the peak workload can result in an application overload, causing the application to crash or become unresponsive. There are numerous documented examples of Internet applications that faced an outage due to an unexpected overload. For instance, the normally well-provisioned Amazon.com site suffered a forty-minute down-time due to an overload during the popular holiday season in November 1999 [69]. Recently ABC's live Internet stream of the Oscars telecast 2014 went down for users across the U.S due to a traffic overload [66]. Similarly overestimating workload causes

significant loss of resources and energy. Therefore, it is a critical issue to provision the peak workload judiciously.

Given the difficulties in predicting peak Internet workloads, one possibility of the application is to employ a combination of dynamic provisioning and request policing to handle workload variations. Dynamic provisioning enables additional resources such as servers to be allocated to an application on-the-fly to handle workload increases, while policing enables the application to temporarily turn away excess requests while additional resources are being provisioned.

Followed by the Markovian model (described in Chapter III), this thesis proposes two possible and generic ways to exploit these information in the context of probabilistic resource provisioning. They can serve as the input of resource management functionalities of the Cloud environment. It is evident that it is not possible to define elasticity without the notion of a time scale. It plays a significant role to define the properties of a system, which is explained in the current section with an example of a homogeneous Poisson process with constant rate (of arrival) Λ. Remind that Λ is the expected number of "arrivals" that occur per unit of time. Fig. 5.1 shows how the probability distribution of the throughput, varies depending on the actual considered time scale τ. Even though the probability distribution of the throughput (for different time scales) are normalized against the mean (number of arrivals) the distribution seems to be significantly different. It indicates that the system is not scale invariant and relies on the analysis scale. Clearly, τ can play an important role for even more complex systems (where the arrivals are far from a simple Poisson process), such as the one proposed in this thesis. The proposed model follows a non-homogeneous poisson arrival process and includes memory in the system which makes it more complicated than a simple memoryless Poisson process. The model also permits a scale dependent characterization description. Therefore, in order to provision resources for such system it becomes imperative to include time resolution.

The Large Deviation Principle (LDP) is capable of automatically integrating the

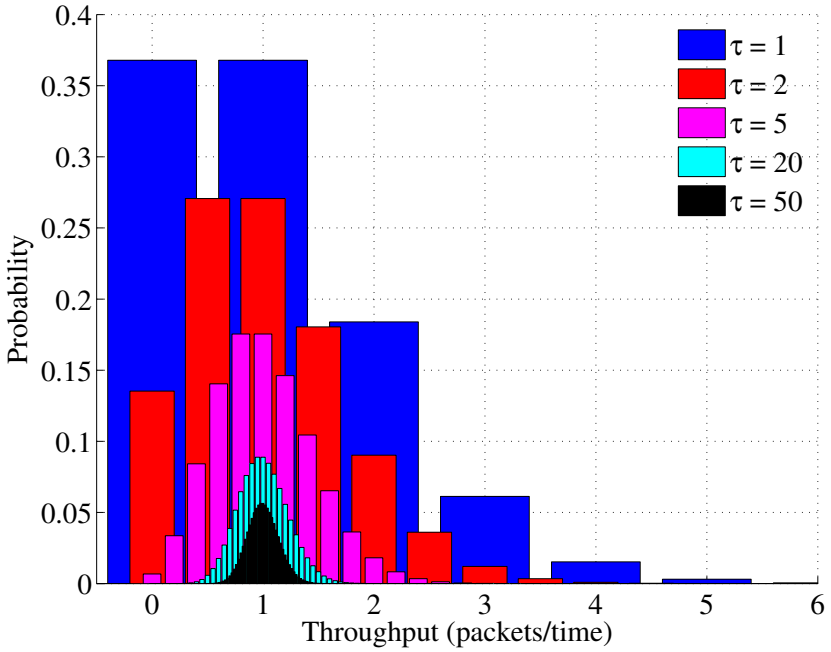


Figure 5.1: Probability distribution of throughput for a homogeneous Poisson process (with rate equal to 1) for different time scales (τ).

time resolution in automatic description of the system. It is to be noted that Markovian processes do satisfy the LDP, but so do some other models as well. Hence, the proposed probabilistic approach is very generic and can be adapted to address various provisioning issues, provided the resource volatility can be resiliently represented by a stochastic process for which the LDP holds true.

5.2 Large Deviation Principle

Consider a continuous-time Markov process $(X_t)_{t\geq 0}$, taking values in a finite state space S, of rate matrix $A = (A_{ij})_{i\in S, j\in S}$. Here, X is a vectorial process $X(t) = (N_I(t), N_R(t)), \forall t \geq 0$, and $S = \{0, \cdots, I_{\max}\} \times \{0, \cdots, R_{\max}\}$. If the rate matrix A is irreducible, then the process X admits a unique steady-state distribution π satisfy-

ing $\pi A = 0$. Moreover, by Birkhoff ergodic theorem, it is known that for any mapping $\Phi : S \to \mathbb{R}$, the sample mean of $\Phi(X)$ at scale τ, i.e. $1/\tau \cdot \int_0^\tau \Phi(X_s)\mathrm{d}s$ converges almost-surely towards the mean of $\Phi(X)$ under the steady-state distribution, as τ tends to infinity. The function Φ is often called the *observable*. Since this work emphasizes on the variations of the current number of users $N_I(t)$, Φ will simply be the function that selects the first component: $\Phi(N_I(t), N_R(t)) = N_I(t)$. The large deviations principle (LDP), which holds for irreducible Markov processes on a finite state space [70], gives an efficient way to estimate the probability for the sample mean calculated over a large period of time τ to be around a value $\alpha \in \mathbb{R}$ that deviates from the almost-sure mean:

$$\lim_{\epsilon \to 0} \lim_{\tau \to \infty} \frac{1}{\tau} \log \mathbb{P} \left\{ \int_0^\tau \Phi(X_s)\mathrm{d}s \in [\alpha - \epsilon, \alpha + \epsilon] \right\} = f(\alpha). \tag{5.1}$$

The mapping $\alpha \mapsto f(\alpha)$ is called the large deviations spectrum (or the rate function). For a given function Φ, it is possible to compute the theoretical large deviations spectrum from the rate matrix A as follows. One first computes, for each values of $q \in \mathbb{R}$, the quantity $\Lambda(q)$ defined as the principal eigenvalue (*i.e.,* the largest) of the matrix with elements $A_{ij} + q\delta_{ij}\Phi(j)$ ($\delta_{ij} = 1$ if $i = j$ and 0 otherwise). Then the large deviations spectrum can be computed as the Legendre transform of Λ:

$$f(\alpha) = \sup_{q \in \mathbb{R}} \{q\alpha - \Lambda(q)\}, \forall \alpha \in \mathbb{R}. \tag{5.2}$$

As described in Eq. (5.1), $\alpha_\tau = \langle i \rangle_\tau$ corresponds in the VoD case, to the mean number of users i observable over a period of time of length τ and $f(\alpha)$ relates to the probability of its occurrence as follows:

$$\mathbb{P}\{\langle i \rangle_\tau \approx \alpha\} \sim e^{\tau \cdot f(\alpha)}. \tag{5.3}$$

Interestingly also, if the process is strictly stationary (*i.e.* the initial distribution is invariant) the same large deviation spectrum $f(\cdot)$ can be estimated from a single trace,

provided that it is "long enough" [7]. This work proceeds as follows: At a scale τ, the trace is chopped into k_τ intervals $\{I_{j,\tau} = [(j-1)\tau, j\tau[,\ j = 1, \ldots, k_\tau\}$ of length τ and have (almost-surely), for all $\alpha \in \mathbb{R}$:

$$f_\tau(\alpha, \epsilon_\tau) = \frac{1}{\tau} \log \frac{\#\left\{j : \int_{I_{j,\tau}} \Phi(X_s)\mathrm{d}s \in [\alpha - \epsilon_\tau, \alpha + \epsilon_\tau]\right\}}{k_\tau} \tag{5.4}$$
$$\text{and } \lim_{\tau \to \infty} f_\tau(\alpha, \epsilon_\tau) = f(\alpha).$$

In practice, for the empirical estimation of the large deviations spectrum, a similar estimator as the one derived in [6] and also used in [44] has been employed. At scale τ, the values of the first and the second order derivatives of $\Lambda_\tau(q)$ (i.e. $\Lambda'_\tau(q)$ and $\Lambda''_\tau(q)$) are computed for each $q \in \mathbb{R}$, where

$$\Lambda_\tau(q) = \tau^{-1} \log \left(k_\tau^{-1} \sum_{j=1}^{k_t} \exp \left(q \int_{I_{j,\tau}} \Phi(X_s)\mathrm{d}s \right) \right).$$

Then, for each value of τ, the number of intervals $I_{j,\tau}$ is counted verifying the condition in expression (5.4). Thus this approach estimate the scale-dependant empirical *log-pdf* $f_\tau(\alpha, \epsilon_\tau)$, with the adaptive choices derived in [6]:

$$\alpha_\tau = \Lambda'_\tau(q) \quad \text{and} \quad \epsilon_\tau = \sqrt{\frac{-\Lambda''_\tau(q)}{\tau}}. \tag{5.5}$$

Now this work illustrates the LDP in the context of the specific VoD use case, where X would correspond to (i, r), the bi-variate Markov process. $\Phi(X)$ is i, the observable and $\frac{1}{\tau}\int_0^\tau \Phi(X_S)\, ds = \langle i \rangle_\tau$ corresponds to the average number of users within a period τ.

Intrinsically, Large Deviation Principle naturally embeds this time scale notion into the statistical description of the aggregated observable at different time resolutions. This chapter aims to demonstrate that the proposed model is able to provide both theoretical and empirical LD spectrums (one empirical spectrum for each time scale)

and these spectrums may be used to get the aggregated observations. Two random traces (one contains buzz and another is buzz-free) have been chosen for this. They illustrate the LD Principle and its relevance in the context of this thesis. Large deviation spectra for real traces have not been used in this chapter. The reason behind this conscious decision is numerical simplicity (computation of rate matrix of the Markov process becomes significantly computation intensive if the matrix size increases due to the high maximum value of i (i_{max}). Computation of the theoretical LD spectrum depends on the rate matrix. Therefore overall process of LD spectrum computation becomes computation intensive as well). The obtained results (see Fig. 5.2) from the two traces (with relatively lower value of i_{max}) show that the theoretical and empirical spectra superimpose, signifying the scale invariant property. Therefore it is possible to use only the empirical spectrum to derive the LD spectrum (Chapter. IV shows the empirical LD spectrum of real traces for $\tau = 0.001$ sec), thus circumventing the computation of the theoretical spectrum. Further research in this direction might include ways to find the numerical methods that can reduce the computational burdens of theoretical LD spectrum estimation.

As expected, the theoretical LD spectra displayed in Fig. 5.2(a) reach their maximum for the same mean number of users. This apex is the almost sure value as described in Section 5.2. As the name suggests almost sure workload ($\alpha_{a.s}$) corresponds to the mean value that is almost surely observed on the trace. More interestingly though, the LD spectrum corresponding to the buzz case, spans over a much larger interval of observable mean workloads than that of the buzz-free case. This remarkable support widening of the theoretical spectrum shows that LDP can accurately quantify the occurrence of extreme, yet rare events.

Plots (b)-(c) of Figure 5.2 compare theoretical and empirical large deviation spectra obtained for the two traces. For each given scale (τ) the empirical estimation procedure yields one LD estimate. These empirical estimates at different scales superimpose for a

given range of α. This is reminiscent of the scale invariant property underlying the large deviation principle. If the supports of the different estimated spectra is investigated it becomes evident that the larger the time scale τ is, the smaller becomes the interval of observable value of α. This is coherent with the fact that for a finite trace-length the probability to observe a number of current viewers, that in average, deviates from the nominal value ($\alpha_{a.s}$) during a period of time (τ) decreases exponentially fast with τ. To fix the ideas, the estimates of plot (c), indicate that for a time scale $\tau = 400\,sec.$, the maximum observable mean number of users is around 5 with probability is $2^{(-0.02\times400)} \approx 39.10^{-4}$ (point A), while it increases up to 9 with the same probability i.e. $2^{(-0.08\times100)}$ (i.e. 39.10^{-4}) for $\tau = 100\,sec.$ (point B).

It is possible to infer the probability distribution function of the random variable $\langle i \rangle_\tau$ (i.e. i calculated over the time interval τ) from the large deviation spectrum $(\alpha, f(\alpha))$. The quantity $e^{\tau f(\alpha)}$ is only proportional to the probability that $\langle i \rangle_\tau$ lies in the interval $I(q)_\tau = [\alpha(q) - \epsilon(q)_\tau, \alpha(q) + \epsilon(q)_\tau]$. As the intervals $I(q)_\tau$ overlap for different q's, it is necessary to compute the $\mathbb{P}(I(q)_\tau \cap I(q')_\tau)$. An algorithm which describes such a procedure has been illustrated in Appendix B.

Fig. 5.3 shows the probability density derived from the procedure in Appendix B and generated out of the large deviation spectrum of plot (c) of Fig. 5.2 for $\tau = 100$ and $\tau = 200$ respectively.

5.3 Probabilistic Provisioning Schemes

Retuning to the VoD use case, two possible schemes for exploiting the Large Deviation description of the system to dynamically provision the allocated resources are sketched here:

- *Identification of the reactive time scale for reconfiguration*: Find a relevant time scale that realizes a good trade-off between the expectable level of overflow associ-

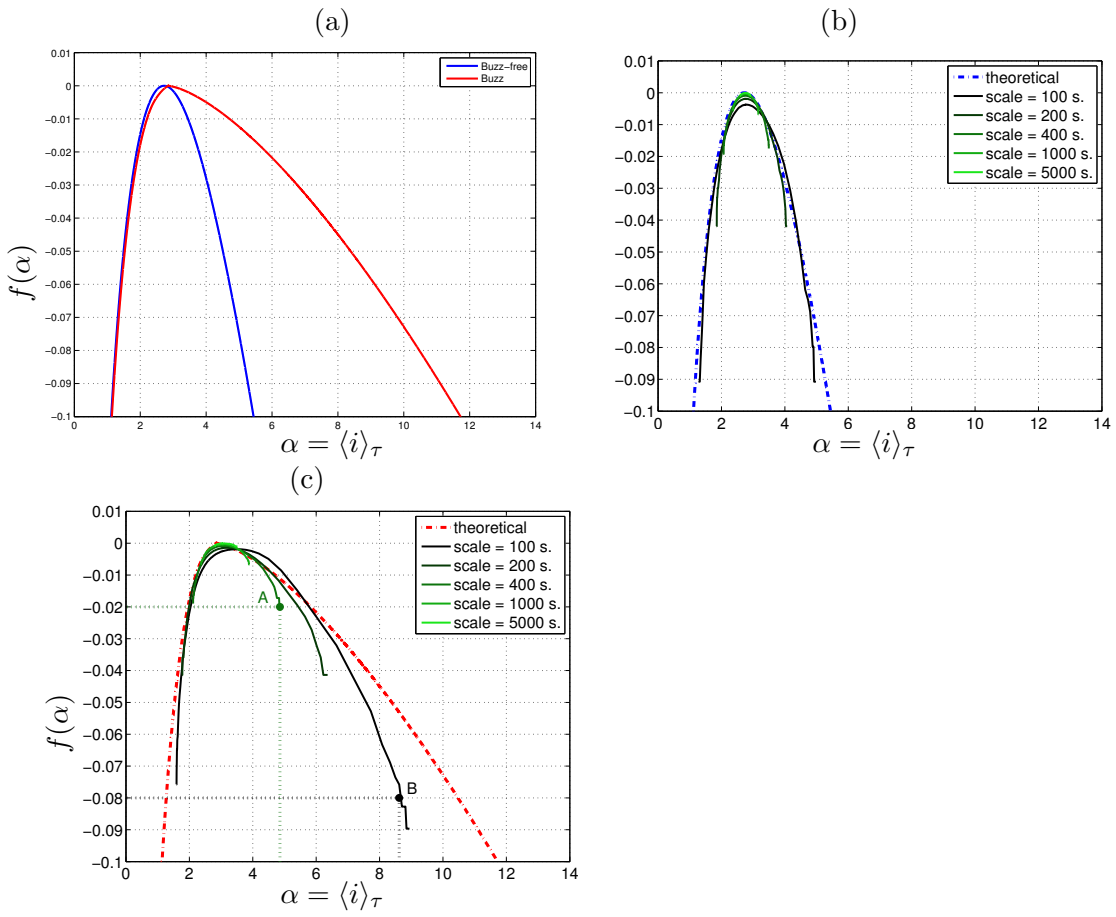


Figure 5.2: Large Deviations spectra corresponding to two traces generated from the proposed model. (a) Theoretical spectra for the buzz free (blue) and for the buzz (red) scenarii. (b) & (c) Empirical estimations of $f(\alpha)$ at different scales from the buzz free and the buzz traces, respectively.

ated to this scale and a sustainable OPEX cost induced by the resources reconfiguration needed to cope with the corresponding flash crowd.

- *Link capacity dimensioning*: Considering a maximum admissible loss probability, find the safety margin that it is necessary to provision on the link capacity, to guarantee the corresponding QoS.

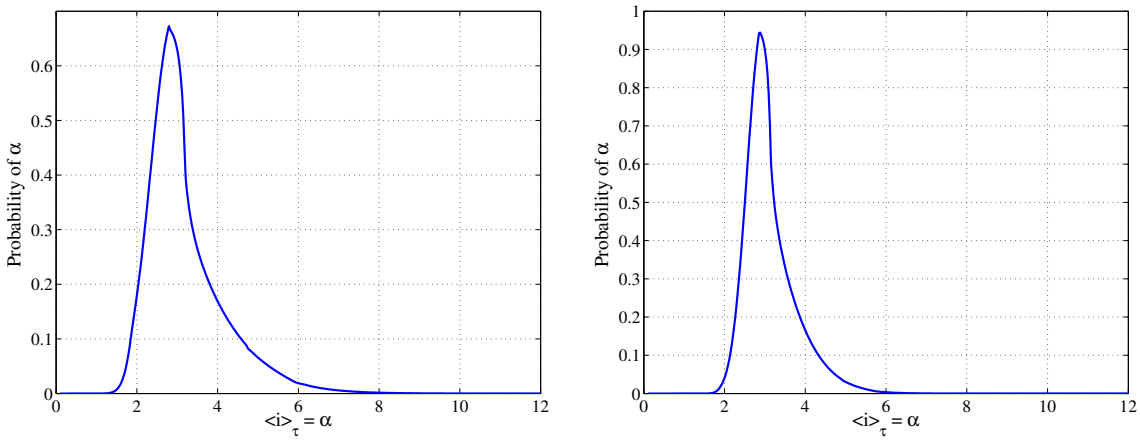


Figure 5.3: Probability density derived from the LD spectrum

5.3.1 Identification of the reactive time scale for reconfiguration

This study considers the case of a VoD service provider who wants to determine the reactivity scale at which it needs to reconfigure its resource allocation. This quantity should clearly derive from a good compromise between the level of congestion (or losses) it is ready to undergo, i.e. a tolerable performance degradation, and the price it is willing to pay for a frequent reconfiguration of its infrastructure. It is to be assumed that the VoD provider has fixed admissible bounds for these two competing factors, having determined the following quantities:

- $\alpha^* > \alpha_{\text{a.s.}}$: the deviation threshold beyond which it becomes worth (or mandatory) considering to reconfigure the resource allocation. This choice is uniquely determined by a CAPEX performance concern.

- σ^*: an acceptable probability of occurrence of these overflows. This choice is essentially guided by the corresponding OPEX cost.

Moreover it is assumed, that the LD spectrum $f(\alpha)$ of the workload process was previously estimated, either by identifying the parameters of the Markov model used to describe the application, or empirically from collected traces. Then, recalling the

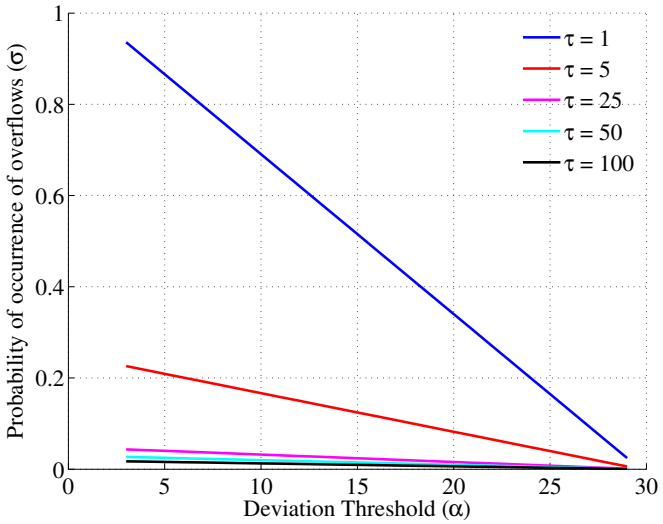


Figure 5.4: Deviation threshold vs. probability of occurrence of overflow for different values of time scale (τ).

probabilistic interpretation that has been surmised in relation (5.3), the minimum reconfiguration time scale τ^* for dynamic resource allocation, that verifies the sought compromise, is simply the solution of the following inequality:

$$\tau^* = \max \left\{ \tau : \mathbb{P}\{\langle i \rangle_\tau \geq \alpha^*\} = \frac{1}{\varrho} \int_{\alpha^*}^{\infty} e^{\tau f_\tau(\alpha)}\, d\alpha \geq \sigma^* \right\}, \tag{5.6}$$

with $f_\tau(\alpha)$ as defined in expression (5.4). ϱ is the normalization constant in Eq. (5.6).

From a more general perspective though, it is possible to see this problem as an under-determined system involving 3 unknowns (α^*,τ^* and σ^*) and only one relation (5.6). Therefore, and depending on the sought objectives, it is possible to fix any other two of these variables and to determine the resulting third so that it abides with the same inequality as in expression (5.6). Fig. 5.4 shows α vs. σ for different values of τ. It illustrates that for a smaller time scale (τ) the operators can guarantee a higher probability of occurrence (σ) for a given deviation threshold (α). But it implies a frequent reconfiguration of resources causing a higher OPEX cost.

5.3.2 Link capacity dimensioning

Next this work considers an architecture dimensioning problem from the infrastructure provider perspective. It is assumed that the infrastructure and the service providers have come to a Service Level Agreement (SLA), which among other things, fixes a tolerable level of losses due to link congestion. This work starts considering the case of a single VoD server and address the following question: What is the minimum link capacity C that has to be provisioned such that it is possible to meet the negotiated QoS in terms of loss probability? Like in the previous case, it is assumed that the estimated LD spectrum $f(\alpha)$ characterizing the application has been priorly identified. A rudimentary SLA would be to guarantee a loss free transmission for the *normal* traffic load only: this loose QoS would simply amount to fix C to the almost sure workload $\alpha_{\text{a.s.}}$. Naturally then, any load overflow beyond this value will result in good-put limitation (or losses, if there is no buffer to smooth out exceeding loads). For a more demanding QoS, the providers are led to determine the necessary safety margin $C_0 > 0$ one has to provision above $\alpha_{\text{a.s.}}$ (i.e. $C = \alpha_{\text{a.s.}} + C_0$) to absorb the exact amount of overruns corresponding to the loss probability p_{loss} that was negotiated in the SLA. From the interpretation of the large deviation spectrum provided in Section 5.2, this margin C_0 is determined by the resolution of the following inequality:

$$C_0 \quad : \quad \frac{1}{\varrho} \int_{\alpha_{\text{a.s.}}+C_0}^{\infty} e^{\tau \cdot f(\alpha)} \, \mathrm{d}\alpha \;\leq\; p_{\text{loss}} \tag{5.7}$$

ϱ is the normalization constant. τ is typically determined in accordance with the available buffer size that is usually provisioned to dampen the traffic volatility.

Based on the reactive time scale τ (fixed by the operator), as long as the server workload remains below C, this resource dimensioning guarantees that no loss occurs. All overrun above this value will produce losses, but it can be ensured that the frequency (probability) and duration of these overruns are such that the loss rate remains

conformed to the SLA. The proposed approach clearly contrasts with resource over-provisioning that does not seek at optimizing the CAPEX to comply with the loss probability tolerated in the SLA.

The same provisioning scheme can straightforwardly be generalized to the case of several applications sharing a common set of resources. To fix the idea, this work considers an infrastructure provider that wants to host K VoD servers over the same shared link. A corollary question is then to determine how many servers K can the fixed link capacity C support, while guaranteeing a prescribed level of losses. If the servers are independent, the probability for two of them to undergo a flash crowd simultaneously is negligible. For ease and without loss of generality, it can be assumed that they are identically distributed and modeled by the same LD spectrum $f^{(k)}(\alpha) = f(\alpha)$ with the same nominal workload $\alpha_{\text{a.s.}}^{(k)} = \alpha_{\text{a.s.}}$, $k = 1, \dots K$. Then, following the same reasoning as in the previous case of a single server, the maximum number K of servers reads:

$$K = \arg\max_{K} \left(C - K \cdot \alpha_{\text{a.s.}}\right) \leq C_0, \tag{5.8}$$

where the safety margin C_0 is defined as in expression (5.7).

Then, depending on the agreed *Service Level Agreements*, the infrastructure provider can easily offer different levels of probability losses (QoS) to its VoD clients, and adapt the number of hosted servers, accordingly.

5.4 Conclusion

Objective of this work is to harness probabilistic methods for resource provisioning in the Clouds. This thesis illustrates this purpose with a Video on Demand scenario, a characteristic service whose demand relies on information spreading. This work proposed a simple, concise and versatile model for generating the workload variations in such context by adopting a constructive approach that captures the users' behavior. A key-point

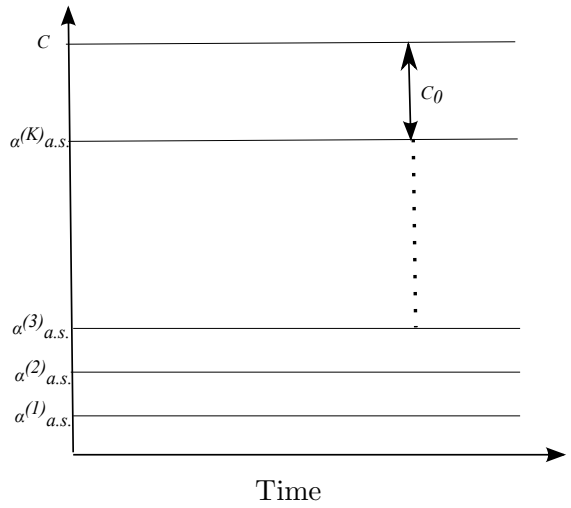


Figure 5.5: Dimensioning K, the number of hosted servers sharing a fixed capacity link C. The safety margin C_0 is determined according to the probabilistic loss rate negotiated in the *Service Level Agreement* between the infrastructure provider and the VoD service provider.

of this model is that it permits to reproduce the workload time series with a Markovian process, which is known to verify a Large Deviation Principle (LDP). This particularly interesting property yields a large deviation spectrum whose interpretation enriches the information conveyed by the standard steady state distribution of the Markovian process. For a given observation (workload trace), LDP allows to infer (theoretically and empirically) the probability that the time average workload, calculated at an arbitrary aggregation scale, deviates from its nominal value (i.e. almost sure value).

This work leveraged this multi-resolution probabilistic description to conceptualize two different management schemes for dynamic resource provisioning. As explained, the rationale is to use large deviation information to help network and service providers together to agree on the best CAPEX-OPEX trade-off. Two major stakes of this negotiation are: *(i)* to determine the largest reconfiguration time scale adapted to the workload elasticity and *(ii)* to dimension VoD server so as to guarantee with utmost probability the Quality of Service imposed by the negotiated Service Level Agreement.

But, as mentioned previously this method is compute intensive (in terms of compu-

tation time) if someone is interested to compute the theoretical LDS for a system with a high value of maximum number of users. But complexity analysis of this method has not been done in course of his work. Therefore future research in this direction can be:

- Complexity analysis of the existing method and finding a better (less compute intensive) approach to compute the theoretical LDS
- Using the similar LDP based concepts to benefit other "Service on Demand" scenarii to be deployed on dynamic cloud environments.

Chapter 6

Conclusion

6.1 Main contributions

This work introduces an epidemic model inspired and continuous time Markov Chain based model for a Video on Demand (VoD) system. The model reproduces workload volatility, especially the buzz effects that can cause significant overload on the VoD applications. Since a workload model can not be exploited without proper calibration, two estimation procedures have also been proposed. Obtained results demonstrate that the calibrated models can exhibit statistically similar properties to a real workload trace. This work also provides a comparison of the two estimation procedures. While the heuristic procedure is based on the model mechanism (providing an intuitive solution) the MCMC procedure is a natural choice for this problem (dealing with hidden states and missing values). But, the experimental results suggest that the MCMC approach can be computationally heavier than the heuristic procedure. This thesis also compares the proposed model with some other existing models in terms of the goodness-of-fit of some statistical properties of real workload traces. Finally it suggests probabilistic resource provisioning approaches based on a Large Deviation Principle (LDP), which characterizes the buzz effects. Specifically this work designs two resource management

policies to accommodate (resource management policies) workload volatility.

6.2 Challenges

A workload model can not be considered pragmatic, unless it is compared against real traces. This thesis faced significant challenges to collect real workload traces for model validation. Only two suitable VoD traces were possible to collect to demonstrate the data model adequacy (see Section. 4.3). Another significant difficulty came from calibrating the model using an MCMC procedure. Since the model has several hidden values (actual number of past viewers, whether the model is in buzz or buzz-free state etc.) it was not trivial to utilize the MCMC framework. Finally there were some technical challenges during implementation of the model on the Grid 5000 test-bed. However, the thesis demonstrates implementation of an efficient server to manage the significant amount of communication among hundreds of nodes, that mimics the user behavior of the VoD system.

6.3 Originality and limitation of the work

This work describes an original as well as a complete framework, which

- introduces a pragmatic, yet a concise model for generating workload of a VoD system
- provides frameworks to calibrate the proposed workload model
- proposes resource management policies to efficiently handle workload volatility

The second originality of this work stems from the analysis of the Large Deviation property of the proposed Markovian model. This thesis leverages these large deviation properties to frame the resource management policies for the overall framework.

However, this work has the following limitations

- the calibration procedures are compute extensive (in terms of computation time) and therefore makes the overall system not suitable for using in an on-line system, where a quick model fitting is an essential criterion
- computation of the theoretical LD spectrum can be significantly heavy for a high value of maximum number of viewers

6.4 Future works

Overall, this thesis provides an useful framework for VoD service providers. However, it night be feasible to enhance the functionality of the overall framework by developing a less compute intensive approach for model calibration of LD spectrum calculation. For this purpose an exhaustive complexity analysis needs to be carried out. Furthermore, the proposed VoD workload model can be adapted to capture the user behavior in a social network (Facebook, Twitter or youtube). Since the social network has a significant influence in modern economy (e.g. advertising a product) and social ecosystem (e.g. the Arab Spring), a constructive model can be highly useful from multiple perspective. Finally, the similar LDP based resource management approaches can also be used to benefit some other Service on Demand applications (one simple example can be simply having an option to tap a smart-phone application and receive a range of services in the form of on-demand labor) to be deployed on dynamic cloud environments.

Appendix A

Proofs of Chapter. 4

Proof of Eq. (4.1)

Let T be a non-negative continuous random variable which represents the waiting time until an event occurs. It can be assumed that the probability density function of T be $f(t)$. Then the cumulative distribution function $F(t) = p(T < t)$ gives the probability that the event has occurred by duration t. Therefore

$$S(t) = p(T \geq t) = 1 - F(t) = \int_t^\infty f(x)\mathrm{d}x \tag{A.1}$$

Now the instantaneous rate of occurrence of an event can be defined as:

$$\lambda(t) = \lim_{\mathrm{d}t \to 0} \frac{p(t \leq T < t + \mathrm{d}t | T \geq t)}{\mathrm{d}t} \tag{A.2}$$

The numerator of Eq. (A.2) is the conditional probability that the event would occur in the interval $(t, t + \mathrm{d}t)$ provided it has not occurred before. The denominator denotes the width of the interval.

The conditional probability of the numerator can be written as the ratio of the joint probability that T is in the interval $(t, t + \mathrm{d}t)$ and $T > t$ to the probability of the condition $T > t$. The former can be written as $f(t)\mathrm{d}t$ for small $\mathrm{d}t$, while the latter is

$S(t)$ by definition. Cancelling dt and passing to the limits it is possible to obtain:

$$\lambda(t) = \frac{f(t)}{S(t)} \tag{A.3}$$

From Eq. (A.1) $f(t)$ is the derivative of $S(t)$. Then it is possible to rewrite the Eq. (A.3) as

$$\lambda(t) = -\frac{\mathrm{d}}{\mathrm{d}t}\log S(t) \tag{A.4}$$

Integrating Eq. (A.4) between 0 and t and given $S(0) = 1$ it is possible to obtain Eq. (4.1):

$$S(t) = p(T \geq t) = \exp\Big(\int_0^t \lambda(x)\mathrm{d}x\Big)$$

Proof of Claim. 1:

Suppose there are n_3 individuals who stopped gossiping within an interval $[0, T]$. It can be assumed that the time at which they stop gossiping follows an uniform distribution. The probability that another person stops gossiping at time t_s within period $(y, y + x]$ is

$$P(t_s \in (y, y + x] \subset (0, T]) = \frac{x}{T} \tag{A.5}$$

Lets denote $A((y, y + x])$ being the number of people who stopped gossiping within the period $(y, y + x]$. Therefore the probability that k people stops gossiping during that period is

$$P(A((y, y + x]) = k) = \frac{n_3!}{k!(n_3 - k)!}\Big(\frac{x}{T}\Big)^k\Big(1 - \frac{x}{T}\Big)^{n_3-k} \tag{A.6}$$

Since, the average number of people stops gossiping within period T is $\lambda = \frac{n_3}{T}$ the Eq. (A.6) can be rewritten as

$$P(A((y, y + x]) = k) = e^{-\lambda x} \cdot \frac{(\lambda x)^k}{k!} \tag{A.7}$$

Eq. (A.7) shows that the number of people stops gossiping within the period $(y, y + x]$

follows a Poisson distribution. Furthermore, let $Z(t_0)$ be the event that one viewer stops gossiping at t_0 and X be the time interval to next person to stop gossiping. Then, Eq. (A.7) can be rewritten as

$$\begin{aligned} P(X > t|Z(t_0)) &= P(A((t_0, t_0 + t]) = 0|Z(t_0)) \\ &= P(A((t_0, t_0 + t]) = 0) \\ &= e^{-\lambda t} \end{aligned} \tag{A.8}$$

Eq. (A.8) shows that the cumulative distribution function and the probability density function of the gossip stopping interval follows an exponential distribution of

$$P(X < t|Z(t_0)) = 1 - e^{-\lambda t} \tag{A.9}$$

$$P(t) = \lambda e^{-\lambda t} \tag{A.10}$$

Appendix B

Algorithm of Chapter. 5

Procedure that returns a vector (1-by-N) of the probability density function of $\langle i \rangle_\tau$ estimated on a uniformly spaced and continuous grid of length N.

Input of this procedure are as follows:

- A vector (1-by-Q) p containing the values $e^{\tau f(\alpha(q))}$ for a given τ and for Q values of q (obtained from the spectrum $f(\alpha)$)
- A vector (1-by-Q) containing the values of $\alpha(q)$
- A vector (1-by-Q) containing the values of $\epsilon(q)_\tau$

The overall procedure can be summarised as follows:

- Compute i_τ which represents uniformly spaced samples of $\langle i \rangle_\tau$ over a grid of length N
- Compute $I(q)_\tau = [\alpha(q) - \epsilon(q)_\tau, \alpha(q) + \epsilon(q)_\tau]$
- For $n = 1$ to N
  - Find the indexes (idx) such that $i_\tau(n-1) \leq \alpha(q)+\epsilon(q)_\tau$ or $i_\tau(n) \geq \alpha(q)-\epsilon(q)_\tau$
  - Compute $r = \frac{i_\tau(n)-i_\tau(n-1)}{I(q_{idx})_\tau(:,2)-I(q_{idx})_\tau(:,1)}$

- Compute $w = \frac{r}{\sum_N r}$
- $\mathbb{P}(n-1) = \sum_N w \cdot p(idx) \cdot r$

Bibliography

[1] A.K. Agrawala, Mohr J.M, and R.M. Bryant. An approach to the workload characterization problem. *IEEE Computer*, 1976.

[2] M. Arlitt and T. Jin. Workload characterization of the 1998 world cup web site. Itechnical report hpl-1999-35r1, HP Labs, 1999.

[3] Martin Arlitt and Tai Jin. Workload characterization of the 1998 world cup web site. Technical report, IEEE Network, 1999.

[4] R. Badonnel, R. State, and O. Festor. Probabilistic management of ad-hoc networks. In *10th IEEE/IFIP NOMS*, 2006.

[5] P. Barford and M. Crovella. Generating representative web workloads for network and server performance evaluation. In *SIGMETRICS*, 1998.

[6] J. Barral and P. Gonçalves. On the estimation of the large deviations spectrum. *Journal of Statistical Physics*, 144(6):1256–1283, 2011.

[7] J. Barral and P. Loiseau. Large deviations for the local fluctuations of random walks. *Stochastic Processes and their Applications*, 121(10):2272–2302, 2011.

[8] A. Barrat, M. Barthelemy, and A. Vespignani. *Dynamical Processes on Complex Networks*. Cambridge University Press, 2008.

[9] A. Bashar, G. P. Parr, S. I. McClean, B. W. Scotney, and D. Nauck. Bard: A novel application of bayesian reasoning for proactive network management. In *10th Annual Conference on the Convergence of Telecommunication, Networking and Broadcasting*, 2009.

[10] D. F. Bernardes, M. Latapy, and F. Tarissan. Inadequacy of sir model to reproduce key properties of real-world spreading phenomena: Experiments on a large-scale p2p system. *Social Network Analysis and Mining (SNAM)*, 2013.

[11] G. Bianchi and R. Melen. The role of local storage in supporting video retrieval services on atm networks. *IEEE Networking*, 1997.

[12] M. Brunner, D. Dudkowski, C. Mingardi, and G. Nunzi. Probabilistic decentralized network management. In *IFIP/IEEE International Symposium on Integrated Network Management*, 2009.

[13] E. Caron, F. Desprez, and A. Muresan. Pattern matching based forecast of non-periodic repetitive behavior for cloud clients. *J. of Grid Comp.*, 2011.

[14] X. Chen and X. Zhang. A popularity-based prediction model for web prefetching. *IEEE Computer*, 2003.

[15] K.C. Cho, T.Y. Kim, and J.S. Lee. User demand prediction-based resource management model in grid computing environment. In *2008 International Conference on Convergence and Hybrid Information Technology*, 2008.

[16] Faban. Faban project web site. http://faban.sunsource.net/.

[17] R. Garcia, X. Paneda, V. Garcia, D. Melendi, and M. Vilas. Statistical characterization of a real video on demand service: User behaviour and streaming-media workload analysis. *Simul Model Pract Th*, 2007.

[18] J. Geweke and H. Tanizaki. Bayesian estimation of state-space models using the metropolis-hastings algorithm within gibbs sampling.

[19] H. Goudarzi and M. Pedram. Multi-dimensional sla-based resource allocation for multi-tier cloud computing systems. In *4th International Conference on Cloud Computing*, 2011.

[20] GRNET. Video traces obtained from grnet, 2011. http://vod.grnet.gr/.

[21] K. P. Gummadi, R. J. Dunn, S. Saroiu, S.D. Gribble, H. M. Levy, and J. Zahorjan. Measurement, modeling, and analysis of a peer-to-peer file-sharing workload. In *ACM symposium on Operating systems principles*, 2003.

[22] R. Gusella. Characterizing the variability of arrival processes with indexes of dispersion. *IEEE JSAC*, 9(2), 1991.

[23] G. Haring. On stochastic models of interactive workloads. 1983.

[24] Httperf. Httperf project web site. http://code.google.com/p/httperf/.

[25] W. Hu et al. Multiple-job optimization in mapreduce for heterogeneous workloads. In *Sixth International Conference on Semantics Knowledge and Grid (SKG)*, 2010.

[26] T. C. K. Hui and C. K. Thanm. Adaptive provisioning of differentiated services networks based on reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics*, 2003.

[27] T.T Huu and J. Montagnat. Virtual resources allocation for workflow-based applications distribution on a cloud infrastructure. In *10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid)*, 2010.

[28] W. Iqbal et al. Adaptive resource allocation for back-end mashup applications on a heterogeneous private cloud. In *International Conference on Electrical En-*

gineering/Electronics Computer Telecommunications and Information Technology (ECTI-CON), 2010.

[29] S.R. Jammalamadaka and E. Taufer. Testing exponentiality by comparing the empirical distribution function of the normalized spacings with that of the original data. *J. Nonparametric Statistics*, 15, 2003.

[30] Jmeter. Jmeter project web site. http://jakarta.apache.org/jmeter/.

[31] S. Kanrar. Analysis and implementation of the large scale video-on-demand system. *IJAIS*, 2(2), 2012.

[32] K. Keahey, M. Tsugawa, A. Matsunaga, and J.A.B Fortes. Sky computing. *Internet Computing*, 2009.

[33] J. Kleinberg. Bursty and hierarchical structure in streams. In *ACM SIGKDD*, 2002.

[34] Z. Kong et al. Mechanism design for stochastic virtual resource allocation in non-cooperative cloud systems. In *4th International Conference on Cloud Computing*, 2011.

[35] P.G. Kulkarni, S. I. McClean, G. P. Parr, and M. M. Black. Deploying mib data mining for proactive network management. In *3rd Intl. IEEE Conference on Intelligent Systems*, 2006.

[36] P.G. Kulkarni, S.I. McClean, G.P. Parr, and M.M. Black. Proactive predictive queue management for improved qos in ip networks. In *IEEE ICN/ICONS/MCL*, 2006.

[37] K. Kumar et al. Resource allocation for real-time tasks using cloud computing. In *20th International Conference on Computer Communications and Networks (ICCCN)*, 2011.

[38] Y.C Lee, W. Chen, A.Y Zomaya, and B.B. Zhou. Profit-driven service request scheduling in clouds. In *10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid)*, 2010.

[39] J. Leskovec, M. McGlohon, C. Faloutsos, N. Glance, and Matthew Hurst. Cascading behavior in large blog graphs. In *7th SIAM International Conference on Data Mining (SDM)*, 2007.

[40] J. Li et al. Adaptive resource allocation for preemptable jobs in cloud systems. In *10th International Conference on Intelligent Systems Design and Applications (ISDA)*, 2010.

[41] V.O.K. Li, W. Lao, X. Qiu, and E. W. M. Wong. Performance model of interactive video-on-demand systems. *IEEE JSAC*, 14, 1996.

[42] W-Y. Lin, G-Y. Lin, and H-Y. Wei. Dynamic auction mechanism for cloud resource allocation. In *10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid)*, 2010.

[43] A. L. Lloyd. Realistic distributions of infectious periods in epidemic models: Changing patterns of persistence and dynamics. *Thoretical Population Biology*, 2013.

[44] P. Loiseau, P. Gonçalves, J. Barral, and P. Vicat-Blanc Primet. Modeling TCP throughput: an elaborated large-deviations-based model and its empirical validation. In *Proceedings of IFIP Performance*, Nov 2010.

[45] X. Lu, J. Lin, L. Zha, and Z. Xu. Vega lingcloud: A resource single leasing point system to support heterogeneous application modes on shared infrastructure. In *9th International Symposium on Parallel and Distributed Processing with Applications (ISPA)*, 2011.

[46] R.T.B. Ma, D. M. Chiu, J.C.S. Lui, V. Misra, and D. Rubenstein. On resource management for cloud users: A generalized kelly mechanism approach. Technical report, Electrical Engineering, 2010.

[47] S. Majumdar. The any-schedulability criterion for providing qos guarantees through advance reservation requests. In *9th IEEE/ACM International Symposium on Cluster Computing and the Grid*, 2009.

[48] S. Majumdar. Resource management on clouds: Handling uncertainties in parameters and policies. *CSI Communications*, 2011.

[49] P. Marshall, K. Keahey, and T. Freeman. Elastic site: Using clouds to elastically extend site resources. In *10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid)*, 2010.

[50] J. Melendez and S. Majumdar. Matchmaking with limited knowledge of resources on clouds and grids. In *International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS)*, 2010.

[51] D. Melendi, R. Garcia, X. G. Paneda, and V. Garcia. Multivariate distributions for workload generation in video on demand systems. *IEEE Comm. Letters*, 13, 2009.

[52] D. Minarolli and B. Freisleben. Utility-based resource allocation for virtual machines in cloud computing. In *IEEE Symposium on Computers and Communications (ISCC)*, 2011.

[53] M. Naldi. A mixture model for the connection holding times in the video-on-demand service. *Performance Evaluation*, 2002.

[54] Netflix. Netflix vod services. https://www.netflix.com/global.

[55] D. Niyato, K. Zhu, and P. Wang. Cooperative virtual machine management for multi-organization cloud computing environment. In *5th International ICST Conference on Performance Evaluation Methodologies and Tools*, 2010.

[56] P. D. O'Neil and G.O. Roberts. Bayesian inference for partially observed stochastic epidemics. *J.R Statist. Soc*, 1999.

[57] S. Pacheco-Sanchez, G. Casale, B. Scotney, S. McClean, G. Parr, and S. Dawson. Markovian workload characterization for QoS prediction in the cloud. In *IEEE Cloud*, 2011.

[58] D. Perez-Palacin, J. Merseguer, and R. Mirandola. Analysis of bursty workload-aware self-adaptive systems. In *ICPE*, 2012.

[59] F.I. Popovici and J. Wilkes. Profitable services in an uncertain world. In *ACM/IEEE Supercomputing Conference*, 2005.

[60] A.G. Prieto, D. Gillblad, R. Steinert, and A. Miron. Toward decentralized probabilistic management. *IEEE Communications Magazine*, 2011.

[61] S.L.G. Quan and S. Ren. On-line scheduling of real time services for cloud computing. In *World Congress on Service*, 2010.

[62] J. Revzina. Possibilities of MMPP processes for bursty traffic analysis. In *RELSTAT*, 2010.

[63] P. Ruth et al. Autonomic adaptation of virtual computational environments in a multi-domain infrastructure. In *International Conference on Autonomic Computing*, 2006.

[64] Sandvine. http://www.sandvine.com/news/pr_detail.asp?ID=312/.

[65] S. Sohail and A. Khanum. Simplifying network management with fuzzy logic. In *IEEE Itnl. Conf. on Communications*, 2008.

[66] T. Spangler. http://variety.com/2014/digital/news/abcs-live-internet-oscar-stream-suffers-nationwide-outage-1201124215/.

[67] W. J. Stewart. *Probability, Markov Chains, Queues, and Simulation: The Mathematical Basis of Performance Modeling*. Princeton Press, 2009.

[68] W-T. Tsai, Shao Q, X. Sun, and J. Elston. Service oriented cloud computing. In *World Congress on Service*, 2011.

[69] B. Urgaonkar, P. Shenoy, A. Chandra, P. Goyal, and T. Wood. Agile dynamic provisioning of multi-tier internet applications. *ACM Trans. on Autonomous and Adaptive Systems*, 2008.

[70] S.R.S. Varadhan. Large deviations. *The Annals of Probability*, 36(2):397–419, 2008.

[71] J. Browne W. *MCMC estimation in MLwiN*. University of Bristol, 2012.

[72] X. Wang et al. Design and implementation of adaptive resource co-allocation approaches for cloud service environments. In *3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*, 2010.

[73] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif. Black-box and gray-box strategies for virtual machine migration. In *4th USENIX conference on Networked systems design and implementation*, 2007.

[74] L. Wu, S.K. Garg, and R. Buyya. Sla-based resource allocation for software as a service provider (saas) in cloud computing environments. In *11th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid)*, 2011.

[75] F. Wuhib and R. Stadler. Distributed monitoring and resource management for large cloud environments. In *FIP/IEEE International Symposium on Integrated Network Management (IM)*, 2011.

[76] Z. Xiaoyun et al. 1000 islands: Integrated capacity and workload management for the next generation data center. In *International Conference on Autonomic Computing*, 2008.

[77] X. You, X. Xu, J. Wan, and D. Yu. Ras-m: Resource allocation strategy based on market mechanism in cloud computing. In *Fourth ChinaGrid Annual Conference*, 2009.

[78] H. Yu, D. Zheng, B. Y. Zhao, and W. Zheng. Understanding user behavior in large-scale video-on-demand systems. In *Proceedings of the 1st ACM SIGOPS/EuroSys European Conference on Computer Systems*, 2006.