
Analysis and design of communication systems in today’s dynamically changing environment face a lot of challenges. Communication systems are not static but are dynamically changing today. Networks can become overloaded at one moment and underutilized at the next. Humans simply cannot respond to changes fast enough; computer programs must be used. Queueing theory has typically been used to address queueing problems of communication networks under the assumption that the inter-arrival times and the service times are renewal processes, and the arrival rate and bandwidth requirement of each customer is constant. In today’s dynamically changing environment, such as in high-speed integrated-services wired and wireless networks and Web service, these assumptions are frequently unjustified. Our research focuses on the performance evaluation and efficient resource dimensioning algorithms of communication systems in a dynamically changing environment. The topics investigated are either new or they have not been studied well in the literature. One of our contributions is providing efficient dimensioning algorithms in closed-form with a constant or near linear complexity.

In this thesis, we considered the following cases: a) The arrival rates of customers are time-varying, b) the bandwidth requirement of a customer is variable, c) integrated streaming and data traffic where the available bandwidth for the data traffic is dynamically changing and d) a statistical multiplexer with autocorrelated and bursty arrival processes.

This work starts with the dynamic analysis and design of a communication link when the arrivals are time-varying. The time-varying arrivals capture the nature of a dynamic traffic environment. Using transient analysis, we dimension a link under time-varying arrivals and transient load. We compare some methods (both numerical and closed-form) in the literature and propose efficient ways to design the system under study.

We next consider the case when the customer’s bandwidth requirements are time-varying. A multi-rate loss model with class-change is proposed and the blocking probability is calculated using an iterative recursive formula. The optimal dimensioning of the capacity of this system is obtained using a formula with a constant cost.
Subsequently we consider a model to analyze the integrated streaming traffic and data traffic, where the data traffic is bursty with a time-varying bandwidth shared with the streaming traffic. Using priority decomposition, we proposed closed-form solution for dimensioning of both streaming and data traffic.

Finally we analyze a statistical multiplexer where the arrival process is the superposition of $n$ independent identical bursty and autocorrelated arrival processes. We construct the superposition of these arrival processes and characterize the departure process from the multiplexer by exact Laplace transform and a two-stage Markov-modulated Poisson process. These results permit us to analyze and dimension the system more efficiently under study.
Analytical Models and Efficient Dimensioning Algorithms for Communication Systems In Randomly Changing Traffic Environments

by

Wenhong Tian

A dissertation submitted to the Graduate Faculty of North Carolina State University in partial fulfillment of the requirements for the Degree of Doctor of Philosophy

Computer Science

Raleigh, North Carolina

2007

Approved By:

Dr. Khalid Harfoush
Dr. William Stewart

Dr. Harry Perros
Chair of Advisory Committee
Dr. Mike Devetsikiotis
To my parents, my brothers and sisters, my own family and friends

—Wenhong Tian
Biography

Wenhong Tian has received the Master of Science from the department of Computer Science at the University of Missouri-Kansas City in 2004. He joined the department of Computer Science of North Carolina State University in the Spring of 2005. Wenhong’s research interests include performance analysis and efficient dimensioning algorithms for wired and wireless networks and Web servers in a randomly changing environment. Wenhong likes to solve complex problems in a simper and more efficient way with a strong desire to make people’s life easier.
Acknowledgements

I would like to thank my advisor, Professor Harry Perros for his continuous support and understanding. He helped me find interesting research topics and guided me through solving difficult problems. I am grateful for his valuable advice and expert insight in this research as well as the composition of this dissertation. I also would like to thank Dr. Michael Devetsikiotis, Dr. William Stewart and Dr. Khalid Harfoush for serving as advisors on my committee and for their constructive input. I would like to express my sincere thanks to my parents, my wife, my brothers and sisters for their encouragement, patience and prayers. I have lots of fun when I play with my son during the interval of study and research. The great thing is that he is humorous and always makes me laugh and shows happiness even right after crying for a while. I also would like to thank many of friends (names are not listed here to save some space, they know who they are.).
Contents

List of Figures vii
List of Tables viii

1 Introduction and Motivation 1
1.1 Research motivation ............................................. 1
1.2 Literature review ................................................. 2
1.3 Thesis organization .............................................. 6

2 The Dynamic Analysis and Design of A Communication link with Stationary and Nonstationary Arrivals 7
2.1 Introduction ......................................................... 8
2.2 The nonstationary Erlang loss system .......................... 9
  2.2.1 Stationary Arrivals ............................................ 10
  2.2.2 Time-varying arrivals ........................................ 11
  2.2.3 Dimensioning a single link .................................. 12
2.3 Exact closed-form solutions for the time-dependent blocking probability ...... 15
  2.3.1 Stationary Arrivals ............................................ 15
  2.3.2 Nonstationary arrivals ........................................ 18
2.4 The truncated Markov process approximation ................. 19
2.5 Numerical solutions ................................................ 21
  2.5.1 Differential equations solver ................................ 21
  2.5.2 The fixed point approximation (FPA) method .............. 21
2.6 The large deviation approach ................................... 23
  2.6.1 Stationary arrivals ............................................ 23
  2.6.2 Nonstationary arrivals ........................................ 24
2.7 More numerical results ........................................... 24
2.8 Conclusions ......................................................... 26

3 Analysis And Provisioning of a Circuit-switched Link with Variable-Demand Customers 28
3.1 Introduction ......................................................... 28
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>The Markov chain of Erlang loss model for a single service center</td>
<td>9</td>
</tr>
<tr>
<td>2.2</td>
<td>The time-dependent blocking probabilities of the stationary arrival for $M/M/8/8$, with offered load $\rho = 2$</td>
<td>11</td>
</tr>
<tr>
<td>2.3</td>
<td>The time-dependent blocking probabilities of the nonstationary arrival where $\lambda(t) = 180 + 50\sin(2(t + 20))$ and $s=220$</td>
<td>12</td>
</tr>
<tr>
<td>2.4</td>
<td>A dimensioning example of a single link with stationary arrival rates</td>
<td>13</td>
</tr>
<tr>
<td>2.5</td>
<td>A dimensioning example of a single link with nonstationary arrival rates</td>
<td>14</td>
</tr>
<tr>
<td>2.6</td>
<td>A comparison among the exact solution (Exact), the truncated Markov process approximation (TR), the Bahadur-Rao approximations (BR) and the large deviations approximations (LD)</td>
<td>26</td>
</tr>
<tr>
<td>2.7</td>
<td>A dimensioning example of a single link with nonstationary arrival rates</td>
<td>27</td>
</tr>
<tr>
<td>4.1</td>
<td>The finite macro state continuous time Markov chain model</td>
<td>53</td>
</tr>
<tr>
<td>4.2</td>
<td>The state space of exact solution</td>
<td>53</td>
</tr>
<tr>
<td>5.1</td>
<td>A typical three-level access network</td>
<td>63</td>
</tr>
<tr>
<td>5.2</td>
<td>A typical dimensioning example of an ADSL access network</td>
<td>80</td>
</tr>
</tbody>
</table>
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Approximation and simulation results for 3 classes customers</td>
<td>38</td>
</tr>
<tr>
<td>3.2</td>
<td>Approximation (Appr.) and simulation (Sim) results for 100 classes customers (1)</td>
<td>38</td>
</tr>
<tr>
<td>3.3</td>
<td>Approximation (Appr.) and simulation (Sim) results for 100 classes customers (2)</td>
<td>39</td>
</tr>
<tr>
<td>3.4</td>
<td>Call blocking comparison among variable-demand service, Max and Min service</td>
<td>40</td>
</tr>
<tr>
<td>3.5</td>
<td>The optimized capacity vs. offered load ($\rho$) for 3-classes traffic</td>
<td>40</td>
</tr>
<tr>
<td>4.1</td>
<td>The analytical and simulated mean waiting time of the data traffic (IPP/M/1 model)</td>
<td>55</td>
</tr>
<tr>
<td>4.2</td>
<td>The optimum capacity for the streaming traffic (with priority) and data traffic (IPP/M/1 model) when the mean waiting time $W=1$ second</td>
<td>56</td>
</tr>
<tr>
<td>4.3</td>
<td>The mean waiting time of IPP/G/1 model with two moments of service time distribution</td>
<td>57</td>
</tr>
<tr>
<td>4.4</td>
<td>The optimized capacity comparison for streaming traffic using two moments</td>
<td>57</td>
</tr>
<tr>
<td>4.5</td>
<td>The capacity comparison of different service time distributions when $W = 0.1$</td>
<td>59</td>
</tr>
<tr>
<td>4.6</td>
<td>Exact and disaggregation results for the blocking probability of the streaming traffic</td>
<td>59</td>
</tr>
<tr>
<td>5.1</td>
<td>Comparison of the mean waiting time $W$ and the mean number of customers $N$ in the system obtained by exact and MMPP$_2$ approximation for $n=10$</td>
<td>83</td>
</tr>
<tr>
<td>5.2</td>
<td>Comparison of the mean waiting time $W$ and the mean number $N$ of customers in the system obtained by exact and MMPP$_2$ approximation for $n=20$</td>
<td>83</td>
</tr>
<tr>
<td>5.3</td>
<td>Comparison of the mean waiting time $W$ and mean number $N$ of customers in the system obtained by exact and MMPP$_2$ approximation for $n=100$</td>
<td>83</td>
</tr>
<tr>
<td>5.4</td>
<td>Comparison of exact and approximate results of the superposition of 20 homogeneous MMPP$_2$</td>
<td>84</td>
</tr>
<tr>
<td>5.5</td>
<td>Comparison of analytical and simulation results of the departure process of MMPP$_2$/M/1</td>
<td>85</td>
</tr>
</tbody>
</table>
5.6 Comparison of ACFs from analytical and simulation results of the departure process of an \( \text{MMPP}_2/\text{M}/1 \) ........................................ 86
5.7 Comparison of exact and approximate (appr) results of the departure process of an \( \text{MMPP}_2/\text{M}/1 \) for low burstiness \( H=0.54, \text{MMPP}_2(0.5, 0.5, 1, 2) \) ........................................ 86
5.8 Comparison of analytical and simulation (appr) results of the departure process of \( \text{MMPP}_2/\text{M}/1 \) for medium burstness \( H=0.73, \text{MMPP}_2 (0.9, 0.1, 10, 1) \) ........................................ 87
5.9 Comparison of exact and approximate (appr) results of the departure process of an \( \text{MMPP}_2/\text{M}/1 \) for high burstness \( H=0.91, \text{MMPP}_2 (0.09, 0.01, 10, 1) \) ........................................ 87
5.10 Comparison of exact and approximate results of the departure process of an \( \text{MMPP}_2/\text{M}/1 \) for low burstness \( C^2=1.1429, \text{MMPP}_2 (0.1, 10.1, 1.1, 1.1) \) ........................................ 88
5.11 Comparison of exact and approximate results of the departure process of an \( \text{MMPP}_2/\text{M}/1 \) for medium burstness \( C^2=15.6341, \text{MMPP}_2 (0.1, 6.1, 0.1, 0.1) \) ........................................ 88
5.12 Comparison of exact and approximate (appr) results of the departure process of an \( \text{MMPP}_2/\text{M}/1 \) for high burstness \( C^2=23.1331, \text{MMPP}_2 (0.1, 6.1, 0.1, 0.1) \) ........................................ 89
5.13 Comparison of analytical and simulation results of the mean number \( (N) \) and response time \( (T) \) in the second node with offered load 0.2 ................. 89
5.14 Comparison of analytical and simulation results of the mean number \( (N) \) and response time \( (T) \) in the second node with offered load 0.9 ................. 90
5.15 Comparison of dimensioning results using equivalent bandwidth approach for 100*IPP and a single \( \text{MMPP}_2 \) approximation ........................................ 90
5.16 Comparison of dimensioning results from exact solution and approximation 10*\( \text{MMPP}_2 \) and a single \( \text{MMPP}_2 \) approximation ........................................ 91
5.17 Comparison of our results of the departure process (Laplace transform) of the IPP/\text{M}/1 with Daley’s results ........................................ 92
Chapter 1

Introduction and Motivation

1.1 Research motivation

In a randomly changing environment, such as time-varying arrivals, variable bandwidth demands, burst traffic integrated with streaming traffic and bursty and correlated arrivals, performance and dimensioning issues become very complicated. Little research has been carried out on dimensioning a system in such an environment to meet quality of service requirement. Our research focuses on analyzing the performance and optimize resource allocation of communication systems in a randomly changing environment.

Queueing theory has typically been used to address queueing problems of the communication networks under the assumption that the inter-arrival times and the service times are renewal processes, and the arrival rate and bandwidth requirement of each customer is constant. In today’s dynamically changing environment, such as packet streams in high-speed integrated-services wired and wireless networks and Web service, these assumptions are frequently unjustified.

We consider the cases where arrival rates of customers are time-varying, customer bandwidth requirement is variable, the arrival process may be correlated, and streaming and data traffic share the same bandwidth.

We first consider the dynamic analysis and design of a circuit-switched communication link where the arrivals are time-varying. The time-varying arrivals capture the
nature of dynamic traffic environment. Using transient analysis, a link is dimensioned under time-varying arrivals and transient load. We find the connection between the stationary and nonstationary analysis through transient analysis. We compare some numerical and closed-form methods reported in the literature and find efficient ways to design the system.

We next consider the case when the customer’s bandwidth requirements are time-varying. A multi-rate loss model with class-change is proposed and the blocking probability is calculated using an iterative recursive formula. The optimal dimensioning of the capacity of this system is obtained using a formula with a constant cost.

Subsequently, we consider a model to analyze the integrated streaming traffic and data traffic, where the data traffic is bursty with a time-varying bandwidth shared with the streaming traffic.

Finally, we analyze a statistical multiplexer where the arrival process is the superposition of \( n \) independent identical bursty and autocorrelated arrival processes. We construct the superposition of these arrival processes and characterize the departure process from the multiplexer by exact Laplace transform and a two-stage Markov-modulated Poisson process. These results permit us to analyze and dimension the system efficiently under study.

1.2 Literature review

Time-dependent (transient) analysis is motivated by the dynamic nature of the traffic. Riordan [47] introduced different methods for the transient analysis of a single service center. The necessary and sufficient conditions for a queueing network to have a transient product-form solution are provided by Taylor and Boucherie [53]. A new dimensioning approach for optical networks under nonstationary arrival rates was introduced by Nayak and Sivarajan [39]. Queueing models with nonstationary arrival rates have been studied extensively by Abdalla and Boucherie [1], Alnowibet [3], Alnowibet and Perros [4], Jagerman [20], Karagiannis et al. [22], Massey and Whitt [36], Massey [37], and Nayak and Sivarajan [39]. The nonstationary blocking probability for an Erlang loss queue was first obtained by Jagerman [20] using the modified offered load (MOL) approach. Massey and Whitt [36] developed analytical bounds on the error between the MOL approximation and the exact solution for an Erlang loss queue with a nonstationary arrival rate. A modified offered load
approximate product-form solution was introduced by Abdalla and Boucherie [1] for mobile networks. A survey for the nonstationary analysis of the Erlang loss queue can be found in Alnowibet and Perros [4]. Mandjes and Ridder [35] proposed large deviation solutions for the transient analysis of the Erlang loss model with a stationary arrival rate. Massey [37] analyzed different queues with time-varying arrival rate for telecommunication models. Nayak and Sivarajan [39] introduced a dynamic dimensioning approach for optical networks under nonstationary arrival rates. Karagiannis et al. [22] showed that the traffic of the internet backbone network can be characterized by a nonstationary Poisson process. Using transient analysis for the stationary arrivals and nonstationary arrivals, we find common approaches (numerical and closed-form) to analyze both cases and dimension the system, This research was reported in Tian and Perros [55] and an application of transient analysis for dimensioning of all-optical networks is reported in Tian [56].

The calculation of call blocking probabilities in circuit-switched networks has been extensively analyzed. However, this has been done under the assumption that the bandwidth allocated to a customer does not change throughout the customer’s service. For instance, Kaufman [21], Roberts [48], and Nilsson et al.[41] developed efficient algorithms for calculating the blocking probabilities of a multi-rate loss queue. In this case, customers belong to different classes and each class is associated with a class-dependent arrival rate, class dependent service rate, and a class-dependent bandwidth requirement expressed in number of servers. However, a class r customer cannot switch classes during its service time, and as a result, it cannot change the number of servers allocated to it. Call blocking probabilities over an entire circuit-switched network have been computed under a variety of assumptions, see for instance Kelly [24], Ross [49], Alnowibet and Perros [3], Washington and Perros [58], under assumptions similar to the above case of a single loss queue. We consider the multirate single loss queue depicting a circuit-switched communication link, this link may be an optical link, or a wired or wireless TDM link. Each server represents a time slot in a TDM link or a subrate stream in an optical link. Class r calls arrive in a Poisson fashion at the rate of $\lambda_r$, and require initially $b_r$ servers. During the service time of the call, the number of servers required may change. The call is not blocked if fewer servers than currently allocated to it are required. However, the call will get blocked if additional servers are required and these servers are not available at that instance. We describe an approximation algorithm for the calculation of the call blocking probability of each class. To the best of our knowledge this queueing system has not been analyzed before. We also use
a provisioning method based on Hampshire et al. [17] to determine the minimum number of required servers.

Both streaming traffic and data traffic may be supported in a wireless link. For example, The Universal Mobile Telecommunications System (UMTS) is a 3G cellular network that supports two broad categories of traffic: streaming traffic and data traffic. Streaming traffic corresponds to the real-time transfer of various signals (e.g., voice, streaming audio/video). This type of traffic is loss sensitive, and stringent transmission bandwidth guarantees are necessary to ensure real-time communication. Data traffic corresponds to the transfer of digital documents (e.g., web pages, emails, stored audio/videos), it is flexible, or “elastic” and bursty in today’s network environment, and the mean delay time (or response time) is a typical performance measure. Another application example of integrated streaming and elastic (data) traffic is GPRS (General Packet Radio Service) implemented in GSM (Global System for Mobile Communications) networks [13]. GPRS is fundamentally based on the hybrid switching principle with two types of traffic: voice calls and packet data. In accordance with current design and traffic management policies, voice calls have priority over data packets and data packets which cannot immediately be transmitted are queued at the source.

Various papers that study the integration of data and streaming traffic have been published recently (for example, Neuts [40], Bonald et al. [6], Borst et al. [7], Boxma et al. [8], Delcoigne et al. [10], Foh et al. [13], Lee et al. [31], and Numez et al. [42]). In terms of modeling approach, a fluid model is proposed in Bonald et al. [6], Borst et al. [7], Delcoigne et al. [10], Numez-Queija [42]) in order to provide closed-form limit results and approximations. These results may serve as performance bounds. Matrix-geometric solution in [40], [12], [13] and [8] is provided to obtain mean waiting time and other quantities for elastic traffic. Neuts [40] used the matrix-geometric approach to analyze the M/M/1 queue in a Markovian environment where the arrival and the service processes are governed by a Markov chain. Using this model of M/M/1 queue in a Markovian environment, the performance of integrated voice and data traffic can be studied. Borst et al. [7] studied a channel-aware scheduling in a wireless system supporting a combination of multi-class streaming and elastic traffic using time-scale decomposition.

In this study, we apply classical Markovian models for both traffic types assuming that streaming traffic has priority over data traffic. Using decomposition, we find an efficient closed-form solution to dimensioning the system. There is little research about dimensioning
the system to satisfy both quality of service (QoS) requirements for streaming traffic, such as loss probability, and data traffic, such as mean waiting time using classical Markov chain approach. The matrix-geometric approach is often used for performance evaluation of single class of streaming and data traffic sharing the same bandwidth. But it does not scale up well. The processing of streaming traffic and data traffic are normally very different in time scale. If we consider that the streaming traffic has priority over data traffic since the streaming traffic is loss sensitive but data traffic is not, then we can treat these two types of traffic as follows. We dimension the system to satisfy the QoS requirement (loss probability) of streaming traffic. In our work, for the streaming traffic, the single class Erlang loss model is used (the multirate loss Erlang model can be used as well). For the data traffic, we consider a bursty arrival process modeled by an interrupted Poisson process (IPP) which captures the dynamic nature of today’s data traffic.

The dimensioning of an ADSL access network is of importance given the traffic carried due to triple play. Specifically, of interest is to determine the size of the upstream and downstream links as a function of the number of ADSL modems supported by the access network. Alternatively, given the size of the upstream and downstream links, determine how many ADSL modems can be supported. The access network can be decomposed into a number of statistical multiplexer. A multiplexer consists of a queue served by a single server. The queue represents the buffer at an output port where all the IP packets wait to be transmitted out, and the server depicts the transmitter. The queue is fed by a large number of arrival streams. What makes this problem difficult to analyze is that the traffic generated by each customer is typically bursty in nature. The multiplexer problem was analyzed in the discrete-time case during the ATM days. Despite the vast literature on ATM multiplexer, this problem was never completely analyzed.

For approximating the superposition process of \( n \) independent and identical interrupted Poisson process (IPP), Whitt [59] introduced a general formula to find the c.d.f (cumulative density function) of the inter-arrival time of the superposed process using the stationary-interval method. Heffes et al. [18] proposed approaches to characterize the statistical multiplexer by considering the Markov-modulated Poisson process as a counting process. Gusella et al. [16] introduced different ways to fit data or a superposed process to an \( M\text{MPP}_2 \). Our approximation of \( n \) IPPs by an \( M\text{MPP}_2 \) is the combination of the above three approaches. As for the approximation of the departure process of an \( M\text{MPP}_2/\text{M}/1 \) queue, Daley [9] developed the Laplace transform of inter-departure times of GI/M/1 queue.
Heffes [18] found the Laplace transform of inter-departure times of GI/M/N queue. Bean et al. [5] claimed that whether the departure process of an MMPP/M/1 can be a MAP (or MMPP) is an open problem (see reference therein). They also conjectured that the departure process of a MAP/PH/1 queue is not a MAP unless the queue is a stationary M/M/1 queue. Yeh et al. [60] introduced a recursive algorithm based on the matrix-geometric solution to compute the moments of the inter-departure times of an MMPP/D/1 queue with complexity at least $O(n^{2.5})$. Heindl [19] proposed a numerical approach whereby the output process of an MMPP/G/1/(K) queue was approximated by a semi-Markov model which was then converted to an $MMPP_2$, this approach is computationally expensive. Lim et al. [33] gave a general framework for the calculation of the Laplace transform of the inter-departure times for a single server queue with Markov renewal input and general service time distribution (MAP/G/1), which is also based on the computationally expensive matrix-geometric solution. No closed-form solution was given. We characterize the departure process of an $MMPP_2/M/1$ queue by finding the explicit Laplace transform of the inter-departure times of the $MMPP_2/M/1$ queue. To the best of our knowledge, such a characterization has not been reported in the literature.

1.3 Thesis organization

The remainder of this dissertation is organized into six chapters. In Chapter 2, we compare five different approaches for the nonstationary Erlang loss model and describe some of the advantages and disadvantages of each method. In Chapter 3, we introduce a modeling and dimensioning approach for a circuit-switched link with variable-demand customers. In Chapter 4, we provide an efficient dimensioning approach of a circuit-switched link which carries integrated streaming and data traffic. In Chapter 5, performance analysis and dimensioning algorithms are proposed for a statistical multiplexer with bursty and correlated arrivals. We conclude our work and discuss future research work in Chapter 6.
Chapter 2

The Dynamic Analysis and Design of A Communication link with Stationary and Nonstationary Arrivals

Most research in queueing theory is typically based on the steady-state analysis. In today’s dynamically changing traffic environment, the steady-state analysis may not provide enough information to operators regarding Quality of Service (QoS) requirements and dynamic design. In addition, the steady state analysis is not practical for nonstationary arrivals. In this Chapter, we consider the time-dependent behavior of a communication link depicted by an Erlang loss queue with stationary and nonstationary arrival rates. The time-dependent analysis for stationary arrival rates captures the dynamic nature of the system during its transient phase. The time-dependent analysis for nonstationary arrivals is of great interest since the arrival rate in most communication systems varies over time.

We review and compare various methods that have been proposed in the literature
for the time-dependent analysis of the nonstationary Erlang loss system for both stationary arrivals and nonstationary arrivals. We classify five practical methods into two categories: (1) closed-form exact solution and closed-form approximation; (2) numerical exact solution and numerical approximation and compare their computation complexity and accuracy. We apply some of these techniques to dimensioning dynamically a single communication system.

2.1 Introduction

In this Chapter, we consider the time-dependent behavior and design of a communication link with stationary and also with nonstationary arrival rates. Time-dependent (transient) analysis is motivated by the dynamic nature of the traffic. Riordan [47] introduced different methods for the transient analysis of a single service center. The necessary and sufficient conditions for a queueing network to have a transient product-form solution are provided by Taylor and Boucherie [53]. A new dimensioning approach for optical networks under nonstationary arrival rates was introduced by Nayak and Sivarajan [39]. Queueing models with nonstationary arrival rates have been studied extensively by Abdalla and Boucherie [1], Alnowibet [3], Alnowibet and Perros [4], Jagerman [20], Karagiannis et al. [22], Massey and Whitt [36], Massey [37], and Nayak and Sivarajan [39]. The nonstationary blocking probability for an Erlang loss queue was first obtained by Jagerman [20] using the modified offered load (MOL) approach. Massey and Whitt [36] developed analytical bounds on the error between the MOL approximation and the exact solution for an Erlang loss queue with a nonstationary arrival rate. A modified offered load approximate product-form solution was introduced by Abdalla and Boucherie [1] for mobile networks. A survey for the nonstationary analysis of the Erlang loss queue can be found in Alnowibet and Perros [4]. Mandjes and Ridder [35] proposed large deviation solutions for the transient analysis of the Erlang loss model with a stationary arrival rate. Massey [37] analyzed different queues with time-varying arrival rate for telecommunication models. Nayak and Sivarajan [39] introduced a dynamic dimensioning approach for optical networks under nonstationary arrival rates. Karagiannis et al. [22] showed that the traffic of the internet backbone network can be characterized by a nonstationary Poisson process.

In this Chapter, we review and compare various techniques that have been reported
in the literature for the calculation of transient blocking probabilities of an Erlang loss queue assuming a stationary and nonstationary arrival rate. We also dimension a communication link, modelled by an Erlang loss queue for both stationary and nonstationary arrivals.

The Chapter is organized as follows. In section 1 we describe the behavior of an Erlang loss queue as a function of time assuming that the arrival rate is either constant or nonstationary, i.e. a function of time \( t \). We also show how an Erlang loss queue can be dimensioned using time-dependent blocking probabilities. In section 2.2, we review closed-form solutions of the transient behavior of an Erlang loss queue assuming constant and nonstationary arrival rates. Section 2.3 reviews an approximation method based on a property of truncated Markov processes, and section 2.4 describes a numerical procedure known as the fixed point approximation (FPA). An alternative approach to dimensioning a communication link, based on the method of large deviations, is presented in section 2.5. Numerical results are given in section 2.6. Finally the conclusions are summarized in section 2.7.

2.2 The nonstationary Erlang loss system

An Erlang loss queue is a system consisting of \( s \) servers and no waiting room. A customer is lost if it arrives at a time when all servers are busy. The loss queue is commonly used to model the telephone network. It has been extensively studied in the stationary case, i.e., assuming that the arrival process is a homogeneous Poisson process, or more generally, an Interrupted Poisson processes, and the service rate is exponentially distributed. (It has been shown that in a loss system, the blocking probability is insensitive to the service distribution but it only depends on its mean). The nonstationary loss queue, where the arrival rate is time-dependent is also of great interest.

![Figure 2.1: The Markov chain of Erlang loss model for a single service center](image)
The rate diagram of a loss queue with nonstationary arrivals \((M_t/M/s/s)\) is shown in FIG. 2.1, where \(s\) is the total number of servers, \(\lambda(t)\) is the time-dependent arrival rate and \(\mu\) is the service rate. We say that the arrival process is stationary if it is time-independent, i.e., \(\lambda(t) = \lambda\), and nonstationary if it is time-dependent (or time-varying). In this case \(\lambda(t)\) is a single continuous or discrete function of time. We discuss these two cases in the following two subsections.

### 2.2.1 Stationary Arrivals

Let us consider a loss queue \(M/M/s/s\) with a time-independent Poisson arrival rate \(\lambda\). Each arrival requests a service that requires an exponential amount of time with mean \(1/\mu\), and it is performed by a single server. The queue has \(s\) identical servers and there is no waiting room. The probability that there are \(n\), \(n=0, 1, \ldots, s\), customers in the queue at time \(t\), \(P_n(t)\), is given by the following set of forward differential equations:

\[
P'_0(t) = \mu P_1(t) - \lambda P_0(t) \tag{2.1}
\]

\[
P'_n(t) = \lambda P_{n-1}(t) + (n+1)\mu P_{n+1}(t) - (\lambda + n\mu)P_n(t), \tag{2.2}
\]

\[
P'_s(t) = \lambda P_{s-1}(t) - s\mu P_s(t) \tag{2.3}
\]

where \(P_0(t) + P_1(t) + P_2(t) + \ldots + P_s(t) = 1\), and \(0 \leq P_n(t) \leq 1\), for \(t \geq 0\) and \(n=0,1,2,\ldots,s\), with initial conditions: \(P_0(0)=1\) and \(P_n(0)=0\), \(n=1,2,3,\ldots,s\).

A numerical example of the time-dependent blocking probability is shown in FIG. 2.2 These probabilities were obtained by solving equations (2.1-2.3) using an ordinary differential equation (ODE) solver. We note that the blocking probability reaches steady-state when \(t = 6\). We also note that the steady-state probability can be very different from that during the transient state. Therefore, dimensioning the network based on the steady-state may result in overprovisioning during the transient period. This may have little impact if the duration of the transient period is very short. However, if the transient period is long, for example, a few months for optical networks, then it may be advantageous to dimension the network using the time-dependent blocking probability instead of the steady-state probability.
2.2.2 Time-varying arrivals

Let us consider a loss queue $M(t)/M/s/s$ with a time-dependent arrival rate $\lambda(t)$. Each arrival requests a service that requires an exponential amount of time with mean $1/\mu$. The probability that there are $n$, $n=0, 1, \ldots, s$, customers in the queue at time $t$, $P_n(t)$, is represented by the following forward differential equations:

\begin{align*}
P'_0(t) &= \mu P_1(t) - \lambda(t) P_0(t) \quad (2.4) \\
P'_n(t) &= \lambda(t) P_{n-1}(t) + (n + 1) \mu P_{n+1}(t) - (\lambda(t) + n\mu) P_n(t), \quad (2.5) \\
P'_s(t) &= \lambda(t) P_{s-1}(t) - s \mu P_s(t) \quad (2.6)
\end{align*}

where $P_0(t) + P_1(t) + P_2(t) + \ldots + P_s(t) = 1$, $t \geq 0$, and $0 \leq P_n(t) \leq 1$, for $t \geq 0$ and $n=0,1,2,\ldots,s$, with initial conditions: $P_0(0)=1$ and $P_n(0)=0$, $n=1, \ldots, s$.

In FIG.2.3, we show a numerical example of the time-dependent blocking probability for a single link obtained assuming a periodic arrival rate function $\lambda(t) = 180 + 50 \sin(2(t+20))$. These probabilities were calculated numerically by solving equations (2.4)-(2.6) using an ODE solver. We note that the blocking probability in this case also has a
Figure 2.3: The time-dependent blocking probabilities of the nonstationary arrival where \( \lambda(t) = 180 + 50\sin(2(t + 20)) \) and \( s=220 \)

transient period followed by repeating periods. The periodic behavior looks like the steady-state behavior of the stationary arrival case but the blocking probabilities follow a repeating pattern.

2.2.3 Dimensioning a single link

A link can be dimensioned using the time-dependent blocking probability for both stationary and nonstationary arrivals. This is done by calculating the number of servers \( s \) so that the blocking probability is under a given threshold at any time \( t \).

Stationary arrivals

Let us consider an Erlang loss queue with \( \lambda = 10 \). The number of servers required so that the blocking probability is less than 0.01 is given in FIG. 2.4. The solid line labelled ‘steady-state dimensioning’ gives the optimum number of servers calculated using the steady-state blocking probability of an Erlang loss model with \( \lambda = 10 \). The dotted line
labelled ‘time-dependent dimensioning’ gives the result using the time-dependent blocking probability of the arrival rate, obtained using differential equations (2.1) to (2.3). We calculate the number of servers iteratively until the blocking probability is less than 0.01. Note that these two curves are the same after the transient phase is over. As we can see, the dimensioning results are quite different for these two scenarios with the time-dependent dimensioning requiring fewer servers.

![Figure 2.4: A dimensioning example of a single link with stationary arrival rates](image)

**Time-varying arrivals**

We consider an example where the arrival rate varies as shown in FIG. 2.4. We assume that time is divided into 12 periods, where each period for instance can be a month. During each period $i$, the arrival rate is constant. In FIG. 2.4, the 12 arrival rates are: $\lambda(t)=[8, 1, 3, 6, 2, 5, 12, 9, 11, 4, 7, 10]$. We dimension the link so that at any time, the blocking probability is less than 0.01. The dimensioning results are also shown in FIG. 2.4, and they were obtained assuming that all servers are free at time $t = 0$. These results were obtained as follows: we first calculate the number of servers for the first period using
equations (2.4)-(2.6), assuming an empty system at time $t = 0$ and service rate $\mu = 1$, so that the nonstationary blocking probability is less than 0.01. This is done as before, in an iterative manner. We repeat this process for the second period assuming that at the beginning of the period the number of customers in it is equal to the average number of customers in the system.

This process is repeated until all 12 periods have been analyzed. We note that we solve this problem by breaking it into periods and analyzing each period separately. Alternatively, we could solve equations (2.4)-(2.6) for the entire 12 period arrival process. In this case, we will not have to approximate the initial condition for each period. However, it is difficult to define $\lambda(t)$ as a single function over the entire 12 periods.

We note that the dimensioning results are very sensitive to the initial conditions and the arrival rates, and the required number of servers follows the arrival process.

Figure 2.5: A dimensioning example of a single link with nonstationary arrival rates
2.3 Exact closed-form solutions for the time-dependent blocking probability

In this section, we review closed-form solutions for the transient analysis of the Erlang loss queue under a constant and nonstationary arrivals.

2.3.1 Stationary Arrivals

In this case, the time-dependent blocking probability can be obtained using equations (2.1-2.3). We have \( P'_n(t) \to 0 \) for all \( n \) as \( t \to \infty \) in the differential equations (2.1-2.3). The differential equations (2.1-2.3) reduce to a set of linear equations from which we can obtain the closed-form solution for the probability \( P_n \) that there are \( n \) customers in the system.

\[
P_n = \lim_{t \to \infty} P\{q(t) = n\} = \frac{\rho^n/n!}{\sum_{i=0}^{s} \rho^i/i!}, n = 0, 1, \ldots, s
\] (2.7)

where \( \rho = \lambda/\mu \), and \( q(t) \) is the number of customers in the system at time \( t \). The probability of blocking \( B_p \) is:

\[
B_p = \lim_{t \to \infty} P\{q(t) = s\} = \frac{\rho^s/s!}{\sum_{i=0}^{s} \rho^i/i!}
\] (2.8)

This is the well-known Erlang B formula. The average number of customers in the system (i.e. the average number of busy servers) is: \( \lim_{t \to \infty} E[q(t)] = E[q] = (1 - B_p)\rho \).

The time-dependent blocking probability function at time \( t \), can be obtained from the differential equations (2.1-2.3). We have:

\[
P_s(t) = \beta e^{(Q^T)t}\alpha
\] (2.9)

where \( \alpha \) is the initial state probability vector, \( \beta \) is an all-zero row vector except that the last entry is 1, and \( Q \) is the infinitesimal generator matrix of the underlying Markov chain,
defined as:

\[
\begin{pmatrix}
-\lambda & \lambda & 0 & \cdots & 0 \\
\mu & -\mu - \lambda & \lambda & \cdots & 0 \\
0 & 2\mu & -2\mu - \lambda & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \lambda & 0 \\
0 & \cdots & (s-1)\mu & -(s-1)\mu - \lambda & \lambda \\
0 & \cdots & 0 & s\mu & -s\mu \\
\end{pmatrix}
\] (2.10)

The computation complexity of equation (2.3) is \(O(s^3)\) using a scaling and squaring algorithm with a Pade approximation [38]. The solution to the differential equations (2.1)- (2.3) can be found using different methods. An elegant closed-form solution can be obtained using Sylvester matrix theorem. Let us assume that we start from an empty system and let \(Q^T\) be the transpose of \(Q\), \(P(t) = [P_0(t), ..., P_s(t)]^T\), and \(P(0) = [1, 0, ..., 0]^T\). Now the system of differential equations (2.1-2.3) can be written in the following form:

\[
P'(t) = Q^T P(t),
\] (2.11)

Solving this equation by applying the property of exponential function, we get

\[
P(t) = e^{(Q^T)t} P(0),
\] (2.12)

Let \(D = e^{Q^Tt} = [d_0, d_1, ..., d_s]\) where \(d_i\) is the column vector of \(D\). Then we have \(P(t) = d_0\) and \(P_i(t) = d_{0,i}\).

**Theorem 1**: All the \((s+1)\) eigenvalues of \(Q\) are real and distinct and one eigenvalue of \(Q\) is zero.

Proof: We first change \(Q\) to a symmetric tridiagonal matrix \(A\) with positive sub-diagonal elements by the similarity transform (this does not change the eigenvalues of \(Q\)). We have: \(A = D^{-1}QD\), where \(D\) is a diagonal matrix. Let us set \(A^*\) as the conjugate transpose of matrix \(A\). Then we know that \(A^* = A\).

We now show that all eigenvalues of \(A\) are real as follow: let \((\lambda, x)\) be the (eigenvalue, eigenvector) pair of matrix \(A\).

\[
Ax = \lambda x,
\] (2.13)

Multiplying both sides of equation (2.13) by \(x^*\), we have

\[
x^*Ax = \lambda x^*x,
\] (2.14)
Taking the transpose of equation (2.14) and noticing that $A^* = A$, we get:

$$x^*Ax = \lambda^*x^*x,$$

(2.15)

where $\lambda^*$ is the conjugate transpose of $\lambda$. Comparing equation (2.14) and (2.15), we see that $\lambda = \lambda^*$, which means that all eigenvalues of $A$ are real.

$A$ is a tridiagonal matrix, so we can compute the characteristic polynomial with a three-term recurrence (just do a determinant expansion) to construct a Sturm sequence. Since off-diagonal elements are positive, the matrix can only have simple eigenvalues (by the property of Sturm sequence). Since $\text{det}(Q) = 0$, then $\text{det}(Q-0^*I) = 0$, which means that zero is one of the eigenvalues of matrix $Q$. This can also be seen by the fact that row sum of matrix $Q$ is zero.

**Theorem 2:** Sylvester’s matrix theorem for distinct roots (eigenvalues) (see Frazer [5]):

If $G(U)$ is any polynomial of the square matrix $U$, and if $x_i$ represents one of the $n$ distinct eigenvalues of $U$, then

$$G(U) = \sum_{i=1}^{n} \frac{G(x_i)\text{Adj}(x_iI - U)}{\prod_{j\neq i}(x_j - x_i)}$$

(2.16)

An important application of Sylverster’s theorem is in finding a closed-form solution for the matrix exponential. The following explains how to use this theorem to get the closed-form solution to our $M/M/s/s$ transient analysis problem. Let $x_0, x_1, ..., x_s$ be the $(s + 1)$ eigenvalues of $Q$. From Sylvester’s theorem we have

$$D = e^{Q^Tt} = \sum_{r=0}^{s} e^{x_r t} \frac{\text{Adj}(x_r I - Q^T)}{\prod_{i\neq r}(x_r - x_i)}$$

(2.17)

where $\text{Adj}(U)$ is the Adjoint matrix of $U$. Especially, the probability that all servers are busy at time $t$ can be further simplified as:

$$P_s(t) = \sum_{r=0}^{s} e^{x_r t} \frac{(-1)^{s+1}\text{det}(M)}{\prod_{i\neq r}(x_r - x_i)}$$

(2.18)
where $\text{det}(M) = \lambda^s$, $\lambda$ is the average arrival rate and $M$ is the submatrix of $Q^T$ with size $s \times s$. We can obtain all the eigenvalues of matrix $Q$ using the fast algorithm with complexity $O(s^2)$, reported in [11]. So the computation complexity of equation (2.12) is roughly $O(s^3)$. From $P_s(t)$ we can also know the steady-state probability. The steady-state probability is just the constant part (which corresponds to the zero eigenvalue of matrix $Q$) of $P_s(t)$. Other quantities of interest such as $P_n(t)$ and average number of busy servers at time $t$ can also be calculated.

### 2.3.2 Nonstationary arrivals

The closed-form solution to differential equations (2.4-2.6) is complex even for fairly small systems with special arrival rate function $\lambda(t)$, see Alnowibet [3]. An explicit solution is provided in Jagerman [20] by using the probability generating functions of the state probabilities and the corresponding binomial moments where the arrival rate function $\lambda(t)$ is considered to be continuous. Following Jagerman [20], we have that the probability of $j$ calls arriving in the time interval $(0, t)$ is given by

$$
\left[ \int_0^t a(u) du \right]^j \frac{j!}{j!} \exp\left( - \int_0^t a(u) du \right)
$$

(2.19)

where $a(t)$ is Poisson-offered load given by $a(t) = \lambda(t)/\mu$. We normalize the service rate $\mu = 1$, so that $a(t)$ is measured in Erlangs. Let us define the Volterra operator $K_r$

$$
K_r f = \int_0^t K_r(t, \tau) f(\tau) d\tau, r = 0, 1, ..., N.
$$

(2.20)

The time-dependent blocking probability that all servers are busy at time $t$ is

$$
P_s(t) = \frac{\gamma(t, 0)}{s!} - K_s(t, \tau) P_s(t)
$$

(2.21)

hence the explicit form of the solution is:

$$
P_s(t) = \frac{\Lambda(t)^s}{s!} - K_s \frac{\Lambda(t)^s}{s!} + K_s^2 \frac{\Lambda(t)^s}{s!} - ..., \quad \text{(2.22)}
$$

where

$$
\Lambda(t) = e^{-t} \int_0^t e^{u} a(u) du
$$

(2.23)
and $K_s$ is a Volterra operator defined by the kernel

$$K_s(t, \tau) = \sum_{j=0}^{s-1} \frac{\gamma(t, 0)^j}{j!} e^{-(s-j)(t-\tau)} \left( \frac{N}{s-j-1} \right) a(\tau)$$

(2.24)

where

$$\gamma(t, \tau) = e^{-t} \int_\tau^t e^u a(u) du$$

(2.25)

Note that $\Lambda(t) = \gamma(t, 0)$. We see that the above explicit solution is quite complicated for an arbitrary arrival rate $\lambda(t)$. The computation complexity of equation (2.22) is approximately $O(s^3)$ depending on how many terms used in the series. In view of this, it is not useful in practice.

### 2.4 The truncated Markov process approximation

The following Corollary holds for truncated reversible Markov process (see Kelly [23]).

**Corollary 1:** If a reversible Markov process $X_t$ with state space $S$ and equilibrium distribution $\Pi(j), j \in S$, is truncated to the set of $S_1 \subset S$, then the resulting Markov process $Y_t$ is reversible in equilibrium and has the equilibrium distribution:

$$\Pi_1(j) = \frac{\Pi(j)}{\sum_{k \in S_1} \Pi(k)}, j \in S_1.$$  

(2.26)

It is interesting to note that the equilibrium distribution of the truncated process is just the conditional (renormalized within the truncated state space) probability of the original process. An efficient way to obtain the stationary distribution of the M/M/s/s queue is to use the fact that the M/M/s/s queue is a truncated process of an $M/M/\infty$, which is a reversible Markov process. Therefore:

$$P\{q = n\} = P\{q_{\infty} = n|q_{\infty} < s\} = \frac{\rho^n/n!}{\sum_{i=0}^{s-1} \rho^i/i!}$$  

(2.27)

For the time-dependent analysis of M/M/s/s queue, we also can apply this truncation property approximately. First let us consider the transient behavior of the $M/M/\infty$ queue Riordan [14]. We have the following differential equations:

$$P_n'(t) = - (\lambda + n\mu) P_n(t) + \lambda P_{n-1}(t) + (n+1)\mu P_{n+1}(t),$$

(2.28)
\[
P_0'(t) = -\lambda P_0(t) + \mu P_1(t), \quad (2.29)
\]

where \( \sum_{i=0}^{\infty} P_i(t) = 1, t \geq 0 \) \( (2.30) \)

and

\[0 \leq P_n(t) \leq 1, \text{for } t \geq 0 \text{ and } n = 0, 1, 2, ..., \infty, \quad (2.31)\]

with initial conditions: \( P_0(0)=1 \) and \( P_n(0)=0, n=1, 2, 3, ..., \infty \). Applying the z-transform approach, we can obtain the transient distribution of \( M/M/\infty \) Riordan \[14\] :

\[
P_n^\infty(t) = \frac{m^n e^{-m}}{n!}, m \equiv m(t) = \rho(1 - e^{-\mu t}). \quad (2.32)
\]

For the time-dependent analysis of an M/M/s/s queue with constant arrival rate, we can apply the truncation property approximately as follows:

\[
P_n^s(t) \approx \frac{P_n^\infty(t)}{\sum_k P_k^\infty(t)}, k \in \{0, 1, ..., s\} \quad (2.33)
\]

Notice that this equation will be close to the exact solution as time increases for small blocking probabilities. Also it is a good approximation for low blocking probabilities. Despite its appealing closed-form solution, equation (2.33) is non-trivial to compute for large value of \( s \) since it involves factorial terms which may cause numerical instability problems such as overflows. So, we can adapt the well-known recursive formula from the steady-state as follows:

\[
B(k + 1, t) = \frac{m(t)B(k, t)}{k + 1 + m(t)B(k, t)} \quad (2.34)
\]

where \( B(s, t) \) is the probability that all \( s \) servers are busy at time \( t \) and \( B(0, t) = 1 \).

The M(t)/M/s/s can also be approximated by truncating the M(t)/M/\infty queue. This method is referred to as the modified offered load (MOL) method and it was first proposed by Jagerman \[20\]. For the M(t)/M/\infty queue, the time-dependent blocking probability is given by:

\[
P_n^\infty(t) = \rho(t)^n e^{-\rho(t)}/n! \quad (2.35)
\]

where \( \rho(t) = e^{-t} \int_0^t \lambda(u)e^udu \). For the M(t)/M/s/s queue, the probability \( P_n(t) \) that there are \( n \) customers in the system is:

\[
P_n^\infty(t) \approx P\{q_\infty(t) = n|q_\infty(t) < s\} = \frac{\rho(t)^n/n!}{\sum_{i=0}^{s} \rho(t)^i/i!} \quad (2.36)
\]
In this case, the following recursive formula can be used:

\[
B(k + 1, t) = \frac{\rho(t)B(k, t)}{k + 1 + \rho(t)B(k, t)}
\]

with \( B(0, t)=1 \) as the initial condition. \( B(s, t) \) is the probability that all \( s \) servers are busy at time \( t \), \( P_s(t) \). The truncated \( M/M/\infty \) provides an exact solution to an \( M/M/s/s \) queue in the steady-state due to the reversibility property. However, this property is lost in the nonstationary case [3]. Hence the truncated \( M(t)/M/\infty \) provides an approximate solution to the \( M(t)/M/s/s \). Massy and Whitt [36] developed analytical bounds on the error between the MOL approximation and the exact solution of the \( M(t)/M/s/s \) queue. The computation complexity of equation (2.36) and (2.39) is \( O(s) \).

Experiments showed that the actual blocking probability of the \( M(t)/M/s/s \) queue should be less than 0.1 in order for the MOL to provide a good approximation, see Alnowibet and Perros [3]. As expected, the MOL underestimates the blocking probability of a loss queue with high load, i.e. when the exact blocking probability is high.

2.5 Numerical solutions

2.5.1 Differential equations solver

Equations (2.1-2.3) and (2.4-2.6) can be solved numerically using an ODE (ordinary differential equation) solver. The numerical results in section 2.2 were obtained using the ODE solver of Matlab 6.5, which can solve efficiently an Erlang loss queue with a few hundreds servers. However, as reported in Moler et al. [38], ODE solver may be very expensive. We also found that ODE solver is not suitable for dimensioning.

Dimensioning of transient Erlang loss model using square-root-safety formula should be possible and simplified.

2.5.2 The fixed point approximation (FPA) method

The fixed point approximation (FPA) method was proposed by Alnowibet and Perros [4]. This method calculates numerically the time-dependent mean number of customers and blocking probability functions in a nonstationary loss queue. The FPA method
was also extended to nonstationary queueing networks of multi-rate loss queues and nonstationary queueing networks with population constraints, see Alnowibet and Perros [4]. The main idea of the FPA method is as follows:

Given a loss queue $M(t)/M/s/s$ with time-dependent rate $\lambda(t)$, the time-dependent average number of customers $E[Q(t)]$ can be expressed as the difference between the effective arrival rate and the departure rate at time $t$. That is:

$$E'[Q(t)] = \lambda(t)(1 - B_p(t)) - \mu E[Q(t)] \quad (2.38)$$

We note that the time-dependent mean number of customers is given by the expression:

$$E[Q(t)] = \rho(t)(1 - B_p(t)) \quad (2.39)$$

where

$$B_p(t) = \frac{\rho(t)^s/s!}{\sum_{i=0}^{s} \rho(t)^i/i!} \quad (2.40)$$

Using equations (2.38)-(2.40), we can calculate the blocking probability iteratively as follows.

1). Choose an appropriate $\Delta t$, final time $T_f$ and tolerance $\epsilon$.
2). Choose initial conditions for $E[Q(t)]$. Set $E[Q(0)] = 0$.
3). Evaluate $\lambda(t)$ at $t=0, \Delta t, 2\Delta t, \ldots, T_f$.
4). Start with an initial blocking probability $B_p^0(t) = 0$, $t=0, \Delta t, 2\Delta t, \ldots, T_f$.
5). Set the iteration counter $k=0$.
6). Solve numerically for $E[Q^k(t)]$ using the following equation:

$$E[Q^k(t + \Delta t)] = E[Q^k(t)] + \lambda(t)(1 - B_p^k(t))\Delta t - \mu E[Q^k(t)]\Delta t.$$  

7). Calculate $\rho^k(t) = E[Q^k(t)]/(1 - B_p^k(t)), t=0, \Delta t, 2\Delta t, \ldots, T_f$.
8). Calculate the blocking probability $B_p^{k+1}(t) = \rho^k(t)^s/s!/(\sum_{i=0}^{s} \rho^k(t)^i/i!), t=0, \Delta t, 2\Delta t, \ldots, T_f$.
9). If $\|B_p^k(t) - B_p^{k+1}(t)\| < \epsilon$, then $B_p^k(t)$ has converged and the algorithm stops. Else, set $k = k + 1$, and go to step 6).

The FPA algorithm does not require a closed-form expression for the arrival rate function. It only requires that the arrival rate function be defined at time points equally spaced by $\Delta t$. In view of this, any arrival rate function can be used despite whether we know its closed-form or not. Since this algorithm discretizes the arrival rate function, the
continuity and differentiability properties of the arrival rate function are not necessary. The computation complexity of this algorithm to find blocking probability is $O(sT_f/\Delta t)$. In all the experiments the FPA results were very close to the exact numerical results or within the simulation confidence intervals. This leads to the conjecture that the blocking probability obtained by FPA is exact (see, Alnowibet and Perros [4]). For dimensioning purpose, we need to slightly modify the algorithm in order to obtain the capacity for any time $t$ given a blocking probability threshold. This can be done by adding an iterative procedure into the main algorithm.

2.6 The large deviation approach

In this section, we obtain an expression for dimensioning the Erlang loss queue using the large deviation method. For stationary arrivals, Mandjes and Ridder [35] have obtained approximate expressions for $P_n(t)$, the probability of having $n$ customers at time $t$. This expression is extended in the case of nonstationary arrivals. The large deviation theory is similar to the Central Limit theory (CLT). The CLT governs random fluctuations only near the mean, which are of the order of $\delta/\sqrt{n}$, where $\delta$ is the standard deviation. Fluctuations which are of the order of $\delta$ are, relative to typical fluctuations, much bigger: they are large deviations from the mean. They happen only rarely, and so the large deviation theory is often described as the theory of rare events, that is, events which take place away from the mean, out in the tails of the distribution. The main idea of the large deviation approach for the nonstationary Erlang loss queue is as follows. An asymptotic regime is obtained by scaling the arrival process. This is done by replacing $\lambda(t)$ with $n\lambda(t)$. The number of sources active at time $t$ are partitioned into the sources that were active at time 0 and are still active at time $t$, and the sources that became active in $(0, t)$ and are still active at time $t$. We then can apply Cramer’s theorem and Chernoff’s formula to obtain the result.

2.6.1 Stationary arrivals

Assuming exponential service time distribution, Mandjes and Ridder [35] obtained the following expression:
\[ P_s(t) \approx e^{s><ln(\gamma(t))>\gamma(t)+1}, \gamma(t) = \lambda_1/\mu(1 - e^{-t}) \]  

(2.41)

where \( \lambda = s\lambda_1 \) and \( s \) is the total number of servers.

\( P_s(t) \) can be better approximated using Bahadur-Rao approximation [35]. We have:

\[ P_s(t) \approx \frac{1}{\sqrt{2\pi s\delta\theta}} e^{s><ln(\gamma(t))>\gamma(t)+1}, \gamma(t) = \lambda_1/\mu(1 - e^{-t}), \]  

(2.42)

where \( \theta = -\log(\gamma(t)) \), \( s\delta^2 = \frac{M''(\theta)}{M(\theta)} \) and \( M(\theta) = e^{\gamma(t)(e^\theta-1)} \).

Mandjes and Ridder [35], pointed out that the Bahadur-Rao approximation (2.42) is more accurate than (2.6.2). We note that the large deviation theory yields simple expressions of the time-dependent blocking probability for stationary arrivals.

### 2.6.2 Nonstationary arrivals

Let us assume that the service time is exponentially distributed with unit mean for nonstationary arrivals. Then expression can be extended as follows:

\[ P_s(t) \approx e^{s><ln(\gamma(t))>\gamma(t)+1} \]  

(2.43)

where \( \gamma(t) = e^{-t} \int_0^t \lambda_1(u)e^udu, \lambda(t) = s\lambda_1(t). \)

The Bahadur-Rao approximation given by (2.43) can be extended as follows:

\[ P_s(t) \approx \frac{1}{\sqrt{2\pi s\delta\theta}} e^{s><ln(\gamma(t))>\gamma(t)+1}, \]  

(2.44)

where \( \gamma(t) = e^{-t} \int_0^t \lambda_1(u)e^udu, \theta = -\log(\gamma(t)) \), \( s\delta^2 = \frac{M''(\theta)}{M(\theta)} \) and \( M(\theta) = e^{\gamma(t)(e^\theta-1)} \). The computation complexity of equation (2.43) and (2.44) is \( O(1) \).

For both stationary and nonstationary cases, the number of servers \( C \) for which the blocking probability is \( \epsilon \) can be obtained using an iterative procedure: starting by the candidate allocation \( C = n_0 \), the candidate allocation is increased by one until the blocking probability is below the threshold \( \epsilon \).

### 2.7 More numerical results

In FIG. 2.6, we give the blocking probability calculated at a specific time \( t = 4.8 \) using four different methods for \( \lambda_1(t) = 0.7 + 0.2\sin(2\pi t) \). The time \( t=4.8 \) was chosen
because by that time the system is out of the transient state for the given periodic arrival rate function. The initial condition was set to an empty system. The blocking probability is plotted in the logarithmic scale against the total number of servers $s$. The graph labeled “LD” shows the results obtained from the large deviation theory using equation (2.43), the graph labeled “BR” gives the results obtained using the Bahadur-Rao equation 2.44, and, the graph labeled “TR” gives the results obtained using equation (2.37) from the truncated Markov process approximation. The exact solution is obtained using the fixed point approximation (FPA).

Running more examples by varying the arrival rate with (high load, medium load, low load), we note that the truncated Markov process approximation provides a very good approximation to the exact solution but underestimates the blocking probabilities. The large deviations approximation differs considerably from the exact blocking probability and is less accurate than the Bahadur-Rao approximation. Because of page limit, we do not provide all the examples here.

In FIG. 2.6, we show a dimensioning example for a communication link over 20 observation periods. We assume that the arrival rate is constant during each period. The values of the arrival rates are given in FIG. 2.6. We calculated the capacity, i.e., the number of servers, iteratively so that at any time the blocking probability is less than 0.005. The dimensioning results are shown in FIG. 2.6, where we assume that all servers are free at time $t=0$. ‘BR’ represents the results obtained using the Bahadur-Rao equation 2.44, and ‘FPA’ gives the results obtained using the fixed point approximation (see, Alnowibet and Perros [4]). The results obtained using the other two methods, i.e. the truncated Markov process approximation and equation 2.44 from the large deviation theory, are not shown because they have a large error. We also carried out a variety of numerical results under different loads, and here we only show a representative sample of these results, since they are all similar. We observed that Bahadur-Rao (BR) approximation is very close to the exact results but it overestimates the capacity. We note that both BR and exact allocated capacity follow the pattern of the arrival rates.
2.8 Conclusions

In this Chapter, we reviewed and compared various time-dependent analysis techniques of a single loss queue with stationary and nonstationary arrivals. The aim of time-dependent analysis is to dimensioning a link in a more efficient way.

It is difficult to answer the question “Which method is the best?”. One method maybe preferable over the others when considering computation complexity and accuracy of the results. We have the following observations:

1) For stationary arrivals, the exact closed-form solution and the truncated Markov process approximation are CPU efficient and easy to implement for medium size systems whereas the large deviation approach is preferred if the system is large.

2) For the nonstationary arrivals:

If the arrival rate is a single continuous function, then the truncated Markov approaches...
process approximation (MOL) and FPA method will be a good choice for medium size systems, and the large deviation approach is a better choice if the system is large.

If the arrival rate is not a continuous function, the FPA method is a better choice. The FPA method can work for both stationary arrivals and nonstationary arrivals and it can be used for medium size systems. For large systems, we may consider using the Bahadur-Rao approximation for dimensioning purposes for both stationary and nonstationary cases since it is the fastest and provides a tight upper bound.

The size of a medium and a large system is relative to the computer used. Our results were obtained on a Pentum(R)4 PC with a 3GHz CPU and a RAM of 504MB. In this context, a medium size system has less than two hundred servers, and a large size system has more than two hundred servers.

Figure 2.7: A dimensioning example of a single link with nonstationary arrival rates
Chapter 3

Analysis And Provisioning of a Circuit-switched Link with Variable-Demand Customers

We consider a single circuit-switched communication link, depicted by a Erlang multi-class loss queue, where a customer may vary its required bandwidth during its service. We obtain approximately the steady-state blocking probability of each class of customer. Comparisons with simulation results show that the approximation solution has a good accuracy. For the proposed model, we also provide an efficient capacity provisioning algorithm.

3.1 Introduction

In circuit-switched communication systems and connection-oriented packet switched networks, a connection is typically allocated a fixed bandwidth which does not vary over the life of the connection. However, in today’s dynamically changing communication networks, the bandwidth allocated to a connection may have to vary in order to accommodate load fluctuations. In this work, we consider the case where the customer’s bandwidth require-
ments change during its service time. This case has been motivated by the Link Capacity Adjustment Scheme (LCAS) in the Data over SONET (Dos) architecture.

Traditional SONET/SDH was optimized to carry voice traffic. It was also defined to carry ATM traffic and IP packets (PoS). Changes in the capacity allocated to a connection are done manually. Recently, a novel architecture has been proposed, referred to as data over SONET/SDH (DoS) which provides a mechanism for the efficient transport of integrated data services.

It utilizes three schemes, namely, the Generic Frame Procedure (GFP), Virtual Concatenation (VCAT), and Link Capacity Adjustment Scheme (LCAS) [30]. GFP is a simple adaptation scheme that extends the ability of SONET/SDH to carrying different types of traffic. Specifically, it permits the transport of frame-oriented traffic, such as Ethernet and IP over PPP. It also permits continuous-bit-rate block-coded data from Storage Area Networks (SAN) transported by networks, such as Fiber Channel, Fiber Connection (FICON), and Enterprise System Connect (ECON).

Virtual concatenation maps an incoming traffic stream into a number of individual subrate payloads. The subrate payloads are switched through the SONET/SDH network independently of each other (see for example, Perros [46]).

Virtual concatenation is only required to be implemented at the originating node where the incoming traffic is demultiplexed into subrate payloads and at the terminating node, where the payloads are multiplexed back to the original stream. The individual payloads might not necessarily be contiguous within the same OC-N payload. Finally, the number of subrate payloads allocated to an application is typically determined in advance. However, the transmission rate of the application may vary over time. In view of this, it can be useful to dynamically vary the number of subrate payloads allocated to an application. This can be done using the link capacity adjustment scheme (LCAS). In LCAS, signaling messages are exchanged between the originating and terminating SONET/SDH node to determine and adjust the number of required subrate payloads. LCAS makes sure that the adjustment process is done without losing any data.

The calculation of call blocking probabilities in circuit-switched networks has been extensively analyzed. However, this has been done under the assumption that the bandwidth allocated to a customer does not change throughout the customer’s service. For instance, Kaufman [21], Roberts [48], and Nilsson et al. [41] developed efficient algorithms for calculating the blocking probabilities of a multi-rate loss queue. In this case, customers
belong to different classes and each class is associated with a class-dependent arrival rate, class dependent service rate, and a class-dependent bandwidth requirement expressed in number of servers. However, a class r customer cannot switch classes during its service time, and as a result, it cannot change the number of servers allocated to it. Call blocking probabilities over an entire circuit-switched network have been computed under a variety of assumptions, see for instance Kelly [24], Ross [49], Alnowibet and Perros [3], Washington and Perros [58], under assumptions similar to the above case of a single loss queue.

In this work, we consider the multirate single loss queue depicting a circuit-switched communication link, this link may be an optical link, or a wired or wireless TDM link. Each server represents a time slot in a TDM link or a subrate stream in an optical link. Class r calls arrive in a Poisson fashion at the rate of $\lambda_r$, and require initially $b_r$ servers. During the service time of the call, the number of servers required may change. The call is not blocked if fewer servers than currently allocated to it are required. However, the call will get blocked if additional servers are required and these servers are not available at that instance. We describe an approximation algorithm for the calculation of the call blocking probability of each class. To the best of our knowledge this queueing system has not been analyzed before.

We also use a provisioning method based on Hampshire et al. [17] to determine the minimum number of required servers.

This work is organized as follows. In section 3.2, we describe in detail the multi-class loss queue under study and how it can be used to model various cases where a customer may change its bandwidth requirements during its service. In section 3.3, we describe the approximation algorithm and in section 3.4, we describe how to calculate the minimum number of servers of the loss system so that the blocking probability of any class is less than a pre-specified value. Numerical examples are given in section 3.5, and finally the conclusions are given in section 3.6.
3.2 The multi-class loss queue with variable-demand customers

Let us consider a multi-class loss queue. There are R classes of traffic, and class $i$ customers ($i=1,2,...,R$) arrive at the loss queue in a Poisson fashion with a class-dependent arrival rate $\lambda_i$ requiring $b_i$ servers. Class $i$ customers receive an exponentially distributed service time with mean $1/\mu_i$. The required number of servers is ordered for convenience, that is, $0 < b_1 \leq b_2 \ldots \leq b_R$. Upon arrival at the loss queue, a customer is blocked if the required number of servers is not available. After an exponentially distributed service with rate $\mu_i$, a class $i$ customer may depart from the system with probability $p_{i0}$ or it may change its class to class $k$ with probability $p_{ik}$ where $\sum p_{ij} = 1$. A class change implies that the customer’s bandwidth requirements change from $b_i$ to $b_k$. If $b_k < b_i$, then the class change is successful and the $b_i - b_k$ remaining unused servers join in the pool of available servers. However, if $b_k > b_i$, then $b_k - b_i$ additional servers are required. The customer is blocked (i.e. lost) if these $b_k - b_i$ additional servers are not all available at that moment. Define $P_1(i,j)=p_{ij}$ which is a matrix with size $R \times (R + 1)$. Let $P$ be submatrix of $P_1$ and $P$ has dimension $R \times R$. We have to assure that $(I - P)$ is invertible so that a customer entering the system eventually exists.

As will be seen below, the analysis of this system permits a large number of servers which allows us to model high-bandwidth circuit-switched links. For instance, an OC-768 link will be modeled by a loss queue with 768 servers where a server represents an OC-1 subrate stream. The analysis of this model also permits a large number of classes. This feature gives the model the required flexibility to depict users with a given bandwidth profile. For instance, let us consider an example where one group of users requires initially 10 servers. This bandwidth requirement is changed to 20 servers and after that to 15 servers. Then, we say that the bandwidth profile of this group of users is: $\{10, 20, 15\}$. This will be modeled using three classes, say 1, 2, and 3, as follows. A customers with this profile arrives at the loss queue with an arrival rate $\lambda_1$ (arrival rates of $\lambda_2$ and $\lambda_3$ are equal to zero) requiring $b_1 = 10$ servers. After an exponential service time with a mean of $1/\mu_1$, a customer changes to class 2, thus requiring $b_2 = 20$ servers with probability $p_{12}=1$. Following an exponential service time with a mean of of $1/\mu_2$, the customer changes to class 3 with probability $p_{23}=1$. Finally, after an exponential service time with a mean of $1/\mu_3$, ...
the customer departs, i.e. $p_{30}=1$. A customer changing from class 1 to 2, may get blocked if the additional 5 servers are not available. However, a customer going from class 2 to 3 will never get blocked since it requires fewer servers than those it held. More complex bandwidth profiles can be constructed by selecting a set of unused classes, and associating each class $i$ with a set of values for $1/\mu_i$, $b_i$, $p_{ij}$. Each set of classes associated with a specific bandwidth profile can be seen as forming a closed super-class within which class changes are allowed in a pre-specified manner. The case where a customer can change bandwidth requirements in a random manner can be readily accommodated.

### 3.3 Calculation of call blocking probabilities

For the classical multi-class loss system without bandwidth adjustments, there are well-known results.

Let us assume that the system has a total of $C$ identical servers (channels or units of bandwidths), and each can provide service to any class of arrivals. Let $n=(n_1, n_2, ..., n_R)$ where $n_r$ is the number of class $r$ customers in the system, and let $b=(b_1, b_2, ..., b_R)$. The total number of busy servers in state $n$ is

$$bn^T = b_1n_1 + b_2n_2 + ... + b_Rn_R.$$  \hspace{1cm} (3.1)

The set of all possible states of the system can be described as

$$S^b = \{ n : bn^T \leq C \}.$$  \hspace{1cm} (3.2)

It is well known that the multi-class loss system has a product-form solution given by:

$$P(n) = \prod_{i=1}^{R} \frac{\rho_i^{n_i}}{n_i!} G^{-1}(\Omega), \forall n \in \Omega$$  \hspace{1cm} (3.3)

where

$$G(\Omega) = \sum_{n \in \Omega} \prod_{i=1}^{R} \frac{\rho_i^{n_i}}{n_i!}$$  \hspace{1cm} (3.4)

and $\rho_i = \lambda_i/\mu_i$. The challenge is to obtain the blocking probability for each class. Computing the blocking probabilities by directly enumerating all possible states of the system requires an $O(C^R)$ amount of time. The direct method is computationally cumbersome and grows exponentially fast even for relatively small systems. Several methods have been
presented in the literature to avoid the exponential complexity of the computations. One of the most powerful methods for obtaining the blocking probabilities was published independently by Kaufman (1981) [21] and Roberts (1981) [48]. The Kaufman-Roberts method is a fast recursive algorithm that has a linear complexity of $O(CR)$. The recursive formula is as follows:

$$w(k) = \frac{1}{k} \sum_{r=1}^{R} \rho_r b_r w(k - b_r), k = 1, 2, ..., C.$$  \hspace{1cm} (3.5)

where $w(0)=1$ and $\rho_r = \lambda_r / \mu_r$. Then, the blocking probability of class r arrivals is given by:

$$B_r = \frac{\sum_{j=C-(b_r-1)}^{C} w(j)}{\sum_{j=0}^{C} w(j)}, r = 1, 2, ..., R.$$  \hspace{1cm} (3.6)

It is interesting to know that this formula can be applied to the single class model, as a fast way of obtaining the blocking probability. Given the blocking probabilities, the average number of class r customers in the system is

$$E[Q_r] = \rho_r (1 - B_r), r = 1, 2, ..., R.$$  \hspace{1cm} (3.7)

The multi-rate loss model with variable-demand customers described in the previous section can be analyzed numerically by setting up the underlying rate matrix and subsequently solving it in order to obtain the stationary probability vector. However, this numerical approach is limited to small size problems due to the complexity involved in setting up the rate matrix. It is also difficult to obtain a closed-form expression because of the variable-demand customers.

In view of these considerations, we solve this loss system approximately as follows.

We assume that when a customer changes its class from $i$ to another class, say class $j$, it simply departs from the loss queue and it re-joins it as a new class $j$ customer. Its departure from the loss queue and its arrival to the loss queue are not synchronized. That is, we simply calculate a new arrival rate for class $i$ customers based on the external arrival rate and all the possible feedbacks due to customers changing their class to class $i$. Specifically, we have that the departure rate of class $i$ customers from the loss model is:

$$\mu_i E[Q_i] = \mu_i \rho_i (1 - B_i) = \lambda_i (1 - B_i)$$  \hspace{1cm} (3.8)

Then, the total class $i$ arrival rate due to feedback from other classes is:

$$\lambda_{hi} = \sum_{k=1}^{R} \bar{X}_k (1 - B_k) p_{ki}$$  \hspace{1cm} (3.9)
where $p_{ki}$ is the probability that a class k customer will change to class i and $\bar{\lambda}_k$ is the total class k effective arrival rate (i.e., external arrival rate plus feedbacks from the other classes).

Thus the total effective arrival rate $\bar{\lambda}_i$ of class-i to the loss model is:

$$\bar{\lambda}_i = \lambda_i + \lambda_{hi} = \lambda_i + \sum_{k=1}^{R} \bar{\lambda}_k (1 - B_k) p_{ki}$$

where $\lambda_i$ is the class i external arrival rate to the loss queue. This equation is often called the traffic equation. The effective arrival rate and the blocking probability of each class are unknown and have to be decided iteratively. The total offered load for each class is given by the following nonlinear matrix equations obtained from (3.10):

$$\bar{\rho} = (I - P^T \bar{B})^{-1} \rho'$$

where $I$ is the identity matrix, $\bar{B} = \text{diag}([1 - B_1, 1 - B_2, \ldots, 1 - B_R])$ is a diagonal matrix, $P$ is the class-changing probability matrix with its elements $P(i,j) = p_{ij}$, $\bar{\rho} = [\bar{\rho}_1, \bar{\rho}_2, \ldots, \bar{\rho}_R]$ where $\bar{\rho}_i = \bar{\lambda}_i / \mu_i$ is the effective offered load of class i and $\rho = [\rho_1, \rho_2, \ldots, \rho_R]$. We can now use (3.11) in expression (3.5) in order to calculate the class blocking probabilities.

The resulting system of equations is solved by a fixed-point procedure summarized below.

**Summary of algorithm**

set small value for degree of accuracy $\epsilon$
do (the following steps)

Step 1: Set initially all blocking probabilities and $\lambda_{hi}$ to be zero
Step 2: compute values for total offered load per $\rho_i = \lambda_i / \mu_i$
Step 3: compute values for blocking probabilities $B_i$ per (6)
Step 4: update values for total offered load per (11)
Step 5: update values for blocking probabilities per (6)

while (relative error of two successive blocking prob. $> \epsilon$ )

end while

The above algorithm for the calculation of call blocking probabilities has a time complexity of $O(\log_2(CR/\epsilon) CR^2)$. This algorithm is scalable in the number of classes. A
35

proof of convergence and complexity for a single-class traffic using the bisection algorithm has been sketched out in [50]. For multi-class case considered in this work, we do not provide a proof of convergence. Through numerous examples, however, we observed that this algorithm converges very fast, often in a few tens of steps.

As is known, the Kaufman-Roberts algorithm is numerically unstable, i.e., it causes overflows when the offered load and/or the total number of servers is very large. This can be avoided by using a dynamic factoring technique. We use a small number $\alpha$ as a scaling factor to avoid potential overflows. If upon inspection, it is found that an overflow would occur in the computation of $w(k)$ in the Kaufman-Roberts formula, all $w(i)$ are scaled, i.e., $w(i)=w(i)\alpha$ for $i=0,1,..k$, so that each $w(i)$ is small enough. The process of dynamic scaling increases the computational costs, but the order of the overall complexity remains unchanged.

We observed that a simpler algorithm can be used in the following two cases:

1). When the total number of servers is very large comparing to the offered loads so that the blocking probability for each class is very small (for example, less than 0.001), equation (3.11) can be approximated by $\bar{\rho} = (I - PT)^{-1}\rho$, i.e., we can set all the blocking probabilities equal to zero. In this case, we can calculate the effective offered load from equation (3.11) without iterating on the blocking probabilities.

2). If total capacity $C$ is very large, then the feedback rate from class $i$ to any class $j$ can be simply expressed as $\lambda_i p_{ij}$. In this case, the solution is simplified as in case (1) above.

The above two cases provide an upper bound on the effective offered load.

3.4 Capacity provisioning

Provisioning optimal total capacity is one of practical ways to meet the blocking probability and other QoS requirements. In this section, we describe how to calculate the minimum number of servers $C$ of the loss model so that the maximum blocking probability of any class is less than a pre-specified value $\epsilon$ for a given load. This permits the blocking probabilities of the remaining classes to also be less than $\epsilon$.

This minimum value of $C$ can be calculated iteratively using the fixed-point algorithm described in the previous section. However, when the required capacity $C$ is
very large, this iterative approach becomes CPU intensive since its time complexity is $O(\log 2(CR/\epsilon)CR^2)$.

It is a long-standing conjecture that the optimal number of servers is of the form $\rho + K \sqrt{\rho}$ for single class traffic where $K$ is a constant depending on the offered load and blocking probability. This approximation yields very accurate results. Indeed, based on extensive sensitivity tests, the actual optimum and approximate values rarely deviate by more than one server, or by more than one percent, whichever is greater (see Grassman [15]). Hampshire et al. [17] obtained the following asymptotic expression (called the square-root-safety formula) for the optimum value of $C$ in the multiclass case:

$$ C = \sum_{i=1}^{R} b_i \bar{\rho}_i + \psi(\min_{1 \leq i \leq R} \frac{\epsilon_i}{b_i}) \sqrt{\sum_{i=1}^{R} b_i^2 \bar{\rho}_i} $$  \hspace{1cm} (3.12)

where $\epsilon_i$ is the blocking probability requirement for class $i$ and $\psi(x)$ is the unique solution of the following differential equation

$$ \psi'(x) = \frac{-1}{(\psi(x) + x)x} ; \psi(\sqrt{2/\pi}) = 0 $$  \hspace{1cm} (3.13)

In [17], the authors suggest to use a lookup table for values of $\psi(x)$ by using a second order Runge-Kutta method to compute $\psi(x)$. However, a lookup table may not be practical if the step size is very small and we do not know the starting point $x$. In this work, we have solved equation (3.13) to obtain

$$ x^{-1} e^{-0.5\psi(x)^2} - \sqrt{2\pi} \text{erf}(0.5^{1/2}\psi(x)) - x\sqrt{0.5\pi} = 0 $$  \hspace{1cm} (3.14)

where $\text{erf}(.)$ function is defined as follow:

$$ \text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_{0}^{x} \exp(-t^2) dt $$  \hspace{1cm} (3.15)

Given $x$, equation (3.14) can be easily solved numerically for $\psi(x)$. Applied to equation (3.12), we obtain the requested total capacity. Because of the asymptotic rule [17], satisfying the requirements provides more than enough capacity for all the other classes. Through many numerical examples, we observed that the minimum capacity $C$ obtained using equation (3.12) is very closed to the exact solution.
3.5 Numerical examples

In this section, we validate the accuracy of our approximation and provide some insights into the multi-rate loss queue with variable-demand customers. We also provide a capacity provisioning example.

The approximation results were compared against simulation data. 95% confidence intervals were also calculated, but since they are extremely small, they are not given in the results below. In Table 3.1, we give the approximate and simulation results of call blocking probabilities for three classes customers with the number of servers $C$ varying from 20 to 50. The following parameters were used: $\rho=\{1,2,3\}, b=\{1,2,3\}, p_{i0}=0.5, i=1, 2, 3$. Any class $i$ customer can change to any other class $j$ customer, including its own, with probability $p_{ij}=0.5/3, j=1, 2, 3$.

Table 3.2 gives similar results for a large problem with 100 classes and 1000 servers. The following traffic parameters were used: $\rho_i=i/1000, b_i=i, p_{i0}=0.5$ and $p_{ij}=0.005$, $i=1,2,..100$. Table 3.3 gives similar results as Table 3.2. The assumptions are the same, with the exception that $\rho_i=i/300, i=1,2,..100$. We observe that the approximation model match the simulation results quite well. Some deviations were observed when the blocking probabilities are high (see for instance, Table 3.1, $C=20$, approximate and simulation results for class 3).

As mentioned above, our algorithm runs very fast. For instance, the approximation results given in Table 3.2 were obtained in 0.363709 seconds in Matlab 7.0.4. However, the simulation needs much longer time. The simulation results in Table 3.2 and 3.3 required around 30000 seconds. (The simulation was implemented in C program on a Pentium(R) 4 CPU 3GHz PC).

Next we consider the case where all customers arriving at the loss queue have the same bandwidth profile. Specifically, new customers arrive at the loss queue as class 1 and require 1 server. After an exponentially distributed service time with mean $1/\mu$, a class-1 customer changes to a class-2 customer with a bandwidth requirement of 2 servers with probability $p_{12}=1$. After another exponentially distributed service time with mean $1/\mu$, the class-2 customer changes to a class-3 customer with a bandwidth requirement of 3 servers with probability $p_{23}=1$. Finally, the class-3 customer departs with probability $p_{30}=1$ after an exponentially distributed service time with mean $1/\mu$. The offered loads are $\rho_1>0$, $\rho_2=0$, $\rho_3=0$, i.e., no external class-2 and class-3 arrivals occur. Given this load profile, we
Table 3.1: Approximation and simulation results for 3 classes customers

<table>
<thead>
<tr>
<th></th>
<th>Appr class-1</th>
<th>Appr class-2</th>
<th>Appr class-3</th>
<th>Sim class-1</th>
<th>Sim class-2</th>
<th>Sim class-3</th>
</tr>
</thead>
<tbody>
<tr>
<td>C=20</td>
<td>0.1129</td>
<td>0.2252</td>
<td>0.3347</td>
<td>0.1176</td>
<td>0.2345</td>
<td>0.3485</td>
</tr>
<tr>
<td>C=25</td>
<td>0.0703</td>
<td>0.1446</td>
<td>0.2221</td>
<td>0.0722</td>
<td>0.1487</td>
<td>0.2285</td>
</tr>
<tr>
<td>C=30</td>
<td>0.0405</td>
<td>0.0856</td>
<td>0.1352</td>
<td>0.0413</td>
<td>0.0874</td>
<td>0.1380</td>
</tr>
<tr>
<td>C=35</td>
<td>0.0219</td>
<td>0.0474</td>
<td>0.0764</td>
<td>0.0225</td>
<td>0.0486</td>
<td>0.0779</td>
</tr>
<tr>
<td>C=40</td>
<td>0.0098</td>
<td>0.0217</td>
<td>0.0359</td>
<td>0.0100</td>
<td>0.0221</td>
<td>0.0366</td>
</tr>
<tr>
<td>C=45</td>
<td>0.0036</td>
<td>0.0081</td>
<td>0.0138</td>
<td>0.0037</td>
<td>0.0083</td>
<td>0.0139</td>
</tr>
<tr>
<td>C=50</td>
<td>0.0010</td>
<td>0.0025</td>
<td>0.0043</td>
<td>0.0011</td>
<td>0.0026</td>
<td>0.0045</td>
</tr>
</tbody>
</table>

Table 3.2: Approximation (Appr.) and simulation (Sim) results for 100 classes customers (1)

<table>
<thead>
<tr>
<th>class</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appr</td>
<td>0.00017</td>
<td>0.00035</td>
<td>0.00054</td>
<td>0.00074</td>
<td>0.00096</td>
</tr>
<tr>
<td>Sim</td>
<td>0.00016</td>
<td>0.00034</td>
<td>0.00053</td>
<td>0.00073</td>
<td>0.00094</td>
</tr>
<tr>
<td>class</td>
<td>30</td>
<td>35</td>
<td>40</td>
<td>45</td>
<td>50</td>
</tr>
<tr>
<td>Appr</td>
<td>0.00119</td>
<td>0.00143</td>
<td>0.00169</td>
<td>0.00196</td>
<td>0.00225</td>
</tr>
<tr>
<td>Sim</td>
<td>0.00117</td>
<td>0.00140</td>
<td>0.00166</td>
<td>0.00192</td>
<td>0.00220</td>
</tr>
<tr>
<td>class</td>
<td>55</td>
<td>60</td>
<td>65</td>
<td>70</td>
<td>75</td>
</tr>
<tr>
<td>Appr</td>
<td>0.00255</td>
<td>0.00287</td>
<td>0.00321</td>
<td>0.00357</td>
<td>0.00395</td>
</tr>
<tr>
<td>Sim</td>
<td>0.00250</td>
<td>0.00281</td>
<td>0.00315</td>
<td>0.00350</td>
<td>0.00387</td>
</tr>
<tr>
<td>class</td>
<td>80</td>
<td>85</td>
<td>90</td>
<td>95</td>
<td>100</td>
</tr>
<tr>
<td>Appr</td>
<td>0.00435</td>
<td>0.00477</td>
<td>0.00522</td>
<td>0.00569</td>
<td>0.00618</td>
</tr>
<tr>
<td>Sim</td>
<td>0.00426</td>
<td>0.00468</td>
<td>0.00512</td>
<td>0.00558</td>
<td>0.00606</td>
</tr>
</tbody>
</table>
Table 3.3: Approximation (Appr.) and simulation (Sim) results for 100 classes customers

<table>
<thead>
<tr>
<th>Class</th>
<th>Appr.</th>
<th>Sim</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.02635</td>
<td>0.02891</td>
</tr>
<tr>
<td>10</td>
<td>0.05229</td>
<td>0.05751</td>
</tr>
<tr>
<td>15</td>
<td>0.07780</td>
<td>0.08573</td>
</tr>
<tr>
<td>20</td>
<td>0.10290</td>
<td>0.11291</td>
</tr>
<tr>
<td>25</td>
<td>0.12758</td>
<td>0.14021</td>
</tr>
<tr>
<td>30</td>
<td>0.15184</td>
<td>0.17061</td>
</tr>
<tr>
<td>35</td>
<td>0.17568</td>
<td>0.19782</td>
</tr>
<tr>
<td>40</td>
<td>0.22208</td>
<td>0.22293</td>
</tr>
<tr>
<td>45</td>
<td>0.24465</td>
<td>0.24923</td>
</tr>
<tr>
<td>50</td>
<td>0.27341</td>
<td>0.27341</td>
</tr>
<tr>
<td>55</td>
<td>0.26679</td>
<td>0.28852</td>
</tr>
<tr>
<td>60</td>
<td>0.30982</td>
<td>0.33069</td>
</tr>
<tr>
<td>65</td>
<td>0.35115</td>
<td>0.35115</td>
</tr>
<tr>
<td>70</td>
<td>0.39684</td>
<td>0.39684</td>
</tr>
<tr>
<td>75</td>
<td>0.44718</td>
<td>0.44718</td>
</tr>
<tr>
<td>80</td>
<td>0.37118</td>
<td>0.37118</td>
</tr>
<tr>
<td>85</td>
<td>0.39081</td>
<td>0.39081</td>
</tr>
<tr>
<td>90</td>
<td>0.41001</td>
<td>0.41001</td>
</tr>
<tr>
<td>95</td>
<td>0.42880</td>
<td>0.42880</td>
</tr>
<tr>
<td>100</td>
<td>0.50767</td>
<td>0.50767</td>
</tr>
</tbody>
</table>

compare the following three bandwidth allocation strategies.

Case 1 (variable-demand policy): bandwidth is allocated on demand whenever a customer changes a class. In this case, a customer may be blocked upon arrival to the loss queue as class-1 customer and each time it changes a class. The class-dependent mean service time is $1/\mu$.

Case 2 (maximum service policy): A class 1 customer is allocated the maximum number of servers, i.e., 3 servers, upon arrival as class 1 customer to the loss queue. The mean service time is: (a) $2/\mu$ in order to keep the product of bandwidth and service-time the same as case 1; or (b) $3/\mu$ so that the arrival will use the same mean service time as case 1. The implication in case (a) is that the customer will take full advantage of the 3 servers allocated to it. In case (b) on the other hand, we assume that the customer follows its bandwidth profile and it uses only the required number of servers. A class 1 customer is blocked if these servers are not available upon arrival.

Case 3 (minimum service policy): A class 1 customer is not allowed to change bandwidth requirements. It is allocated the minimum number of customers, i.e., 1 server for a service time $6/\mu$ in order to keep the product of bandwidth and service-time the same as cases 1 and 2. A customer is blocked if no server is available upon arrival.

Case 1 was analyzed using our approximation algorithm, whereas cases 2 and 3 were analyzed using the Erlang loss formula for single class traffic. In order to facilitate the
Table 3.4: Call blocking comparison among variable-demand service, Max and Min service

<table>
<thead>
<tr>
<th>ρ1</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
<th>21</th>
<th>22</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case1</td>
<td>0.0495</td>
<td>0.0691</td>
<td>0.0915</td>
<td>0.1126</td>
<td>0.1336</td>
<td>0.1541</td>
<td>0.1740</td>
<td>0.1932</td>
</tr>
<tr>
<td>Case2a</td>
<td>0.0805</td>
<td>0.1109</td>
<td>0.1430</td>
<td>0.1755</td>
<td>0.2075</td>
<td>0.2384</td>
<td>0.2680</td>
<td>0.2959</td>
</tr>
<tr>
<td>Case2b</td>
<td>0.3093</td>
<td>0.3472</td>
<td>0.3817</td>
<td>0.4131</td>
<td>0.4417</td>
<td>0.4678</td>
<td>0.4917</td>
<td>0.5136</td>
</tr>
<tr>
<td>Case3</td>
<td>0.0270</td>
<td>0.0539</td>
<td>0.0874</td>
<td>0.1238</td>
<td>0.1606</td>
<td>0.1963</td>
<td>0.2302</td>
<td>0.2620</td>
</tr>
</tbody>
</table>

Table 3.5: The optimized capacity vs. offered load (ρ) for 3-classes traffic

<table>
<thead>
<tr>
<th>Offered load</th>
<th>Method</th>
<th>Capacity</th>
<th>Offered load</th>
<th>Method</th>
<th>Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>[0.14 0.29 0.43]</td>
<td>Appr.</td>
<td>15</td>
<td>[10,20,30]</td>
<td>Appr.</td>
<td>301</td>
</tr>
<tr>
<td>[0.14 0.29 0.43]</td>
<td>Asmp</td>
<td>13</td>
<td>[10,20,30]</td>
<td>Asmp</td>
<td>300</td>
</tr>
<tr>
<td>[1,2,3]</td>
<td>Appr.</td>
<td>46</td>
<td>[40,80,120]</td>
<td>Appr.</td>
<td>1088</td>
</tr>
<tr>
<td>[1,2,3]</td>
<td>Asmp</td>
<td>44</td>
<td>[40,80,120]</td>
<td>Asmp</td>
<td>1087</td>
</tr>
<tr>
<td>[3,6,9]</td>
<td>Asmp</td>
<td>105</td>
<td>[100,200,300]</td>
<td>Asmp</td>
<td>2629</td>
</tr>
</tbody>
</table>

Comparison among these three cases, we calculate the average call blocking probability for the case 1 as follows:

\[ B_{avg} = \frac{\sum_{r=1}^{R} \bar{\rho}_r b_r B_r/ \sum_{r=1}^{R} \bar{\rho}_r b_r}{R} \]  

(3.16)

We show the results in Table 3.4 for various values of the class-1 offered load \( \rho_1 \) for a total capacity \( C=100 \).

We note that the variable demand policy outperforms the maximum and minimum service policies when the offered load is medium or large. This observation holds for many other similar examples (not reported here).

Finally, in Table 3.5 we show the minimum required number of servers for a 3-class Erlang loss queue, so that the blocking probability is less than 0.01 for all three classes. The required bandwidth for the three classes is \( b=[1,2,3] \), \( p_{ij}=0.5 \) for \( i=1,2,3 \) and \( p_{ij}=0.5/3 \), \( j=1,2,3 \) and \( i=1,2,3 \). The external offered load \( \rho_1, \rho_2, \rho_3 \) was varied. For each set of value of \( \rho \), we computed the minimum required servers using equation 3.12 (labeled as ‘Asmp’) and also using our algorithm in an iterative manner as explained at the beginning of section 3.4 (labeled as ‘Appr.’).
3.6 Conclusion

In this Chapter, we described a model for calculation of call blocking probabilities in a multi-rate Erlang loss queue where the customers are allowed to change their bandwidth requirements during their service. Comparisons against simulation data showed that the algorithm has a good accuracy. The model was also used to evaluate different allocation policies and capacity provisioning.
Chapter 4

Efficient Dimensioning of
Integrated Streaming and Data Traffic

4.1 Introduction

In this Chapter, we study the problem where the total bandwidth of a link is shared by streaming traffic (real time traffic such as voice or video etc) and data traffic (non-real time variable bit rate traffic). Integrating streaming traffic and data traffic presents a unique dimensioning problem. We dimension the system to satisfy both quality of service (QoS) requirements for streaming traffic, such as loss probability, and data traffic, such as mean waiting (delay) time. An Erlang loss model is used to study the streaming traffic and a single queue with bursty arrival is used to study the data traffic under priority decomposition.
4.2 Related work

Both streaming traffic and data traffic may be supported in a wireless link. For example, Universal Mobile Telecommunications System (UMTS) is a 3G cellular network that supports two broad categories of traffic: streaming traffic and data traffic. Streaming traffic corresponds to the real-time transfer of various signals (e.g., voice, streaming audio/video). This type of traffic is loss sensitive, and stringent transmission bandwidth guarantees are necessary to ensure real-time communication. Data traffic corresponds to the transfer of digital documents (e.g., web pages, emails, stored audio/videos), it is flexible, or “elastic” and bursty in today’s network environment, and the mean delay time (or response time) is a typical performance measure. Another application example of integrated streaming and data traffic is GPRS (General Packet Radio Service) implemented in GSM (Global System for Mobile Communications) networks [13]. GPRS is fundamentally based on the hybrid switching principle with two types of traffic: voice calls and packet data. In accordance with current design and traffic management policies, voice calls have priority over data packets and data packets which cannot immediately be transmitted are queued at the source.

Various papers that study the integration of data and streaming traffic have been published recently (for example, Neuts [40], Bonald et al. [6], Borst et al. [7], Boxma et al. [8], Delcoigné et al. [10], Foh et al. [13], Lee et al. [31], and Numez et al. [42]). In terms of modeling approach, a fluid model is proposed in Bonald et al. [6], Borst et al. [7], Delcoigné et al. [10], Numez-Queija [42]) in order to provide closed-form limit results and approximations. These results may serve as performance bounds. Matrix-geometric solution in [40], [12], [13] and [8] is provided to obtain mean waiting time and other quantities for data traffic. Neuts [40] used the matrix-geometric approach to analyze the M/M/1 queue in a Markovian environment where the arrival and the service processes are governed by a Markov chain. Using this model of M/M/1 queue in a Markovian environment, the performance of integrated voice and data traffic can be studied. Borst et al. [7] studied a channel-aware scheduling in a wireless system supporting a combination of multi-class streaming and data traffic using time-scale decomposition.

In this study, we apply classical Markovian models for both traffic types assuming that streaming traffic has priority over data traffic. Using decomposition, we find an efficient closed-form solution to dimensioning the system. There is little research about dimensioning the system to satisfy both quality of service (QoS) requirements for streaming traffic, such
as loss probability, and data traffic, such as mean waiting time using classical Markov chain approach. The matrix-geometric approach is often used for performance evaluation of single class of streaming and data traffic sharing the same bandwidth. But it does not scale up well. The processing of streaming traffic and data traffic are normally very different in time scale. If we consider that the streaming traffic has priority over data traffic since the streaming traffic is loss sensitive but data traffic is not, then we can treat these two types of traffic as follows. We dimension the system to satisfy the QoS requirement (loss probability) of streaming traffic. Let $C_s$ be the required capacity. Then we dimension the system to satisfy mean waiting time requirement of the data traffic using the spare capacity leftover by the streaming traffic. Let $C_e$ be the required capacity. The final capacity of the system is the maximum of $(C_s, C_e)$.

In this Chapter, for the streaming traffic, the single class Erlang loss model is used (the multirate loss Erlang model can be used as well). For the data traffic, we consider a bursty arrival process modelled by an interrupted Poisson process (IPP) which captures the dynamic nature of today’s data traffic. There is a single server with infinite waiting room (buffer) to serve the data traffic and the service rate reflects the idle capacity of the link. The data traffic is served using the first-come-first-service (FIFO) discipline.

This Chapter is organized as follows. In section 4.3, the integration of streaming and data traffic model is introduced and the decomposition with priority consideration is proposed in section 4.4. Section 4.5 discusses the dimensioning of streaming traffic, the bursty traffic model and dimensioning of data traffic is introduced in section 4.6. We present some numerical examples in section 4.7, and finally, the conclusions are given in section 4.8.

### 4.3 Integration of streaming and data traffic with a unique dimensioning problem

We consider a wired or wireless link which supports both streaming traffic and data traffic, i.e., two types of traffic share the total bandwidth. For example, if the total bandwidth is 2Mbps, there are 256 channels of 8kbps, each represented by a server. For streaming traffic, the Erlang loss model is applied. The data traffic is transmitted using the free bandwidth leftover by the streaming traffic. To capture the bursty nature of data traffic, we use the IPP model.
traffic, we model the arrival process as an interrupted Poisson process. We assume that all awaiting data packets are in single server queue and we use an IPP/G/1 model for data traffic. Integrating streaming traffic and data traffic presents a unique dimensioning problem in that we want to dimension the total link bandwidth so that the loss probability for streaming traffic and mean delay time for data are both satisfied.

4.4 Priority decomposition and local stability

The average holding times for most flows is of the order of several minutes while most data traffic typically lasts a few seconds. We assume that the dynamics of streaming traffic (loss sensitive) take place on a much slower time scale (may be up to three magnitudes) than those of data traffic (delay insensitive). More specifically, we assume that the data traffic reaches statistical equilibrium while the number of active streaming calls remain unchanged. This assumption, called the quasi-stationary regime by Boxma [8] and Borst et al. [7], is shown to be reasonable in practice (see also by Lee [31]). Under quasi-stationary regime, a complete separation of time scales occurs when the data traffic approximately reaches steady-state in between changes in the population of streaming users.

Also, streaming traffic is loss sensitive so that it will be given priority over data traffic which can be delayed in practice. Assuming streaming traffic has priority over data traffic, we can decompose these two traffic by considering that the service bandwidth for data traffic will be the free bandwidth left by the streaming traffic. We call this approach priority decomposition (called time-scale decomposition in some papers, for example, Boxma [8] and Borst et al. [7]). If we assume that the arrival process for streaming traffic is Poisson, and the calls that find all servers occupied are blocked, then in this case, the Erlang loss model can be used to study the performance and dimensioning of streaming traffic. The mean number of free servers for data traffic is

$$C_e = C_s - \lfloor \rho_s(1 - B_s) \rfloor,$$

where $\lfloor x \rfloor$ is the smallest integer greater than or equal to $x$, and $C_s, \rho_s, B_s$ are bandwidth, offered load and blocking probability for streaming traffic respectively. $B_s$ is defined by the Erlang loss formula:

$$B_s = \frac{\rho_s^{C_s} / C_s!}{\sum_{i=0}^{C_s} \rho_s^i / i!}$$ (4.1)

and $\rho_s(1 - B_s)$ is the mean number of busy servers for streaming traffic (this can be obtained by using Little’s relation). The distribution of busy number of servers is a truncated Poisson
distribution assuming that the arrival process is Poisson.

Using this mean bandwidth for the data traffic, we can study the performance of the data traffic and dimensioning the total bandwidth to meet both QoS requirements of streaming traffic and data.

Notice that for local stability of the data traffic, the average arrival rate of the data traffic should be less than the mean service rate. To meet the local stability, some bandwidth of the link has to be reserved for the data traffic so that the offered load to the data traffic is less than unity.

4.5 The dimensioning of streaming traffic

If the streaming traffic has priority over data data traffic, then the streaming traffic can be studied separately from data traffic. (The case that both streaming traffic and data traffic share the bandwidth without priority will be discussed in the following section).

We consider a single class of streaming traffic, but the dimensioning can be extended to the multi-class case. Borst et al. in [7] studied a channel-aware scheduling in a wireless system supporting a combination of multi-class streaming and data traffic using a fluid-flow model. A dimensioning algorithm for the multi-class case was described in Chapter 3, also Tian and Perros [57].

4.6 Dimensioning data traffic with bursty arrivals

We model the data traffic as an interrupted Poisson process (IPP).

The IPP is a special case of the Markov-modulated Poisson process (MMPP) which is a Poisson process whose instantaneous rate is itself a stationary random process which varies according to an irreducible $n$ state Markov chain. It is characterized by the infinitesimal generator matrix $Q$ of the underlying Markov process and by $\Lambda$ of the superposed arrival rate matrix (refer to Fischer et al. [12] for a thorough investigation of the MMPP).

In the IPP, input traffic source alternates between two states (the On and Off state). During the On state, the source generates traffic at rate $\lambda$, and in the Off state, no traffic is generated at all. The On and Off periods are exponentially distributed with the
average rates of $\delta_1$ and $\delta_2$ respectively. Therefore, an individual On/Off source is defined by three parameters, $\lambda$, $\delta_1$ and $\delta_2$.

The IPP corresponds to the special case of MMPP for which $Q$ and $\Lambda$ have the form:

$$Q = \begin{bmatrix} -\delta_1 & \delta_1 \\ \delta_2 & -\delta_2 \end{bmatrix}; \Lambda = \begin{bmatrix} \lambda & 0 \\ 0 & 0 \end{bmatrix}.$$  \hspace{1cm} (4.2)

The $Q$ and $\Lambda$ determine the steady-state probability vector $\Pi$

$$\Pi = (\Pi_1, \Pi_2) = \left( \frac{\delta_2}{\delta_1 + \delta_2}, \frac{\delta_1}{\delta_1 + \delta_2} \right).$$ \hspace{1cm} (4.3)

The mean arrival rate $\lambda_a$ is

$$\lambda_a = \frac{\lambda \delta_2}{\delta_1 + \delta_2}. $$ \hspace{1cm} (4.4)

This process is shown to be a very useful model for data, voice and video over the Internet, where bursty arrivals of packets occur for a period of time followed by an idle interval. The IPP process is also characterized by the index of dispersion count (IDC), which is the variance of the number of arrivals divided by the mean number of arrivals in an interval. $IDC_{IPP}$ that measures the burstiness of the arrival process. Notice that in terms of inter-arrival times, the squared coefficient of variation (SCV) $C^2_{IPP} = \text{Var}(X)/E(X)^2$ is same as the IDC. Heffes and Lucantoni [18] derived the following formula for the IDC of a 2-state MMPP process:

$$IDC_{IPP} = 1 + 2 \frac{\lambda \delta_1}{(\delta_1 + \delta_2)^2}. $$ \hspace{1cm} (4.5)

From this equation we can observe that the value of $IDC_{IPP}$ is affected by the arrival rate and the duration of the ON and OFF periods. IDC can be used to characterize the burstiness of a given process and also to quantify dispersion from the Poisson process which has $IDC = 1$. Notice that in the following sections we use IPP to model the bursty arrival process but a 2-state MMPP can be also used in the proposed solution.

### 4.6.1 Integrating streaming and data traffic without priority

The case where streaming and data traffic share the total bandwidth without priority, can be modelled using Neuts’ matrix-geometric solution for the M/M/1 model in the Markovian random environment (see Neuts [40] for details). The idea is to build a 3-dimensional Markov chain $(i, j, k)$ where $i=0,1,\ldots$, is the number of data traffic packets in
the queue (including the one in service), \( j=0,1,\ldots \), is the number of channels available for the data traffic, and \( k \) indicates the state of the IPP. The algorithm can be summarized as follows:

1). Set total servers as \( c \), the arrival rate of streaming traffic and data traffic as \([\lambda_s, \lambda_e]\) respectively and service rate of single channel for streaming and data traffic as \([\mu_s, \mu_e]\) respectively.

2). Set \( Q \) as the infinitesimal generator matrix of the Markovian environment:

\[
Q = \begin{bmatrix}
-\delta_1 - c\mu_s & \delta_1 & C\mu_s & 0 & \cdots \\
\delta_2 & -\delta_2 - c\mu_s & 0 & c\mu_s & \cdots \\
\lambda_s & 0 & -\lambda_s - \delta_1 - (c-1)\mu_s & \delta_1 & \cdots \\
0 & \lambda_s & \delta_2 & -\lambda_s - \delta_2 - (c-1)\mu_s & \cdots \\
& & & & -\lambda_s - \delta_2
\end{bmatrix}
\] (4.6)

3). Set \( \Lambda_e = [\lambda, 0, \lambda, 0, \ldots] \) as the vector of arrival rate of data traffic;

4). Set \( \mu_e = (1\mu, 2\mu, \ldots, c\mu) \) as the vector of service rate of data traffic.

5). Using Neuts’ formalization, we first calculate the rate matrix \( R \), from the matrix equation

\[
R^2\Delta(\mu_e) + R(Q - \Delta(\Lambda_e + \mu_e)) + \Delta(\Lambda_e) = 0 \] (4.7)

6). The stationary probability vector \( x \) of the queue size is given by

\[
x_k = \pi (I - R)R^k, \text{ for } k \geq 0
\] (4.8)

where \( I \) is the identity matrix and \( \pi \) is given by solving \( \pi Q = 0 \) and \( \pi e = 1 \).

7). Using queue size vector \( x \), we can find the mean queue size and mean delay of the data traffic and loss probability for streaming traffic. For dimensioning, we can use this matrix-geometric solution iteratively to find optimal number of total bandwidth \( c \).

Foh et al. [13] used this approach to evaluate the performance of GPRS (General Packet Radio Service). To dimension the system to meet both QoS requirement of streaming and data traffic, an iterative process has to be applied to the above matrix-geometric solution. However, it can be very expensive to use this process for dimensioning since this process involves matrix multiplication and inversion with complexity at least \( O(c^3) \) where \( c \) is the total capacity.
Aslo if there are multi-class of streaming traffic, matrix-geometric solution cannot be applied.

4.6.2 Integrating streaming and data traffic with priority

When we consider that the streaming traffic has a priority (pre-emptive) over the data traffic, then data traffic can be modeled as IPP/M/1 queue where the service capacity is the mean free bandwidth left by the streaming traffic.

The IPP/M/1 can be analyzed using the matrix-geometric approach which is often used to find steady-state probability distribution by considering two-dimensional Markov chain \((n_1, n_2)\) where \(n_1\) is the indication of ON/OFF and \(n_2\) is the number of customers in the queue. The matrix-geometric solution can be summarized as follows (\(I\) is identity matrix and \(e\) is vector with unity elements):

1. Set \(\Pi = [\pi_0, \pi_1, \ldots]\) as the steady-state probability (\(\pi_k\) indicates that there are \(k\) customers in the system.)

2. Set \(B_0 = Q \cdot \Lambda, A_0 = \Lambda; A_1 = Q \cdot \mu I - \lambda, A_2 = \mu I\) and \(B_1 = A_2\) where \(Q\) and \(\Lambda\) are given in previous section.

3. Solving \(R^2A_2 + RA_1 + A_0 = 0\) for \(R\).

This can be done by using iterative procedure \(R^k = A_0(R^{k-1}A_2 + A_1)^{-1}\) with \(R^0\) as zeros matrix.

4. Use \(\pi_0(B_0 + RB_1) = 0\) and \(\pi_0(I - R)^{-1}e = 1\) as boundary conditions to obtain \(\pi_0\)

5. Then \(\pi_k = \pi_0 R^k\).

Once \(\Pi\) is obtained, we can study the quantities of interests such as the mean waiting time and the probability of waiting. The mean queue size can be computed using the stationary probability vector \(\Pi\), and mean delay can be found by Little’s law. We have:

\[
N_q = \sum_{i=1}^{i=\infty} \pi_i k \quad (4.9)
\]

\[
W_q = \frac{N_q}{\lambda_0} \quad (4.10)
\]

This approach, however, involves matrix multiplication and power, iterative numerical solution which can be costly when used for dimensioning purpose. Alternatively the IPP/M/1
queue can be solved as a GI/M/1 queue. Quantities of interest such as the queueing probability and the mean waiting time can be obtained in closed-form which are more efficient for dimensioning purpose. It is well known that the IPP is statistically identical to hyperexponential distribution with two phases (called $H_2$). The parameters of the $H_2$ can be calculated as follows from IPP assuming that $(\delta_1, \delta_2, \lambda)$ are transition rate from ‘Off’ to ‘On’, transition rate from ‘On’ to ‘Off’ and arrival rate at ‘On’ state respectively, (see [27] and [12]) :

$$
\mu_1 = 0.5(\lambda + \delta_1 + \delta_2 + \sqrt{(\lambda + \delta_1 + \delta_2)^2 - 4\lambda \delta_2}) \tag{4.11}
$$

$$
\mu_2 = 0.5(\lambda + \delta_1 + \delta_2 - \sqrt{(\lambda + \delta_1 + \delta_2)^2 - 4\lambda \delta_2}) \tag{4.12}
$$

$$
p = (\lambda - \mu_2)/(\mu_1 - \mu_2) \tag{4.13}
$$

The above equations can also be obtained using the Laplace transform matching developed in Chapter 5, which is more efficient.

The probability density function (p.d.f.) of the $H_2$ is

$$
p\mu_1 e^{\mu_1 x} + (1 - p)\mu_2 e^{\mu_2 x} \tag{4.14}
$$

The Laplace transform of the hyperexponential distribution $H_2$ is:

$$
A(s) = \left[ \frac{p\lambda_1}{(\lambda_1 + s)} \right] + \left[ \frac{(1 - p)\lambda_2}{(\lambda_2 + s)} \right] \tag{4.15}
$$

Using the well known results for G/M/1 (or $H_2$/M/1) in Kleinrock [25], we can solve the following equation

$$
\sigma = A(\mu - \mu \sigma) \tag{4.16}
$$

Under the stable condition that $\rho = \lambda/\mu < 1$, the equation 4.16 will always have a unique root $0 < \sigma < 1$.

Combining 4.15 and 4.16, we obtain a third degree polynomial in $\sigma$ equal to zero. Since $\sigma = 1$ is always a solution ($A(0) = 1$), we divide the equation by $(\sigma - 1)$. Then $\sigma$ is the solution to the quadratic equation

$$
\sigma^2 - (\lambda_1 + \lambda_2 + \mu)\sigma/\mu + [\lambda_1 \lambda_2 + \mu(p\lambda_1 + (1-p)\lambda_2)]/\mu^2 = 0 \tag{4.17}
$$

where $\mu$ is the mean service rate. Since $0 < \sigma <= 1$, we obtain $\sigma$

$$
[(\lambda_1 + \lambda_2 + \mu) - \sqrt{(\lambda_1 + \lambda_2 + \mu)^2 - 4[\lambda_1 \lambda_2 + \mu(p\lambda_1 + (1-p)\lambda_2)]}]/(2\mu). \tag{4.18}
$$
\( \sigma \) is the queueing probability (or waiting probability) seen by arbitrary arrivals. For the given parameters of arrival process, we can dimensioning the system capacity (the service rate \( \mu \)) to satisfy the delay probability requirement. Then the mean waiting time of the IPP/M/1 will be:

\[
W = \frac{\sigma}{(\mu(1 - \sigma))}
\]  
(4.19)

and the pdf of the waiting time (unconditional) is (see [25]):

\[
f_w(t) = 1 - \sigma \exp(-\mu(1 - \sigma)t)
\]  
(4.20)

Given parameters for the arrival process and QoS requirements (loss probability and mean delay for streaming traffic and data traffic respectively), we can then dimension the network capacity as follows: 1). Use the equation (4.2) to (4.5) to find the minimum capacity \( C_s \) for streaming traffic. 2). Use the equation (4.16) of the mean waiting time to solve for the minimum capacity for data traffic \( C_e \). 3). Use the max \((C_s, C_e)\) as the capacity for the whole system. Algorithm for dimensioning data traffic:

- step 1: solve \( \sigma = A(\mu(1 - \sigma)) \) for \( \sigma \) which is the function of \( \mu \).
- step 2: using \( W = \frac{\sigma}{\mu(1-\sigma)} \) to find \( \mu \) for given \( W \) and \( \sigma \) from step 2.

### 4.6.3 Bursty data traffic with general service time distribution

In the previous section we assumed that the service time of the data traffic is exponentially distributed with a rate equal to the mean free bandwidth left by the streaming traffic. In this section, we model the service time by a general distribution characterized by the first two moments of the free bandwidth.

When the IPP is offered to a single server with a service time Laplace transform \( H(s) \) and finite first two moments, \( m_1 \) and \( m_2 \), the Laplace transform of the waiting time \( D(s) \) is given in [12] by applying MMPP/G/1 queueing model. We have:

\[
D(s) = s(1 - \rho)g[sI + Q - \Lambda(1 - H(s))]^{-1}e
\]  
(4.21)

and the mean waiting time \( W_{IPP} \) is given in [12]

\[
\frac{1}{\rho} \left[ \frac{1}{2(1 - \rho)}(2\rho + \lambda_a m_2 - 2m_1((1 - \rho)g + m_1 \Pi \lambda)(Q + e\Pi)^{-1} \lambda_a) - 0.5\lambda_a m_2 \right]
\]  
(4.22)
where $e$ is a $2 \times 1$ vector of ones, $\lambda_e = (\lambda_1, 0)$, $\rho = \lambda_a/\mu$ is the offered load, and $g=[g_1, 1-g_1]$ is a vector that can be found by an algorithm provided in [12].

A more efficient way to find $g$ is provided in [51] by solving the following equation for $g_1$:

$$\frac{\delta_1 - g_1(\delta_1 + \delta_2)(1 + \frac{1}{g_1} - \frac{1}{g_1 - 1}) + exp\left(\frac{(1 - g_1)\lambda_1\delta_1 - g_1\lambda_2\delta_2}{(1 - g_1)g_1(\lambda_1 - \lambda_2)}\right)}{\lambda_1 - \lambda_2} = 1. \quad (4.23)$$

A fast numerical iterative procedure can then be applied to dimension the system to meet the mean delay requirement $W_{IPP}$ using equation (4.22).

If the mean and variance (or second moment) of the inter-arrival times and the service times are known, then an upper bound of the mean waiting time in the queue is given by the expression (using result from GI/G/1 queue)

$$W_{IPP} \leq \frac{\lambda_a(\sigma_A^2 + \sigma_S^2)}{2(1 - \rho)} \quad (4.24)$$

where $\sigma_A^2$ and $\sigma_S^2$ are the variance of interarrival time and service time distribution respectively, $\lambda_a$ is the mean arrival rate, $\rho$ is the offered load which is product of mean arrival rate and mean service time. It is known that this upper bound will work well when the offered load is closed to 1 (heavy traffic approximation). We may also dimensioning the system to meet the mean delay time requirement by using this upper bound iteratively.

### 4.7 Time-varying capacity for streaming traffic

In mobile networks and other networks, it is frequently the case that the total available resources are time-varying. For example, the capacity of a wireless link may vary due to Rayleigh fading and the capacity of a call center may vary due to time-varying allocated staffing. A finite state Markov chain is often used to model the behavior of the time-varying resources, see Konrad et al. [26] and references therein. In this section, we introduce an accurate and efficient approach to find performance measurement of interests.

Let us have a look at the direct method. Consider an Erlang loss system whose capacity varies over time as modeled by the finite state Markov chain (FSMC) shown in Figure 4.1. Let assume that the system can have $K$ different capacities, namely $S_1$, $S_2$, ..., $S_K$. We will assume that state $k$ is associated with $S_k$ servers. For simplicity, we assume that transitions happen only between neighboring states, though this can be generalized to permit a transition from a state to any other state. We shall refer to the states in Figure 4.1 as the macro states.
The basic idea is to define a 2-dimensional Markov chain \((n_1, n_2)\) shown in Figure 4.1, where \(n_1\) indicates the number of customers in the system and \(n_2\) shows the capacity, i.e., the total number of servers of the system. The exact solution can be obtained using a numerical procedure, such as block Gaussian elimination algorithm introduced in Tian [54] can be used to find the total blocking probability. The block Gaussian elimination algorithm has time complexity \(O(Km^3)\) where \(m\) is the maximum number of servers possible and \(K\) is the number of macro states. The exact solution is impractical for large systems. In the following we introduce an efficient decomposition algorithm which has complexity \(O(K)\). The exhaustive discipline is assumed. That is, if the capacity of the system is reduced, say
by a server, at a time when all servers are busy, then a server will complete its service before it leaves the pool of servers.

The decomposition and aggregation process is as follows:

step 1). Each set of states associated with a macro state is an Erlang system with \( S_k \) servers. There are \( K \) such systems. For each such loss system \( k, k=1,\ldots, K \), the Erlang loss formula (single node) can be used to find the probability distribution and the blocking probability \( b_k \).

step 2). Compute the steady-state probability of the macro state based on the rate diagram given in Figure 4.1. Let \( p_i \) be the probability that the system is in the state \( i, i=1,\ldots,K \).

step 3). Obtain the total blocking probability in the system by \( BP = \sum_{i=1}^{K} p_ib_i \).

Numerical examples show that this decomposition approach produces results very close to the exact solution when the nearly complete decomposability condition (NCD) is satisfied. This is often true for our problem since the transitions between different macro states are very small to avoid fluctuation/instabilities in practice. From many numerical solutions, we observed that under the condition that the blocking probability is small for the loss model (for example less than 0.01) and the transition rate between different macro states (capacity change) is small, approximation results will be very close to the exact solutions.

If the streaming traffic has priority over data traffic, we may find the mean number of free servers using following formula:

\[
m = \sum_{i=1}^{K} (S_i - \rho(1 - b_i))p_i \tag{4.25}\]

where \( \rho \) is the offered load. Then this mean number of free server may be applied to the data traffic. Alternatively, we can analyze the data traffic for each macro state \( k \). We can then aggregate the results by unconditioning on the macro state. Dimensioning for the streaming and data traffic can be done within each macro state. The results can be aggregated by unconditioning on the macro states.

4.8 Numerical examples

In this section, we provide some numerical examples to validate our dimensioning approach. Note that in all the examples, the mean waiting time is measured in seconds.
Table 4.1: The analytical and simulated mean waiting time of the data traffic (IPP/M/1 model)

<table>
<thead>
<tr>
<th>$\rho_s$</th>
<th>$C_s$</th>
<th>$\rho_e$</th>
<th>$IDC_{IPP}$</th>
<th>$W(Ana)$</th>
<th>$W(Sim)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>84.3</td>
<td>101</td>
<td>0.5225</td>
<td>5.8889</td>
<td>0.1348</td>
<td>0.1350</td>
</tr>
<tr>
<td>84.3</td>
<td>101</td>
<td>0.6175</td>
<td>6.7778</td>
<td>0.2150</td>
<td>0.2152</td>
</tr>
<tr>
<td>84.3</td>
<td>101</td>
<td>0.7125</td>
<td>7.6667</td>
<td>0.3518</td>
<td>0.3521</td>
</tr>
<tr>
<td>84.3</td>
<td>101</td>
<td>0.8075</td>
<td>8.5556</td>
<td>0.6274</td>
<td>0.6276</td>
</tr>
<tr>
<td>84.3</td>
<td>101</td>
<td>0.8550</td>
<td>9.0000</td>
<td>0.9020</td>
<td>0.9021</td>
</tr>
<tr>
<td>84.3</td>
<td>101</td>
<td>0.9025</td>
<td>9.4444</td>
<td>1.4452</td>
<td>1.4454</td>
</tr>
<tr>
<td>84.3</td>
<td>101</td>
<td>0.9500</td>
<td>9.8889</td>
<td>3.0237</td>
<td>3.0240</td>
</tr>
<tr>
<td>84.3</td>
<td>101</td>
<td>0.9595</td>
<td>9.9778</td>
<td>3.7844</td>
<td>3.7847</td>
</tr>
<tr>
<td>84.3</td>
<td>101</td>
<td>0.9690</td>
<td>10.0667</td>
<td>5.0122</td>
<td>5.0115</td>
</tr>
<tr>
<td>84.3</td>
<td>101</td>
<td>0.9785</td>
<td>10.1556</td>
<td>7.2971</td>
<td>7.3279</td>
</tr>
</tbody>
</table>

And for the waiting times obtained via simulation, we only give the mean but not provide the confidence intervals since the confidence intervals are very small.

**Example 1:** We compare the mean-waiting-time of the data traffic obtained from the analytical and simulation results. In this example, the blocking probability for streaming traffic is set to be 0.01. We use formulae (4.2)-(4.5) to find the optimal capacity $C_s = 101$ for streaming traffic when the offered load is $\rho_s = 84.3$. For the data traffic, the mean packet length is set to 512 bytes (a typical TCP packet) and each channel has a bandwidth of 8 Kbps so that each packet will take $512*8/(8*1024)=0.5$ seconds (with service rate 2 packets/per channel). The parameters for the IPP are: $\delta_1 = 10, \delta_2 = 5$ and the offered load for data traffic is varied. In Table 1.1, we compare the mean-waiting-time $W$ for data traffic obtained using the IPP/M/1 model. $[W(Ana), W(Sim)]$ is the analytical and simulation result respectively. This is a typical result. We observed that the analytical results match the simulation very well through many examples.

**Example 2:** We compare the optimal capacity obtained from analytical and simulation results. In this example, the blocking probability for the streaming traffic is set to 0.01, and we use formulae (4.2)-(4.5) to find the optimal capacity for the streaming traffic. For the data traffic, the IPP parameters are $\delta_1 = 10, \delta_2 = 5$ and $\lambda_e$ is varying. We find the capacity for the data traffic so that the mean waiting time $W = 1$ by varying the $\lambda_e$. Each channel for the data traffic is 4 Kbps so that the service rate of each channel is normalized to be one. $[C_e(Ana), C_e(Sim)]$ is respectively the analytical and simulation results shown
Table 4.2: The optimum capacity for the streaming traffic (with priority) and data traffic (IPP/M/1 model) when the mean waiting time $W=1$ second

<table>
<thead>
<tr>
<th>$\rho_s$</th>
<th>$C_s$</th>
<th>$\lambda_e$</th>
<th>$IDC_{IPP}$</th>
<th>$C_e(Ana)$</th>
<th>$C_e(Sim)$</th>
<th>Max($C_e + \rho_s \times 0.99, C_s$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>84.3</td>
<td>101</td>
<td>10</td>
<td>1.8889</td>
<td>4</td>
<td>4</td>
<td>101</td>
</tr>
<tr>
<td>84.3</td>
<td>101</td>
<td>20</td>
<td>2.7778</td>
<td>8</td>
<td>9</td>
<td>101</td>
</tr>
<tr>
<td>84.3</td>
<td>101</td>
<td>30</td>
<td>3.6667</td>
<td>12</td>
<td>12</td>
<td>101</td>
</tr>
<tr>
<td>84.3</td>
<td>101</td>
<td>40</td>
<td>4.5556</td>
<td>16</td>
<td>16</td>
<td>101</td>
</tr>
<tr>
<td>84.3</td>
<td>101</td>
<td>50</td>
<td>5.4444</td>
<td>20</td>
<td>20</td>
<td>103</td>
</tr>
<tr>
<td>84.3</td>
<td>101</td>
<td>60</td>
<td>6.3333</td>
<td>24</td>
<td>24</td>
<td>107</td>
</tr>
<tr>
<td>84.3</td>
<td>101</td>
<td>70</td>
<td>7.2222</td>
<td>28</td>
<td>28</td>
<td>111</td>
</tr>
<tr>
<td>84.3</td>
<td>101</td>
<td>80</td>
<td>8.1111</td>
<td>32</td>
<td>32</td>
<td>115</td>
</tr>
<tr>
<td>84.3</td>
<td>101</td>
<td>90</td>
<td>9.0000</td>
<td>36</td>
<td>36</td>
<td>119</td>
</tr>
<tr>
<td>84.3</td>
<td>101</td>
<td>100</td>
<td>9.8889</td>
<td>40</td>
<td>40</td>
<td>123</td>
</tr>
</tbody>
</table>

in Table 4.2. We observe that the analytical results match the simulation results very well in this case.

**Example 3**: We compare analytical and simulation results of the mean waiting time using the IPP/G/1 model with general service time distribution with two moments available. In this example, $C_s=101$, $\rho_s = 84.3$, $\delta_1 = 10$, $\delta_2 = 5$, and $\lambda$ is varied from 10 to 100 so that $IDC_{IPP}$ changes. In Table 4.3, we compare the mean waiting time from the analytical approach and simulation. It is observed that the analytical and simulation results match very well.

**Example 4**: We compare the optimal capacity obtained using the IPP/M/1 model and IPP/G/1 model in the work. In this example, $C_s=101$, $\rho_s = 84.3$, $\delta_1 = 10$, $\delta_2 = 5$, and $\lambda$ is varied from 10 to 60 so that $IDC_{IPP}$ changes.

In this case we find the optimal capacity for the data traffic using a one- and two-moment approximation of the service time distribution, that is, we use IPP/M/1 and IPP/G/1. The two moments are obtained from the occupancy distribution of the Erlang loss model assuming the streaming traffic only. Let $m_1$ be the first moment and $m_2$ the second moment. $\delta_1 = 10$, $\delta_2 = 5$. The service rate of each data channel is normalized to 1. The optimal capacity for the data traffic using one or two moments is given in Table 4.4 for $W=0.5$ and $W=1$. The optimal capacity for the streaming traffic is also given.

In general, the optimal capacity obtained using two moments is larger than that obtained using the first moment (especially when mean waiting time is smaller than 1). For
Table 4.3: The mean waiting time of IPP/G/1 model with two moments of service time distribution

<table>
<thead>
<tr>
<th>$\rho_s$</th>
<th>$C_s$</th>
<th>$IDC_{IPP}$</th>
<th>Ana W</th>
<th>Sim W</th>
<th>Relative error(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>84.3</td>
<td>101</td>
<td>1.8889</td>
<td>3.4690</td>
<td>3.6274</td>
<td>4.37</td>
</tr>
<tr>
<td>84.3</td>
<td>101</td>
<td>2.7778</td>
<td>6.8326</td>
<td>6.9161</td>
<td>1.21</td>
</tr>
<tr>
<td>84.3</td>
<td>101</td>
<td>3.6667</td>
<td>10.2164</td>
<td>10.2748</td>
<td>0.57</td>
</tr>
<tr>
<td>84.3</td>
<td>101</td>
<td>4.5556</td>
<td>13.6205</td>
<td>13.6665</td>
<td>0.34</td>
</tr>
<tr>
<td>84.3</td>
<td>101</td>
<td>5.4444</td>
<td>17.0451</td>
<td>17.0836</td>
<td>0.23</td>
</tr>
<tr>
<td>84.3</td>
<td>101</td>
<td>6.3333</td>
<td>20.4905</td>
<td>20.5239</td>
<td>0.16</td>
</tr>
<tr>
<td>84.3</td>
<td>101</td>
<td>7.2222</td>
<td>23.9567</td>
<td>23.9866</td>
<td>0.12</td>
</tr>
<tr>
<td>84.3</td>
<td>101</td>
<td>8.1111</td>
<td>27.4440</td>
<td>27.4712</td>
<td>0.1</td>
</tr>
<tr>
<td>84.3</td>
<td>101</td>
<td>9</td>
<td>30.9526</td>
<td>30.9777</td>
<td>0.08</td>
</tr>
<tr>
<td>84.3</td>
<td>101</td>
<td>9.8889</td>
<td>34.4827</td>
<td>34.5061</td>
<td>0.07</td>
</tr>
</tbody>
</table>

Table 4.4: The optimized capacity comparison for streaming traffic using two moments

<table>
<thead>
<tr>
<th>$\rho_s$</th>
<th>$C_s$</th>
<th>$(m_1)$ (W=0.5)</th>
<th>$(m_1, m_2)$ (W=0.5)</th>
<th>$(m_1)$ (W=1)</th>
<th>$(m_1, m_2)$ (W=1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>27.24</td>
<td>38</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>34.10</td>
<td>46</td>
<td>9</td>
<td>10</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>60.72</td>
<td>75</td>
<td>13</td>
<td>17</td>
<td>11</td>
<td>13</td>
</tr>
<tr>
<td>75.02</td>
<td>91</td>
<td>17</td>
<td>23</td>
<td>15</td>
<td>18</td>
</tr>
<tr>
<td>84.3</td>
<td>101</td>
<td>21</td>
<td>31</td>
<td>19</td>
<td>23</td>
</tr>
<tr>
<td>129.06</td>
<td>148</td>
<td>26</td>
<td>40</td>
<td>23</td>
<td>28</td>
</tr>
</tbody>
</table>
dimensioning purpose we should use both moments if available.

**Example 5**: In Table 4.5, we compare the mean waiting time in the IPP/G/1 queue assuming the following four different service time distributions, all with the same mean rate of 1.0:

1. A deterministic service time with constant service time \(d\) and its Laplace transform is \(\exp(-sd)\):

2. A 4-stage Erlang distribution, \(E_4\), with density distribution

\[
b(x) = \frac{\mu(\mu x)^3 \exp(-\mu x)}{3!}
\]

(4.26)

and Laplace transform \(\mu^4/(s + \mu)^4\):

3. An exponential distribution with density function \(b(x) = \mu \exp(-\mu x)\) and Laplace transform \(\mu/(s + \mu)\).

4. A hyperexponential distribution \(H_2\) with density function

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

(4.27)

and balanced mean such that \(p/\mu_1 = (1 - p)/\mu_2\) and we set \(p = 0.2\).

These four distributions are listed in an increasing order of randomness (defined as the squared coefficient of variation, i.e., \(\text{variance}/\text{mean}^2\)). The squared coefficients of variation are \([0, 0.25, 1, 2.125]\) respectively for \([\text{constant, } E_4, \text{exponential, } H_2]\) distributions respectively.

In this example, \(C_s=101, \rho_s = 84.3, \delta_1 = 10, \delta_2 = 5,\) and \(\lambda\) is varied from 10 to 100 so that \(IDC_{IPP}\) will change.

We compare the optimal capacity for the data traffic for different service time distributions when mean waiting time \(W = 0.1\), results are shown in Table 4.5.

We observe that the mean-waiting-time is increasing as the randomness of service time distributions is increasing when other traffic parameters are the same. This is consistent with the theoretical analysis.

**Example 6**: In this example, we compare the results of exact and decomposition approach when the total capacity for streaming traffic is time-varying. We consider 9 macro states where the total available bandwidth (number of servers) \(S\) varies from 27 to 3 by increments of 3, \(\lambda=1\). We randomly generate a transition rate matrix \(Q\) with nine states.

Notice that the transition probability \(P\) can be obtained by converting the rate matrix to probability matrix \(P = I + \frac{Q}{\max_i q_{ii}}\) where \(I\) is the identity matrix and \(\max_i q_{ii}\) is
Table 4.5: The capacity comparison of different service time distributions when $W = 0.1$.

<table>
<thead>
<tr>
<th>$\rho_s$</th>
<th>$C_s$</th>
<th>$IDC_{1PP}$</th>
<th>$W(\text{constant})$</th>
<th>$W(E_4)$</th>
<th>$W(\text{Exp})$</th>
<th>$W(H_2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>84.3</td>
<td>101</td>
<td>1.8889</td>
<td>10</td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>84.3</td>
<td>101</td>
<td>2.7778</td>
<td>17</td>
<td>18</td>
<td>18</td>
<td>20</td>
</tr>
<tr>
<td>84.3</td>
<td>101</td>
<td>3.6667</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
</tr>
<tr>
<td>84.3</td>
<td>101</td>
<td>4.5556</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>84.3</td>
<td>101</td>
<td>5.4444</td>
<td>38</td>
<td>39</td>
<td>40</td>
<td>41</td>
</tr>
<tr>
<td>84.3</td>
<td>101</td>
<td>6.3333</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
</tr>
<tr>
<td>84.3</td>
<td>101</td>
<td>7.2222</td>
<td>52</td>
<td>53</td>
<td>54</td>
<td>55</td>
</tr>
<tr>
<td>84.3</td>
<td>101</td>
<td>8.1111</td>
<td>59</td>
<td>60</td>
<td>61</td>
<td>62</td>
</tr>
<tr>
<td>84.3</td>
<td>101</td>
<td>9</td>
<td>66</td>
<td>67</td>
<td>68</td>
<td>69</td>
</tr>
<tr>
<td>84.3</td>
<td>101</td>
<td>9.8889</td>
<td>73</td>
<td>74</td>
<td>75</td>
<td>76</td>
</tr>
</tbody>
</table>

Table 4.6: Exact and disaggregation results for the blocking probability of the streaming traffic.

<table>
<thead>
<tr>
<th>Exact</th>
<th>disaggregation</th>
<th>Relative error %</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0176</td>
<td>0.0173</td>
<td>1.7341</td>
</tr>
<tr>
<td>0.0117</td>
<td>0.0116</td>
<td>0.8621</td>
</tr>
<tr>
<td>0.0039</td>
<td>0.0038</td>
<td>2.6316</td>
</tr>
<tr>
<td>0.0013</td>
<td>0.0012</td>
<td>0.8929</td>
</tr>
<tr>
<td>0.0034</td>
<td>0.0033</td>
<td>3.0303</td>
</tr>
<tr>
<td>0.0195</td>
<td>0.0192</td>
<td>1.5625</td>
</tr>
<tr>
<td>0.0118</td>
<td>0.0116</td>
<td>1.7241</td>
</tr>
<tr>
<td>0.0010</td>
<td>0.0010</td>
<td>0</td>
</tr>
<tr>
<td>0.0176</td>
<td>0.0174</td>
<td>1.1494</td>
</tr>
</tbody>
</table>

the largest diagonal element (absolute value) of rate matrix $Q$. Similarly the probability matrix can be converted to rate matrix $Q$. Exact and approximate results for the blocking probability of the streaming traffic are shown in Table 4.6.

From many numerical comparisons, we observed that under the condition that the blocking probability is small for the loss model (for example less than 0.01) and the transition rate between different macro states (capacity change) is small, the disaggregation results will be very close to the exact solutions.
4.9 Conclusion

In this Chapter, performance evaluation and dimensioning approaches for the integrated streaming and data traffic are presented. We have proposed efficient solutions to dimension the integrated streaming and data traffic by using priority decomposition (or time-scale decomposition as referred to in some literature). Comparing the analytical results with simulation results, we found that they matched very well.

An efficient and accurate decomposition/aggregation approach is proposed when the total capacity for streaming traffic is time-varying. The case that the arrival process of the data traffic is more general (for example the arrival process is a Markov modulated Poisson processes) can be an interesting extension.
Chapter 5

Performance Analysis and Efficient Provisioning of Statistical Multiplexer with Correlated and Bursty Arrivals

5.1 Introduction

In this Chapter, we analyze a statistical multiplexer consisting of a single-server queue. The service time is assumed to be exponentially distributed and the queue size is infinite. \( n \) IPP identical arrival streams flow into the multiplexer.

The statistical multiplexer has been widely studied in the open literature. In this Chapter, we first approximate the superposition of the \( n \) IPP arrival streams by a two-stage MMPP (referred to as \( MMPP_2 \)) and then we solve the resulting queue as an \( MMPP_2/M/1 \) queue. The analysis of the queue is done numerically. Subsequently, we obtain the exact Laplace transform solution of the departure process from the statistical multiplexer and
develop an efficient method for approximating the departure process from the statistical multiplexer by an $MMPP_2$. Extensive numerical results have validated the accuracy of this approximation.

The main contribution in this Chapter are the efficient approximation of $n$ IPPs ($MMPP_2$) by a single $MMPP_2$ and the exact Laplace transform and $MMPP_2$ approximation of the departure process of the $MMPP_2/M/1$ queue. These results permit us to analyze a number of statistical multiplexer linked in series, as they arise for instance in an ADSL access network, see FIG. 5.1. Such a network consists of subscribers linked to a DSLAM via the telephone links. Groups of DSLAMs are linked to a metro Ethernet switch and groups of Ethernet switches are linked to a broadband remote access server (BRAS), which is connected to the Internet. In this network we can recognize three statistical multiplexer. The first one represents a DSLAM that serves a number of homes, the second represents a metro Ethernet switch that serves a group of DSLAMs, and the third one represents the BRAS that serves all the metro Ethernet switches. Such a network of statistical multiplexer can be decomposed to individual multiplexer, which can then be analyzed in isolation. The key to this analysis is the characterization of the departure process from a statistical multiplexer. This is one of main objectives of this Chapter. Analysis and dimensioning examples are also provided in the numerical example section.

The multiplexer problem was analyzed extensively in the discrete-time case within the context of ATM networks. Despite the vast literature on ATM multiplexer, this problem was never completely analyzed. The multiplexer problem in the continuous-time case, has not been satisfactorily analyzed as well.

The arrival process and departure process to/from a multiplexer can be autocorrelated. For example, the arrival process of a queueing node can be the superposition of $n$ interrupted Poisson processes (IPP), which is known to be autocorrelated in general. The departure process of a single server queue with auto-correlated arrival processes, such as Markov arrival process (MAP) is not a MAP in general. Failure to take the auto-correlation in both the arrival process and the departure process into consideration can lead to serious underestimation of the performance measures. In this work, we first provide an efficient way to approximate the superposition of $n$ IPP (or $MMPP_2$) arrival process by an $MMPP_2$. We then find the closed-form Laplace transform of the departure process of an $MMPP_2/M/1$
Figure 5.1: A typical three-level access network

queue and we finally approximate the departure process by another $MMPP_2$ using the Laplace transform matching technique, to the best of our knowledge, this is an original result. Numerical comparisons with simulation results show that our approximation of the superposition of $n$ IPP (or $MMPP_2$) by another $MMPP_2$ is very accurate. Also simulation results of the departure process match very well with our $MMPP_2$ approximation of the departure process.
5.2 The Laplace transform of the inter-arrival times of an

**MMPP$_2$**

In this section we obtain the Laplace transform of the p.d.f. of the inter-arrival
time of a two-state Markov modulated Poisson process, hereafter referred to as **MMPP$_2$**. Let

$$Q = \begin{bmatrix} -\delta_1 & \delta_1 \\ \delta_2 & -\delta_2 \end{bmatrix} \text{ and } \Lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \quad (5.1)$$

The **MMPP$_2$** is a renewal process only in very special cases (such as IPP). This
may be intuitively seen as follows (see Fischer et al. [12]). Consider two consecutive arrivals
in an **MMPP$_2$**. Suppose the state of the continuous-time Markov chain $J(t)$ is 1 at the
first arrival and 2 at the second arrival. In between these arrivals there is a first transition
from state 1 to state 2 via several steps, followed by a geometric number of returns to state
2 during which no arrivals occur, followed by an arrival in state 2. Clearly, each of these
distributions depends on 1 and 2. Thus the time between arrivals is not exponential and
the distribution of the time between the $(k-1)$st and $k$th arrivals depends on the state of
$J(t)$ at the time of the $(k-1)$st and the $k$th arrivals.

The Laplace transform of the p.d.f. of the inter-arrival time of an **MMPP$_2$** can
be obtained as follows. A transition probability matrix for the embedded Markov renewal
process is defined in Fischer et al. [12]: let $J_0$ be the state of $J(t)$ at time $t=0$ and $X_0=0$.
Associate with the $k$th arrival of the MMPP the corresponding state $J_k$ of the underlying
Markov process as well as the time $X_k$ between the $(k-1)$st and the $k$th arrivals. Then the
sequence $(J_n, X_n), n \geq 0$ is a Markov renewal sequence with transition probability matrix

$$f(x) = \int_0^x [e^{x(Q-\Lambda)}u] du = (I - e^{x((Q-\Lambda) t)})(\Lambda - Q)^{-1}\Lambda \quad (5.2)$$

Define

$$\lambda = [\lambda_1, \lambda_2]', \pi = [\delta_2, \delta_1]/(\delta_1 + \delta_2), p = \pi\Lambda/(\pi\lambda). \quad (5.3)$$

where $\pi$ is the stationary vector of matrix $Q$ and $p$ is the stationary vector of matrix
$(\Lambda - Q)^{-1}\Lambda$ which is called the transition probability matrix of the Markov chain embedded
at arrival epochs (see for example Fischer et al. [12]).
The Laplace transform matrix $F(s)$ of the transition probability matrix $f(x)$ is then given by

$$F(s) = E[exp(-sX)] = (sI - Q + \Lambda)^{-1}\Lambda$$  \hspace{1cm} (5.4)$$

where $I$ is the identity matrix. The Laplace transform of the p.d.f. of the inter-arrival time of the $MMPP_2$ can be obtained as follows:

$$f_{MMPP_2}^*(s) = pF(s)e = p(sI - Q + \Lambda)^{-1}\Lambda e \hspace{1cm} (5.5)$$

After some calculations we have:

$$f_{MMPP_2}^*(s) = \left(\delta_2\lambda_1^2 + \delta_1\lambda_2^2\right)s + \delta_2\lambda_1\lambda_2 + 2\delta_1\lambda_2\delta_2\lambda_1 + \delta_1^2\lambda_2^2 + \delta_1\lambda_2^2\lambda_1 \hspace{1cm} (5.6)$$

or

$$f_{MMPP_2}^*(s) = \frac{\left(\delta_2\lambda_1^2 + \delta_1\lambda_2^2\right)s + \lambda_1\lambda_2 + \lambda_1\delta_2 + \lambda_2\delta_1}{s^2 + (\lambda_1 + \lambda_2 + \delta_1 + \delta_2)s + \lambda_1\lambda_2 + \lambda_1\delta_2 + \lambda_2\delta_1} \hspace{1cm} (5.7)$$

The moments of the inter-arrival times in an $MMPP_2$ can be obtained by differentiating the above Laplace transform. These moments can be used in a moment matching approach to fit an $MMPP_2$ model from collected data or superposition of different arrival processes. We will show application examples later. Since the IPP is a special case of the $MMPP_2$, the Laplace transform of the inter-arrival times of an IPP with parameters $(\delta_1, \delta_2, \lambda_1)$ can be obtained by setting $\lambda_2 = 0$. We have:

$$f_{IPP}^*(s) = \frac{\lambda_1(s + \delta_2)}{s^2 + (\lambda_1 + \delta_1 + \delta_2)s + \lambda_1\delta_2} \hspace{1cm} (5.8)$$

To the best of our knowledge, this is perhaps the most efficient way to obtain Laplace transform of p.d.f of inter-arrival time of IPP.

### 5.3 The probability density function of an $MMPP_2$ as a counting process

In the above section, we derived the Laplace transform of the inter-arrival times of an $MMPP_2$. In this section, we will consider an $MMPP_2$ as a counting process and derive its probability density function and z-transform. Let $N_t$ be the number of arrivals in $(0,t]$ and $J_t$ the state of the Markov process at time $t$. Now let

$$P_{ij}(n,t) = Pr\{N_t = n, J_t = j | N_0 = 0, J_0 = i\} \hspace{1cm} (5.9)$$
be the \((i, j)\)-entry of a matrix \(P(n, t)\). The matrices \(P(n, t)\) satisfy the (forward) Chapman-Kolmogorov equations
\[
P'(n, t) = P(n, t)(Q - \Lambda) + P(n - 1, t)\Lambda, \quad n \geq 1, \quad t \geq 0
\] (5.10)
with \(P(0, 0) = I\) (identity matrix). The matrix generating function \(P^*(z, t) = \sum_{n=0}^{\infty} P(n, t)z^n\) then satisfies
\[
\frac{d}{dt} P^*(z, t) = P^*(z, t)(Q - \Lambda) + zP^*(z, t)\Lambda,
\] (5.11)
with \(P^*(z, 0) = I\) so that \(P^*(z, t)\) is explicitly given by
\[
P^*(z, t) = \exp((Q - (1 - z)\Lambda)t) \quad (5.12)
\]
For an \(MMPP_2\) with parameters \((\delta_1, \delta_2, \lambda_1, \lambda_2)\), we have:
\[
P^*(z, t) = \begin{bmatrix}
\frac{\delta_2 + \delta_1 e^{(\delta_1 + \delta_2)t}}{\delta_1 + \delta_2} + e^{\lambda_1(1 - z)t} & \frac{-\delta_1(1 - z)}{\delta_1 + \delta_2} \\
\frac{-\delta_1 e^{-(\delta_1 + \delta_2)t} - 1}{\delta_1 + \delta_2} & \frac{\delta_1 + \delta_2 e^{(\delta_1 + \delta_2)t}}{\delta_1 + \delta_2} + e^{\lambda_2(1 - z)t}
\end{bmatrix}
\] (5.13)
Similar to the Laplace transform of the inter-arrival times, the z-transform of the probability density function of the counting process of an \(MMPP_2\) can be obtained as follows:
\[
f(z, t) = pP^*(z, t)e
\] (5.14)
where \(p\) and \(e\) are defined as before. After some simplifications, we obtain:
\[
f(z, t) = 1 + \frac{\delta_1 \lambda_2}{\delta_1 \lambda_2 + \delta_2 \lambda_1} e^{(z-1)\lambda_1 t} + \frac{\delta_2 \lambda_1}{\delta_1 \lambda_2 + \delta_2 \lambda_1} e^{(z-1)\lambda_2 t}
\] (5.15)
We know that the probability density function of the Poisson process (as a counting process) with average arrival rate \(\lambda\) is
\[
Pr(N_t = k) = \frac{e^{-\lambda t}(\lambda t)^k}{k!}
\] (5.16)
From this, we obtain its z-transform (probability generating function) as follows:
\[
f_1(z, t) = \sum_{k=0}^{\infty} \frac{e^{-\lambda t}(\lambda t)^k}{k!}z^k = e^{(z-1)\lambda t}
\] (5.17)
Comparing this to equation 5.15, we find that the z-transform of the \(MMPP_2\) has two Poisson terms; its physical interpretation is obvious. By inverting its z-transform, we obtain its probability density function as follows:
\[
f(n, t) = \Delta(n) + \frac{\delta_1 \lambda_2}{\delta_1 \lambda_2 + \delta_2 \lambda_1} (\lambda_1 t)^k e^{-\lambda_1 t} \frac{1}{k!} + \frac{\delta_2 \lambda_1}{\delta_1 \lambda_2 + \delta_2 \lambda_1} (\lambda_2 t)^k e^{-\lambda_2 t} \frac{1}{k!}
\] (5.18)
where $\Delta(n)$ has $z$-transform equals to 1.

Let $\alpha = \frac{\delta_1 \lambda_2}{\delta_1 \lambda_2 + \delta_2 \lambda_1}$, we can simplify the $f(n, t)$ as follows:

$$f(n, t) = \alpha \frac{(\lambda_1 t)^k e^{-\lambda_1 t}}{k!} + (1 - \alpha) \frac{(\lambda_2 t)^k e^{-\lambda_2 t}}{k!} + \Delta(n)$$  \hspace{1cm} (5.19)

The $z$-transform of the p.d.f. of the superposed $n$ homogeneous $MMPP_2$ process is:

$$F(z, t) = (f_1(z, t))^n$$  \hspace{1cm} (5.20)

The number of arrivals of the superposed traffic in an observation interval can be obtained by inverting the $z$-transform.

### 5.4 The superposition of $n$ interrupted Poisson processes

Generally speaking, there are two ways we can study the packet stream generated by a single source: a) study the number of arrivals during a fixed time interval, and b) study the inter-arrival times between successful packet arrivals. Also, the superposition of $n$ independent identical distributed (IID) arrival processes can be studied using various techniques such as: (1) study the number of arrivals (for example, counts) in an interval $(0,t)$; (2) characterize the inter-arrival time distribution (such as the probability density distribution (p.d.f.)); (3) approximate the $n$ IPP by an $MMPP_2$. In this Chapter, we will combine the above three methods to characterize the superposition of $n$ independent identical interrupted Poisson processes (IPP). It is well known that a single IPP is a renewal process with independent identical distributed (IID) inter-arrival times but the superposition of $n$ IPP is not a renewal process in general. In this section, we will obtain the p.d.f. of the superposed process in closed-form and then compute its moments. We will approximate this superposition with an $MMPP_2$ in the following section.

Whitt in [59] introduced the general formula to find the c.d.f (cumulative density function) of the inter-arrival times of the superposed process using the stationary-interval method of the superposed process.

Let the stationary interval in the $i$th component process have a c.d.f $F_i$ with $j$th moment $\mu_{ij}$ and let the intensity of the $i$th counting process be $\lambda_i = \mu_{i1}$. Then the mean $\mu$ and its c.d.f $F$ satisfy

$$\mu^{-1} = \lambda = \lambda_1 + \lambda_2 + ... + \lambda_n$$  \hspace{1cm} (5.21)
and

$$1 - F(x) = \sum_{i=1}^{n} (\lambda_i/\lambda)(1 - F_i(x)) \prod_{j \neq i} \lambda_j \int_{x}^{\infty} [1 - F_j(s)] ds.$$  \hspace{1cm} (5.22)

Let $T$ be a random variable with the above c.d.f. Then for $k \geq 1$, its moments can be obtained as follows:

$$E[T^k] = \int_{0}^{\infty} kx^{k-1}[1 - F(x)] dx$$  \hspace{1cm} (5.23)

For the superposition of $n$ homogeneous IPPs, if we set the c.d.f. excessive function

$$F_e(t) = \lambda_1 \int_{0}^{t} [1 - F_1(u)] du, t \geq 0,$$  \hspace{1cm} (5.24)

where $F_1(t)$ is the c.d.f. of a single arrival process which can be an arbitrary MMPP process. Applying above formula (5.22)-(5.24), we obtain the c.d.f of the inter-arrival times in the superposed process as

$$F(t) = 1 - (1 - F_1(t))(1 - F_e(t))^{n-1}, t \geq 0,$$  \hspace{1cm} (5.25)

and we can obtain moments of the inter-arrival times of superposed process accordingly once the c.d.f of a single IPP is given.

We know (for example, from Kuczura [27]) that the p.d.f of a single IPP can be written as

$$f(t) = pe^{-\mu_1 t} + (1-p)e^{-\mu_2 t}, t \geq 0,$$  \hspace{1cm} (5.26)

where $\mu_1$, $\mu_2$ and $p$ are obtained from $(\lambda, \delta_1, \delta_2)$, the three parameters of IPP

$$\mu_1 = 0.5(\lambda + \delta_1 + \delta_2 + \sqrt{(\lambda + \delta_1 + \delta_2)^2 - 4\lambda\delta_2})$$  \hspace{1cm} (5.27)

$$\mu_2 = 0.5(\lambda + \delta_1 + \delta_2 - \sqrt{(\lambda + \delta_1 + \delta_2)^2 - 4\lambda\delta_2})$$  \hspace{1cm} (5.28)

$$p = (\lambda - \mu_2)/(\mu_1 - \mu_2)$$  \hspace{1cm} (5.29)

From the p.d.f of a single IPP, we can obtain its c.d.f and the c.d.f $F_e(t)$ readily, then we can compute the c.d.f of the inter-arrival time of superposed process using above formula.

What is the computation complexity to obtain the moments of the inter-arrival times of the superposed process? Using today’s software such as Matlab and Maple, for a few hundreds of independent arrival processes, only a few seconds are needed to obtain the
first three moments and variance exactly. In 1980s, researchers often used approximations to obtain these moments, see for example Whitt [59] and references therein. So the computation complexity can be considered as constant for the given parameters of individual arrival.

What can we do with the p.d.f of the superposed process and its moments? We can approximate the superposed process by moment matching and we discuss this in the following section.

5.5 Approximation of the superposition of $n$ IPPs by an \textit{MMPP}$_2$

Traditionally, the arrival process to a queue is often considered as a renewal process. However, the superposed arrival process of $n$ independent IPP (or \textit{MMPP}$_2$) is not a renewal process since it is correlated. Many studies are based on direct Markov chain approach for the superposition, see Fischer et al. [12]. However, this approach leads to a large number of states. For instance, the superposition of $n$ homogeneous IPPs (\textit{MMPP}$_2$) is a $(n+1)$-state Markov chain and the superposition of $n$ heterogeneous IPPs (\textit{MMPP}$_2$) results to a $2^n$-state Markov chain. It is intractable to solve the problem using direct Markov chain approach for large $n$.

In this section, we approximate the superposed process by an \textit{MMPP}$_2$ which captures the autocorrelation of the superposed arrival processes by matching moments and the index of dispersion counts (IDC).

The first moment ($m_1$) and second moment ($m_2$) of the inter-arrival times of an \textit{MMPP}$_2$ can be obtained by differentiating the Laplace transform of the p.d.f. of the inter-arrival time distribution given by (5.6). We have:

\begin{equation}
 m_1 = \frac{\delta_1 + \delta_2}{\delta_1 \lambda_1 + \delta_2 \lambda_2} \tag{5.30}
\end{equation}

\begin{equation}
 m_2 = \frac{2(\delta_1^2 + \delta_1 \lambda_1 + 2 \delta_1 \delta_2 + \delta_2 \lambda_2 + \delta_2^2)}{(\delta_1 \lambda_2 + \lambda_1 \delta_2 + \lambda_1 \lambda_2)(\lambda_1 \delta_2 + \lambda_2 \delta_1)} \tag{5.31}
\end{equation}
However, the third moment \( (m_3) \)

\[
m_3 = \frac{6(\delta_1^3 + 3\delta_2\delta_1^2 + 2\delta_1^2\lambda_1 + 2\delta_2\delta_1\lambda_2 + 3\delta_1\delta_2^2 + 2\delta_2\delta_1\lambda_1 + \delta_1\lambda_2^2 + \lambda_2^2\delta_2 + 2\delta_2^2\lambda_2 + \delta_1^3)}{\left(\delta_1\lambda_2 + \lambda_1\delta_2 + \lambda_1\lambda_2\right)^2}\]  

(5.32)

is a little complicated to use for moment matching. Another parameter that can be applied is the index of dispersion count (IDC). Considering the process as a counting process, the number of arrivals in an interval of length \( t \) can be obtained from generating functions (see for example [12]). The time-dependent variable IDC\((t)\) is defined as the variance of the number of arrivals in an interval of length \( t \) divided by the mean number of arrivals in \( t \):

\[
IDC(t) = \frac{Var(N_t)}{E(N_t)} \]  

(5.33)

where \( N_t \) indicates the number of arrivals in an interval of length \( t \). The IDC has been so defined in order that its value is 1 for all \( t \) for a Poisson process. Heffes and Lucantoni [18] derived the following formula for the IDC of an \( MMPP_2 \) process:

\[
IDC(t) = 1 + \frac{2\delta_1\delta_2(\lambda_1 - \lambda_2)^2}{((\delta_1 + \delta_2)^2(\lambda_1\delta_2 + \lambda_2\delta_1))} - \frac{2\delta_1\delta_2(\lambda_1 - \lambda_2)^2(1 - exp(-(\delta_1 + \delta_2)t))}{\left(\delta_1 + \delta_2\right)^3(\lambda_1\delta_2 + \lambda_2\delta_1)t} \]  

(5.34)

The asymptote of the IDC is

\[
IDC_\infty = 1 + \frac{2\delta_1\delta_2(\lambda_1 - \lambda_2)^2}{\left(\delta_1 + \delta_2\right)^2(\lambda_1\delta_2 + \lambda_2\delta_1)}, \]  

(5.35)

and it is straightforward to verify that (see Gusella et al. [16])

\[
\frac{IDC_\infty - IDC_{t_0}}{IDC_\infty - 1} = 1 - exp(-(\delta_1 + \delta_2)t_0) \]  

(5.36)

The quantity \( r = \delta_1 + \delta_2 \) can be interpreted as the “rate” at which the IDC approaches its asymptote (see Gusella et al. [16]). This equation can be used to estimate \( r \) for a superposed arrival process (or measured arrival process) since the lefthand side can be easily evaluated from a point at \( t_0 \) on the IDC and the estimated IDC asymptote; \( r \) can then be obtained by solving equation 5.35 numerically. Notice that the time point \( t_0 \) should be chosen carefully (not near to the starting point) to have a better result.

Summarizing, we have four parameters namely the first moment \( m_1 \), the second moment \( m_2 \), the index of dispersion count (IDC) and the rate \( r \). So we can use these four parameters to match a \( MMPP_2 \) with four variables \( (\delta_1, \delta_2, \lambda_1, \lambda_2) \) by solving the four equations. We already discussed how to obtain the first moment, second moment and rate \( r \) from the superposed
arrival processes. Now we will discuss how to obtain the IDC of the superposed arrival processes. For the superposition of \(n\) independent Markov arrival process (for example IPP), the IDC(t) is defined

\[
IDC_n(t) = \frac{\sum_{i=1}^{n} \text{Var}(N_i(t))}{\sum_{i=1}^{n} E(N_i(t))}
\]  

(5.37)

where \(E(N_i(t))\) and \(\text{Var}(N_i(t))\) are the mean number of arrivals and its variance in the given interval for source \(i\). If the individual arrival process is homogeneous and independent, we know that \(IDC_n(t) = IDC(t)\), i.e., IDC of superposed processes will be same as the the single source. This is a nice feature for us to take advantage of when matching the superposed arrival processes to a single \(\text{MMPP}_2\).

5.5.1 When does the superposition of \(n\) IPP Processes become Poisson?

The Poisson process can be seen as a superposition of an infinite number of independent processes, for example, binomial ON/Off processes. It is a well known conjecture that the superposition of a large (or infinite) number of uniformly sparse processes can be approximated by a Poisson process (see Palm [44]). This can be proved to be true for the superposition of infinite number of IPP arrival processes. But the question is that how large the number \(n\) should be for the superposed arrival process to be Poisson? Or if there is error, how can we quantify it? There is some research along these lines (see Whitt [59] and references therein). The basic idea is to obtain the first few moments and variance of the superposed arrival process and then compare to the corresponding ones of the Poisson process. The variance is often used as the indicator of the difference between superposed arrival process and Poisson arrival process. (It’s known that the variance of the inter-arrival times of Poisson process is 1 which is a nice feature to compare). Applying our derived explicit superposition functions from the previous section, we can obtain the variance of the superposed arrival process, \(V_n\) and then compare it to 1. We have:

\[
V_n = \sum_{i=1}^{n} \text{Var}_i = \frac{E[T^2] - (E[T])^2}{(E[T])^2}
\]  

(5.38)

We can defined the absolute difference function as follow:

\[
d = |V_n - 1|
\]  

(5.39)
Then this difference function can also be considered as the error function. For given error
threshold \( d \), we can find an optimal number \( n \) of individual independent arrival processes
which will approximate a Poisson arrival process with absolute error \( d \).

5.6 The autocorrelation function of the superposed arrival
processes

Let us define sequences inter-arrival times (random variables) \( X_1, X_2, ..., X_n, ... \).
The term burstiness has often been used in the literature to characterize the relative vari-
ability of a given traffic when compared to that of a Poisson process. (We know that the
variance divided by the mean number of arrivals in an interval of length \( t \) is 1 for Poisson
process). The term correlation usually refers to the dependence that exists among succes-
sive packet inter-arrival times or to the dependence in the average packet arrival rates in
successive time intervals. To measure the dependence between these random variables, the
autocovariance function (or the autocorrelation function) can be applied.

The superposed arrival process of \( n \) independent arrival process in many cases
(even for homogeneous IPP) will not be renewal process, i.e., the inter-arrival time will not
be independent but correlated. Once we approximate the superposed arrival process as an
\( M M P P_2 \), we can analyze the autocovariance quantitatively using autocovariance function
\( \text{Cov}[k] \) (see [12] for example) as follow:

\[
\text{Cov}[k] = E[(X_1 - E(X_1))(X_{k+1} - E(X_{k+1}))]
\]

\[
\text{Cov}[k] = p(\Lambda - Q)^{-2}\Lambda[(\Lambda - Q)^{-1}\Lambda]^{k-1} - ep(\Lambda - Q)^{-2}\Lambda e = A\sigma^k
\]

where

\[
A = \frac{(\lambda_1 - \lambda_2)^2\delta_1\delta_2}{(\lambda_2\delta_1 + \lambda_1\delta_2)^2(\lambda_1\lambda_2 + \lambda_2\delta_1 + \lambda_1\delta_2)}
\]

\[
\sigma = \frac{\lambda_1\lambda_2}{\lambda_1\lambda_2 + \lambda_2\delta_1 + \lambda_1\delta_2}
\]

We can see that the correlation is null when one of the arrival rates is zero (for
example IPP). The autocorrelation coefficient function of an \( M M P P_2 \) can be computed as
follows:

\[
\rho[k] = \frac{E[X(t)X(t + k)] - m_1E[X(t + k)]}{\sigma^2}
\]
where $E[X]$ is the expectation of the random variable $X$ which can be computed using its definition equation and $(m_1, \sigma)$ is the first moment and variance of inter-arrival times of $MMPP_2$ respectively.

### 5.6.1 Self-similarity nature of the $MMPP_2$ process

It is known that $MMPP_2$ is a good model for the superposition of packet voice streams, see Lazarou et al. [29] and reference therein. The model can be used to capture the high variability of traffic over a range of small time scales. Let $H$ be the Hurst parameter. For general self-similar processes, the Hurst parameter measures the degree of “self-similarity”. If $0.5 < H < 1$, then the process has a long-range dependence (LRD), and if $0 < H \leq 0.5$, then it has a short-range dependence (SRD). $H$ is widely used to capture the intensity of long-range dependence of a random process. The closer $H$ is to 1, the more long-range dependent the traffic is, and vice versa [29]. Lazarou et al. [29] found the following relationship between $H$ and IDC(t) as follows:

$$H_v(t) = \frac{d}{dt} \left( \log(\text{IDC}(t)) \right) + 1$$

where $H_v(t)$ is called index of variability and $d(\log(\text{IDC}(t)))$ is the local slope of the IDC curve at each $t$ when plotted in log-log scale. Note that the index of variability is so defined in order that for a long-range dependent process $H_v(t)=H \in (0.5, 1)$ for $t \geq t_0 \geq 0$. If a process is exactly self-similar, then $H_v(t)=H \in (0.5, 1)$ for all $t$. The index of variability can be thought of as the Hurst parameter defined at each time scale. For $MMPP_2$, the index of variability (or Hurst parameter) can be then obtained easily since IDC(t) is given in equation (5.34). $H_v(t)$ is explicitly given in [29] as

$$H_v(t) = 0.5 \left\{ \frac{A[1 - (1 + rt)e^{-rt}]}{1 + rA} t - A(1 - e^{-rt}) + 1 \right\}$$

where $r = \delta_1 + \delta_2$ and $A=\frac{2\delta_2(\lambda_1 - \lambda_2)^2}{r^2\lambda_0}$. So that

$$H_v(\infty) = 0.5$$

We can then measure the self-similarity nature of $MMPP_2$. 


5.7 Analysis of the \textit{n*IPP/M/1} queue

The \textit{n*IPP/M/1} queue can be analyzed numerically using the matrix-geometric method. In addition, a numerical block-matrix-power approach for the analysis of an \textit{n IPP/M/m + r} model was described in [54], which is exact and efficient for medium size \textit{m + r} (for example less than 200). Since it involves matrix multiplication and inversion, its complexity is at least \(O(mn^3)\) which is computationally expensive for larger size problem.

An alternative way to analyze the \textit{n*IPP/M/1} queue is to approximate the \textit{n*IPP} by an \textit{MMPP}_2 and then analyze the resulting \textit{MMPP}_2/M/1 queue numerically using the matrix-geometric method.

We provide some numerical examples of approximating \textit{n*IPP} by an \textit{MMPP}_2 to a single server queue in the numerical example section.

5.8 The departure process of \textit{MMPP}_2/M/1 queue

In this section, we characterize the departure process of an \textit{MMPP}_2/M/1 queue by finding the explicit Laplace transform of the inter-departure times of the \textit{MMPP}_2/M/1 queue. To the best of our knowledge, such a characterization has not been reported in the literature. Daley [9] developed the Laplace transform of inter-departure times of GI/M/1 queue. Heffes [18] found the Laplace transform of inter-departure times of GI/M/N queue. Bean et al. [5] claimed that whether the departure process of an MMPP/M/1 can be a MAP (or MMPP) is an open problem (see also reference therein). They also conjectured that the departure process of a MAP/PH/1 queue is not a MAP unless the queue is a stationary M/M/1 queue. Yeh et al. [60] introduced a recursive algorithm based on matrix-geometric solution to compute the moments of the inter-departure times of MMPP/D/1 queue with complexity at least \(O(n^{2.5})\). Heindl [19] proposed a numerical approach whereby the output process of an MMPP/G/1/(K) queue was approximated by a semi-Markov model which was then converted to an \textit{MMPP}_2, this approach is computational expensive. Lim et al. [33] gave a general framework for the calculation of the Laplace transform of the inter-departure times for a single server queue with Markov renewal input and general service time distribution (MAP/G/1), which is also based on the computationally expensive matrix-geometric solution. No closed-form solution was given.

Below, we obtain a closed-form solution of the Laplace transform of the departure
process of $MMPP_2/M/1$ queue. We then equate the obtained Laplace transform to the Laplace transform of an $MMPP_2$ from where we obtain the four parameters for $MMPP_2$ $(\delta_1, \delta_2, \lambda_1, \lambda_2)$. We call this method the Laplace transform matching method. The $MMPP_2$ characterization of the departure process from an $MMPP_2/M/1$ queue is approximate, as in the subsequent section, we show that the output process of an $MMPP_2/M/1$ is not an $MMPP_2$ (or MAP) in general. However, through extensive numerical examples, given at the end of this Chapter, we show that the $MMPP_2$ characterization is a very good approximation.

The Laplace transform of the probability density function of the inter-departure time from an $MMPP_2/M/1$ is of the form:

$$D(s) = (1 - \pi_0 e)H(s) + \pi_0 B(s)e$$

where $H(s)$ is the Laplace transform of service time distribution, $e$ is the unity vector, $\pi_0$ was obtained by Lucantoni [34] using the embedded Markov chain approach:

$$\pi_0 = \frac{1 - \rho}{\lambda_m} g(A - Q)$$

$\pi_0$ is defined as the probability that a departure leaves the system behind with no customer and $\rho$ is the mean offered load, $\lambda_m$ is the mean arrival rate and $g$ is defined as $gG = g$ and $ge = 1$, and for $MMPP_2$,

$$G = \begin{bmatrix} 1 - G_1 & G_1 \\ G_2 & 1 - G_2 \end{bmatrix} = \begin{bmatrix} 1 - x & x \\ \frac{\delta_2 x}{\delta_1 + (\lambda_1 - \lambda_2)x} & 1 - \frac{\delta_2 x}{\delta_1 + (\lambda_1 - \lambda_2)x} \end{bmatrix}$$

where $x$ satisfies

$$x = 1 - \frac{\delta_2 x}{\delta_1 + (\lambda_1 - \lambda_2)x} - H(\delta_1 + \delta_2) + \lambda_1 x + \frac{\lambda_2 \delta_2 x}{\delta_1 + (\lambda_1 - \lambda_2)x}$$

For the $MMPP_2$, $G_{ij}$ is the probability that a busy period starting with the $MMPP_2$ in state $i$ ends in state $j$. For $MMPP_2$, we can find $(G_1, G_2)$ explicitly using the definition for $G$ as:

$$G = \int_0^{\infty} e^{(Q - A + AG)t} dt$$

After simplification, we obtain $G_2$ as the smallest positive root of the following equation

$$(\lambda_2 \lambda_1 - \lambda_2^2) y^3 + (\mu(\lambda_1 - \lambda_2) - \lambda_2(\lambda_1 - \delta_1 - 2\delta_2 - \lambda_2) + \lambda_1 \delta_2) y^2 + (-\delta_2(\delta_1 + \mu + \delta_2 + \lambda_1 - 2\lambda_2)) y + \delta_2^2$$

$$= 0$$
and $G_1$ is given by

$$G_1 = \frac{\delta_1 G_2}{\lambda_2 G_2 + \delta_2 - \lambda_1 G_2}$$  \hspace{1cm} (5.54)

We note that there appears to be an error in the expression for $G_1$ in \[12\] where an iterative process for solving $G_1, G_2$ is described. Once $(G_1, G_2)$ are obtained, stationary vector $g$ can be obtained explicitly.

$B(s)$ in expression (5.50) is defined as the Laplace transform of the idle time distribution function matrix where $B_n(t)_{ij}$ is the probability that a departure left the system empty and the arrival process in state $i$, the next departure occurs no later than time $t$ with the arrival process in state $j$, leaving $n$ customers in the system given a departure at time 0. The explicit form for $B(s)$ is obtained using the equation in Lucantoni \[34\]

$$B(s) = [sI - (Q - \Lambda)]^{-1}\Lambda A(s)$$  \hspace{1cm} (5.55)

and

$$A(s) = \int_{0}^{\infty} e^{-st} e^{Qt} h(t) dt$$  \hspace{1cm} (5.56)

where $h(t)$ is the probability density function of service time. For an $MMPP_2$, $e^{Qt}$ can be found through matrix exponential as

$$e^{Qt} = e^{\pi} - \frac{e^{(\delta_1 + \delta_2)t}}{\delta_1 + \delta_2} Q$$  \hspace{1cm} (5.57)

Then $D(s)$ has a closed-form solution for the given $H(s)$ and $B(s)$. We observe that $D(s)$ is not the Laplace transform of an $MMPP_2$ but of mixture of negative exponential distributions. After simplification, $D(s)$ has the following form in general:

$$D(s) = \frac{a s^2 + b s + c}{s^3 + d s^2 + e s + f} = \frac{a_1}{s + \mu_1} + \frac{b_1}{s + \mu_2} + \frac{c_1}{s + \mu_3}$$  \hspace{1cm} (5.58)

where $a, b, c, d, e, f$ is given respectively as follows:

$$a = \mu (1 - \Pi_0 e)$$  \hspace{1cm} (5.59)

$$b = \mu \delta_1 (1 - \pi_0 e) + \delta_2 (1 - \pi_0 e) + \lambda_1 + \lambda_2 - \pi_0 \lambda_1 - \pi_0 \lambda_2$$  \hspace{1cm} (5.60)

$$c = \mu (\lambda_2 \lambda_1 + \lambda_1 \delta_2 + \delta_1 \lambda_2)$$  \hspace{1cm} (5.61)

$$d = \delta_1 + \delta_2 + \lambda_1 + \lambda_2 + \mu$$  \hspace{1cm} (5.62)

$$e = \delta_1 \lambda_2 + \delta_2 \lambda_1 + \lambda_1 \lambda_2$$  \hspace{1cm} (5.63)
\[ f = \mu(\delta_1 \lambda_2 + \delta_2 \lambda_1 + \lambda_1 \lambda_2) \]  

(5.64)

and \((a_1, b_1, c_1)\) can be obtained using partial fraction expansion. For example, \(a_1\) is the value of \((s + \mu_1)D(s)\) for \(s=-\mu_1\).

We note that this is the Laplace transform of a hyper-exponential distribution with three stages. Therefore, the inverse of \(D(s)\), i.e., the p.d.f. of the inter-departure time is as follows:

\[ d(t) = a_1 e^{-\mu_1 t} + b_1 e^{-\mu_2 t} + c_1 e^{-\mu_3 t} \]  

(5.65)

where \((a_1, b_1, -c_1)\) and \((\mu_1, \mu_2, \mu_3)\) are all positive real numbers. We have observed numerically that the term \(c_1 \exp(-\mu_3 t)\) is very small in most cases and therefore we can approximate \(D(s)\) as follows:

\[ D(s) = \frac{a_1}{s + \mu_1} + \frac{b_1}{s + \mu_2} \]  

(5.66)

This is equivalent to the Laplace transform of a hyperexponential distribution with two terms.

Comparing \(D(s)\) to the Laplace transform of the p.d.f. of an \(MMPP_2\), we obtain the following equations

\[ \frac{\delta_2 \lambda_2^2 + \delta_1 \lambda_2^2}{\lambda_1 \delta_2 + \delta_1 \lambda_2} = a, \lambda_1 \lambda_2 + \lambda_1 \delta_2 + \lambda_2 \delta_1 = b, \lambda_1 + \lambda_2 + \delta_1 + \delta_2 = c \]  

(5.67)

Together with a normalized equation for the transition rates \(\delta_1 + \delta_2\), we can find the four parameters \((\lambda_1, \lambda_2, \delta_1, \delta_2)\) for an \(MMPP_2\). If the term \(c_1 \exp(-\mu_3 t)\) is larger than 10% of the total weight, we can use moments matching to find another \(MMPP_2\) to approximate the departure process.

We have compared the analytical results with simulation results obtained by fitting the inter-departure times to an \(MMPP_2\) using an algorithm introduced in Li et al. [32].

As discussed in the numerical section below, we found that the approximate results match the simulation results very well in many different cases. Some typical results are provided in the numerical section.

Notice that the above approximation also works for an \(MMPP/G/1\) queue if the Laplace transform of the service time distribution is known.
5.8.1 The departure process of the $\text{MMPP}_2/D/1$ queue

In the same spirit as the above section, we can find the Laplace transform of the inter-departure time of an $\text{MMPP}_2/G/1$ for a general service time distribution with Laplace transform $H(s)$. For example, in many cases, the service time is considered as constant. A constant service time $d$ can be considered as having Laplace transform $H(s) = \exp(-sd)$ and its p.d.f. as Delta function $\Delta(t - d)$. We obtain $D(s)$ as follows:

1) Using expression for $A(s)$ and $B(s)$ and parameters for $\text{MMPP}_2$, we obtain $b(s) = x_0 B(s) e$ which has following form:

$$b(s) = \frac{e^{-sd(a_2s + b_2)}}{(s^2 + s\delta_2 + s\lambda_2 + s\delta_1 + \delta_1\lambda_2 + \lambda_1 s + \lambda_1\delta_2 + \lambda_1\lambda_2)} \quad (5.68)$$

2) Using equation for $D(s)$ and value for $\pi_0$, we can obtain $D(s)$ which has following form in general:

$$D(s) = \frac{e^{-sd(a_2s^2 + b_2s + c_2)}}{(s^2 + s\delta_2 + s\lambda_2 + s\delta_1 + \delta_1\lambda_2 + \lambda_1 s + \lambda_1\delta_2 + \lambda_1\lambda_2)} \quad (5.69)$$

which can be simplified as

$$D(s) = \frac{e^{-sd(a_2s^2 + b_2s + c_2)}}{s^3 + d_2s^2 + e_2s + f_2} \quad (5.70)$$

In general, for $\text{MMPP}_2/G/1$ queue, $D(s)$ will be the following form

$$D(s) = \frac{H(s)(a_2s^2 + b_2s + c_2)}{(s^2 + s\delta_2 + s\lambda_2 + s\delta_1 + \delta_1\lambda_2 + \lambda_1 s + \lambda_1\delta_2 + \lambda_1\lambda_2)} \quad (5.71)$$

where $(a_2, b_2, c_2)$ depend on $\pi_0$ and four parameters of the $\text{MMPP}_2$.

5.9 Dimensioning approach of an ADSL access network

An ADSL access network can be modeled by the queueing network, shown in FIG. 5.2. The queueing network represents the upstream traffic (i.e. from the subscribers to the Internet) and it consists of three levels of statistical multiplexers. The first level consists of a number of multiplexers each representing a DSLAM serving a number of subscribers. The second level consists of multiplexers each representing a metro Ethernet switch serving a
number of DSLAMs. The third level consists of a single multiplexer representing the BRAS serving all the metro Ethernet switches. The dimensioning of this access network is done by dimensioning of each multiplexer in isolation as follows:

1) Each subscriber is represented by an identical IPP. In view of this, a DSLAM multiplexer is modeled as $n^*IPP/M/1$ queue, where the packet transmission time is assumed exponentially distributed. The departure process from each DSLAM multiplexer is approximated by an $MMP_2$.

2) A metro Ethernet switch multiplexer is modeled as an $k^*MMP_2/M/1$ queue, where $k$ is the number of DSLAMs served by a metro Ethernet switch. The superposition of $k$ $MMP_2$ sources is approximated by a single $MMP_2$ so that an Ethernet switch multiplexer is analyzed as an $MMP_2/M/1$ queue, whose departure process is also approximated by an $MMP_2$.

3) The BRAS multiplexer is analyzed in the same way as a metro Ethernet switch multiplexer. Since all the multiplexers are modeled as a single server queue with an $MMP_2$ arrival, classic dimensioning expressions for $MMP_2/M/1$ can be applied. These expressions are summarized below.

### 5.9.1 Mean waiting time

A closed-form solution for mean waiting time of MMPP/G/1 queue is provided in Fischer et al. [12]. We can use it to dimension the speed of the multiplexer's outgoing link to meet the mean waiting time requirement. We introduced the dimensioning formula in the previous Chapter in equation (4.22) to (4.24).

### 5.9.2 Equivalent bandwidth

The equivalent bandwidth can be used to calculate the required bandwidth of the multiplexer's outgoing link. A commonly used expression for the equivalent bandwidth (see [45]) is based on the assumption that the source is an interrupted fluid process (IFP) characterized by the triplet $(R, r, b)$, where $R$ is its peak bit rate, $r$ the fraction of time the source is active, defined as the ratio of the mean length of the on period divided by the
sum of the mean on and off periods, and $b$ is the mean duration of the on period. Let us assume that the source feeds a finite-capacity queue with a constant service time, and let $K$ be the size of the queue expressed in bits. The service time is equal to the time it takes to transmit out a cell. Then equivalent bandwidth $e$ is given by the expression:

$$e = \frac{a - K + \sqrt{(a - K)^2 + 4Kar}}{2a} R \quad (5.72)$$

where $a = b(1 - r)Rln(1/\epsilon)$ and $\epsilon$ is the packet loss probability. For given parameters $(\delta_1, \delta_2, \lambda)$ of IFP (or IPP), $r = \delta_1/(\delta_1 + \delta_2)$. The method of simply adding up the equivalent bandwidth requested by each connection may lead to under-utilization of the link, i.e., more bandwidth may be allocated for all the connections than it is necessary. The following approximation for the equivalent bandwidth of $n$ sources corrects the over-allocation.

Figure 5.2: A typical dimensioning example of an ADSL access network
problem:

\[ c_{\text{min}} = \min \{ \rho + \sigma R \sqrt{-2 \ln(\epsilon)} - \ln(2\pi), \sum_{i=1}^{n} e_i \} \]  

(5.73)

where \( \rho \) is the average bit rate of all the sources, \( e_i \) is the equivalent bandwidth of the \( i \)th source, calculated using expression above, and \( \sigma \) is the sum of the standard deviation of the bit rate of all the sources, and it is equal to

\[ \sigma = \sum_{i=1}^{n} \sqrt{r_i(R_i - r_i)} \]  

(5.74)

These expressions can be used to dimension the upstream link of DSLAM (i.e. from DSLAM to metro Ethernet switch).

As for the second level of multiplexer, its departure process is the arrival process to the metro Ethernet switch. In view of this, the arrival process to a multiplexer representing metro Ethernet switch is the superposition of \( k \ MMP P_2 \) where \( k \) is the number of DSLAM linked to it. This superposition is approximated by a single \( MMP P_2 \), so that the multiplexer can be analyzed as an \( MMP P_2/M/1/K \) queue. The upstream link (i.e., from the metro Ethernet switch to BRAS) can be dimensioned using the following approach, see Schwartz [52] and reference therein.

For an \( MMP P_2/M/1/K \) queue, let the blocking probability requirement be \( P_L \), and set \( \theta = \frac{1}{K} \ln(1/P_L) \), then the required link capacity, referred to as the effective capacity, is given by the follow equation

\[ C_e = \frac{1}{\theta} z(Q + (e^\theta - 1)\Lambda)R \]  

(5.75)

where \( z(A) \) is the largest eigenvalue of matrix \( A \). For the single \( MMP P_2 \) arrival process, it is reported in [52] that the effective capacity is very accurate comparing to the exact solution. If there are \( n \) independent \( MMP P_2 \) (or IPP), the effective capacity has additive property, i.e., the total effective capacity is the sum of the individual effective capacity. This is very helpful for capacity dimensioning though it is considered to give conservative estimates in some cases.

The third level multiplexer can be dimensioned following the same approach as in the second level multiplexer.

It is well known (from many numerical examples) that the dimensioning results obtained using equivalent bandwidth approach are conservative. However, since in most cases,
we are considering dimensioning a network to meet 3-5 years traffic growth, conservative results may be acceptable.

5.10 Numerical examples

In this section, we provide some numerical examples to validate our proposed approach on approximating the superposition of \( n \) IPP arrivals and approximating the departure process of the \( MMPP_2/M/1 \) process.

We observed that the queueing performance such as the mean number in the system and response time can be very different for IPP and \( MMPP_2 \) arrival process, even if IPP has the same moments of the inter-arrival time as the \( MMPP_2 \). This is because the IPP and \( MMPP_2 \) have different auto-variance functions (The IPP has an auto-variance equal to zero) which will cause the queueing performance to be different.

Example 1: In this example, we consider an \( n \times IPP/M/1 \) queue with \( n = 10 \) and service rate \( \mu \) is changing so that the offered load of the single server queue is changing from 0.1 to 0.9. The single IPP has parameters \((\delta_1, \delta_2, \lambda) = (0.9, 0.1, 1.0)\) which makes it very bursty. We analyze this queue exactly using matrix-geometric procedure. We also approximate the superposition of \( n \times IPP \) by an \( MMPP_2 \) using our proposed approach and subsequently analyze this queue as an \( MMPP_2/M/1 \) queue using the matrix-geometric solution. The \( MMPP_2 \) has parameters \((r_1, r_2, \lambda_1, \lambda_2) = (0.2712, 0.7288, 9.5786, 7.4446)\). The error is the relative error in percentage computed by \((-\text{exact}-\text{approximation})/\text{exact} \times 100\%\). The results are given in Table 5.1. We observe that the exact results and approximation results match very well.

Example 2: This is the same example as above, only \( n = 20 \). The results are given in Table 5.2. The equivalent \( MMPP_2 \) has parameters \((r_1, r_2, \lambda_1, \lambda_2) = (0.3271, 0.6729, 18.9355, 16.0759)\).

Example 3: same as in example 1, with \( n = 100 \). The equivalent \( MMPP_2 \) has parameters \((r_1, r_2, \lambda_1, \lambda_2) = (0.4180, 0.5820, 92.5426, 86.4603)\).

The results are given in Table 5.3.

We observed that:

1). The auto-covariance is \((0.0236, 0.0166, 0.6852e-003)\) for \( n = (10, 20, 100) \) respectively. That is, the auto-covariance is decreasing when \( n \) is increasing. This is consistent
Table 5.1: Comparison of the mean waiting time $W$ and the mean number of customers $N$ in the system obtained by exact and $MMPP_2$ approximation for $n=10$

<table>
<thead>
<tr>
<th>$\rho$</th>
<th>$W$ (Exact)</th>
<th>$W$ ($MMPP_2$)</th>
<th>Error(%)</th>
<th>$N$ (Exact)</th>
<th>$N$ ($MMPP_2$)</th>
<th>Error(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.0230</td>
<td>0.0225</td>
<td>2.17</td>
<td>0.1230</td>
<td>0.1225</td>
<td>0.41</td>
</tr>
<tr>
<td>0.2</td>
<td>0.1101</td>
<td>0.1063</td>
<td>3.45</td>
<td>0.3101</td>
<td>0.3063</td>
<td>1.23</td>
</tr>
<tr>
<td>0.3</td>
<td>0.2896</td>
<td>0.2805</td>
<td>3.14</td>
<td>0.5896</td>
<td>0.5805</td>
<td>1.54</td>
</tr>
<tr>
<td>0.4</td>
<td>0.5975</td>
<td>0.5841</td>
<td>2.24</td>
<td>0.9975</td>
<td>0.9841</td>
<td>1.34</td>
</tr>
<tr>
<td>0.5</td>
<td>1.0976</td>
<td>1.0819</td>
<td>1.43</td>
<td>1.5976</td>
<td>1.5819</td>
<td>0.98</td>
</tr>
<tr>
<td>0.6</td>
<td>1.9225</td>
<td>1.9061</td>
<td>0.85</td>
<td>2.5225</td>
<td>2.5061</td>
<td>0.65</td>
</tr>
<tr>
<td>0.7</td>
<td>3.3863</td>
<td>3.3702</td>
<td>0.48</td>
<td>4.0863</td>
<td>4.0702</td>
<td>0.39</td>
</tr>
<tr>
<td>0.8</td>
<td>6.4375</td>
<td>6.4222</td>
<td>0.24</td>
<td>7.2375</td>
<td>7.2222</td>
<td>0.21</td>
</tr>
<tr>
<td>0.9</td>
<td>15.825</td>
<td>15.8108</td>
<td>0.09</td>
<td>16.725</td>
<td>16.7108</td>
<td>0.08</td>
</tr>
</tbody>
</table>

Table 5.2: Comparison of the mean waiting time $W$ and the mean number $N$ of customers in the system obtained by exact and $MMPP_2$ approximation for $n=20$

<table>
<thead>
<tr>
<th>$\rho$</th>
<th>$W$ (Exact)</th>
<th>$W$ ($MMPP_2$)</th>
<th>Error(%)</th>
<th>$N$ (Exact)</th>
<th>$N$ ($MMPP_2$)</th>
<th>Error(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.0086</td>
<td>0.0085</td>
<td>0.26</td>
<td>0.1172</td>
<td>0.1169</td>
<td>1.16</td>
</tr>
<tr>
<td>0.2</td>
<td>0.0416</td>
<td>0.0403</td>
<td>0.99</td>
<td>0.2833</td>
<td>0.2805</td>
<td>3.12</td>
</tr>
<tr>
<td>0.3</td>
<td>0.1143</td>
<td>0.1093</td>
<td>0.89</td>
<td>0.5286</td>
<td>0.5186</td>
<td>4.37</td>
</tr>
<tr>
<td>0.4</td>
<td>0.2485</td>
<td>0.2379</td>
<td>2.37</td>
<td>0.8971</td>
<td>0.8758</td>
<td>4.27</td>
</tr>
<tr>
<td>0.5</td>
<td>0.4794</td>
<td>0.4630</td>
<td>2.25</td>
<td>1.4588</td>
<td>1.4260</td>
<td>3.42</td>
</tr>
<tr>
<td>0.6</td>
<td>0.8745</td>
<td>0.8538</td>
<td>1.76</td>
<td>2.3491</td>
<td>2.3077</td>
<td>2.37</td>
</tr>
<tr>
<td>0.7</td>
<td>1.5914</td>
<td>1.5684</td>
<td>1.18</td>
<td>3.8828</td>
<td>3.8368</td>
<td>1.45</td>
</tr>
<tr>
<td>0.8</td>
<td>3.1041</td>
<td>3.0804</td>
<td>0.67</td>
<td>7.0082</td>
<td>6.9609</td>
<td>0.76</td>
</tr>
<tr>
<td>0.9</td>
<td>7.7868</td>
<td>7.7637</td>
<td>0.28</td>
<td>16.4736</td>
<td>16.4274</td>
<td>0.30</td>
</tr>
</tbody>
</table>

Table 5.3: Comparison of the mean waiting time $W$ and mean number $N$ of customers in the system obtained by exact and $MMPP_2$ approximation for $n=100$

<table>
<thead>
<tr>
<th>$\rho$</th>
<th>$W$ (Exact)</th>
<th>$W$ ($MMPP_2$)</th>
<th>Error(%)</th>
<th>$N$ (Exact)</th>
<th>$N$ ($MMPP_2$)</th>
<th>Error(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.0012</td>
<td>0.0012</td>
<td>0.00</td>
<td>0.1123</td>
<td>0.1123</td>
<td>0.00</td>
</tr>
<tr>
<td>0.2</td>
<td>0.0057</td>
<td>0.0057</td>
<td>0.00</td>
<td>0.257</td>
<td>0.257</td>
<td>0.12</td>
</tr>
<tr>
<td>0.3</td>
<td>0.0152</td>
<td>0.015</td>
<td>0.00</td>
<td>0.452</td>
<td>0.4502</td>
<td>0.40</td>
</tr>
<tr>
<td>0.4</td>
<td>0.0331</td>
<td>0.0324</td>
<td>0.00</td>
<td>0.7313</td>
<td>0.724</td>
<td>1.00</td>
</tr>
<tr>
<td>0.5</td>
<td>0.0663</td>
<td>0.064</td>
<td>0.00</td>
<td>1.1627</td>
<td>1.1403</td>
<td>1.93</td>
</tr>
<tr>
<td>0.6</td>
<td>0.1288</td>
<td>0.1235</td>
<td>0.00</td>
<td>1.8875</td>
<td>1.8345</td>
<td>2.81</td>
</tr>
<tr>
<td>0.7</td>
<td>0.2533</td>
<td>0.2436</td>
<td>0.00</td>
<td>3.2331</td>
<td>3.1364</td>
<td>2.99</td>
</tr>
<tr>
<td>0.8</td>
<td>0.5362</td>
<td>0.5226</td>
<td>0.00</td>
<td>6.1616</td>
<td>6.026</td>
<td>2.20</td>
</tr>
<tr>
<td>0.9</td>
<td>1.4533</td>
<td>1.4386</td>
<td>0.00</td>
<td>15.4328</td>
<td>15.2855</td>
<td>0.95</td>
</tr>
</tbody>
</table>
with the known results that superposition of infinite (or very large) number of IPPs will be a Poisson process which has auto-covariance equal to zero. We actually can find a $n$ so that the superposition of $n$ IPP will be close to a Poisson process.

2). The queueing performance (the mean waiting time and number in the system) is decreasing when $n$ is increasing and other parameters are the same. This means the auto-covariance affects/dominates the queueing performance

3). For the same $n$, when the offered load is increasing, the mean waiting time and number in the system are both increasing; this is consistent with theoretical results.

**Example 4:** In this example, we show the results of the superposition of $n$ homogeneous $MMPP_2$. We use the single $MMPP_2$ with parameters $(r_1, r_2, \lambda_1, \lambda_2)=(0.9, 0.1, 10, 1)$.

We obtain the infinitesimal generater $Q$ and $\Lambda$. Using matrix-geometric approach, we found exact solution for the $n * MMPP_2/M/1$ queue for $n=20$. Through moments and IDC, we obtained an $MMPP_2$ with parameters $(0.6966, .3034, 56.3226, 30.0507)$.

Using the $MMPP_2$ as the input to an $MMPP_2/M/1$ queue, we compute the mean waiting time and mean number of customers in the system. The exact solution (which is obtained by simulating two tandem $MMPP_2/M/1$ queues) and the approximate results are shown in Table 5.4.

We observe that the approximate $MMPP_2$ slightly overestimates the mean number and mean waiting times in the system in many cases.

**Example 5:** In this example, we study the accuracy of the analytical results and simulation results for the departure process of an $MMPP_2/M/1$ queue. We do this as

---

**Table 5.4:** Comparison of exact and approximate results of the superposition of 20 homogeneous $MMPP_2$

<table>
<thead>
<tr>
<th>$\rho$</th>
<th>$n$(Exact)</th>
<th>$n$(Appr)</th>
<th>Error(%)</th>
<th>$W$(Exact)</th>
<th>$W$(Appr)</th>
<th>Error(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.1125</td>
<td>0.1125</td>
<td>0.00</td>
<td>0.0003</td>
<td>0.0003</td>
<td>0.00</td>
</tr>
<tr>
<td>0.2</td>
<td>0.2584</td>
<td>0.2584</td>
<td>0.00</td>
<td>0.0015</td>
<td>0.0015</td>
<td>0.00</td>
</tr>
<tr>
<td>0.3</td>
<td>0.4585</td>
<td>0.4584</td>
<td>0.02</td>
<td>0.0042</td>
<td>0.0042</td>
<td>0.00</td>
</tr>
<tr>
<td>0.4</td>
<td>0.7574</td>
<td>0.7533</td>
<td>0.54</td>
<td>0.0095</td>
<td>0.0094</td>
<td>1.05</td>
</tr>
<tr>
<td>0.5</td>
<td>1.2993</td>
<td>1.2736</td>
<td>1.98</td>
<td>0.0203</td>
<td>0.021</td>
<td>3.33</td>
</tr>
<tr>
<td>0.6</td>
<td>2.385</td>
<td>2.4284</td>
<td>1.79</td>
<td>0.0481</td>
<td>0.0469</td>
<td>2.49</td>
</tr>
<tr>
<td>0.7</td>
<td>5.0443</td>
<td>5.2608</td>
<td>4.29</td>
<td>0.1143</td>
<td>0.12</td>
<td>4.99</td>
</tr>
<tr>
<td>0.8</td>
<td>11.6971</td>
<td>12.6715</td>
<td>8.33</td>
<td>0.2868</td>
<td>0.3122</td>
<td>8.86</td>
</tr>
<tr>
<td>0.9</td>
<td>34.5901</td>
<td>36.5382</td>
<td>5.63</td>
<td>0.8866</td>
<td>0.9373</td>
<td>5.72</td>
</tr>
</tbody>
</table>

---
Table 5.5: Comparison of analytical and simulation results of the departure process of $MMPP_2/M/1$

<table>
<thead>
<tr>
<th>$\rho$</th>
<th>Output (Ana)</th>
<th>Output (Sim)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$[\delta_1, \delta_1, \lambda_1, \lambda_2]$</td>
<td>$[\delta_1, \delta_1, \lambda_1, \lambda_2]$</td>
</tr>
<tr>
<td>0.2</td>
<td>[0.9425, 0.0575, 4.9307, 0.9873]</td>
<td>[0.9219, 0.0781, 4.9013, 1.0167]</td>
</tr>
<tr>
<td>0.3</td>
<td>[0.8797, 0.1203, 3.0128, 0.9386]</td>
<td>[0.8649, 0.1351, 3.0059, 0.9456]</td>
</tr>
<tr>
<td>0.4</td>
<td>[0.7442, 0.2558, 2.0996, 0.8684]</td>
<td>[0.7305, 0.2696, 2.0993, 0.8687]</td>
</tr>
<tr>
<td>0.5</td>
<td>[0.5084, 0.4916, 1.6267, 0.7513]</td>
<td>[0.4979, 0.5022, 1.6288, 0.7492]</td>
</tr>
<tr>
<td>0.6</td>
<td>[0.2664, 0.7336, 1.3960, 0.5887]</td>
<td>[0.2612, 0.7388, 1.3983, 0.5864]</td>
</tr>
<tr>
<td>0.7</td>
<td>[0.8745, 0.1255, 0.4169, 1.2869]</td>
<td>[0.8790, 0.1210, 0.4145, 1.2893]</td>
</tr>
<tr>
<td>0.8</td>
<td>[0.9456, 0.0544, 0.2614, 1.2316]</td>
<td>[0.9474, 0.0526, 0.2599, 1.2331]</td>
</tr>
<tr>
<td>0.9</td>
<td>[0.9806, 0.0194, 0.1292, 1.2000]</td>
<td>[0.9814, 0.0186, 0.1281, 1.2011]</td>
</tr>
</tbody>
</table>

follows. We consider a tandem queueing network with two nodes. A very bursty $MMPP_2$ arrival process with parameters ($\delta_1=0.98$, $\delta_2=0.02$, $\lambda_1=10$, $\lambda_2=1.0$) is considered as the input to the first node (single server with infinite buffer size and FIFO). We change the service rate $\mu$ so that the offered load is varying from 0.2 to 0.9. The departure process from the first node is fed into the second node which is a single server infinite capacity queue, with exponentially distributed service time with a rate of $\mu$. Our validation includes two parts. First, we compare the departure process from the first node as characterized by an $MMPP_2$ using our proposed approach against simulation results obtained using an algorithm to fit an $MMPP_2$ due to in Li et al. [32] and reference therein. Secondly, we compare the mean number of customers and the mean waiting time in the second node obtained by simulation and by solving numerically the second node as an $MMPP_2/M/1$ queue, where $MMPP_2$ is the departure process from the first node as characterized above. The results are presented below. $O(Sim)$ and $O(Ana)$ with parameters ($\delta_1$, $\delta_2$, $\lambda_1$, $\lambda_2$) are the departure processes from simulation and analytical results respectively. $N(Sim)$ and $N(Ana)$ are the mean number of customers in the system from simulation and analytical results respectively, and $T(Sim)$, $T(Ana)$ are mean response times from simulation and analytical results respectively.

Example 6: We also compute the auto-covariance function (ACF) for the departure process of an $MMPP_2/M/1$ queue. The analytical and simulation results are shown in Table 5.6.

Auto-covariance function (ACF) for an $MMPP_2$ is given in Fischer et al. [12] as

$$ACF_{MMPP_2}(k) = p(\Lambda - Q)^{-2}\Lambda(((\Lambda - Q)^{-1}\Lambda)^{k-1} - ep)(\Lambda - Q)^{-2}\Lambda e; \quad (5.76)$$
Table 5.6: Comparison of ACFs from analytical and simulation results of the departure process of an $MMPP_2/M/1$

<table>
<thead>
<tr>
<th>$\rho$</th>
<th>$ACF_1$(Ana)</th>
<th>$ACF_1$(Sim)</th>
<th>$ACF_3$(Ana)</th>
<th>$ACF_3$(Sim)</th>
<th>IDC(Ana)</th>
<th>IDC(Sim)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>0.0752</td>
<td>0.0782</td>
<td>0.0482</td>
<td>0.0489</td>
<td>2.3880</td>
<td>2.6458</td>
</tr>
<tr>
<td>0.3</td>
<td>0.0566</td>
<td>0.0569</td>
<td>0.0280</td>
<td>0.0278</td>
<td>1.7665</td>
<td>1.8104</td>
</tr>
<tr>
<td>0.4</td>
<td>0.0416</td>
<td>0.0413</td>
<td>0.0153</td>
<td>0.0150</td>
<td>1.4877</td>
<td>1.4968</td>
</tr>
<tr>
<td>0.5</td>
<td>0.0290</td>
<td>0.0286</td>
<td>0.0075</td>
<td>0.0073</td>
<td>1.3241</td>
<td>1.3248</td>
</tr>
<tr>
<td>0.6</td>
<td>0.0187</td>
<td>0.0184</td>
<td>0.0032</td>
<td>0.0031</td>
<td>1.2157</td>
<td>1.2145</td>
</tr>
<tr>
<td>0.7</td>
<td>0.0109</td>
<td>0.0105</td>
<td>0.0011</td>
<td>0.0010</td>
<td>1.1411</td>
<td>1.1375</td>
</tr>
<tr>
<td>0.8</td>
<td>0.0050</td>
<td>0.0048</td>
<td>2.2902e-004</td>
<td>2.1833e-004</td>
<td>1.0821</td>
<td>1.0799</td>
</tr>
<tr>
<td>0.9</td>
<td>0.0014</td>
<td>0.0013</td>
<td>1.8425e-005</td>
<td>1.7290e-005</td>
<td>1.0356</td>
<td>1.0369</td>
</tr>
</tbody>
</table>

Table 5.7: Comparison of exact and approximate (appr) results of the departure process of an $MMPP_2/M/1$ for low burstiness $H=0.54$, $MMPP_2(0.5,0.5,1,2)$

<table>
<thead>
<tr>
<th>$\rho$</th>
<th>$m_1$(exact)</th>
<th>$m_2$(exact)</th>
<th>$m_3$(exact)</th>
<th>$m_1$(appr)</th>
<th>$m_2$(appr)</th>
<th>$m_3$(appr)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>0.8475</td>
<td>1.6174</td>
<td>1.2521</td>
<td>0.8238</td>
<td>1.5453</td>
<td>1.2770</td>
</tr>
<tr>
<td>0.3</td>
<td>0.8475</td>
<td>1.5955</td>
<td>1.2215</td>
<td>0.8416</td>
<td>1.5773</td>
<td>1.2268</td>
</tr>
<tr>
<td>0.4</td>
<td>0.8475</td>
<td>1.5730</td>
<td>1.1902</td>
<td>0.8451</td>
<td>1.5654</td>
<td>1.1920</td>
</tr>
<tr>
<td>0.5</td>
<td>0.8475</td>
<td>1.5503</td>
<td>1.1587</td>
<td>0.8463</td>
<td>1.5465</td>
<td>1.1593</td>
</tr>
<tr>
<td>0.6</td>
<td>0.8475</td>
<td>1.5276</td>
<td>1.1270</td>
<td>0.8468</td>
<td>1.5252</td>
<td>1.1272</td>
</tr>
<tr>
<td>0.7</td>
<td>0.8475</td>
<td>1.5048</td>
<td>1.0953</td>
<td>0.8491</td>
<td>1.5119</td>
<td>1.0969</td>
</tr>
<tr>
<td>0.8</td>
<td>0.8475</td>
<td>1.4820</td>
<td>1.0635</td>
<td>0.8483</td>
<td>1.4855</td>
<td>1.0645</td>
</tr>
<tr>
<td>0.9</td>
<td>0.8471</td>
<td>1.4578</td>
<td>1.0316</td>
<td>0.8480</td>
<td>1.4617</td>
<td>1.0326</td>
</tr>
</tbody>
</table>

And the $IDC(IDC_\infty)$ for an $MMPP_2$ is given in equation (5.34).

Example 7: Comparison of exact and approximate (appr) results of the departure process of an $MMPP_2/M/1$ for low burstiness $H=0.54$. Table 5.7 gives a comparison of the first three moments of the approximation $MMPP_2$ of the departure process (appr.) against exact data (from Laplace transform). The parameters of the input $MMPP_2$ are (0.5, 0.5, 1, 2). The approximation results match the exact results very well with a maximum relative error of less than a few percent points.

Example 8: A similar comparison as in the above example is given in Table 5.8. for medium burstness $H=0.73$. The parameters of the input $MMPP_2$ are (0.9, 0.1, 10, 1). The approximation results match exact results very well with maximum relative error of less than a few percent points.

Example 9: A similar comparison to example 7 and 8 for high burstiness $H=0.91$
Table 5.8: Comparison of analytical and simulation (appr) results of the departure process of $\text{MMPP}_2/\text{M}/1$ for medium burstness $H=0.73$, $\text{MMPP}_2(0.9, 0.1, 10, 1)$

<table>
<thead>
<tr>
<th>$\rho$</th>
<th>$m_1$ (exact)</th>
<th>$m_2$ (exact)</th>
<th>$m_3$ (exact)</th>
<th>$m_1$ (appr)</th>
<th>$m_2$ (appr)</th>
<th>$m_3$ (appr)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.5263</td>
<td>0.8901</td>
<td>2.4269</td>
<td>0.5289</td>
<td>0.8904</td>
<td>2.4270</td>
</tr>
<tr>
<td>0.2</td>
<td>0.5263</td>
<td>0.8709</td>
<td>2.3623</td>
<td>0.5748</td>
<td>0.8798</td>
<td>2.3647</td>
</tr>
<tr>
<td>0.3</td>
<td>0.5263</td>
<td>0.8365</td>
<td>2.2382</td>
<td>0.5307</td>
<td>0.8373</td>
<td>2.2385</td>
</tr>
<tr>
<td>0.4</td>
<td>0.5263</td>
<td>0.7977</td>
<td>2.0875</td>
<td>0.5277</td>
<td>0.7980</td>
<td>2.0876</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5263</td>
<td>0.7578</td>
<td>1.9199</td>
<td>0.5269</td>
<td>0.7579</td>
<td>1.9199</td>
</tr>
<tr>
<td>0.6</td>
<td>0.5263</td>
<td>0.7174</td>
<td>1.7378</td>
<td>0.5263</td>
<td>0.7173</td>
<td>1.7378</td>
</tr>
<tr>
<td>0.7</td>
<td>0.5263</td>
<td>0.6766</td>
<td>1.5420</td>
<td>0.5265</td>
<td>0.6767</td>
<td>1.5420</td>
</tr>
<tr>
<td>0.8</td>
<td>0.5263</td>
<td>0.6358</td>
<td>1.3328</td>
<td>0.5264</td>
<td>0.6358</td>
<td>1.3328</td>
</tr>
<tr>
<td>0.9</td>
<td>0.5263</td>
<td>0.5949</td>
<td>1.1103</td>
<td>0.5263</td>
<td>0.5949</td>
<td>1.1103</td>
</tr>
</tbody>
</table>

Table 5.9: Comparison of exact and approximate (appr) results of the departure process of an $\text{MMPP}_2/\text{M}/1$ for high burstness $H=0.91$, $\text{MMPP}_2(0.09, 0.01, 10, 1)$

<table>
<thead>
<tr>
<th>$\rho$</th>
<th>$m_1$ (exact)</th>
<th>$m_2$ (exact)</th>
<th>$m_3$ (exact)</th>
<th>$m_1$ (appr)</th>
<th>$m_2$ (appr)</th>
<th>$m_3$ (appr)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.5263</td>
<td>0.9499</td>
<td>2.7942</td>
<td>0.5266</td>
<td>0.9499</td>
<td>2.7943</td>
</tr>
<tr>
<td>0.2</td>
<td>0.5263</td>
<td>0.9390</td>
<td>2.7551</td>
<td>0.5263</td>
<td>0.9504</td>
<td>2.7958</td>
</tr>
<tr>
<td>0.3</td>
<td>0.5263</td>
<td>0.8947</td>
<td>2.5875</td>
<td>0.5274</td>
<td>0.8949</td>
<td>2.5876</td>
</tr>
<tr>
<td>0.4</td>
<td>0.5263</td>
<td>0.8465</td>
<td>2.3907</td>
<td>0.5266</td>
<td>0.8465</td>
<td>2.3907</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5263</td>
<td>0.7979</td>
<td>2.1770</td>
<td>0.5264</td>
<td>0.7979</td>
<td>2.1771</td>
</tr>
<tr>
<td>0.6</td>
<td>0.5263</td>
<td>0.7492</td>
<td>1.9476</td>
<td>0.5264</td>
<td>0.7492</td>
<td>1.9477</td>
</tr>
<tr>
<td>0.7</td>
<td>0.5263</td>
<td>0.7004</td>
<td>1.7027</td>
<td>0.5263</td>
<td>0.7004</td>
<td>1.7027</td>
</tr>
<tr>
<td>0.8</td>
<td>0.5263</td>
<td>0.6516</td>
<td>1.4421</td>
<td>0.5263</td>
<td>0.6516</td>
<td>1.4421</td>
</tr>
<tr>
<td>0.9</td>
<td>0.5263</td>
<td>0.6028</td>
<td>1.1662</td>
<td>0.5263</td>
<td>0.6028</td>
<td>1.1662</td>
</tr>
</tbody>
</table>

is given in Table 5.9. The parameters of the input $\text{MMPP}_2$ are $(0.09, 0.01, 10, 1)$. The approximation results match the exact results very well with a maximum relative error of less than 2 percent.

The squared coefficient of variation $C^2$ (equals to the variance divided by the squared mean) can be used as an alternative indicator of burstiness.

Example 10: Table 5.10 gives a comparison of the exact and approximate results of the departure process of an $\text{MMPP}_2/\text{M}/1$ for low burstness $C^2=1.1429$.

Example 11: Table 5.11 gives a comparison of the exact and approximate results of the departure process of an $\text{MMPP}_2/\text{M}/1$ for medium burstness $C^2=15.6341$.

Example 12: Table 5.12 gives a comparison of the exact and approximate results
Table 5.10: Comparison of exact and approximate results of the departure process of an MMPP\(_2\)/M/1 for low burstness \(C^2=1.1429\), MMPP\(_2\) (0.1, 10.1, 1.1, 1.1)

<table>
<thead>
<tr>
<th>(\rho)</th>
<th>(m_1) (exact)</th>
<th>(m_2) (exact)</th>
<th>(m_3) (exact)</th>
<th>(m_1) (appr)</th>
<th>(m_2) (appr)</th>
<th>(m_3) (appr)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.1961</td>
<td>0.2333</td>
<td>0.6133</td>
<td>0.1961</td>
<td>0.2333</td>
<td>0.6133</td>
</tr>
<tr>
<td>0.2</td>
<td>0.1961</td>
<td>0.2303</td>
<td>0.6038</td>
<td>0.1961</td>
<td>0.2304</td>
<td>0.6038</td>
</tr>
<tr>
<td>0.3</td>
<td>0.1961</td>
<td>0.2240</td>
<td>0.5833</td>
<td>0.1961</td>
<td>0.2254</td>
<td>0.5835</td>
</tr>
<tr>
<td>0.4</td>
<td>0.1961</td>
<td>0.2134</td>
<td>0.5480</td>
<td>0.1961</td>
<td>0.2293</td>
<td>0.5517</td>
</tr>
<tr>
<td>0.5</td>
<td>0.1961</td>
<td>0.1968</td>
<td>0.4915</td>
<td>0.1961</td>
<td>0.2190</td>
<td>0.4973</td>
</tr>
<tr>
<td>0.6</td>
<td>0.1961</td>
<td>0.1768</td>
<td>0.4213</td>
<td>0.1961</td>
<td>0.1814</td>
<td>0.4226</td>
</tr>
<tr>
<td>0.7</td>
<td>0.1961</td>
<td>0.1539</td>
<td>0.3388</td>
<td>0.1961</td>
<td>0.1555</td>
<td>0.3392</td>
</tr>
<tr>
<td>0.8</td>
<td>0.1961</td>
<td>0.1292</td>
<td>0.2475</td>
<td>0.1961</td>
<td>0.1299</td>
<td>0.2478</td>
</tr>
<tr>
<td>0.9</td>
<td>0.1961</td>
<td>0.1036</td>
<td>0.1497</td>
<td>0.1961</td>
<td>0.1038</td>
<td>0.1498</td>
</tr>
</tbody>
</table>

Table 5.11: Comparison of exact and approximate results of the departure process of an MMPP\(_2\)/M/1 for medium burstness \(C^2=15.6341\), MMPP\(_2\) (0.1, 6.1, 0.1, 0.1)

<table>
<thead>
<tr>
<th>(\rho)</th>
<th>(m_1) (exact)</th>
<th>(m_2) (exact)</th>
<th>(m_3) (exact)</th>
<th>(m_1) (appr)</th>
<th>(m_2) (appr)</th>
<th>(m_3) (appr)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.3226</td>
<td>1.7297</td>
<td>25.4266</td>
<td>0.3226</td>
<td>1.7297</td>
<td>25.4266</td>
</tr>
<tr>
<td>0.2</td>
<td>0.3226</td>
<td>1.7221</td>
<td>25.3080</td>
<td>0.3226</td>
<td>1.7222</td>
<td>25.3080</td>
</tr>
<tr>
<td>0.3</td>
<td>0.3226</td>
<td>1.7098</td>
<td>25.1100</td>
<td>0.3226</td>
<td>1.7104</td>
<td>25.1100</td>
</tr>
<tr>
<td>0.4</td>
<td>0.3226</td>
<td>1.6732</td>
<td>24.5224</td>
<td>0.3226</td>
<td>1.6793</td>
<td>24.5248</td>
</tr>
<tr>
<td>0.5</td>
<td>0.3226</td>
<td>1.7346</td>
<td>25.5020</td>
<td>0.3226</td>
<td>1.5687</td>
<td>22.8340</td>
</tr>
<tr>
<td>0.6</td>
<td>0.3226</td>
<td>1.3573</td>
<td>19.3963</td>
<td>0.3226</td>
<td>1.3686</td>
<td>19.4018</td>
</tr>
<tr>
<td>0.7</td>
<td>0.3226</td>
<td>1.0920</td>
<td>15.0360</td>
<td>0.3226</td>
<td>1.0946</td>
<td>15.0370</td>
</tr>
<tr>
<td>0.8</td>
<td>0.3226</td>
<td>0.7993</td>
<td>10.1860</td>
<td>0.3226</td>
<td>0.8002</td>
<td>10.1860</td>
</tr>
<tr>
<td>0.9</td>
<td>0.3226</td>
<td>0.5115</td>
<td>5.3428</td>
<td>0.3226</td>
<td>0.5118</td>
<td>5.3429</td>
</tr>
</tbody>
</table>
Table 5.12: Comparison of exact and approximate (appr) results of the departure process of an $MMPP_2/M/1$ for high burstness $C^2=23.1331$, $MMPP_2 (0.1, 6.1, 0.1, 0.1)$

<table>
<thead>
<tr>
<th>$\rho$</th>
<th>$m_1$(exact)</th>
<th>$m_2$(exact)</th>
<th>$m_3$(exact)</th>
<th>$m_1$(appr)</th>
<th>$m_2$(appr)</th>
<th>$m_3$(appr)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.5263</td>
<td>0.9499</td>
<td>2.7942</td>
<td>0.5266</td>
<td>0.9499</td>
<td>2.7943</td>
</tr>
<tr>
<td>0.2</td>
<td>0.5263</td>
<td>0.9504</td>
<td>2.7958</td>
<td>0.5772</td>
<td>0.9491</td>
<td>2.7581</td>
</tr>
<tr>
<td>0.3</td>
<td>0.5263</td>
<td>0.8947</td>
<td>2.5875</td>
<td>0.5274</td>
<td>0.8949</td>
<td>2.5876</td>
</tr>
<tr>
<td>0.4</td>
<td>0.5263</td>
<td>0.8465</td>
<td>2.3907</td>
<td>0.5266</td>
<td>0.8465</td>
<td>2.3907</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5263</td>
<td>0.7979</td>
<td>2.1770</td>
<td>0.5264</td>
<td>0.7979</td>
<td>2.1771</td>
</tr>
<tr>
<td>0.6</td>
<td>0.5263</td>
<td>0.7492</td>
<td>1.9476</td>
<td>0.5264</td>
<td>0.7492</td>
<td>1.9477</td>
</tr>
<tr>
<td>0.7</td>
<td>0.5263</td>
<td>0.7004</td>
<td>1.7027</td>
<td>0.5263</td>
<td>0.7004</td>
<td>1.7027</td>
</tr>
<tr>
<td>0.8</td>
<td>0.5263</td>
<td>0.6516</td>
<td>1.4421</td>
<td>0.5263</td>
<td>0.6516</td>
<td>1.4421</td>
</tr>
<tr>
<td>0.9</td>
<td>0.5263</td>
<td>0.6028</td>
<td>1.1662</td>
<td>0.5263</td>
<td>0.6028</td>
<td>1.1662</td>
</tr>
</tbody>
</table>

Table 5.13: Comparison of analytical and simulation results of the mean number ($N$) and response time ($T$) in the second node with offered load 0.2

<table>
<thead>
<tr>
<th>$\rho$(1st node)</th>
<th>N(Ana)</th>
<th>N(sim)</th>
<th>Error(%)</th>
<th>T (ana)</th>
<th>T(sim)</th>
<th>Error(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>0.3065</td>
<td>0.3100</td>
<td>1.13</td>
<td>0.2349</td>
<td>0.2525</td>
<td>6.97</td>
</tr>
<tr>
<td>0.3</td>
<td>0.2742</td>
<td>0.2746</td>
<td>0.15</td>
<td>0.2307</td>
<td>0.2244</td>
<td>2.81</td>
</tr>
<tr>
<td>0.4</td>
<td>0.2635</td>
<td>0.2634</td>
<td>0.04</td>
<td>0.2226</td>
<td>0.2195</td>
<td>1.41</td>
</tr>
<tr>
<td>0.5</td>
<td>0.2583</td>
<td>0.2582</td>
<td>0.04</td>
<td>0.2186</td>
<td>0.2168</td>
<td>0.83</td>
</tr>
<tr>
<td>0.6</td>
<td>0.2552</td>
<td>0.2552</td>
<td>0.00</td>
<td>0.2161</td>
<td>0.2151</td>
<td>0.46</td>
</tr>
<tr>
<td>0.7</td>
<td>0.2533</td>
<td>0.2532</td>
<td>0.04</td>
<td>0.2151</td>
<td>0.2139</td>
<td>0.56</td>
</tr>
<tr>
<td>0.8</td>
<td>0.2519</td>
<td>0.2518</td>
<td>0.04</td>
<td>0.2137</td>
<td>0.2131</td>
<td>0.28</td>
</tr>
<tr>
<td>0.9</td>
<td>0.2508</td>
<td>0.2508</td>
<td>0.00</td>
<td>0.2127</td>
<td>0.2123</td>
<td>0.19</td>
</tr>
</tbody>
</table>

of the departure process of an $MMPP_2/M/1$ for high burstness $C^2=23.1331$.

In examples 10, 11 and 12, we obtain similar results as in the previous three examples but using different burstness indicators. From Table 5.10–5.12, we observed that the exact moments and approximate moments match very well, with a maximum relative error in all cases less than 9 percent.

Example 13: In this example, we feed the departure process from the $MMPP_2/M/1$ node to a second single server infinite-capacity node with exponentially distributed service times. We analyze this node numerically and compare the mean number and the mean response time in the system against simulation. The service rate of the second queue is set to 11.7926 so that the offered load in the second queue is 0.2. The results are shown in Table 5.13.
Table 5.14: Comparison of analytical and simulation results of the mean number \((N)\) and response time \((T)\) in the second node with offered load 0.9

<table>
<thead>
<tr>
<th>(\rho(1st\ node))</th>
<th>(N(Ana))</th>
<th>(N\ (sim))</th>
<th>Error(%)</th>
<th>(T\ (ana))</th>
<th>(T\ (sim))</th>
<th>Error(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>15.0754</td>
<td>16.1806</td>
<td>6.83</td>
<td>12.4189</td>
<td>12.2580</td>
<td>1.31</td>
</tr>
<tr>
<td>0.3</td>
<td>12.3049</td>
<td>12.4876</td>
<td>1.46</td>
<td>10.3561</td>
<td>10.2039</td>
<td>1.49</td>
</tr>
<tr>
<td>0.4</td>
<td>11.0707</td>
<td>11.1063</td>
<td>0.32</td>
<td>9.3556</td>
<td>9.2521</td>
<td>1.12</td>
</tr>
<tr>
<td>0.5</td>
<td>10.3563</td>
<td>10.3579</td>
<td>4.05</td>
<td>8.7643</td>
<td>8.6976</td>
<td>0.77</td>
</tr>
<tr>
<td>0.6</td>
<td>9.8908</td>
<td>9.8853</td>
<td>0.06</td>
<td>8.3752</td>
<td>8.3333</td>
<td>0.50</td>
</tr>
<tr>
<td>0.7</td>
<td>9.5764</td>
<td>9.5613</td>
<td>0.16</td>
<td>8.1315</td>
<td>8.0789</td>
<td>0.65</td>
</tr>
<tr>
<td>0.8</td>
<td>9.3321</td>
<td>9.3229</td>
<td>0.10</td>
<td>7.9162</td>
<td>7.8880</td>
<td>0.36</td>
</tr>
<tr>
<td>0.9</td>
<td>9.1482</td>
<td>9.1427</td>
<td>0.06</td>
<td>7.7576</td>
<td>7.7406</td>
<td>0.22</td>
</tr>
</tbody>
</table>

Table 5.15: Comparison of dimensioning results using equivalent bandwidth approach for 100*IPP and a single \(MMPP_2\) approximation

<table>
<thead>
<tr>
<th>(\delta_1)</th>
<th>(C_e(1*IPP))</th>
<th>(100C_e\ (100*IPP))</th>
<th>(c_{min})</th>
<th>(C_{MMPP_2})</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>9.985178</td>
<td>998.5178</td>
<td>152.3608</td>
<td>150.0003</td>
</tr>
<tr>
<td>0.2</td>
<td>9.985181</td>
<td>998.5181</td>
<td>203.1478</td>
<td>200.0008</td>
</tr>
<tr>
<td>0.3</td>
<td>9.985185</td>
<td>998.5185</td>
<td>232.7351</td>
<td>300.3092</td>
</tr>
<tr>
<td>0.4</td>
<td>9.985190</td>
<td>998.5190</td>
<td>248.8043</td>
<td>399.9840</td>
</tr>
<tr>
<td>0.5</td>
<td>9.985198</td>
<td>998.5198</td>
<td>253.9349</td>
<td>500.0080</td>
</tr>
<tr>
<td>0.6</td>
<td>9.985209</td>
<td>998.5209</td>
<td>248.8043</td>
<td>598.7573</td>
</tr>
<tr>
<td>0.7</td>
<td>9.985227</td>
<td>998.5227</td>
<td>232.7351</td>
<td>699.2919</td>
</tr>
<tr>
<td>0.8</td>
<td>9.985263</td>
<td>998.5263</td>
<td>203.1479</td>
<td>799.9974</td>
</tr>
<tr>
<td>0.9</td>
<td>9.985371</td>
<td>998.5371</td>
<td>152.3609</td>
<td>900.9523</td>
</tr>
</tbody>
</table>

**Example 14:** This is similar to the previous example, only the offered load to the second queue is 0.9. Results are shown in Table 5.14.

We note that the auto-covariance of the departure process from the first node decreases as the offered load increases. This explains why the mean number of customers in the system and mean response time in the second queue decreases as its offered load increases. That is, the queueing performance of the queue depends on the auto-covariance as well as the offered load.

**Example 15:** A dimensioning example for DSLAM multiplexer is shown in Table 5.15. The IPP parameters are: \(\delta_1\) varies from 0.1 to 0.9 by an increment of 0.1, \(\delta_2=1-\delta_1\), \(\lambda=1\) and the mean ‘On’ time is \(b=0.1\) seconds. The peak rate \(R_p=25Mbps\). The buffer size \(K\) was set to 2.048M bps. and the loss probability requirement is less than \(P_L=10^{-6}\). From
Table 5.16: Comparison of dimensioning results from exact solution and approximation \(10^*MMP P_2\) and a single \(MMPP_2\) approximation

<table>
<thead>
<tr>
<th>(W)</th>
<th>(10^*MMP P_2) (exact)</th>
<th>(1^* MMPP_2) (Appr.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0001</td>
<td>934.5882</td>
<td>999.8405</td>
</tr>
<tr>
<td>0.0005</td>
<td>467.2941</td>
<td>499.9203</td>
</tr>
<tr>
<td>0.0013</td>
<td>311.5294</td>
<td>333.2802</td>
</tr>
<tr>
<td>0.0027</td>
<td>233.6471</td>
<td>249.9601</td>
</tr>
<tr>
<td>0.0052</td>
<td>186.9176</td>
<td>199.9681</td>
</tr>
<tr>
<td>0.0095</td>
<td>155.7647</td>
<td>166.6401</td>
</tr>
<tr>
<td>0.0179</td>
<td>133.5126</td>
<td>142.8344</td>
</tr>
<tr>
<td>0.0381</td>
<td>116.8235</td>
<td>124.9801</td>
</tr>
<tr>
<td>0.1152</td>
<td>103.8431</td>
<td>111.0934</td>
</tr>
</tbody>
</table>

Table 5.17, we can see that the effective bandwidth of the single \(MMPP_2\) is between the \(100^*IPP\) and \(\epsilon_{min}\) for different cases.

**Example 16**: A dimensioning example for the BRAS multiplexer is shown in Table 5.16. We consider dimensioning the output link (transmission rate) of BRAS to meet a mean delay requirement \(W\). In this example, \(10^*MMP P_2\) (i.e., 10 Ethernet switches are connected to BRAS) is considered as the input to BRAS. We approximate the \(10^*MMP P_2\) each with parameters \((0.4481, 0.5519, 12.7, 6.671)\), by a single \(MMPP_2\) with parameters \((0.9922, 0.0078, 196.981, 92.645)\) and compare the dimensioning results. Both exact results and approximation results are obtained using the matrix-geometric approach.

Notice that the exact and approximation results in Table 5.16 is with the unit same as the peak rate of arrival streams. For example, if each arrival stream has peak rate 25Mbps, both the exact and approximation results should be multiplied by 25Mbps.

### 5.10.1 Validation for the departure process of an IPP/M/1 queue

Using different approach, Daley in [9] found the Laplace transform of the inter-departure time of an IPP/M/1 queue (with service rate \(\mu\)) as follows:

\[
D_{ipp}(s) = \frac{\mu}{s + \mu} - \frac{\mu(1 - \sigma)}{s - \mu(1 - \sigma)} A_{ipp}(s)
\]

where \(\sigma\) is the root of \(\sigma = A_{ipp}(\mu(1 - \sigma))\) and \(A_{ipp}(s)\) is the Laplace transform of the inter-arrival time of an IPP, which is obtained in section 2. To validate our results for an IPP/M/1 (a special case of \(MMPP_2\)), we compared our results to Daley’s results for different IPPs.
Table 5.17: Comparison of our results of the departure process (Laplace transform) of the IPP/M/1 with Daley’s results

<table>
<thead>
<tr>
<th>( \lambda )</th>
<th>Input (IPP)</th>
<th>Output (ours)</th>
<th>Output (Delay)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( [0.09, 0.01, \lambda] )</td>
<td>Laplace transform</td>
<td>Laplace transform</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>( .74586(s+1.3442)(s+0.99745e-2) )</td>
<td>( .74586(s+1.3442)(s+0.99745e-2) )</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>( .53154(s+2.1064)(s+0.99471e-2) )</td>
<td>( .53154(s+2.1064)(s+0.99471e-2) )</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>( .96178(s+5.1229)(s+0.99325e-2) )</td>
<td>( .96178(s+5.1229)(s+0.99325e-2) )</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>( .98075(s+4.1090)(s+0.99217e-2) )</td>
<td>( .98075(s+4.1090)(s+0.99217e-2) )</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>( .98786(s+5.1020)(s+0.99202e-2) )</td>
<td>( .98786(s+5.1020)(s+0.99202e-2) )</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>( .99215(s+6.0990)(s+0.99171e-2) )</td>
<td>( .99215(s+6.0990)(s+0.99171e-2) )</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>( .99509(s+7.0950)(s+0.99148e-2) )</td>
<td>( .99509(s+7.0950)(s+0.99148e-2) )</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>( .99783(s+8.0900)(s+0.99127e-2) )</td>
<td>( .99783(s+8.0900)(s+0.99127e-2) )</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>( .99877(s+9.0913)(s+0.99112e-2) )</td>
<td>( .99877(s+9.0913)(s+0.99112e-2) )</td>
<td></td>
</tr>
</tbody>
</table>

with different parameters, the two results match exactly except for some numerical errors caused by the computation tool. Our approach has identical computation complexity as Daley’s approach. However, Daley’s approach only works for an IPP arrival process while our method works for general \( MMPP2 \).

Some results are provided in the Table 5.15 where \( (\delta_1, \delta_2)=(0.09, 0.01) \) and \( \lambda \) is varying from 1 to 9. We also verified that the departure process of an M/M/1 (obtained by making the \( MMPP2 \) to have the same arrival rates at both states) is a Poisson process with same average rate as arrival process. This is well known result.

5.11 Conclusions

In this Chapter, we developed an efficient way to approximate the superposition of \( n \) IPP (or \( n \) \( MMPP2 \)) arrival process by an \( MMPP2 \) through moments and IDC matching. We then find the closed-form Laplace transform of the departure process of an \( MMPP2/M/1 \) queue. Using Laplace transform matching, we approximated the departure process of an \( MMPP2/M/1 \) by an \( MMPP2 \). The approximate analytical results match the simulation results very well. Based on these results, we give examples of dimensioning...
of an ADSL access network. It will be an interesting topic to extend the current work to the $MMPP_2/M/N$ queue.
Chapter 6

Summary and Future Work

6.0.1 Summary of Research Contributions

In this thesis, performance evaluation and optimal dimensioning algorithms of communication systems in a dynamically changing environment were developed. We considered the cases where the arrival rates of customers are time-varying, the bandwidth requirement of each customer is variable, the arrival process is correlated and integrated streaming and data traffic where the available bandwidth for data traffic is dynamically changing.

In Chapter 2 we compared different dimensioning approaches for nonstationary arrivals and proposed an efficient dimensioning algorithm for different cases.

In Chapter 3 we developed an original model for variable-demand customers in a circuit-switched link, and proposed an iterative recursive algorithm for performance analysis and provided capacity dimensioning approach for this model.

In Chapter 4 we proposed an efficient dimensioning approach for integrated streaming and data traffic using decomposition.

In Chapter 5 we obtained analytical approximations for the superposition of $n$ homogeneous IPP (or $MMPP_2$) and characterized the departure process of an $MMPP_2/M/1$ queue as an $MMPP_2$. The characterization of the departure process of an $MMPP_2/M/1$ queue by another $MMPP_2$ was obtained using an efficient method based on Laplace transform matching.

There are connections among the four chapters. It is interesting to note that
the most efficient dimensioning approach for nonstationary arrivals is the one described in Chapter 2, i.e., the square-root-safety dimensioning formula, which can be applied to both stationary and nonstationary cases. The results from the square-root-safety dimensioning formula are very close to the fixed point approximation (FPA) [4].

The square-root-safety dimensioning formula is also used in Chapter 4 for the streaming traffic. We note that the square-root-safety dimensioning formula can be applied to multi-class streaming traffic too.

The bursty traffic model of the superposition of $n$ homogeneous IPPs ($MMPP_2$) in Chapter 5 can be applied to the data traffic model introduced in Chapter 4. In this sense, the model presented in Chapter 4 can be further generalized.

6.0.2 Future Directions

There are some interesting extensions that we will consider in the future:

1). In Chapter 2, only a single class of traffic is considered for the time-varying arrivals. It will be interesting to extend this model to multi-class cases.

2). In Chapter 3, we can assume that bandwidth allocation is adaptive such as in a wireless network to accommodate more customers when congestion occurs. The performance evaluation and efficient dimensioning of this case is another interesting topic.

3). For the integrated streaming and data traffic in Chapter 4, we only considered the single-class traffic case. The multi-class traffic extension for both streaming and data traffic can be an interesting topic. Also, the streaming traffic can be bursty too.

4). For the data traffic, in Chapter 4, we only considered the infinite buffer size case. The dimensioning of an MMPP/G/1/K queue (i.e. with a finite buffer) will be an interesting extension.

5). In Chapter 5, we approximated $n$ independent homogenous IPPs ($MMPP_2$) by an $MMPP_2$. It will be interesting to consider the case of non-homogenous IPPs (or $MMPP_2$). This is known to be a very difficult problem.

6). The departure process of an $MMPP_2/M/1$ queue was studied in Chapter 5. It will be interesting to study the finite buffer case and obtain the departure process of an $MMPP_2/M/N$ queue.

7). Finally results obtained in Chapter 5 can be applied to dimension an end-to-end packet-switched connection, such as an LSP in an multi-protocol labeled switch (MPLS).
network.
Bibliography


