Approximation to the stationary distribution of information flows in a communication network

CAO YI

(B.Math, Nanjing University)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF SCIENCE

(MANAGEMENT)

DEPARTMENT OF DECISION SCIENCE

NUS BUSINESS SCHOOL

NATIONAL UNIVERSITY OF SINGAPORE

Aug, 2005

Acknowledgement

First and foremost, I would like to take this opportunity to express my sincere appreciation to my supervisors, **Assoc Prof Ou Jihong** and **Asst Prof Ye Hengqing**. They introduced me into the area of queueing network modelling. Their enlightening instruction, plus their comprehensive collection of research materials, allows me to deeply understand my research topic. I am immensely grateful to them for their time and effort on this thesis in spite of their heavy schedule and other responsibilities. Their interest and enthusiasm in this research topic have made it possible for the completion of this thesis.

I am also grateful to **A/P Thompson Teo, A/P Chou Cheng Feng, Ms. Ang Chin Teng, Ms. Lim May Lin**, and of course my supervisors, for the time you shared with me in personal conversation and sincere communication, and your greatest support.

Lastly, I dedicate this thesis to my parents, who continually encourage and support me to pursue my goal, who love me most and who I love most.

Contents

Summary

In this study we propose a procedure to approximately compute the stationary distribution of the number of transmitting information flows in a communication network.

The flows arrive to the network according to Poisson processes with exponentially distributed flow volumes, and traverse through a fixed path of transmission links in the network. The links have finite transmission capacities which are allocated to the information flows concurrently transmitting in the network according to some dynamic bandwidth sharing rule, which ensures the stability of the total number of information flows ongoing in the network.

The procedure is based on dynamic approximation of the bandwidths allocated to concurrent information flows in the network. Numerical examples show that the procedure produces the numerical solution of the network within 2% of the true values.

Chapter 1

Introduction

1.1 Background

This thesis proposes an approximation algorithm for computing the stationary joint distribution of the number of ongoing connections (or information flows) in a communication network. This type of communication network is widely used to model the modern data transmitting communication network such as today's Internet, rather than the telecommunication network and manufacture job shop, etc. In the latter cases, the traditional queueing network such as the Jackson network or the BCMP network is used.

In recent years, the increasing volume of digital media file transmitting in the network and the heavy visiting rate to some news websites upon the occurrence of some worldwide events such as the 911 have deteriorated the previously high performance of the Internet, because the current network traffic control mechanism is designed for the small file transmission situation, rather than today's demanding situation. Therefore the network is once again put up on the researchers' table.

An abstract framework of this communication network comprises a set of routes con-

necting a pair of nodes that are the possible source and destination of information flows, which can be voice conversations in a telephone network, or the digital documents in a data network like the Internet, and a set of transmission links. A simplified example of this communication network with two links and three routes is illustrated in Figure 1.1, associated with its abstraction in Figure 1.2. Each route carrying an amount of information flows traverses through a fixed subset of links; while each link has a transmission bandwidth capacity, which will by some dynamic bandwidth allocation rule, be shared among the routes that traverse through it. The bandwidth allocated to each route is uniquely determined accordingly, facilitating the transmission of these information flows.

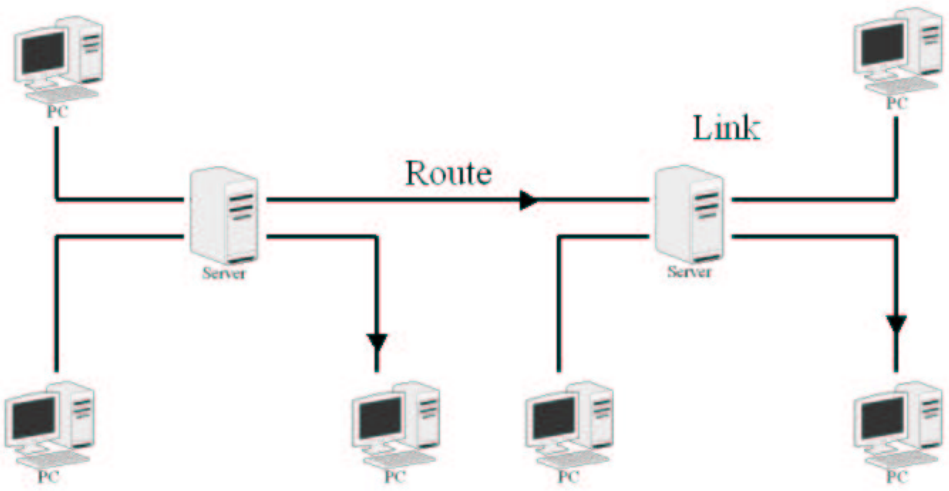

Figure 1.1: Communication Network

When an information flow carrying an amount of data arrives on a given route, a connection is established on that route. After the transmission is finished, the connection is terminated. The same as the traditional queueing network such as the manufacturing or service network, the communication network can be characterized by the fluctuation of the number of ongoing connections on each route in the network. However, different from a job in a manufacturing job shop or a customer in a service system that visits the

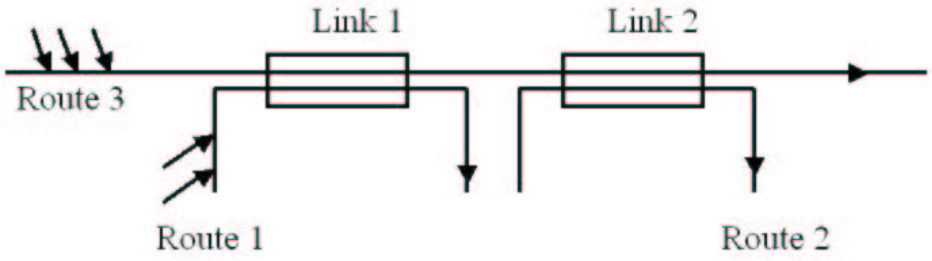

Figure 1.2: Communication Network Model

service stations along its route one at a time, an information flow in the communication network takes up resources simultaneously at all the links along its transmission route.

A fundamental issue about this communication network is how to allocate the link's bandwidth capacity to the routes that traverse through it. If we imagine the transmission process on each route as a queueing system, it can be seen that the bandwidth allocation according to some bandwidth allocation rules determines the service rate associated with each queue (route).

An Additive Increase Multiplicative Decrease bandwidth allocation algorithm is implemented in the TCP (the traffic control protocol) of the Internet (see Chiu and Jian 1989, Chiu 2000). However, it is observed that the TCP algorithm favors shorter round trip time. Bertsekas and Gallager (1992) discussed the Max-min fairness bandwidth allocation algorithm which intended to maximize the minimum bandwidth allocated to each route.

Kelly (1997, 1998) proposed the concept of proportional fairness bandwidth allocation and developed a decentralized algorithm to implement it. The objective of the allocation rule was to maximize the overall utility of the bandwidth allocations by assuming each route had the logarithmic utility function.

Mo and Walrand (2000) generalized the above results. They proposed a general form of optimization problem that solved the bandwidth allocations. They referred it as the α proportional fairness allocation. The Max-min and Kelly's proportional fairness allocations are then the special cases of the α allocation rule.

These fairness bandwidth allocation rules are critical to another important aspect of the network, the stability of the network: whether or not the bandwidth allocated to each route is enough to digest the workload. It may be intuitive that the normal offered load condition is sufficient, that is the total traffic load on each link is within the link's capacity. Unfortunately, Bonald and Massoulie (2000) presented some examples showing that for some priority bandwidth allocation rules, the condition is insufficient. However, many studies show that when the various fairness allocation rules are applied instead, the normal offered load condition is sufficient. See De Veciana et.al (2001)'s discussion for the max-min fairness allocation, Bonald and Massoulie (2000)'s for the general α fairness bandwidth allocation, and Ye (2003)'s for the more general utility maximizing bandwidth allocation under general traffic conditions.

1.2 Motivation

Although many studies have been conducted for this communication network, compared to the rich analytical results for the traditional queueing network, little has been available for this network. Lying at the bottom line of those analytical results is the stationary distribution of the queueing length of each queue in the network. However, even for most traditional queueing networks, the analytical solutions are not permitted. It adds on extra difficulty for this communication network due to its special bandwidth allocation

characteristics. It is the complexity introduced by the bandwidth allocation rule in the network that precludes to derive the simple closed form solution, e.g. a product form solution, for the stationary distribution of the system. In particular, Chiu (2000) showed the solution is not of product form for a particular network example.

Up to now, only for a few networks with simple structure and certain bandwidth allocation rule, the closed form solutions are derived (see Fayolle et.al 2001). Masoullie and Roberts (1998) showed the closed from solution for the linear network, Bonald and Massoulie (2000) further found the solution for the grid network by solving the same full balance equations. They suggested that the closed form solution for the network that violates the strict underlying assumptions is unavailable.

Instead, we may resort to the numerical solution by solving the Markov transition rate matrix. However, when the state space is too large, such as a network with too many routes, solving the huge matrix is impractical due to the "curse of dimensionality".

It thus stimulates our interests to design an approximation method to fill the gap, as what has long been done for those traditional non-product form queueing networks. The underlying idea of our algorithm is to decompose the network into disjoint routes with each one being represented by an $M/M/1$ processor sharing (PS) queue. The service capacity is random in that it is dynamically approximated by taking into account the interdependence of the bandwidth allocations on all the other transmission routes. In particular, the transmission bandwidth on each route is estimated based on the current states of all the other routes in the network. The procedure is then iterative: it first computes the marginal distribution of the number of ongoing connections on one route, which provides the base to compute the joint distribution of two routes, etc.

The same idea of decomposition approach was developed to approximately compute

the stationary distributions of the traditional queueing network that does not permit the closed form solution. The seminal works include Bitran and Tirupati (1988) and Whitt (1983), and the more recent ones can be found in Whitt (1995) and (1999). The procedure we construct here is a first attempt to compute (*approximately*) the stationary distribution for a queueing network with the simultaneous resource consumption (SRC) characteristics. The decomposition approach is modified here in that each isolated queue is not statically separated from the others but rather dynamically linked in the estimation of the processing capacity.

1.3 Research contribution

As we have mentioned, the data transmitting communication network such as the Internet is once again a hot topic today. In recent years, the increasing volume of digital file transmitting in the network and the heavy visiting rate to some websites have largely deteriorated the previously high performance of the Internet, because the current network traffic control mechanism is not suitable for today's demanding situation. Thus improvements are introduced, such as the new bandwidth allocation rules other than the TCP.

Consequently how to evaluate the performance of the network in the context of these new improvements becomes an urgent subject. But the very few analytical results available up to now is disappointing. Although we have the closed form solutions for some simple networks, there is still no clues how the solution looks like for the general network.

Our approximation method is trying to fill the gap. It provides a very accurate numerical solution to this network. Numerical examples indicate that the approximation error falls within a very small margin of the true solution. As another feasible method,

even the most effective modern statistical method, namely the Gibbs Sampling method under-performs. Thus we can expect that those communication networks with the modest size could now be solved with a high degree of accuracy. To best of our knowledge our method is the first general approximation procedure that provides the numerical solution to this communication network.

Our algorithm has two advantages over those analytical results that are currently available. One is that it is independent of the specific structure of the network in that it can be applied to any such communication network without adjusting the algorithm to accommodate its specific structure. The network structure is automatically reflected in the dynamic bandwidth allocation rule, a subfunction in our algorithm.

Another advantage is that it is independent of the specific bandwidth allocation rule. The bandwidth allocation rule is packaged in a sub-function and called by the main function in our algorithm. This feature is of practical use. Because those newly developed bandwidth allocation rules can be tested here in terms of their distinctive impact on the network performance. We just modify the subfunction to accommodate the specific rule.

There are some practical usages as well. For example, in a large network, the accurate solution of the system is not the first concern. The network administrator is concerning with the bottleneck of the network. In this case, we pursue the speed of the solution rather than the accuracy by introducing larger truncation error. Then the marginal distribution of each route, which is more accurate than the joint distribution of the system when the truncation error is large, will provide information about the dynamics of each route, indicating where the network is in heavy traffic condition and where light traffic. This result is definitely not achievable through the inefficient simulation method, or any other local approximation methods that isolate routes for tractability.

Given the information of the traffic on each route, we can further adjust the settings

of the algorithm, such that the truncation on each route is treated individually. This adjustment will improve the computational efficiency as well as the accuracy of the solution of the system.

1.4 Organization of the thesis

The organization of this thesis is as follows. The following chapter of literature review will provide the well round background of our study. We will first study the communication network which is the subject of this study. The network structure, various bandwidth allocation rules, and the current achievement of some analytical results will be covered. Since the analytical results for the communication network are relatively rare, we will resort to the traditional queueing networks such as the Jackson and BCMP network to search for insights from their rich numerical approximation toolbox. The most effective approximation methods for the non-product form queueing network will be reviewed. Finally some modern statistical tools developed in the last decade as a very effective way to compute the complex probability distribution will be briefly introduced. In particular, we will briefly investigate the modern sampling method, namely the Gibbs sampling method, which makes computing the complex probability distribution easy by using the modern computational power.

Chapter 3 formulates the framework of the communication network under study, and discusses the various issues that are critical to the network. In Chapter 4, we proposes the approximation algorithm to compute the solution of the network numerically. The modified Gibbs Sampling which provides an alternative method, other than the ineffective simulation method, to derive the benchmark solution of the network for comparison purpose is the subject of Chapter 5. Numerical results are presented in Chapter 6, in

which we compare the results of our approximation algorithm with those from the Gibbs sampling method and the rare analytical results. Chapter 7 concludes our study.

Chapter 2

Literature Review

This chapter consists of three parts. Section 2.1 discusses this communication network under study. Various issues critical to the network and some current achievements will be covered. Section 2.2 resorts to the rich set of the approximation methods for the traditional queueing network to look for insights. Two major approximation methods for the non-product form queueing network are discussed in detail. The development of the modern statistical method as an alternative but very effective method to compute a complex probability distribution is the subject in Section 2.3. Section 2.4 summaries this chapter.

2.1 Communication network

The data transmission communication network such as the Internet has been there for a decade. In recent years, it is observed that the increasing volume of large file transmitting in the network and the heavy network traffic have seriously deteriorated the previously high performance of the network, because the current network traffic control mechanism is designed for the small file transmission situation, not suitable for today's demanding

situation. Therefore the network is once again a hot topic.

2.1.1 Bandwidth allocation

One of the fundamental questions related to improving the network performance in the new environment is how to allocate each link's bandwidth capacity among those transmission routes that traverse through it, such that the network can effectively handle the workload on each route.

Bertsekas and Gallager (1992) discussed the Max-min bandwidth allocation algorithm which intended to maximize the minimum bandwidth allocated to each route such that the minimum transmission rate is improved. It was later proved that this allocation is the fairest bandwidth rule. (De Veciana et.al 2001)

Kelly (1997, 1998) proposed another, namely the Proportional fairness rule. This bandwidth allocation maximized the overall utility of the network by assuming a logarithmic utility function. From the mathematic perspective, Kelly's study suggested that the bandwidth allocation could be obtained by solving an optimization problem, pre-assuming the number of ongoing connections on each route was fixed.

Later on, this idea was further developed by Mo and Walrand (2000). They considered a more general optimization problem. The corresponding bandwidth allocation was referred as the α proportional fairness bandwidth allocation rule. This allocation rule includes a wide range of allocation rules, such as the max-min rule, Kelly's proportional fairness rule, etc. Moreover, a weighting factor w_r was introduced into the optimization problem of the α allocation rule. (see Bonald and Massoulie 2000)

Ye (2003) considered a more general bandwidth allocation rule, named the U- utility maximizing allocation rule, based on Kelly (2001) and Low (2003)'s work. This rule maximized a more general form utility function aiming to approximate the current TCP

allocation rule.

2.1.2 Stability conditions

Another important issue is the stability condition of the network, under which the mean number of ongoing connections on each route will remain finite, not grow into infinite in the long run. Intuitively, it is expected that the normal capacity constrain on each link is a sufficient condition, which is also referred as the normal offered load condition.

Unfortunately, Bonald and Massoulie (2000) showed for some networks with the priority bandwidth allocation rules, this condition is insufficient. They concluded that in the absence of the fairness prerequisite, the bandwidth allocation rules of Pareto efficiency was not sufficient to guarantee the stability of the network under the normal traffic condition.

According to their suggestion, the stability problem was then studied when some certain fairness bandwidth allocation rule was applied. Some recent results were found in Massoulie and Roberts (1998) for Kelly's rule, De Veciana et.al (2001) for max-min rule. Bonald and Massoulie (2000) provided the stability results under the general α bandwidth allocation rule, by using a Fluid model. Ye (2003) provided similar stability results for a more general bandwidth allocation, the U-utility maximizing allocation. Ye et.al (2003) extended the results to the network with general stationary arrival process.

2.1.3 Stationary distribution

Rich studies are conducted in the static context, in that the number of ongoing connections on each route is fixed during the period of study. Little has been done to study the stochastic behavior of the network (Massoulie and Roberts 1998), of which the fundamental question is the dynamic of the state of the system.

In rare cases where the full balance equations are applicable, the system is solved

by using the traditional Markov chain technique. For example, Massoulie and Roberts (1998) provided the close form solution for the linear network under Kelly's Proportional fairness rule. Bonald and Massoulie (2000) extended the above results to Grid network, the generalization of the linear network.

They added on that the analytical result was not available for the more general network where the strict underlying assumptions were not satisfied. See Fayolle et.al (2001) for the similar comments and their study of the approximation method to the star shaped network.

2.2 Approximation methods

Long before the fast development of the communication network, the traditional queueing network has been extensively studied since Jackson's seminar work (Jackson 1957,1963), and later the BCMP theory (Baskett et.al 1975). These networks have an attractive property that the stationary joint distribution of the system could be explicitly expressed in a product form. But in more general cases where the local balance equations are not available, most queueing networks do not permit the product form solution. Only by approximation methods can we obtain an approximated solution. Among them, the most effective approximation method, namely the decomposition method, borrowed the underlying idea of Jackson's product form solution.

2.2.1 Decomposition method

Although the queueing network is difficult to analyze in a whole, it can be divided into several small subnetworks, in the extreme case each subnetwork consisting of only one queue. Then each subnetwork is analyzed individually. Finally by taking into account the interaction between the different subnetworks, the individual results are combined together to obtain the approximated solution to the entire network.

Based on this idea, this method is widely used when the queues of the network can be divided into weakly interrelated groups. The advantage of this method is that it requires little on the computational time which is independent of the size of the entire network. While the disadvantage is that the uncertainty of the accuracy level of the solution remains, and the convergence of the solution is not guaranteed (Gelenbe and Pujolle 1987).

As noted by Harrison and Petal (1993), the decomposition method gave a very accu-

rate approximation when the system was almost of product form. It also gave remarkably good results even when many product form assumptions were violated.

The most often used decomposition method is found in Kuehn (1979), later in Bitran and Dasu (1990), and is especially applied to the open queueing network. It first decomposes the network into a set of single queues, and then analyzes the effective input and output process of each queue. The interaction among the separated queues is reflected by square of coefficient of variations of arrival processes at each queue. (see Whitt 1983,1984's a series of superior work and the reference therein)

With some normal assumptions, the method consists of three steps: flow aggregating (see Bitran and Tirupati 1988, Whitt 1982,1983), flow analysis (see Pujolle and Ai 1986), flow splitting (see Whitt 1984, Disney and Konig 1985). Finally by combining the three steps, we obtain the system of linear equations (see Bitran and Tirupati 1988) that solve the effective arrival rates and their interrelationship, which are used to compute the various performance measures of the system (see Albin 1984, Bitran and Dasu 1990).

Whitt (1983) developed a software, namely the QNA (Queueing Network Analyzer) to implement the above procedures. The advantage of the QNA over other similar solvers such as the PANACEA is that it requires only renewal arrivals rather than Poisson arrivals as in the other solvers, thus the modelling error is largely eliminated. The drawback is that it assumes un-correlated and un- autocorrelated arrivals, thus may encounters difficulties in the heavy traffic bottleneck situation. (see Kim et.al 2000, Suresh and Whitt 1990, and Whitt 1995).

Bitran and Tirupati (1988) considered the decomposition method for the multiple product network with deterministic routing, where the interaction among the different types of product streams is a concern in the splitting step. They proposed a way to take

into account this interaction.

2.2.2 Diffusion approximation

The decomposition method is successful for the queueing network with normally discrete arrivals. When the arrivals are intensive, each increase or decrease of the population comparing to the total population is relatively small. Thus it is suggested by Harrison (1985), Reiman (1984) to model the population as a Brownian Motion. Because of its similarity to the diffusion equations for the ideal gas, this approximation method is as well called the diffusion approximation.

It was shown (Reiman 1984) that under the heavy traffic condition, the J- dimension queue length process associating with a certain type of open J- dimension queueing network, when properly normalized, converges to a corresponding reflected Brownian Motion (RBM) with drift.

Many studies have been done to identify the underlying RBM model of the single class queueing network, and to convert parameters of the network to the inputs of the corresponding RBM model. See Harrison and Williams (1987) and Harrison et.al (1990) for the survey of work on open and close queueing network respectively.

The justification of this approximation method is based on the "heavy traffic limit theorem". For example, Dai and Dai (1999) proved the theorem be valid for the finite buffer single class queueing network.

For two dimensional RBMs, the analytical solutions were derived in Harrison et.al (1985), Foddy (1983). In higher dimension, RBMs with exponential form solutions were identified in Harrison and Williams (1987), Williams (1987). In general, we have to identify and then solve a set of Partial Differential Equations (PDEs) to obtain the numerical solution.

Dai and his colleagues have done a lot of work on identifying a unique set of PDEs, and exploring efficient methods to solve the RBM model, see Dai and Harrison (1991,1992), Dai et.al (1994)' SBD method, Harrison and Nguyen (1990)'s QNET, Chen et.al (2002)'s using of the finite element method for the finite buffer network. Recent applications of the RBM model to multiple product queueing network can be found in Chen et.al (2001).

2.3 Gibbs sampling method

In our experiments, we found that the simulation results are unsatisfying for providing the benchmark solution to evaluate the accuracy of our approximation algorithm, in absence of analytical solution of the network. The reason is that although in practice, the simulation method is most of time used to provide the benchmark of mean queue length, in our case the solution of the system is a set of huge number of values. Thus to obtain an accurate solution by conducting the naive simulation method is not satisfying. In order to quantify the accuracy of our approximation algorithm, we resort to the more effective modern statistical method, in particular the Gibbs sampling method (Germen and German 1984, Liu 1996), which provides the benchmark solution.

Gibbs sampling was first developed by Geman and Geman (1984) for simulating posterior distribution in image reconstruction. As a family member of the modern sampling methods (the Monte Carlo Markov Chain technique), it gives a way to approximate the probability distributions through sampling. In particular, it gives a convenient way to sample from a complex distribution.

The Gibbs technique can be theoretically justified by the Monte Carlo Markov Chain (MCMC) theory, and it does do an excellent job in assisting statisticians to compute the posterior marginal distribution efficiently and accurately. For the complicated applica-

tions and additional references, see Casella and George (1992), Gelfand et.al (1990), and Gelfand and Smith (1990) and references therein.

For the case of discrete sample space, Liu (1996) modified the naive Gibbs sampler, and proved that the modified Gibbs sampler was statistically more efficient than the random scan Gibbs sampler (a type of the naive Gibbs sampler). It was essentially a random sampler, which updated the sample in each cycle with an acceptance probability, as that of the MCMC sampler.

2.4 Chapter summary

This chapter reviews a rich collection of the literatures relative to our study. For the communication network that once again becomes a hot topic in recent years, the bandwidth allocation rule, stability condition, and the stationary solution are among the most important issues of theoretical and practical interests. At present time, the bandwidth allocation issue is under extensive studies, but the stationary solution remains to be a challenging problem.

To search for the approximation method to compute that stationary solution which is similar to that of the the traditional queueing network, classic Jackson network provides the theoretical insights to our understanding of a network system. Although it fails to fit into more realistic network, its product form solution does suggest the availability of the decomposition approximation method for a wide range of the traditional networks. Up to now, the decomposition method still dominates the approximation area of most of real life networks due to its relatively easy formation. In the parallel side, reflected Brownian motion approximation is used to approximate the solution of networks with heavy traffic feature. Modern statistical method such as the MCMC and Gibbs sampling methods

provides a new angle to look at the difficult problem.

Chapter 3

The Communication Network Model

In this chapter, we will introduce the communication network model, which is often used to model today's data transmission network such as the Internet, WAN, LAN, etc. The fundamental issues about this network will be covered, including the bandwidth allocation rule, the stability condition, and some analytical results available up to date.

3.1 The Network framework

The communication network comprises a set of L transmission links, which provide the bandwidth for the information flows transmitting on the network; and a set of routes r's carrying information flows, with each one be a non- empty subset of L, in the sense that each route r traverses a set of links l's. Denote the set of all possible routes as R such that $r \in R$, with a total of M routes. Conversely, let a fixed $0-1$ incidence matrix $A = (A_{lr}, l \in L, r \in R)$ indicate which links are in a particular route, and let $R(l)$ indicate all the routes that have link l on their path.

A simple abstraction of a real-life communication network (linear network) is illustrated in Figure 3.1. This network consists of two links and three routes, with route 1

and 2 traverse through link 1, 2 respectively, and route 3 through both link 1 and 2.

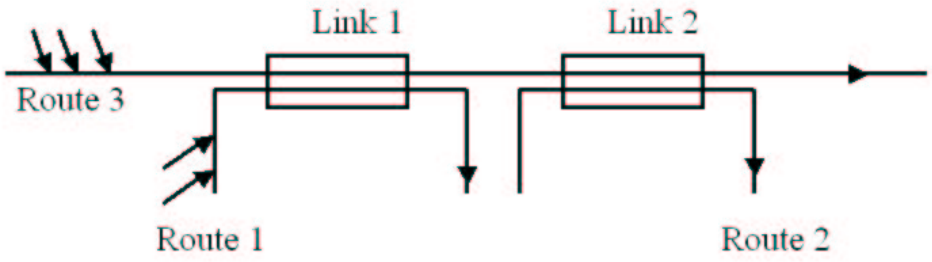

Figure 3.1: Linear network model

Another basic network is an extension of the linear network, the grid network, Figure 3.2. It consists of several horizontal routes r_k and vertical routes r_l, (in the linear network case, there are only one horizontal route: the longest route traverses all links).

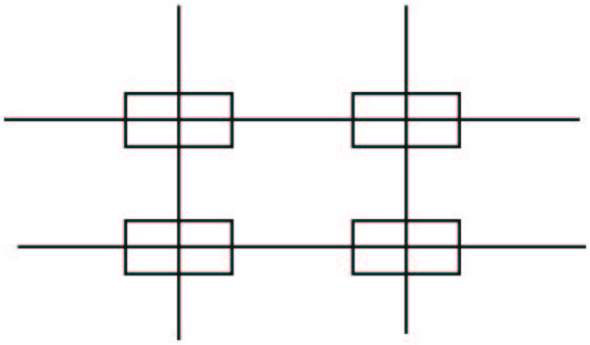

Figure 3.2: Grid network model

The third basic network most often studied is the cyclic network in Figure 3.3 . In this simple cyclic network, it consists of 6 links and 6 routes, with each route traverses 3 links in a symmetric fashion.

Information flows arrive to route r according to a Poisson process with rate λ_r; and the flow's volume (i.e. the size of the file that will be transmitted) is an iid exponential random variable with mean v_r^{-1}. On each route, an arriving information flow will be

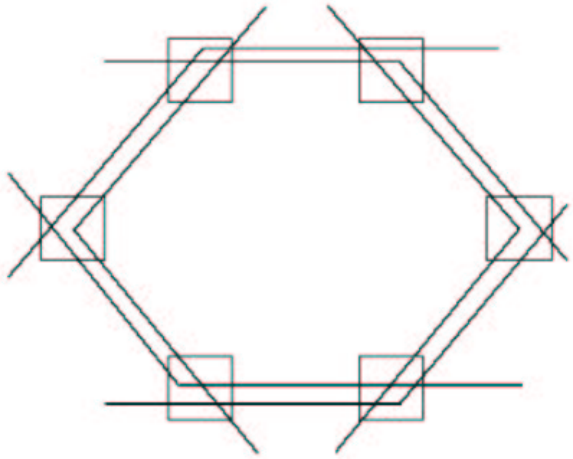

Figure 3.3: Cyclic network model

immediately transmitted through the link, no matter whether there is any other flows being transmitting. In other words, the flows are transmitted simultaneously upon their arrival. The above transmission mechanism can be well modelled by a traditional $M/M/1$ Processor Sharing (PS) queue in that the service capacity (the bandwidth here) is equally shared among the present ongoing information flows on that route, with the bandwidth Λ_r being determined by the bandwidth allocation rule (defined later).

Note that the difference of this type of communication network with the traditional queueing network lies only in that the transmitting information flows in route 3 *simultaneously consume* the link capacity on both link 1 and 2 which lie on the path of that route. In other words, every bit of the flow that has been transmitted through link 1 will immediately goes to link 2 and be transmitted as if the link 1 and 2 are seamlessly linked; rather than being transmitted by only one of the link 1 and 2 at a time, like a customer being served by two bank tells one by one. Thus the bandwidth for route 3 is in effect restricted to be the minimum of bandwidth given by link 1 and 2.

The number of information flows being transmitting on each route fluctuates. It

increases when a new flow arrives, decreases when a flow is completely transmitted through the link(s) on the its path. Same as for the traditional queueing network, one of the fundamental questions concerning to evaluating the network performance is the stationary distribution of the number of the ongoing transmitting flows in the network (the analogy to the number of queueing customers in a queueing network). At time $t > 0$, let $n_r(t)$ denote the number of flows that are currently transmitting on route r, and $n(t) = \{n_r(t) : r \in R\}$ be the vector of the numbers of ongoing flows in the network. We are interested in the probability distribution of the $n(t)$ in the long run.

3.2 Bandwidth allocation rule

We now consider a remaining issue that completes the construction of the network, that is how the link allocates its bandwidth capacity to each route that traverses through it. Each link $l \in L$ has a bandwidth capacity $C_l > 0$, that is the volume of information flow that can be transmitted through the link per unit of time. The bandwidth capacity of each link is allocated to the routes that traverse through this link according to some *dynamic* bandwidth allocation rules in the sense that the rule re-adjusts the allocation as the number of transmitting information flows on each route fluctuates.

In particular, let $\Lambda(n(t)) = \{\Lambda_r(n(t)) : r \in R\}$ denote the bandwidth allocated to each route, determined by the generic bandwidth allocation rule, with $\Lambda_r(n(t))$ be the amount of bandwidth allocated to route r at time t when the numbers of transmitting flows on all routes in the network are $n(t) = \{n_r(t), r \in R\}$. Here we implicitly assume the allocation rule depends only on the number of transmitting flows on each route at time t, which may ignore some realistic consideration, but is widely adopted in the present literatures.

People have developed some bandwidth allocation rules that achieve these two goals as

much as possible, such as the max-min fairness rule (Bertsekas and Gallager 1992), which maximizes the minimum bandwidth allocated to each route such that to some extent the minimum service rate is guaranteed.

Later on Kelly proposed another fairness rule, the proportional fairness rule, that mathematically corresponds to an optimization model that determines the bandwidth allocations. Now almost all bandwidth allocation could be determined by solving this optimization model as extended by Mo and Wolrand (2000), including some rules developed after Kelly's rule. That optimization model reads:

$$\begin{aligned} &\text{Max} \sum_{r \in R} U(n_r, \Lambda_r) \\ &\text{subject to} \sum_{r \in R(l)} \Lambda_r \leq C_l \text{ for } l \in L \\ &\Lambda_r = 0, \text{ if } n_r = 0 \end{aligned}$$

Kelly's proportional fairness rule corresponds to the logarithm utility maximization function.

Given n_r's , the bandwidth allocation rule based on this optimization model is aimed to maximize the overall utility of the network. The constrain simply states that the bandwidth allocations cannot exceed the link's capacity. Therefore the specific forms of the utility functions differentiate those various bandwidth allocation rules. For example Kelly's proportional fairness rule maximizes the following utility function:

$$\text{Max} \sum_{r \in R} n_r log(\Lambda_r / n_r)$$

That reads the rule maximizes the total benefit all over the flows in the network by assuming each flow possess logarithm utility upon the bandwidth it shares.

Mo and Walrand (2000) developed a very general utility function, the α- proportional fairness rule, that includes many specific cases we have mentioned. It takes the form as

following:

$$\text{Max } \sum n_r \frac{\Lambda_r^{1-\alpha}}{1-\alpha}$$

with α be an adjustable parameter. This framework includes many bandwidth allocation rules, such as the max-min fairness allocation as $\alpha \to \infty$, Kelly's proportional fairness rule as $\alpha \to 0$, and the potential delay allocation as $\alpha \to 2$.

It should be mentioned here that the determination of the bandwidth for each route by solving the optimization problem is conducted in a static context in that the number of currently transmitting flows on each route is pre-fixed. Although it is hard to believe this scheme could really be implemented in a realistic network, it does not impair the fundamental study of the network (see Ye et.al 2003). Thus to study the dynamic behavior of the network (that is the fluctuation of the number of transmitting flows), we take this mechanism as granted, that is the bandwidth allocation is immediately determined by re-solving the optimization problem once the number of transmitting flows on any route fluctuates, and remains unchange until the next fluctuation occurs.

For the three basic networks we have presented before, we can obtain their weighted α proportional fairness bandwidth allocation in a close form by solving the optimization problem. These three network cases will be used to test our approximation algorithm later in the chapter of numerical study, thus it is worth here to derive their bandwidth allocations respectively. (Note that for Kelly's proportional fairness rule, it is simply that $w_i = 1, \alpha = 1$)

Here we list the bandwidth allocation only for the linear network with unit capacity links (that is $C_l = 1$), (more on the bandwidth allocations for the grid, cyclic network later). The weighted α proportional fairness bandwidth are:

$$\begin{aligned}\Lambda_0 &= \frac{(w_0 n_0^\alpha)^{1/\alpha}}{(w_0 n_0^\alpha)^{1/\alpha} + (\sum w_r n_r^\alpha) 1/\alpha} \\ \Lambda_r &= 1 - \Lambda_0\end{aligned}$$

where

Λ_0: bandwidth rate allocated to the go-through-all-links route R_0

Λ_r: bandwidth rate allocated to each go-through-one-link route R_r

n_r: number of transmitting information flows on route R_r

w_r: weighting factor for route R_r

For the grid network as well with unit capacity links, the α proportional fairness bandwidth rates are derived in the same way:

$$\begin{aligned}\Lambda_k &= \frac{(\sum_{k=1}^K w_k n_k^\alpha)^{1/\alpha}}{(\sum_{k=1}^K w_k n_k^\alpha)^{1/\alpha} + (\sum_{l=1}^L w_l n_l^\alpha)^{1/\alpha}} \\ \Lambda_l &= 1 - \Lambda_k\end{aligned}$$

with n_k be the number of transmitting flows on the horizontal route R_k, and n_l be that on the vertical route R_l.

For the cyclic network, which consists of $2L$ links and $2L$ routes of length L, and route l crosses link $l+1, \ldots, l+L$, the α proportional fairness allocation is:

$$\Lambda_r = \frac{(w_l n_l^\alpha + w_{l+L} n_{l+L}^\alpha)^{1/\alpha}}{\sum_{l=1}^L (w_l n_l^\alpha + w_{l+L} n_{l+L}^\alpha)^{1/\alpha}},\ 1 \le l \le 2L$$

3.3 Stationary distribution

Up to now most of studies are focused on the static analysis of the network, that is to assume the number of transmitting flows on each route is fixed during the period of

study, such as the determination of the bandwidth allocation. The dynamic behavior of the network is not yet studied. What if the state (the number of transmitting flows on each route) fluctuates? How does this fluctuation affect the network performance?

As the same for the traditional queueing system, the state of the system fluctuates with the time. Given the stability condition $\sum_{r \in R(l)} \rho_r < C_l$, which guarantees the long run stationary status of the network exists under the various fairness bandwidth allocation rules, we are interested in the long run behavior of the system, that is the stationary distribution of these states.

Continuous time Markov chain technique is still the root to derive any exact solution of a queueing system other than various approximation methods. The communication network studied here is also the case.

By assuming Poisson arrivals on each route, and iid exponential flow volume, the state of the system can be modelled as a continuous time Markov chain as follows:

The state space is well defined, that is $\vec{n}(t) = \{n_r(t), r \in R\}$. The transition rates depend on the state through the bandwidth allocation rule ($\Lambda_r(n)$), as well as on the arrival rates (λ_r), mean flow volume (μ_r^{-1}). Specifically (see Ye 2003):

$$q(n, n') = \begin{cases} \lambda_r & n' = n + e_r \\ \mu_r \cdot \Lambda_r(\vec{n}) & n' = n - e_r \text{ and } n_r \geq 1 \\ 0 & \text{otherwise} \end{cases}$$

where $\Lambda_r(\vec{n})$ is the bandwidth rate allocated to route R_r, depending on the states of the system ($\vec{n}$), and e_r is a vector having the same dimension with $\vec{n}(t)$, with 1 at the r-th position and 0 for all the others.

For the linear network under Kelly's proportional fairness allocation rule, the transition

rates from state $\vec{n}$ to $\vec{n}'$ are:

$$\begin{aligned} q(\vec{n}, \vec{n} + e_i) &= \lambda_i \\ q(\vec{n}, \vec{n} - e_0) &= \mu_0 \frac{n_0}{n_0 + \sum_i^L n_i} \\ q(\vec{n}, \vec{n} - e_i) &= \mu_i (1 - \frac{n_0}{n_0 + \sum_i^L n_i}) \end{aligned}$$

Note that $\frac{n_0}{n_0 + \sum_i^L n_i}$ is the bandwidth rate allocated to the longest route R_0, i.e. Λ_0 in the linear network case, where $\alpha = 1, w_i = 1$. (see section 3.2)

For the grid network with the same settings, let the state be $\{(x_k, y_l)\}$, representing the pair of horizontal and vertical routes, the transition rates are then:

$$\begin{aligned} q\left((x, y) \to (x + e_k, y)\right) &= \lambda_k \\ q\left((x, y) \to (x - e_k, y)\right) &= \mu_k \frac{\sum x_k}{\sum x_k + \sum y_l} \end{aligned}$$

It is then proved by Massoulie and Roberts (1998) that for the linear network with unit capacity link ($C_l = 1$) and under Kelly's proportional fairness rule, if the normal offered load condition is satisfied, that is: $Max_{1 \le i \le L}\ \rho_0 + \rho_i < 1$, where $\rho_i = \lambda_i / \mu_i$ is the traffic load on route i, then the process $n(t)$ is reversible, and has stationary distribution:

$$\pi(n_0, \ldots, n_L) = C^{-1} \binom{\sum n_i}{n_0} \prod \rho^{n_i}$$

$$where\ C = \frac{(1 - \rho_0)^{L-1}}{\prod_1^L (1 - \rho_0 - \rho_i)}$$

where C is the normalization constant.

For the grid network, the similar result could be derived by solving the balance equations of the continuous time Markov chain. The Markov process (x_k, y_l) is reversible, and with the stationary distribution:

$$\pi(x, y) = C^{-1} \binom{\sum x_k + \sum y_l}{\sum x_k} \prod \rho_k^{x_k} \prod \rho_l^{y_l}$$

$$where\ C = \sum_{k=1}^{K}\sum_{l=1}^{L} \frac{\rho_k^{K-1}\rho_l^{L-1}}{\prod_{k'\neq k}(\rho_k - \rho_{k'})\prod_{l'\neq l}(\rho_l - \rho_{l'})} \frac{1}{1-\rho_k-\rho_l}$$

However, for the networks other than the simple linear and grid network, to derive the close form solution by solving the balance equations of the Markov chain is impossible. Even for the linear and grid network, if other bandwidth allocation rule than Kelly's proportional fairness rule is adopted, the close form solution is as well impossible. See comments by Massoulie and Roberts (1998) and Bonald and Massoulie (2000), and Fayolle et.al (2001)'s review.

Even trying to derive the numerical solution by solving the Markov transition rate matrix is impractical. One of the reasons is that the size of the transition rate matrix increases exponentially with the number of routes and number of states. For example, for a simple linear network with K routes, and let the state (the number of transmitting flows) on each route be the same: $\{0, 1, \ldots, N-1\}$. Thus the state of the system as a whole (that is to consider the routes jointly) is a K dimensional vector $(n_1, \ldots, n_K)$ with $n_i = 0, 1, \ldots, N-1$. The size of the transition rate matrix is then N^K *by* N^K. When N and K are too large, it far exceeds the capability of the computer to solve the matrix.

3.4 Chapter summary

In this chapter, we formulate the framework of the communication network under study. It consists of a set of routes and links, on which the very important bandwidth allocations are determined by solving an optimization problem. Next the state of the system, namely the number of transmitting flows on each route, is modelled by a continuous time Markov chain, with the transition rates be determined by the arrival rates and bandwidth allocations exclusively. By solving the balance equations of the Markov chain, the closed form solution to the stationary distribution of the state of the system is derived for only a few

networks up to date. In general, by solving the Markov transition rate matrix numerically is impractical as well.

Chapter 4

Approximation Procedure

It is impossible to derive a closed form solution for more general network cases, and even impractical to solve the equation system involving the huge transition rate matrix. As for the tradition queueing network, approximation methods are more often considered as an effective way to deal with various complex queueing networks. Among them, the decomposition method is probably the best. Thus in this section, we propose an approximation algorithm in a decomposition fashion, that approximates the exact numerical solution of the network.

4.1 Modelling processor sharing queues

In particular, the algorithm decomposes the network into disjoint transmission routes with each one being represented by an $M/M/1$ processor sharing (PS) queue. Different from the traditional queueing networks, where a set of queues are interrelated by the effective arrival process and departure processes, this collection of processor sharing queues have independent arrival and departure processes, but are interrelated by their service rates.

Specifically, consider a linear network for example (see Figure 4.1, Figure 4.2). As

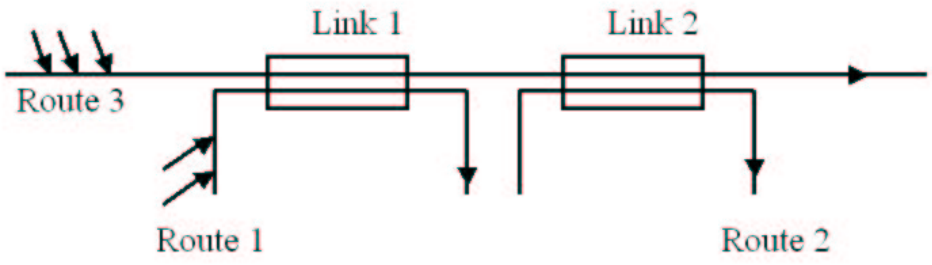

Figure 4.1: Linear network model

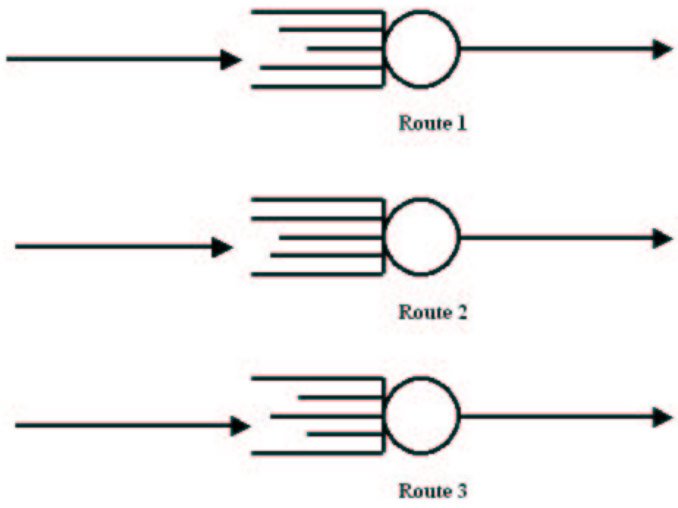

Figure 4.2: Modelling as a set of PS queues

have been mentioned in the Section 3.1, each route in the network could be considered as a processor sharing queue. Information flows arrive on each route according to a Poisson process, simultaneously be served (processor sharing), and then leave (flow be transmitted, connection terminated). On the path of the Route 3, the link 1 and 2 are in effect seamlessly linked (simultaneous resource consumption), thus Route 3 has essentially only one server with the service rate being determined by the smaller bandwidth given by link 1 and 2.

The service rate of each queue (each route) is equivalent to the bandwidth allocated to that route. Note that every time when the number of transmitting flows on each route fluctuates (due to a new arrival or a finished transmission), the bandwidth on all routes are totally re-allocated by solving that optimization problem given the altered state $n(t) = \{n_1(t), \ldots, n_k(t)\}$.

The bottom line of this approximation algorithm is the decomposition principle, that means to analyze each queue individually as if all the other queues do not exist. Since the arrival process of each queue is an independent Poisson process, the only remaining uncertainty of each queue is its service rate, which dynamically fluctuates with the state of the system. Thus in our algorithm, each queue's service rate is dynamically approximated by taking into account the interdependence of the bandwidth allocations on all the other transmission routes. In particular, the transmission bandwidth on each route is calculated conditioning on the concurrent status of the queueing lengths on all the other routes in the network.

4.2 Approximation algorithm

We first define the following notations used below to describe the procedure:

$$\begin{aligned}
S_{-r} &= \text{the set of routes} \{1, 2, \cdots, r\}, r = 1, 2, \cdots, M; \\
S_{+r} &= \text{the set of routes} \{r+1, r+2, \cdots, M\}, r = 1, 2, \cdots, M-1; \\
\pi_{r|S_{+r}}(n_r|S_{+r}) &= \text{the conditional marginal distribution of } n_r(t) \\
&\quad \text{for given numbers of flows on routes in } S_{+r}, r = 1, 2, \cdots, M-1; \\
\pi_{S_{-r}|S_{+r}}(n_1, \cdots, n_r|S_{+r}) &= \text{the conditional joint distribution of } (n_1(t), \cdots, n_r(t)) \\
&\quad \text{for given numbers of flows on routes in } S_{+r}, r = 1, 2, \cdots, M-1; \\
\pi(n_1, n_2, \cdots, n_M) &= \text{the stationary distribution of the system } n(t);
\end{aligned}$$

Firstly, we order the set of queues (which queue or route comes first does not matter because of the symmetry of the system), and divide them at queue r into two parts: the S_{-r}, S_{+r}. Then we compute the conditional distribution $\pi(n_1, \ldots, n_r|n_{r+1}, \ldots, n_M)$, given the state on the latter set of queues, i.e. $S_{+r}(= n_{r+1}, \cdots)$. Here the joint distribution is calculated simply by the conditional probability formula:

$$\pi(n_1, \ldots, n_r|S_{+r}) = \pi(n_1, \ldots, n_{r-1}|n_r, S_{+r})\pi(n_r|S_{+r})$$

where the marginal distribution $\pi(n_r|S_{+r})$ needs to be calculated first.

For example, we first calculate $\pi(n_1|n_2, \cdots)$ using a given formula (more on this formula in Step 1 of the algorithm). Then we need to find a way to calculate $\pi(n_2|n_3, \cdots)$, i.e. $\pi(n_2|S_{+2})$, which is where our algorithm contributes (Step 2-A). Finally, the joint distribution of the first two routes, given the states on S_{+2} fixing, is simply:

$$\pi(n_1, n_2|S_{+2}) = \pi(n_1|n_2, \cdots)\pi(n_2|S_{+2})$$

Working in this fashion, of course a backward fashion, we can finally obtain $\pi(n_1, \cdots, n_M)$, the unconditional distribution of the system. The key point lies in how to calculate the

$\pi(n_r|S_{+r})$. Below we provide the details for each step of the algorithm. (Next we use *queue* and *route* interchangeably)

Step 1 Compute $\pi_{1|S_{+1}}(n_1|S_{+1})$

We begin from the first queue. Here we will also meet the most important part of our algorithm.

For a single $M/M/1$ processor sharing queue, where the arrival process is a Poisson process with rate λ, the service rate is the exponential random variable with mean $E[x]$, or rate $\mu = E[x]^{-1}$, the stationary distribution is easy to find:

$$P(n = k) = C^{-1}\frac{\lambda^k}{\mu^k}$$

where C is the normalization constant.

However in our case, even we fix the state on routes $2, \ldots, M$, every time when the state on route 1 fluctuates, the bandwidth has to be re-allocated. Thus the service rate for queue 1 is not fixed as μ in the single $M/M/1$ processor sharing queue case. Therefore when $n_1 = i$, and the state on all the other routes fixed, the service rate $\mu_i = \min\{\Lambda_1(i, \cdot), C_l\}$ (C_l is the capacity of the link where route 1 traverses.)

Thus from the result of Roberts (2004), when the state of route $2, \ldots, M$ fixed, the stationary distribution of queue 1 is readily derived from the above M/M/1 processor sharing queue's formula with a replacement of μ:

$$P(n = k) = C^{-1}\frac{\lambda^k}{\prod_{i=1}^{k}\Lambda_1(i, \cdot)}$$

Using the above result, we can formulate the first step of our algorithm as follows: For given the number of flows on route in S_{+1}, we approximate (in the sense that we artificially fixed the state on all the other routes) the dynamic behavior of $n_1(t)$ as a

processor sharing queue with arrival rate λ_1 and service rate $\Lambda_1(n_1, S_{+1})$. Specifically, we have

$$\pi_{1|S_{+1}}(n_1|S_{+1}) = \frac{\lambda_1^{n_1}}{\prod_{k=0}^{n_1}\Lambda_1(k,S_{+1})}\left(\sum_{n=0}^{\infty}\frac{\lambda_1^{n}}{\prod_{k=0}^{n}\Lambda_1(k,S_{+1})}\right)^{-1},\ n_1 = 0,1,\cdots,$$

The latter term of the right hand is just a normalization factor.

Step 2-A Compute $\pi_{2|S_{+2}}(n_2|S_{+2})$

For route 2, we again want to model the dynamic behavior of $n_2(t)$ as a processor sharing queue as we did for route 1. However, even we fix the state on routes in S_{+2}, the service rate when the state on route 2 is $n_2 = i$, cannot be uniquely determined since the number of flows on route 1 still randomly fluctuates. Thus here we make the state of route 1 fixed in the sense of taking the expectation of Λ_2 upon the fluctuation of n_1. (see Figure 4.3) Specifically we approximate the service rate of route 2 with the following mean value, based upon the probability distribution $\pi_1(n_1|S_{+1})$ which is what we have just calculated:

$$E[\Lambda_2(\cdot, n_2, S_{+2})] = \sum_{n_1=0}^{\infty}\Lambda_2(n_1, n_2, S_{+2})\pi_{1|S_{+1}}(n_1|n_2, S_{+2})$$

Having the service rate, we can apply the single processor sharing queue formula on route 2 to derive the marginal distribution of n_2:

$$\pi_{2|S_{+2}}(n_2|S_{+2}) = \frac{\lambda_2^{n_2}}{\prod_{k=0}^{n_2}E[\Lambda_2(\cdot,k,S_{+2})]}\left(\sum_{n=0}^{\infty}\frac{\lambda_2^{n}}{\prod_{k=0}^{n}E[\Lambda_2(\cdot,k,S_{+2})]}\right)^{-1}, n_2 = 0,1,\cdots$$

Step 2-B Compute $\pi_{S_{-2}|S_{+2}}(n_1, n_2|S_{+2})$

From Step 1 where we calculated the conditional distribution of n_1 given n_2 and S_{+2}, and Step 2-A where we obtain the marginal distribution of n_2 (given S_{+2} as well), the joint distribution of n_1 and n_2 is readily obtained from the conditional probability formula:

$$\pi_{S_{-2}|S_{+2}}(n_1, n_2|S_{+2}) = \pi_{1|S_{+1}}(n_1|n_2, S_{+2})\pi_{2|S_{+2}}(n_2|S_{+2})$$

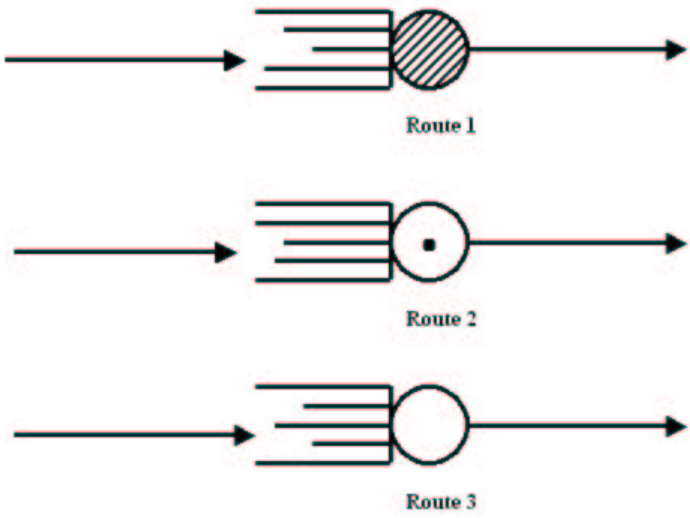

Figure 4.3: Taking $E[\Lambda_2(\cdot, n_2, S_{+2})]$ upon fluctuation on route 1

.

Thus we have built the key part of our approximation algorithm. The principle of fixing S_{+2}, and fixing n_1 by taking expectation, then computing the marginal distribution of n_2 is then readily extended to latter routes. See Figure 4.4. For example for a 3 routes network, the above procedures (step 1 and step 2-AB) are conducted for each state of n_3 repeatedly, that is for $n_3 = 0$, we calculate $\pi(n_1, n_2|n_3 = 0)$ using the above procedure; then for $n_3 = 1$, we calculate $\pi(n_1, n_2|n_3 = 1)$, and so forth. When they are readily prepared, we can obtain the service rate $\Lambda_3(\cdot, \cdot, n_3)$ for route 3 through fixing n_1, n_2 in terms of taking their expectation:

$$\Lambda_3(\cdot, \cdot, n_3) = \sum_{n_1, n_2} \Lambda_3(n_1, n_2, n_3)\pi(n_1, n_2|n_3)$$

Then the single processor sharing queue formula is applied on route 3 to derive its marginal distribution $\pi(n_3)$. Finally we multiply it with the calculated conditional distribution $\pi(n_1, n_2|n_3)$ to obtain the joint distribution of the 3 queue system as a whole.

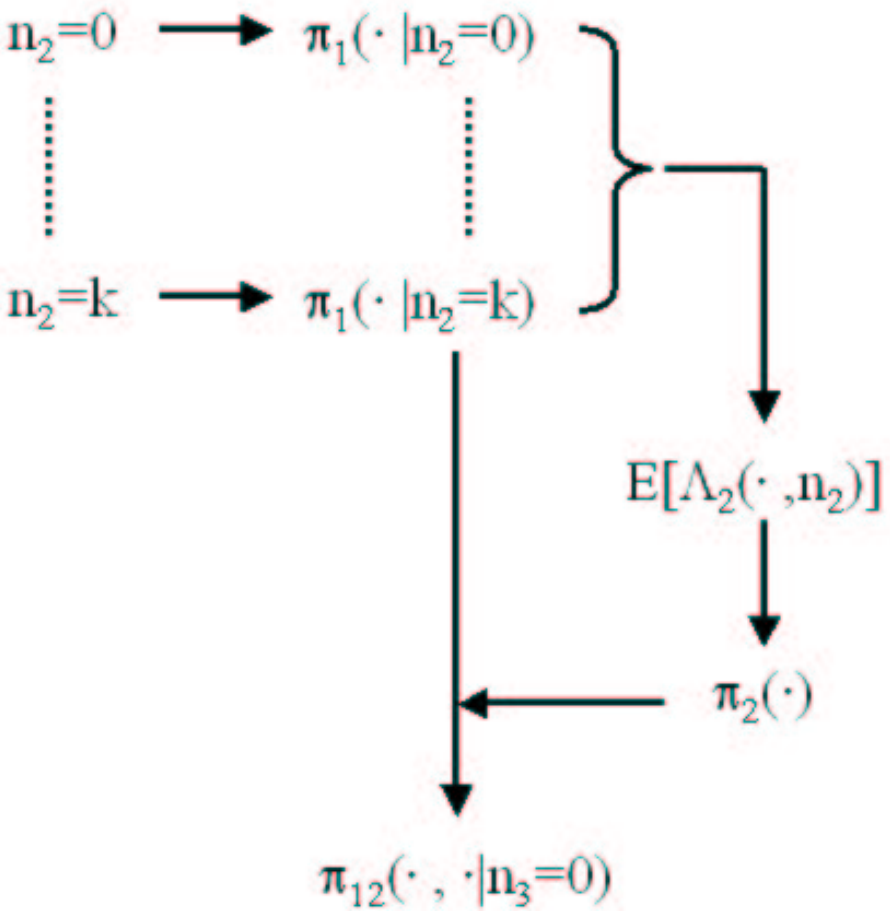

Figure 4.4: Flow chart of the Algorithm

The above procedure can be formulated into a general form:

Step (r+1)-A Compute $\pi_{r+1|S_{+(r+1)}}$ after obtaining $\pi_{S_{-r}|S_{+r}}$ in *Step r-B*

We approximate the service rate on route $r+1$ with its expected value:

$$\begin{aligned} & E[\Lambda_{r+1}(\cdots, n_{r+1}, S_{+(r+1)})] \\ = & \sum_{n_1,\cdots,n_r} \Lambda_{r+1}(n_1,\cdots,n_r,n_{r+1},S_{+(r+1)})\pi_{S_{-r}|n_{r+1},S_{+(r+1)}}(n_1,\cdots,n_r|n_{r+1},S_{+(r+1)}) \end{aligned}$$

Then compute the marginal distribution of queue $r+1$ based on its service rate exclusively:

$$\begin{aligned} & \pi_{r+1|S_{+(r+1)}}(n_{r+1}|S_{+(r+1)}) \\ = & \frac{\lambda_{r+1}^{n_{r+1}}}{\prod_{k=0}^{n_{r+1}} E[\Lambda_{r+1}(\cdots,k,S_{+(r+1)})]} \left(\sum_{n=0}^{\infty} \frac{\lambda_{r+1}^{n}}{\prod_{k=0}^{n} E[\Lambda_{r+1}(\cdots,k,S_{+(r+1)})]}\right)^{-1} \end{aligned}$$

$$n_{r+1} = 0, 1, \cdots,$$

Step (r+1)-B Compute $\pi_{S_{-(r+1)}|S_{+(r+1)}}(n_1,\cdots,n_{r+1}|S_{+(r+1)})$

The joint distribution $\pi_{S_{-(r+1)}|S_{+(r+1)}}(n_1,\cdots,n_{r+1}|S_{+(r+1)})$ uses what have been calculated already:

$$\begin{aligned} & \pi_{S_{-(r+1)}|S_{+(r+1)}}(n_1,\cdots,n_{r+1}|S_{+(r+1)}) \\ = & \pi_{S_{-r}|S_{+r}}(n_1,\cdots,n_r|n_{r+1},S_{+(r+1)})\pi_{r+1|S_{+(r+1)}}(n_{r+1}|S_{+(r+1)}) \end{aligned}$$

Finally, in **Step M-B** we will obtain $\pi(n_1, n_2, \cdots, n_M)$.

In summary, the algorithm works in the following way:

Step 1: Compute $\pi_{1|S_{+1}}(n_1|S_{+1})$

$$\pi_{1|S_{+1}}(n_1|S_{+1}) = \frac{\lambda_1^{n_1}}{\prod_{k=0}^{n_1}\Lambda_1(k,S_{+1})}\left(\sum_{n=0}^{\infty}\frac{\lambda_1^{n}}{\prod_{k=0}^{n}\Lambda_1(k,S_{+1})}\right)^{-1},\ n_1 = 0,1,\cdots,$$

Step 2-A: Compute $\pi_{2|S_{+2}}(n_2|S_{+2})$

$$E[\Lambda_2(\cdot, n_2, S_{+2})] = \sum_{n_1=0}^{\infty} \Lambda_2(n_1, n_2, S_{+2})\pi_{1|S_{+1}}(n_1|n_2, S_{+2})$$

$$\pi_{2|S_{+2}}(n_2|S_{+2}) = \frac{\lambda_2^{n_2}}{\prod_{k=0}^{n_2} E[\Lambda_2(\cdot, k, S_{+2})]} \left(\sum_{n=0}^{\infty} \frac{\lambda_2^n}{\prod_{k=0}^{n} E[\Lambda_2(\cdot, k, S_{+2})]} \right)^{-1}, n_2 = 0, 1, \cdots$$

Step 2-B: Compute $\pi_{S_{-2}|S_{+2}}(n_1, n_2|S_{+2})$

$$\pi_{S_{-2}|S_{+2}}(n_1, n_2|S_{+2}) = \pi_{1|S_{+1}}(n_1|n_2, S_{+2})\pi_{2|S_{+2}}(n_2|S_{+2})$$

Step (r+1)-A Compute $\pi_{r+1|S_{+(r+1)}}$

$$\begin{aligned} & E[\Lambda_{r+1}(\cdots, n_{r+1}, S_{+(r+1)})] \\ = & \sum_{n_1, \cdots, n_r} \Lambda_{r+1}(n_1, \cdots, n_r, n_{r+1}, S_{+(r+1)})\pi_{S_{-r}|n_{r+1}, S_{+(r+1)}}(n_1, \cdots, n_r|n_{r+1}, S_{+(r+1)}) \end{aligned}$$

$$\begin{aligned} & \pi_{r+1|S_{+(r+1)}}(n_{r+1}|S_{+(r+1)}) \\ = & \frac{\lambda_{r+1}^{n_{r+1}}}{\prod_{k=0}^{n_{r+1}} E[\Lambda_{r+1}(\cdots, k, S_{+(r+1)})]} \left(\sum_{n=0}^{\infty} \frac{\lambda_{r+1}^n}{\prod_{k=0}^{n} E[\Lambda_{r+1}(\cdots, k, S_{+(r+1)})]} \right)^{-1} \end{aligned}$$

$$n_{r+1} = 0, 1, \cdots,$$

Step (r+1)-B Compute $\pi_{S_{-(r+1)}|S_{+(r+1)}}(n_1, \cdots, n_{r+1}|S_{+(r+1)})$

$$\begin{aligned} & \pi_{S_{-(r+1)}|S_{+(r+1)}}(n_1, \cdots, n_{r+1}|S_{+(r+1)}) \\ = & \pi_{S_{-r}|S_{+r}}(n_1, \cdots, n_r|n_{r+1}, S_{+(r+1)})\pi_{r+1|S_{+(r+1)}}(n_{r+1}|S_{+(r+1)}) \end{aligned}$$

Up to $r + 1 = M$, we obtain $\pi(n_1, n_2, \cdots, n_M)$.

END

Note that in the formula that computes the marginal distribution of n_r, we in fact replace the normalization constant C with a finite truncation sum instead of the infinity sum:

$$\begin{aligned}\pi_{r|S_{+r}}(n_r|S_{+r}) &= \frac{\lambda_r^{n_r}}{\prod_{k=0}^{n_r} E[\Lambda_r(\cdots,k,S_{+r})]} C^{-1} \\ &= \frac{\lambda_r^{n_r}}{\prod_{k=0}^{n_r} E[\Lambda_r(\cdots,k,S_{+r})]} \left(\sum_{n=0}^{N} \frac{\lambda_r^{n}}{\prod_{k=0}^{n} E[\Lambda_r(\cdots,k,S_{+r})]} \right)^{-1}\end{aligned}$$

The finite sum is always smaller than the infinite sum. Thus when dividing by the underestimated normalization constant, the marginal probability distribution is in effect consistently overestimated slightly, that is every probability value of the state will be overestimated by a same marginal percentage. And the effect will be passed on to the final joint distribution by the intermediate multiplication operations. We anticipate that the overestimation effect will diminish as N includes more and more number of states. (that $N \to \infty$ in effect) In the section of numerical examples, we will verify this judgement.

4.3 Chapter Summary

In this chapter, we proposed an approximation algorithm to compute the stationary distribution of the communication network numerically. We model the network as a set of processor sharing queues, which have independent arrival processes, only interrelated through their service rates which are determined by the bandwidth allocations. The algorithm works in a decomposition's fashion: It first computes the conditional distribution of the first set of queues given the states on the second set of queues, then it provides the basis to compute the marginal distribution of second set of queues. By multiplying the marginal and conditional distributions, we get the joint distribution of the system as a whole. Of course the computational time increases exponentially with the number of states of the system.

Chapter 5

Gibbs Sampling Method

In the cases where close form solution of the network are available, we can compare our approximation results with the close form solutions directly. However such cases are rare, only the linear network and grid network with unit capacity links. In absence of the closed form solutions, we have to find some benchmark to evaluate our approximation solutions.

Simulation is the most often used benchmark to evaluate a queueing system. It does a good job to produce the mean value estimations for the system. However in our case, what we are computing are not several mean values, but N^R probability values for the total states of the system. That is, essentially N^R parameters (e.g. mean value) need to be evaluated. As verified by our numerical experiments (see the chapter of the numerical examples), the simulated results are not satisfying in this situation due to the huge number of values it has to evaluate. Marginal improvement of the simulation results requires huge extra computational effort, thus it is inefficient, and impractical.

We notice that the simulation method is essentially a sampling method by mimicking the realistic physical system. Specifically, we generate the periods of time when the system stays in state i, sum them up, then divide it by the whole period of time the system operates to obtain the long run average percentage of time in state i, taking it as

the stationary probability value of the system in state i. The merit of this method is that we do not require any information about the probability distribution of the system before doing the simulation.

If fortunately we have some information about the distribution of the system beforehand, then we could resort to some computationally intensive statistical methods (i.e. the Monte Carlo Markov Chain technique, MCMC) that have become very popular since the last decade. These methods provide another simulation (i.e. sampling) scheme to find the probability distribution of the system, but work in a much more efficient and effective manner than the engineer's simulation method. In particular, these methods directly generate (sample) the states repeatedly to establish a large enough pool of samples (i.e. states), then count the frequency of every state $\vec{i}$, taking it as the stationary probability of state $\vec{i}$.

In this chapter we will use a very effective method of the MCMC toolbox, the Gibbs sampling method, to obtain the benchmark solution for the network. It thus helps the comparison of the numerical solutions for any general network cases where the closed form solutions are not available. Next we just list the final Gibbs sampling algorithm, the detailed deduction is left in the thesis.

Next we will first introduce the generic Gibbs sampling method, and then some modifications, in order to make it more efficient for our cases.

5.1 Generic Gibbs sampling method

The generic Gibbs sampling method samples the states, requiring only the information about the conditional distribution of the system. Specifically, the Gibbs sampling method

is to sample a random vector

$$U = (U_1, \ldots, U_k)$$

having a joint distribution $F(u)$ in an iterative fashion. Here $F(u)$ is either unknown or very complicated, but for each i, the conditional distribution $F(u_i|u_1, \ldots, u_{i-1}, u_{i+1}, \ldots, u_k)$, or in short

$$U_i|(U_1, \ldots, U_{i-1}, U_{i+1}, \ldots, U_k)$$

is known and relatively easy to sample from. (*Sampling* simply means to transform a computer generated random number (a uniform r.v) into any one of the feasible values of the random variable by using its probability distribution function.)

The sampling process works in the following way:

Step 0: Start from an arbitrary value $u_0 = (u_{01}, \ldots, u_{0k})$ from the feasible value set of the U vector. (Here the first subscript 0 denotes the 0-th cycle of the procedure, and the second one denotes the 1-th element of the vector.) Let it be the outcome of the first round sampling.

Step 1: Compute the first cycle:

1. First, u_{01} is updated by sampling U_1, with realized value u_{11}, from the conditional distribution:

$$U_1|(U_2 = u_{02}, \ldots, U_k = u_{0k})$$

2. Then update u_{02} by simulating U_2, with realized value u_{12}, from the condition distribution

$$U_2|(U_1 = u_{11}, U_3 = u_{03}, \ldots, U_k = u_{0k})$$

Note that now U_1 is using the updated value u_{11}, not u_{01}.

3. Continue in this fashion until u_{0k} is updated by sampling U_k, with realized value u_{1k}, from the conditional distribution

$$U_k|(U_1 = u_{11}, \ldots, U_{k-1} = u_{1,k-1})$$

Note that at any point, we always use the most recently updated value for any random variable, so that when updating u_{05}, we use $(u_{11}, \ldots, u_{14})$ and $(u_{06}, \ldots, u_{0k})$.

At this point, the first cycle of the Gibbs sampling procedure is finished. We record the result as:

$$\vec{u_1} = (u_{11}, \ldots, u_{1k})$$

Here the $\vec{u_1}$ is the realized value of the random vector U. Since this is the first round sampling of U, we record U as $U^{(1)}$. If we repeat this procedure many times, we could expect u_1s are not necessarily the same, but we use the same $U^{(1)}$ to denote these realized random values from the first round sampling, the same case as for a single random variable.

Step 2: Then the second cycle begins: Further updating u_{11} by sampling U_1 from the conditional distribution

$$U_1|(U_2 = u_{12}, \ldots, U_k = u_{1k})$$

with realized value u_{21}; then updating u_{12} by sampling U_2 from the conditional distribution

$$U_2|(U_1 = u_{21}, U_3 = u_{13}, \ldots, U_k = u_{1,k})$$

with realized value u_{22}. Continue in this fashion, the procedure generates the second cycle

$$u_2 = (u_{21}, \ldots, u_{2k})$$

Step n: Continue in this fashion, the sampler generates the n-th cycle, u_n and the corresponding random vector $U^{(n)}$.

END

Under fairly general conditions, the distribution of the random vector $U^{(n)}$ converges to that of U, the original distribution, that is

$$U^{(n)} \to U$$

for large enough n. In other words if we sample $U^{(i)}$ for a large number of cycles, the procedure will generate a random vector whose distribution is very the same as that of U. We then count the frequency of the vector $(U_1 = s_1, \ldots, U_k = s_k)$ from the sample set $\{\vec{u}_n : n = 1, 2 \ldots\}$ as the best estimation of the probability distribution of U in the steady state:

$$P(\vec{U} = \vec{s}) = \frac{1}{N - M + 1} \sum_{j=M}^{N} I_{\vec{s}}(u_j)$$

$$I_{\vec{s}}(\vec{u}) = \begin{cases} 1 & \text{if } \vec{u} = \vec{s} \\ 0 & \text{otherwise} \end{cases}$$

5.2 Modified Gibbs sampler

We call the conditional distribution $U_i|(U_1, \ldots, U_{i-1}, U_{i+1}, \ldots, U_k)$ a sampler, in the sense that we generate (sample) $u_j = (u_{j,1}, \ldots, u_{j,k})$ from it.

Different samplers can be constructed, but have very distinctive degree of efficiency. Up to now, the best sampler for the case of discrete random variable is due to Liu (1996). Next we will use Liu's sampler to achieve the better efficiency. First suppose the current cycle is j, the sample is $u_j = (u_{j,1}, \ldots, u_{j,k})$, we modify the updating procedure of the generic Gibbs method in cycle j as follow:

Step j: Randomly choose i from $\{1, \ldots, k\}$, and let $\{y_i\}$ be the set of the feasible values of U_i other than $u_{j,i}$, and do a sampling from this set with the probability distribution:

$$P(y_i) = \frac{\pi(y_i|u_{[-i]})}{1 - \pi(u_{j,i}|u_{[-i]})}$$

where $u_{[-i]} = (u_{j,1}, \ldots, u_{j,i-1}, u_{j,i+1}, \ldots, u_{j,k})$,

and $\pi()$ is just the $U_i|(U_1, \ldots, U_{i-1}, U_{i+1}, \ldots, U_k)$.

Suppose the sample we draw is Y. We update $u_{j,i}$ with Y with the following acceptance probability:

$$min\left\{1, \frac{1-\pi(u_{j,i}|u_{[-i]})}{1-\pi(Y|u_{[-i]})}\right\}$$

otherwise $u_{j,i}$ remains unchanged. That reads:

$$u_{j,i} = \begin{cases} Y & \text{with probability } min\left\{1, \frac{1-\pi(u_{j,i}|u_{[-i]})}{1-\pi(Y|u_{[-i]})}\right\} \\ u_{j,i} & \text{otherwise} \end{cases}$$

Now the cycle j finishes, the sample u_j is updated.

Note there is another difference between the generic Gibbs sampling method and the modified Gibbs sampling besides of the sampler aspect of the two methods. In each cycle, the former method updates $u_{j,1}, \ldots, u_{j,k}$ totally and subsequently; while the latter method updates only one ($u_{j,i}$) by randomly choosing from the k candidates. Thus it is expected the latter method will require more cycles.

5.3 Multiple path sampling

Another shortfall of the generic Gibbs sampling method is that we have to sample a large number of cycles in order to count the frequency, and since which point the samples are stationary is not clear. Gelfand (1990) suggested the above Gibbs sampling procedure be conducted on several independent parallel paths simultaneously. This improvement produces faster convergence of the estimators.

In particular, we will run the sampling on m independent parallel paths simultaneously, on each path we sampling n cycles as the generic Gibbs sampling, that is we construct

the following m independent parallel sampling chains:

$$j = 1 : \ u_1^{(1)} \to u_2^{(1)} \ldots \to u_n^{(1)} \to \ldots$$

$$j = 2 : \ u_1^{(2)} \to u_2^{(2)} \ldots \to u_n^{(2)} \to \ldots$$

$$\vdots$$

$$j = m : \ u_1^{(m)} \to u_2^{(m)} \ldots \to u_n^{(m)} \to \ldots$$

where $u_n^{(j)} = (u_{n,1}^{(j)}, \ldots, u_{n,k}^{(j)})$ is the n- th cycle of sample on the j-th independent parallel path.

The probability estimation of each state is then simply averaged over all these paths:

$$P(\vec{U} = \vec{s}) = \frac{1}{(N - M + 1)m} \sum_{j=1}^{m} \sum_{i=M}^{N} I_{\vec{s}}(u_i^{(j)})$$

Now only one question remains: What is the condition sampler $U_i|(U_1, \ldots, U_{i-1}, U_{i+1}, \ldots, U_k)$ in our case? That is exactly the conditional distribution $\pi_{1|S_{+1}}(n_1|S_{+1})$ in the Step 1 of our approximation algorithm (Note that the order of those processor sharing queues does not matter, thus we can put any queue at the first), the conditional distribution of the number of flows on any single route given the fixed numbers of flows on all the other routes. That reads:

$$\begin{aligned} & N_i|(N_1, \ldots, N_{i-1}, N_{i+1}, \ldots, N_k) \sim \pi_{1|S_{+1}}(n_1|S_{+1}) \\ = \ & \frac{\lambda_i^{n_i}}{\prod_{k=0}^{n_i} \Lambda_i(k, S_{+1})} \cdot C^{-1} \end{aligned}$$

where N_i represents the number of flows on route i.

5.4 Modified Gibbs sampling algorithm

The modified Gibbs sampling method is constructed as follow: Let

$$\begin{aligned} \pi() &= \pi(N_i|N_1,\ldots,N_{i-1},N_{i+1},\ldots,N_k) \\ &= \frac{\lambda_i^{n_i}}{\prod_{k=0}^{n_i}\Lambda_i(k,S_{+1})}\cdot C^{-1} \end{aligned}$$

For $m = 1, 2, \ldots, M$, doing the following:

- **Initialize:**

  Starting from an arbitrary value $u_1^{(m)} = (u_{1,1}, \ldots, u_{1,k})$, with $u_{1,i}$ be from the feasible value set of N_i. We drop (m) next for simplicity.

- **For** $j = 1, 2, \ldots, N$**,** do the following:

  1. Randomly choose i from $\{1, \ldots, k\}$, and let $\{y_i\}$ be the set of the feasible values of U_i other than $u_{j,i}$, and do a sampling from $\{y_i\}$ according to the probability distribution:
  $$P(y_i) = \frac{\pi(y_i|u_{[-i]})}{1-\pi(u_{j,i}|u_{[-i]})}$$
  where $u_{[-i]} = (u_{j,1}, \ldots, u_{j,i-1}, u_{j,i+1}, \ldots, u_{j,k})$.

  2. Suppose the sample we draw is Y. We update $u_{j,i}$ with the following acceptance rule:
  $$u_{j,i} = \begin{cases} Y & \text{with probability } min\left\{1, \frac{1-\pi(u_{j,i}|u_{[-i]})}{1-\pi(Y|u_{[-i]})}\right\} \\ u_{j,i} & \text{otherwise} \end{cases}$$

  3. Set $u_{j+1}^{(m)} = u_j^{(m)}$

Count the frequency of every state $\vec{s}$:

$$P(\vec{N} = \vec{s}) = \frac{1}{NM}\sum_{j=1}^{M}\sum_{i=1}^{N} I_{\vec{s}}(u_i^{(j)})$$

$$where\ I_{\vec{s}}(\vec{u}) = \begin{cases} 1 & \text{if } \vec{u} = \vec{s} \\ 0 & \text{otherwise} \end{cases}$$

END

5.5 Chapter summary

In this chapter, instead of using the naive simulation method, we resort to the modern statistical sampling method to provide a benchmark to evaluate the numerical solution of our approximate algorithm. The modified Gibbs sampling method based on Liu (1996)'s modification of the generic Gibbs sampling method tailored to the discrete random variables provides the best solution in this communication network context.

Chapter 6

Numerical Study

In this chapter, we present in section 6.1 three network models where analytical solutions of the stationary distribution are available, and compare the results of our approximation method and the Modified Gibbs sampling method with these exact solutions. Next in section 6.2 we consider two network models where analytical solutions are not available, and apply our approximation method and Modified Gibbs method to them. The reason that we use instead the Modified Gibbs sampling method as the benchmark solution has been mentioned in the chapter of the Gibbs sampling method.

To measure the accuracy of our approximation algorithm, in addition to the naive one-to-one comparison, we may consider some often used error estimators based on the set of discrete probability values we have calculated, such as the CAE, RMSE, etc. (see the definition below) However, in experiments, it is observed that these error estimators are not reliable because they are diluted significantly after being divided by the huge number of the values (N), meaning that they become too small. Only the RCSE and MAPE are the reliable error estimators that we will thus use to evaluate the accuracy of the approximation results.

Mean Absolute Error	$MAE = \frac{1}{N}\sum_1^N \epsilon_i$
Cumulative Absolute Error	$CAE = \sum \epsilon_i$
Root Mean Squared Error	$RMSE = \sqrt{\frac{\sum \epsilon_i^2}{N}}$
Root Cumulative Squared Error	$RCSE = \sqrt{\sum \epsilon_i^2}$
Mean Absolute Percentage Error	$MAPE_2 = \frac{1}{N}\sum \frac{\epsilon_i}{\lvert x_i\rvert}$, for all $x_i > 0.01$
Mean Absolute Percentage Error	$MAPE_3 = \frac{1}{N}\sum \frac{\epsilon_i}{\lvert x_i\rvert}$, for all $x_i > 0.001$

From the definition, we can see that from the geometrical perspective, the RCSE represents the straight line distance between the calculated value (a vector) and the true value in the N- dimensional Euclid space, with the i-th coordinate of the vector be $\epsilon_i = |\hat{x}_i - x_i|$. And notice that $RCSE > max\{\epsilon_i\}$.

In addition to the numerical comparison results, we add on the graphical comparison to test the validity of the numerical results. In particular, we will compare the distribution of the percentage error (rather than the absolute error) of each calculated value, and the marginal distribution of each route.

In the following examples, we denote the k routes: $r_1, \ldots, r_k$ with Poisson arrivals of information flows with arrival rates: $p_1, \ldots, p_k$ respectively, and with flow volume of iid exponential distribution with mean 1 . Kelly's proportional fairness allocation rule is applied exclusively in all cases.

Some additional technical specifications are that: we truncate each n_i (the number of ongoing flows on route i) at the 95th percentile of its realized values according to simulation; and for the Modified Gibbs sampling method, we set $M = 1000$ parallel independent paths, each with $N = 30$ cycles sampling, such that the computational time is modest, and the accuracy level is at 0.001.

6.1 Networks with analytical solution

In this section, we consider three networks where exact analytical solutions are available, which are the standard (that is of unit capacity link) linear network, the standard grid network, and the single bottleneck; and compare our approximation results and Modified Gibbs results with the exact solutions. These examples of network models are not chosen for simplicity, but rather for their representativeness. As suggested in Bonald and Massoulie (2000), we consider networks whose performance are limited by several critical bottlenecks, and where the bottlenecks share similar characteristics of link capacity and offer load. In the cases where bottlenecks have widely different characteristics, one could expect the performance is driven by bottlenecks which are imposed the tightest capacity constraint, and then the preceding results could be applied by ignoring those unconstrained bottlenecks.

6.1.1 Standard linear network

Consider a standard linear network with two unit capacity links and three routes. see Figure 6.1. The link's capacity is by Kelly's proportional fairness rule, shared among those routes that traverse through it. The bandwidth allocated to each route is then equally shared among the information flows transmitting on that route.

Assume the information flows arrive as a Poisson process, each carrying an amount of data as an iid exponential random variable with mean 1, and the Kelly's proportional fairness bandwidth allocation rule, the exact solution is given by Massoulie and Roberts (1998):

$$\pi(n_1, \ldots, n_k) = C \binom{\sum n_i}{n_k} \prod p_i^{n_i}$$

$$where\ C = \frac{(1-\rho_0)^{L-1}}{\prod_1^L (1-\rho_0-\rho_i)}$$

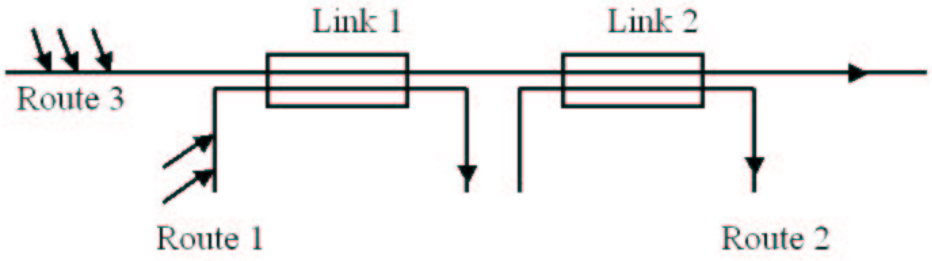

Figure 6.1: The Linear network

where C is the normalization constant, and n_k represents the number of ongoing flows on the longest route.

Error Measure	Approximation	Gibbs	generic Gibbs	Simu(1K/2K/3K)
MAE	4.63E-06	2.22E-04	-	-
CAE	1.00E-03	4.80E-02	-	-
RMSE	3.20E-05	9.53E-04	-	-
RCSE	4.70E-04	1.40E-02	1.6E-02	3/2/1.5E-02
$MAPE_3$	0.1%	13%	18%	38/26/19%
$MAPE_2$	0.1%	4.7%	7%	15/9/7%

Table 6.1: Error estimation for linear network

The numerical results of our approximation algorithm and Gibbs sampling method are summarized in Table 6.1, for arrival rates $(0.2, 0.2, 0.2)$. The *Gibbs* refers to the Modified Gibbs method, which is indeed more accurate than the *generic Gibbs*, i.e. the plain Gibbs method without Liu (1996)'s Modification. It is further confirmed that the averaged error estimations MAE and RMSE are largely diluted, with both seeming too small at the level of $10^{-5} \sim 10^{-6}$. It appears that the errors are numerically insignificant, suggesting excellent approximation results.

RCSE:

However, as we have pointed out, the error estimator RCSE is instead the more reasonable estimator in our case. The approximation algorithm produces a RCSE at the level of 10^{-4}. In other words, this suggests that the calculated value (a vector in n^k dimensional space) deviates from the true value at a distance of 10^{-4}. In plain English, if the true value is 0.5, the approximated result is within $(0.5 - 0.0001, 0.5 + 0.0001)$. The Gibbs method yields a RCSE at the level of 10^{-2}.

MAPE:

On the relative error side, the approximation algorithm produces an excellent result with the percentage error averaged only at 0.1% (MAPE), based on the comparison with all true values that are larger than 10^{-3}. This extreme small MAPE may be resulted from the very low arrival rate on each route. For even smaller true values that are below 10^{-3}, such as those below 10^{-5}, 10^{-8} or even lower, the MAPE's are still 0.1% without any tendency to increase.

For the Modified Gibbs method, the average percentage error (MAPE) is at 4.7% compared with true values that are larger than 10^{-2}, and about 13% for true values that are larger than 10^{-3}. The generic Gibbs sampling method produces larger errors: $7\%MAPE_2, 18\%MAPE_3$ respectively. The acceptable 4% MAPE at the 10^{-2} level suggests that $M = 1000$ parallel paths and $N = 30$ cycles sampling are good enough to provide a practically accurate result in our case. Whereas the 13% error requires more sampling either by increasing the number of parallel paths or the number of cycles to reduce the relative error for small true values (those within $(10^{-3}, 10^{-2})$). For example, when we sample $10,000$ cycles in each parallel path, the Gibbs method produces $6\%MAPE_3$, a large improvement.

We notice the simulation method is inefficient here due to the large number of values to be evaluated: with 1000 simulated arrivals on each route, the $MAPE_2 = 15\%$; when simulate 3000 arrivals, the $MAPE_2$ approaches 6%, close to the Modified Gibbs result, but costs much more computational time.

Distribution of individual Percentage Errors:

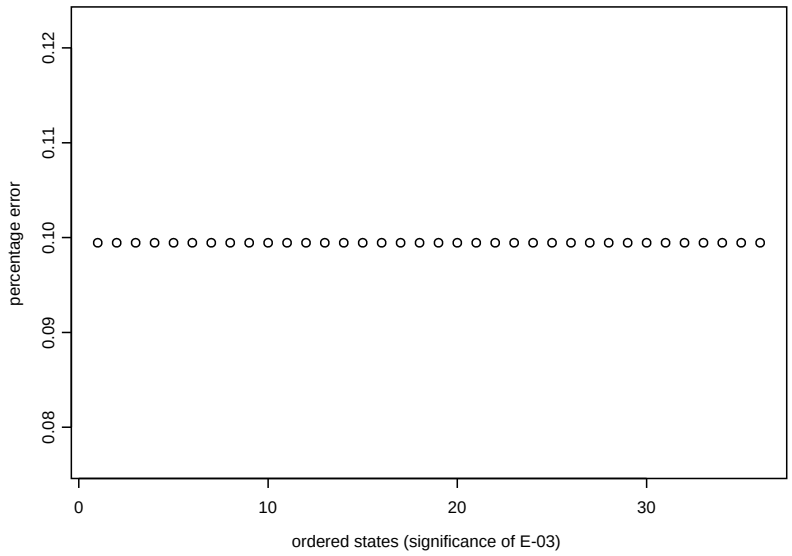

Figure 6.2: PE distribution of linear network

As the supplement to the MAPE and other aggregated error estimations, further insights could be obtained from Figure 6.2, the detailed distribution of the percentage errors (rather than the absolute error) of our approximation results for the values larger than 10^{-3}. The x-axis represents those states $(n_1, \ldots, n_k)$ whose associated values are larger than 10^{-3}, being labelled into one dimension; y-axis is the percentage error of each individual approximated value compared with its true counterpart.

Surprisingly, the percentage error is the same! Fixing at the 0.1% level, regardless

of the specific state. In other words, whatever the state is, $(n_1, n_2, n_3) = (2, 3, 4)$, or $= (3, 4, 5)$, the errors of our approximated results compared with the true solution are always 0.1%. This is consistent with the previous observation that the MAPE of the approximation result does not change no matter how significant the underlying values are, that is, we get the same MAPE for values that are larger than either 10^{-2}, or 10^{-3}, etc.

The fact that the percentage errors of each individual values are the same, in other words, our approximation results consistently overestimate the true values for a same percentage, must suggest something. Recall that in the chapter of building the approximation algorithm, we use a finite sum to approximate the normalization constant:

$$\begin{aligned}\pi_{r|S_{+r}}(n_r|S_{+r}) &= \frac{\lambda_r^{n_r}}{\prod_{k=0}^{n_r} E[\Lambda_r(\cdots, k, S_{+r})]} C^{-1} \\ &= \frac{\lambda_r^{n_r}}{\prod_{k=0}^{n_r} E[\Lambda_r(\cdots, k, S_{+r})]} \left(\sum_{n=0}^{N} \frac{\lambda_r^{n}}{\prod_{k=0}^{n} E[\Lambda_r(\cdots, k, S_{+r})]} \right)^{-1}\end{aligned}$$

Thus it is expected that our approximation results will consistently overestimate the true values for a same margin. And we anticipate that the margin will diminish as we increase N gradually.

We test this judgement by increasing N (that to reduce the truncation error). The result is shown in Figure 6.3.

We now see the effect, that the percentage error decreases to $1.205880E-004\%$, almost zero, when we increase the number of state. As we increase the number of state further, the percentage error indeed diminishes.

The findings may be summarized as that for standard network such as the linear network (later we will see the same for the grid network), our approximation algorithm is numerically accurate (see Figure 6.3), if excluding the implementation error, i.e. approximating the normalization constant with a finite sum, although it is impossible in practice.

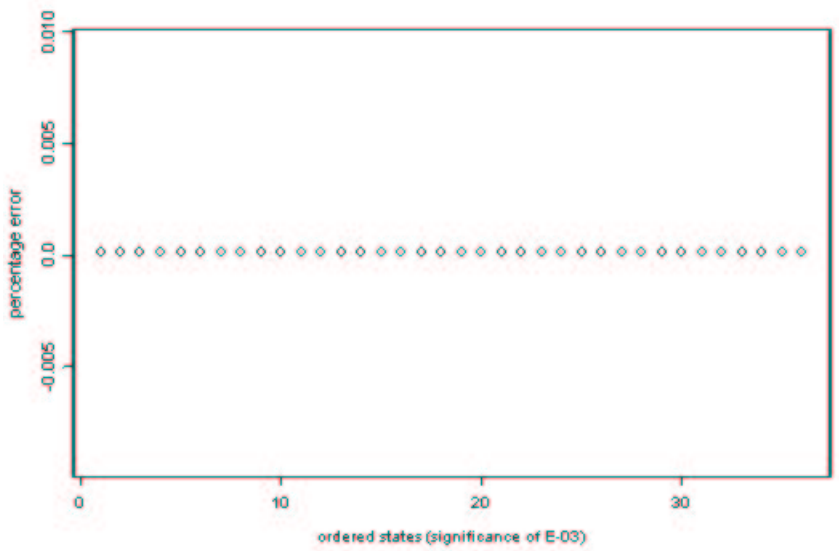

Figure 6.3: PE distribution after reducing truncation error

Marginal distribution of each route:

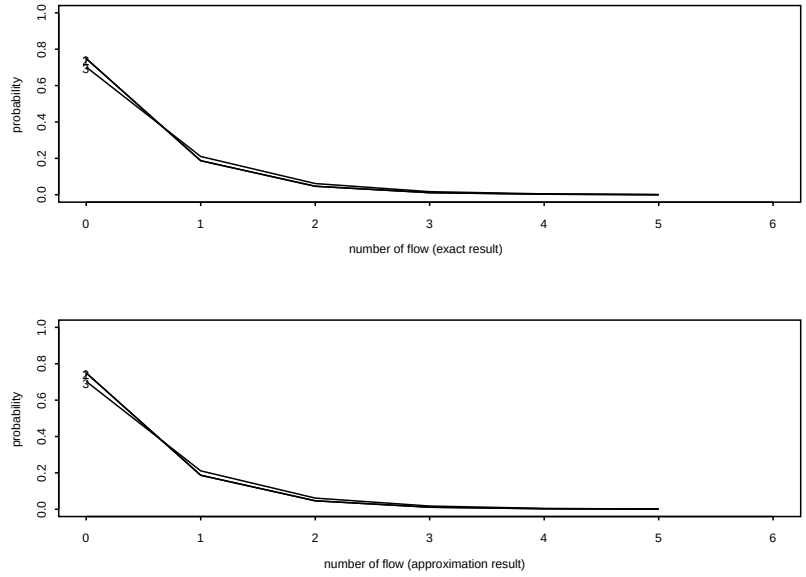

Figure 6.4: Marginal distribution comparison of linear network

The marginal distribution of each route is another very important concern to the

system. The marginal distribution of each route is simply:

$$\pi_i(n_i) = \sum_{for\ all\ n_j, n_j \neq n_i} \pi(n_1, \ldots, n_k)$$

In Figure 6.4, we present the comparison of the marginal distributions of all the three routes, under the approximation method (the lower panel), and the exact results (the upper panel) respectively. The curves in the two panels are basically the same, suggesting that the approximation method yields even more accurate results for each individual queues in the network.

6.1.2 The grid network

Next we considered the standard grid network, essentially a generalization of the standard linear network. See Figure 6.5 for a four route, four unit capacity link case.

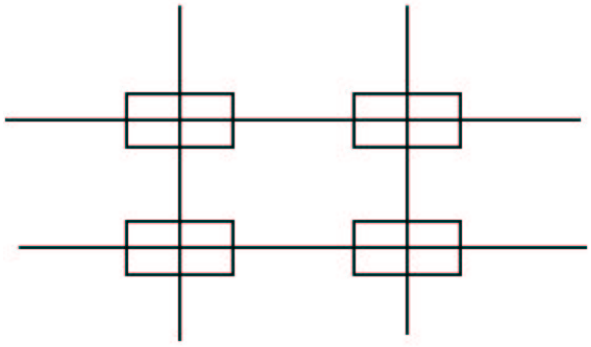

Figure 6.5: The grid network

Under the same assumptions, its exact solution is given by Bonald and Massoulie (2000):

$$\pi(x, y) = C \binom{\sum x_k + \sum y_l}{\sum x_k} \prod p_k^{x_k} \prod p_l^{y_l}$$

$$where\ C = \sum_{k=1}^{K} \sum_{l=1}^{L} \frac{\rho_k^{K-1} \rho_l^{L-1}}{\prod_{k' \neq k} (\rho_k - \rho_{k'}) \prod_{l' \neq l} (\rho_l - \rho_{l'})} \frac{1}{1 - \rho_k - \rho_l}$$

where x_k is the number of flows on the horizontal route k, y_l is that on vertical route l.

Error Measure	Approximation	Gibbs
RCSE	4.00E-03	1.00E-02
$MAPE_3$	2	13
$MAPE_2$	2	4

Table 6.2: Error estimation for grid network I

Error Measure	Approximation	Gibbs
RCSE	4.00E-03	1.10E-02
$MAPE_3$	1.42	14
$MAPE_2$	1.42	4

Table 6.3: Error estimation for grid network II

The results with two sets of different input rates are summarized in Table 6.2 for arrival rates $(0.2, 0.3, 0.2, 0.3)$, and Table 6.3 for $(0.1, 0.3, 0.1, 0.3)$ respectively. The RCSE is still very small, at the level of 10^{-3} for the approximation method, and 10^{-2} for the Gibbs method.

In addition, the MAPE of our approximation method remains at about 2% in both cases; and the Modified Gibbs method at the $4\%, 14\%$ level respectively as well. It is worth noting that on the relative error side, MAPEs of the Gibbs method remain unchanged compared with the results for the linear network with 4.7% and 13% respectively. This confirms the fact that the Gibbs sampling method is a very general method as well as the simulation method, regardless of the specific problems.

For certain truncation setting, the percentage error distribution of the approximation results, e.g. for the first set of input rates $(0.2, 0.3, 0.2, 0.3)$, tells the same story as that

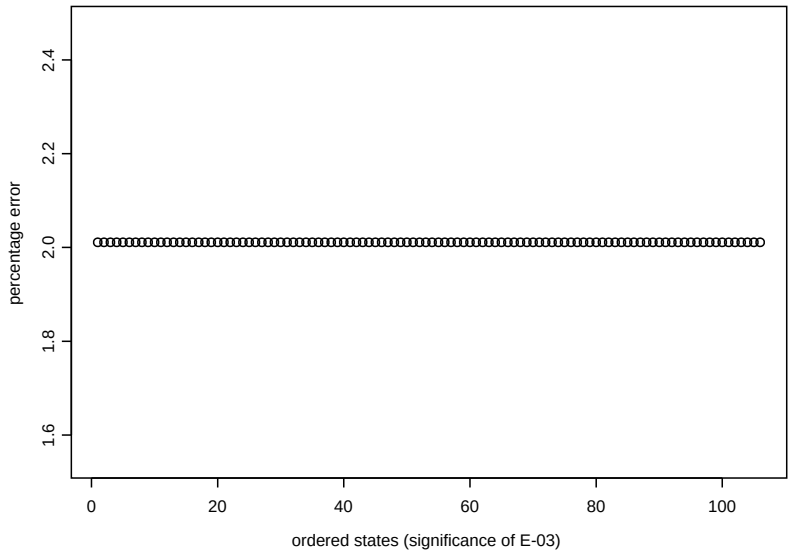

Figure 6.6: PE distribution of grid network

of the linear network, see Figure 6.6. This time the result consistently overestimates the true value by 1.42%.

The marginal distributions are shown in Figure 6.7, indicating close enough curves with the true ones.

6.1.3 Single bottleneck

Another rarely available network that has analytical solution is the simple single bottleneck, which consists of only one link. It is also a special case of the classic BCMP network, see Figure 6.8 for a three route example. In this simple case, the link's capacity is thus proportionally shared among the three routes in accordance with the number of transmitting flows on each route, and subsequently equally shared among all the transmitting flows within each route.

The analytical solution of stationary distribution is simply given by the BCMP for-

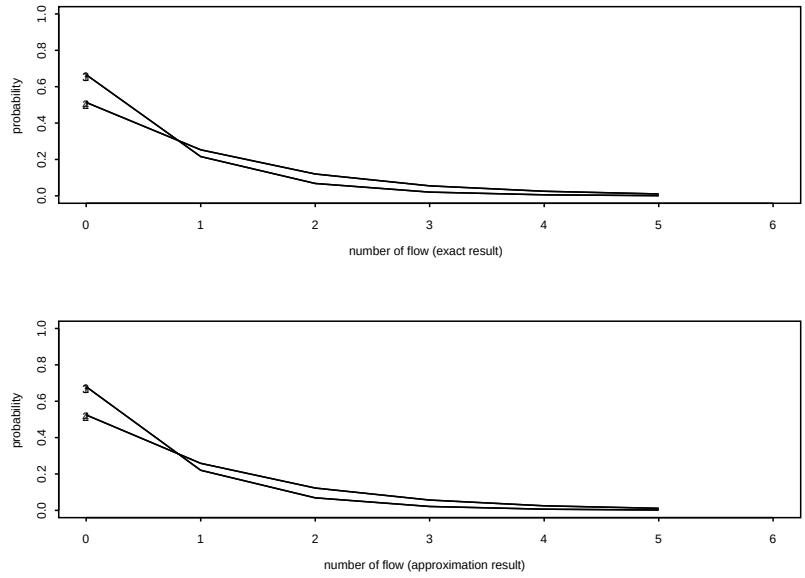

Figure 6.7: Marginal distribution comparison of grid network

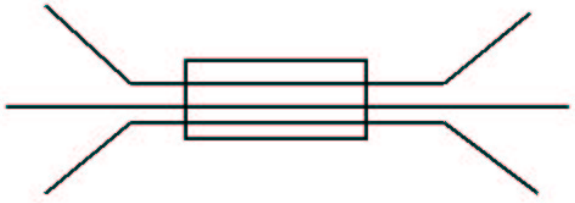

Figure 6.8: Single bottleneck

mula:

$$\pi(n_1, \ldots, n_k) = (1 - \sum p_i)(\sum n_i)! \prod \frac{p_i^{n_i}}{n_i!}$$

where n_i is the number of transmitting flows on route i.

Error Measure	Approximation	Gibbs
RCSE	7.00E-03	1.70E-02
$MAPE_3$	2.1	13
$MAPE_2$	2.1	4

Table 6.4: Error estimation for single bottleneck I

Error Measure	Approximation	Gibbs
RCSE	3.00E-03	1.90E-02
$MAPE_3$	0.73	14
$MAPE_2$	0.73	4

Table 6.5: Error estimation for single bottleneck II

The numerical results of our approximation algorithm and Gibbs method are summarized in Table 6.4 for arrival rates , and Table 6.5 with two sets of different arrival rates $(0.2, 0.2, 0.3), (0.1, 0.2, 0.3)$ respectively as input parameters.

The approximation algorithm produces a RCSE at the level of 10^{-3}, and the Gibbs method yields a value at the level of 10^{-2}.

On the relative error side, the approximation algorithm produces an excellent result with the percentage error averaged only at 2.1% (MAPE), (and even lower 0.73% for the second case), based on the comparison with all true values that are larger than 10^{-3}. For the Modified Gibbs method, the average percentage error (MAPE) is at 4% compared with true values that are larger than 10^{-2}, and about $12 \sim 13\%$ for true values that are larger than 10^{-3}.

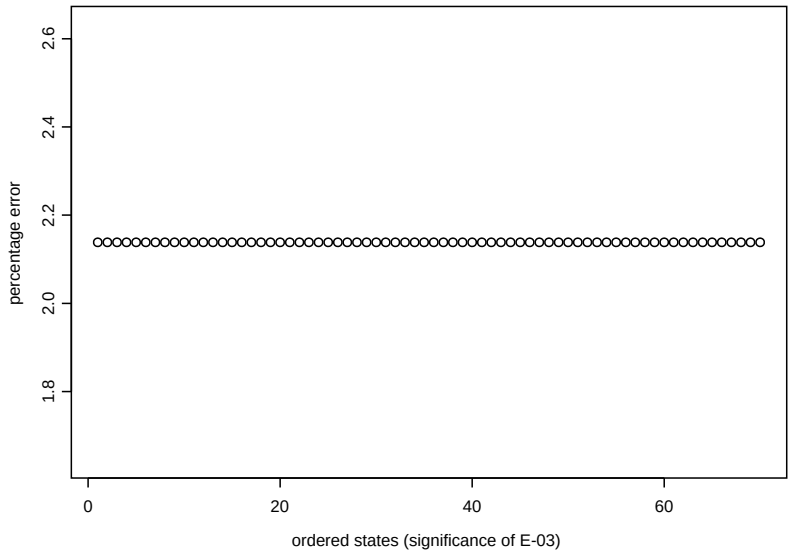

Figure 6.9: PE distribution of single bottleneck

The percentage error distribution is shown in Figure 6.9. The approximation result consistently overestimates 2.1% as we can expected.

The marginal distribution of each route is shown in Figure 6.10, (for the $(0.2, 0.2, 0.3)$ input rates). Note it is expected because of the same arrival rates on two routes, the marginal distributions of the two routes are exactly the same, thus they overlap in the Figure. The curves in the two panels are still basically the same, suggesting the approximation method yields even more accurate results for each individual queues in the network.

6.2 Network without analytical solution

In this section, we consider a cyclic network (Figure 6.11) where the exact bandwidth allocation is available, but the exact solution is not. We will compare our approxima-

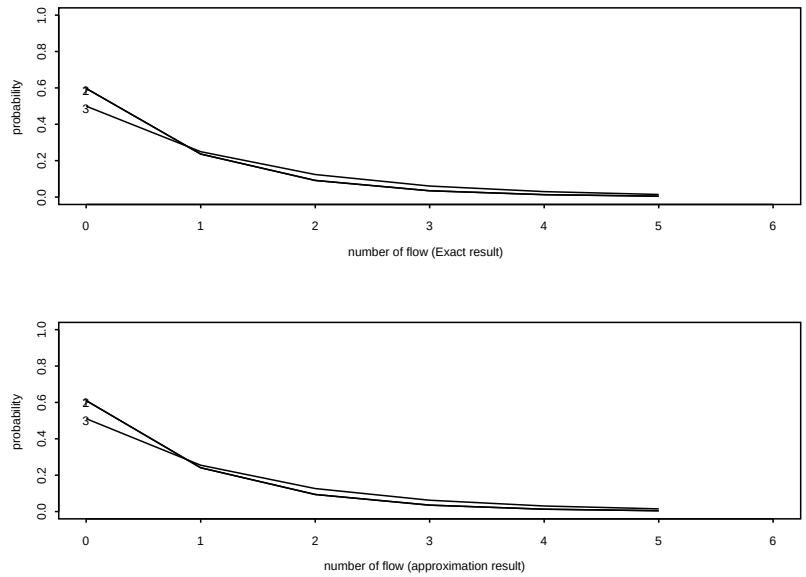

Figure 6.10: Marginal distribution comparison of single bottleneck

tion method with the Gibbs sampling method. In the non-exact solution case, instead of comparing the difference between the approximated values and the true values, we compare the approximated values with Gibbs sampling values which is similar to the simulation, and evaluate the difference using the preceding error estimators. If both of these two methods approximate the true values, we shall expect not much divergence from the preceding results.

The result is listed in Table 6.6, for arrival rates $(0.2, 0.2, 0.2, 0.2, 0.2, 0.2)$. For each MAPE, the left hand side value is the percentage error calculated based on the approximated values, that is to divide the absolute error by the approximated values, and the right hand side is based on Gibbs values.

We can see that the results $5\%MAPE_2$ and $19\%MAPE_3$are quite consistent with the preceding results: $4\%MAPE_2$ and $13\%MAPE_3$ of the Gibbs sampling method, suggesting that the approximation method does approximate the true values at a slight margin

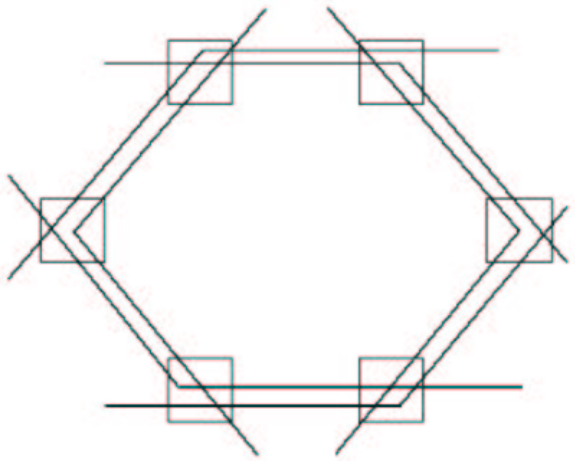

Figure 6.11: The cyclic network

Error Measure	Approximation − Gibbs
MAE	1.21E-05
CAE	5.65E-01
RMSE	5.32E-05
RCSE	1.15E-02
$MAPE_3$	19 / 22
$MAPE_2$	5.7 / 5.5

Table 6.6: Error estimation for the cyclic network

near the $1 \sim 2\% MAPE$, whereas the Gibbs method approximates the true values at $4\% MAPE_2$ regardless of the specific type of network. The slightly higher $19\% MAPE_3$ may be due to inaccuracy from the increased dimensionality and the 2% inaccuracy of the approximated values. Thus it is not surprised that the result is very close to the Gibbs results in preceding examples. In addition, the RCSE which is at the level of 10^{-2} also is quite close to the preceding results for the linear, grid and single bottle networks.

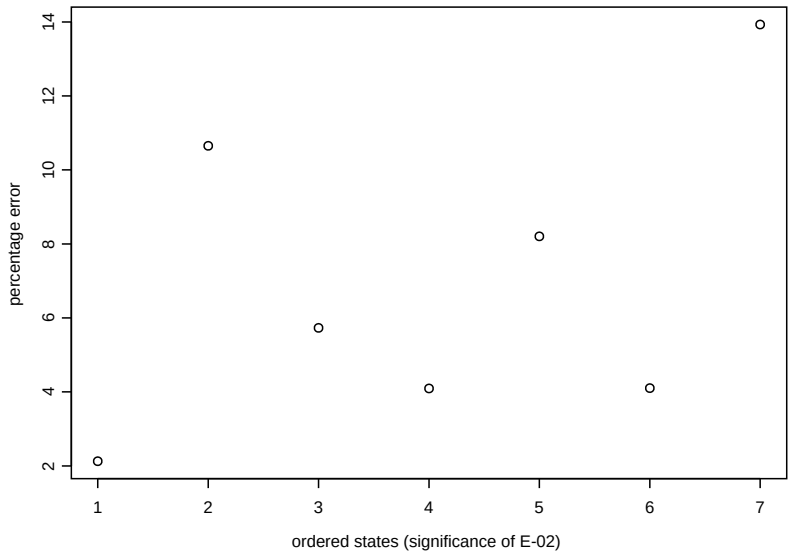

Figure 6.12: PE distribution of cyclic network

The percentage error distribution of the approximation result compared with the Gibbs result for those values larger than 10^{-2} is shown in Figure 6.12. In this case, we see relatively wide error distribution, due to the randomness of the Gibbs sampling results.

The Marginal distribution comparison of these two methods is shown in Figure 6.13. The six marginal distributions are exactly the same, overlapping at a single curve, all because of the symmetric network structure and the same input rates.

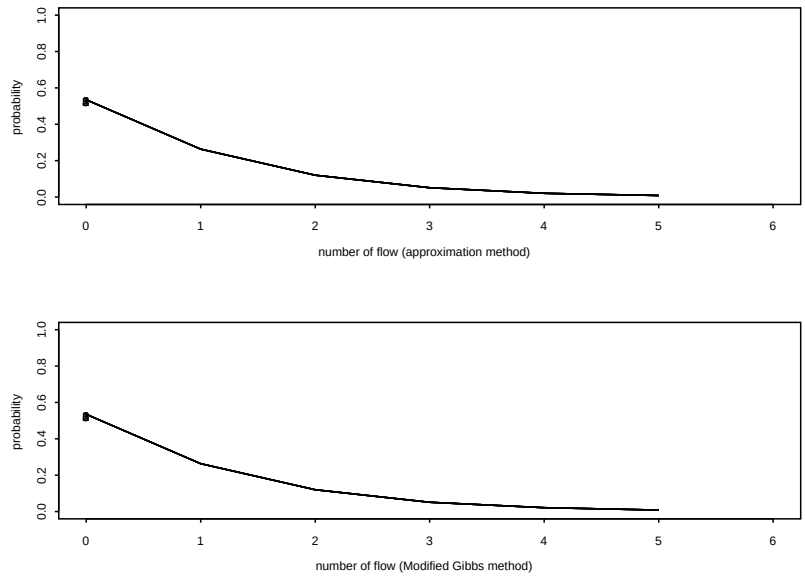

Figure 6.13: Marginal distribution comparison of cyclic network

6.3 Truncation error

In previous section, we showed that as the truncation error diminishes, the percentage error approaches zero as well. In this section, we consider how bad the truncation error could be if we do not reduce it purposely, that is what is the worst impact of truncation of n_i on the accuracy. We conduct the experiments under the heavy traffic load condition, namely the $\sum p_i \to 1$, where the truncation error has a major effect.

First we consider two extreme cases of a linear network (see Figure 6.14) where arrival rate on exactly one route is 0. The first one is $(p_1, p_2, p_3) = (0.8, 0, 0.15)$ where the rates p_1, p_3 differ significantly. In this case, the maximum number of flows on route 1 and route 3 are $100, 25$ respectively, the 99th percentile of the outcomes according to simulation. We choose to truncate N_1, N_2 at lower values for computational savings, and compare the MAPEs under different scenarios, such that we will see the impact of truncation on the

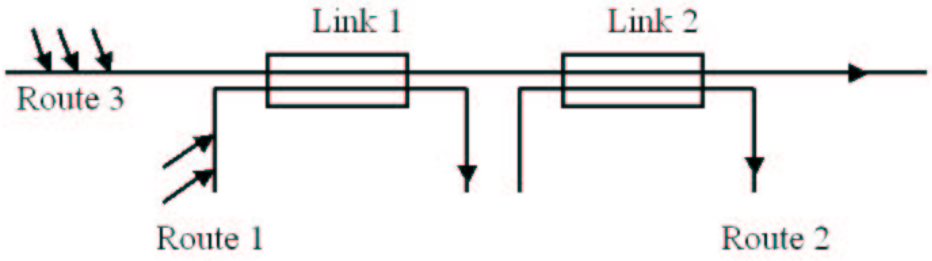

Figure 6.14: Linear network under heavy traffic

accuracy of the approximation results.

N1	N3	MAPE
100	25	0.11%
70	25	1.25%
60	25	2.31%
50	25	4.52%
40	25	8.84%

Table 6.7: Impact of truncation error I

N1	N3	MAPE
70	70	0.12%
60	60	0.33%
50	50	0.88%
40	40	2.35%
30	30	6.36%

Table 6.8: Impact of truncation error II

The results are shown in Table 6.7. We see that N_1 decreases by 10 each time, which incur larger truncation error, while the MAPE only increases marginally.

For the second case where $(p_1, p_2, p_3) = (0.5, 0, 0.45)$, where the rates p_1, p_3 are close to each other, similar comparison results are summarized in Table 6.8. In this case, even we truncate the $N1, N2$ half from 70 to 30, the MAPE is still practically acceptable. Thus it is worth to sacrifice some accuracy to obtain significant computational savings.

Finally we consider when $p_2 \neq 0$, that is in a higher dimension, how is the truncation effect. Suppose $(p_1, p_2, p_3) = (0.5, 0, 0.4), (0.5, 0.5, 0.4)$, named set1 and set2 respectively. The result is summarized in Table 6.9. For each truncation, the MAPE of the three route network (set2) is about as 2.5 times large as that of set1, a two route network, suggesting the increasing dimensionality makes the approximation results worse under the same truncation settings.

N1 (and N2)	N3	MAPE	
		set1	set2
40	40	0.05%	-
30	30	0.37%	0.90%
20	20	2.47%	6.62%
10	10	18.20%	48%

Table 6.9: Impact of truncation error III

6.4 Chapter summary

In this chapter, we present some numerical examples to compare the approximation results with those of the Modified Gibbs method and exact results, if available. In all these cases, it is observed that some error estimators such as RCSE and MAPE provide more reliable comparison results for the accuracy of the approximation algorithm. In particular, the RCSE of the approximation results maintain at the level of 10^{-3}, indicating the deviation

from the exact results is at the 0.001 level in absolute terms, and the MAPE suggests a quite small, less than 2% relative error within the true values. Further investigation suggests that our approximation algorithm consistently overestimates the true values only at a margin, due to the truncation error involved in the algorithm. The overestimation diminished as we reduce the truncation error, suggesting that the approximation algorithm is quite accurate.

The analysis of truncation error suggests that the approximation algorithm is robust. The percentage error MAPE increases only marginally when we truncate the state in modest range. This provides room for computational savings.

Chapter 7

Conclusion

7.1 Overview

This concluding chapter provides a summary of the findings as well as the implications of the results. The limitations of the study are discussed next and finally suggestions for future research will round up the study.

7.2 Summary of findings

According to the results of the numerical analysis, our algorithm provides a highly accurate approximation for some simple networks, as comparing with their analytic solutions. The percentage error is limited below 2%. One of the sources of error is from the approximation of the normalization constant. If excluding this error factor, the percentage error is further significantly reduced.

The Modified Gibbs method provides slightly larger error of 4%, depending on the number of rounds of sampling we set. But it provides the benchmark solutions for those complex networks where the analytic solutions are not available. Thus we can evaluate the

accuracy of the approximation solutions, and come to the conclusion that the algorithm is widely applicable.

7.3 Implications of study

This study suggests that due to the special characteristic of this type of network, instead of resorting to traditional Markov Chain technique, we can solve it in a decomposition fashion. Our approximation algorithm proved the feasibility.

Based on the numerical study on networks with analytic solutions, and later numerical comparison with Gibbs method on some non-analytic-solution networks, we firmly believe that this algorithm is applicable in solving a wide range of networks which share the same characteristics. In other words, this decomposition method indeed provides another way to approach this type of network.

7.4 Limitations of study

The algorithm is designed for network with Poisson arrivals, each carrying exponentially distributed file size, because only for these cases, analytic results are abundant for use. Thus theses analytic results largely reduce the modelling error. But this simplification cause concerns about the applicability of the algorithm to more realistic network cases, where numerous observations show that most of time the arrival process is definitely not Poisson, and the file size distribution is heavy tailed.

However, the analytic results are not available for those more complex networks, thus our numerical studies are mainly conducted on a few simple networks, where the analytic solutions provide the benchmark for evaluation of accuracy.

On the other hand, the computation effort of the algorithm exponentially increases

with the number of states of the system. Thus in our study the algorithm is implemented for networks with relatively light traffic. For heavy traffic cases, the usefulness of the algorithm is limited.

7.5 Suggestions for future research

Future research on this subject may go to two directions. One is to lessen the assumptions of Poisson arrivals, exponential file size. This may require introducing additional steps to numerically approximate the realistic arrival process and its impacts. In these cases, simulation results may be used as benchmark solutions.

For the simple light traffic networks, our algorithm provides very accurate approximation to the solution, i.e. the probability of every state. For more practical cases, where the heavy traffic dominates, the probability of the intervals of states, i.e. $P(\vec{n} \in A_i)$ is instead of more interest. Thus another improvement is to design methods calculating the cumulative probabilities, i.e. $P(n_1 < x, n_2 < y, \cdots)$, instead of the discrete point probabilities.

Bibliography

[1] Albin, S.L. (1984). " Approximating a point process by a renewal process, II: superposition arrival processes to queues". *Oper.Res. 32*, 1133-1162.

[2] Baskett, F. , Chandy,K.M., Muntz, R.R. and Palacios, F.G. (1975). " Open, Closed and Mixed networks of queues with different classes of customers". *J.ACM. 22*, 248-260.

[3] Bertsekas, D. and R. Gallager. (1992). " Data networks". *Prentice-Hall, Englewood Cliffs, NJ.*

[4] Bitran, G.R. and Dasu, S (1990). " A review of open queueing network models of Manufacturing systems". *WP No.3229-90-MSA, MIT.*

[5] Bitran, G.R. and Tirupati, D. (1988). " Multiproduct Queueing Networks with Deterministic Routing: Decomposition Approach and the Notion of Interference". *Management Science 34*, 75-100.

[6] Bolch, G., Greiner, S, de Meer, H, and Trivedi, K.S. (1998). " Queueing networks and Markov chains". *John Wiley & Sons, Inc.*

[7] Bonald T. and Massoulie L. (2000). " Impact of Fairness on Intrnet Performance". *Manuscript.*

[8] Burman, D.Y and Smith, D.R (1986). " An asymptotic analysis of a queueing system with markov-modulated arrivals".*Operations Research 34*, 105-119.

[9] Burman, D.Y. and Smith, D.R (1983). " A light traffic theorem for multi-server queues".*Math Operations Research 8*, 15-25.

[10] Casella, G. and George, E.I. (1992). " Explaining the Gibbs sampler".*Tech Report BU1098M, Biometrics Unit, Cornell University*.

[11] Chen, Hong Dai, J. G. , Dai, Wanyang, and Shen, Xinyang (2002). " The Finite Element Method for Computing the Stationary Distribution of an SRBM in a Hypercube with Applications to Finite Buffer Queueing Networks".*Queueing Systems 42*, 33-62.

[12] Chen, Hong, Shen, Xingyang and Yao, D (2001). " Brownian approximations of multiclass open queueing networks".*Working paper*.

[13] Chiu, D. M. (2000). " Some Observations on Fairness of Bandwidth Sharing".*Proceedings of ISCC 2000*.

[14] Chiu,D.M. and Jain,R. (1989). " Analysis of the increase and decrease algorithms for congestion avoidance in computer networks".*Computer Networks and ISDN Systems 17*,1-14.

[15] Courtois, P.J. (1977). " Decomposability: queueing and computer system applications".*Academic Press, New York*.

[16] Dai, J. G. and Dai, W.Y. (1999). " A heavy traffic limit theorem for a class of open queueing networks with finite buffers".*Queueing Systems 32*, 5-40.

[17] Dai, J. G. and Harrison, J. M. (1991). " Steady-state analysis of RBM in a rectangle: numerical methods and a queueing application".*Annals of Applied Probability 1*, 16-35.

[18] Dai, J. G. and Harrison, J. M. (1992). " Reflected Brownian motion in an orthant: numerical methods for steady-state analysis".*Annals of Applied Probability 2*, 65-86.

[19] Dai, J. G. and Nguyen V. (1994). " On the Convergence of Multiclass Queueing Networks in Heavy Traffic".*Annals of Applied Probability 4*,26-42.

[20] Dai, J. G., Nguyen, Vien and Reiman, Martin I. (1994). " Sequential Bottleneck Decomposition: An Approximation Method for Generalized Jackson Networks".*Operations Research 42*,119-136 .

[21] De Veciana, G., T. J. Lee and T. Konstantopoulos. (2001). " Stability and performance analysis of networks supporting elastic services".*IEEE/ACM Transactions on Networking 9*, 2-14 .

[22] Disney, R.L. and Konig, D (1985). " Queueing networks: a survey of their random processes".*SIAM review 27*, 335-403.

[23] Fayolle, G., Arnaud de La Fortelle, Jean-Marc Lasgouttes, Laurent Massoulie, James Roberts (2001). " Best-Effort Networks: Modeling and Performance Analysis Via Large Networks Asymptotics".*IEEE INFOCOM 01.*

[24] Foddy, M. (1983). " Analysis of Brownian motion with drift, confined to a quadrant by oblique reflection".*PH.D. dissertation, Dept. of Mathematics, Stanford University.*

[25] Gelenbe, E.and Pujolle, G. (1987). " Introduction of queueing networks".*Jon Wiley & Sons.*

[26] Gelfand, A., Hills, S., Racine-Poon, A., and Smith, A. (1990). " Illustration of Bayesian inference in normal data models using Gibbs sampling".*J.Amer.Statist.Assoc. 85*,972-985.

[27] Gelfand, A. and Smith, A. (1990). " Sample based approaches to calculating marginal densities".*J.Amer.Statist.Assoc. 85*, 398-409.

[28] Geman, S. and Geman, D. (1984). " Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images".*IEEE Trans. Pattern Anal Mach Intelligence 6*, 721-741.

[29] Gordon,W.J. and Newell,G.F. (1967). " Closed queueing systems with exponential servers".*Operations Research 15*, 254-265.

[30] Harrison, J. M. and Williams, R. J. (1987). " Multidimensional reflected Brownian motions having exponential stationary distributions".*Annals of Prob. 15*,115-137.

[31] Harrison, J. M. , Williams, R. J. and Chen H. (1990). " Brownian models of closed queueing networks with homogeneous customer populations".*Stochastics 29*, 37-74.

[32] Harrison, J. M. , Landau, H. and Shepp, L. A. (1985). " The stationary distribution of reflected Brownian motion in a planar region".*Annals Prob. 13*, 744 -757.

[33] Harrison, J. M. and Nguyen, V. (1990). " The QNET method for two-moment analysis of open queueing networks".*Queueing Systems: Theory and Applications 6*, 1-32.

[34] Harrison, J. M. and Williams, R. J. (1987). " Brownian models of open queueing networks with homogeneous customer population".*Stochastics 22*, 77-115.

[35] Harrison, J.M. (1985). " Brownian Motion and Stochastic Folw System".*John Wiley & Sons, New York*.

[36] Harrison,P.G. and Petal,N.M. (1993). " Performance modelling of communication networks and computer architectures".*Addison Wesley publishers Ltd.*

[37] Jackson, J.R. (1957). " Networks of waiting lines".*Oper. Res. 5*,518-521.

[38] Jackson, J.R. (1963). " Jobshop like queueing system".*Management Science 10*, 131-142.

[39] Kelly, F.P. (2001). " Mathematical modeling of the Internet".*in B. Engquist and W. Schmid (ed.), Mathematics Unlimited - 2001 and Beyond, Springer-Verlag, Berlin*,685-702.

[40] Kelly, F.P. (1975). " Networks of queues with customers of different types".*J.Appl.Prob. 12*, 542-554.

[41] Kelly, F.P. (1997). " Charging and rate control for elastic traffic".*European Transactions on Telecommunications 8*, 33-37.

[42] Kelly, F.P., A. Maulloo and D. Tan. (1998). " Rate control in communication networks: shadow prices, proportional fairness and stability".*Journal of the Operational Research Society 49*, 237-252.

[43] Kim, S., Muralidharan, R., and O'Cinneide, C.A. (2000). " Taking Account of Correlation between Streams in Queueing Network Approximations".*Working paper, Purdue University*.

[44] Kramer,W and Langenbach-Belz, M. (1976). " Approximate formulae for general single systems with single and bulk arrivals".*In proc 8th Int Teletraffic Congress, Melbourne*.

[45] Kuehn, P.J (1979). " Approximate analysis of general queueing networks by decomposition".*IEEE Trans Comm , Com-27*, 113-126.

[46] Liu, Jun (1996). " Peskun's Theorem and A Modified Discrete-State Gibbs Sampler".*Biometrika 83*, 681-682.

[47] Low, S. (2003). " A Duality Model of TCP and Queue Management Algorithms".*IEEE/ACM Transactions on Networking, to appear.*

[48] Marshall, K.T. (1968). " Some Inequalities in Queueing".*Operations Research 16*, 651-665.

[49] Massoulie L and Roberts J.W (1998). " Bandwidth sharing and admission control for elastic traffic".*ITC Specialist Seminar, Yokohama.*

[50] Massoulie, L. and Roberts, J. (2002). " Bandwidth Sharing: Objectives and Algorithms".*IEEE/ACM Transactions on Networking 10*, 320-328.

[51] Mo, J. and J. Walrand. (2000). " Fair end-to-end window-based congestion control".*IEEE/ACM Transactions on Networking 8*, 556-567.

[52] Pujolle, G. and Ai, W. (1986). " A solution for multiserver and multiclass open queueing networks".*INFOR. 24*, 221-230.

[53] Reiman, M. I. (1990). " Asymptotically Exact Decomposition Approximations for Open Queueing Networks".*Oper. Res. Letters 6*, 363-370.

[54] Reiman, M.I. (1984). " Open queueing networks in heavy traffic".*Math Oper Res. 9*, 441-458.

[55] Reiman, M.I. and Simon, B (1986). " Light traffic limits of sojourn time distributions in Markovian queueing networks".*Commun. Statist.-Stochastic Models 4*, 191-233.

[56] Reiman, M.I. and Simon, B (1989). " Open queueing systems in light traffic".*Math of Oper Res. 14*, 26-59.

[57] Roberts, J.W. (2004). " A survey of statistical bandwidth sharing".*Computer Network 45*, 319-332.

[58] Suresh, S. and Whitt, W. (1990). " The Heavy-Traffic Bottleneck Phenomenon in Open Queueing Networks".*Operations Research Letters 9*, 355-362.

[59] Walrand, J (1988). " An introduction to queueing networks".*Prentice-Hall.*

[60] Whitt, W (1983). " The queueing network analyzer".*Bell System tech J. 62*, 2779-2815.

[61] Whitt, W (1982). " Approximating a point process by a renewal process: two basic methods".*Oper Res. 30*, 125-147.

[62] Whitt, W. (1999). " Decomposition Approximations For Time-Dependent Markovian Queueing Networks".*Operations Research Letters 24*, 97-103.

[63] Whitt, W. (1995). " Variability functions for parametric-decomposition approximations of queueing networks".*Management Science 41*, 1704-1715.

[64] Whitt, W. (1984). " Approximations for departure processes and queues in series".*Naval Research Logistics Quarterly 31*, 499-521.

[65] Whitt, Ward (1984). " On Approximations for Queues, I: Extremal Distributions".*AT&T Bell Laboratories Technical Journal 63*, 115-138.

[66] Whitt, Ward. (1995). " Variability Functions for Parametric-Decomposition Approximations of Queueing Networks".*Management Science 41*, 1704-1715.

[67] Williams, R. J. (1987). " Reflected Brownian motion with skew symmetric data in a polyhedral domain".*Probability Theory and Related Fields 75*, 459-485.

[68] Ye H.Q., Ou J.H., and Yuan X.M. (2003). " Stability of Data Networks: Stationary and Bursty Model".*To appear in Operations Research*.

[69] Ye, H.Q. (2003). " Stability of Data Networks Under An Optimization-Based Bandwidth Allocation".*IEEE Transactions on Automatical control 48*, 1238-1242.