Performance Modeling of Communication Networks with Markov Chains

Synthesis Lectures on Communication Networks

Editor

Jean Walrand, *University of California, Berkeley*

Synthesis Lectures on Communication Networks is an ongoing series of 50- to 100-page publications on topics on the design, implementation, and management of communication networks. Each lecture is a self-contained presentation of one topic by a leading expert. The topics range from algorithms to hardware implementations and cover a broad spectrum of issues from security to multiple-access protocols. The series addresses technologies from sensor networks to reconfigurable optical networks.

The series is designed to:

- Provide the best available presentations of important aspects of communication networks.
- Help engineers and advanced students keep up with recent developments in a rapidly evolving technology.
- Facilitate the development of courses in this field.

Performance Modeling of Communication Networks with Markov Chains

Jeonghoon Mo
2010

Communication Networks: A Concise Introduction

Jean Walrand, Shyam Parekh
2010

Path Problems in Networks

John S. Baras, George Theodorakopoulos
2010

Performance Modeling, Loss Networks, and Statistical Multiplexing

Ravi R. Mazumdar
2009

Network Simulation

Richard M. Fujimoto, Kalyan S. Perumalla, George F. Riley
2006

Performance Modeling of Communication Networks with Markov Chains

Jeonghoon Mo

www.morganclaypool.com

ISBN: 9781598299175 paperback
ISBN: 9781598299182 ebook

DOI 10.2200/S00269ED1V01Y201004CNT005

A Publication in the Morgan & Claypool Publishers series
SYNTHESIS LECTURES ON COMMUNICATION NETWORKS

Lecture #5
Series Editor: Jean Walrand, *University of California, Berkeley*
Series ISSN
Synthesis Lectures on Communication Networks
Print 1935-4185 Electronic 1935-4193

Performance Modeling of Communication Networks with Markov Chains

Jeonghoon Mo
Yonsei University, Seoul, Korea

SYNTHESIS LECTURES ON COMMUNICATION NETWORKS #5

MORGAN & CLAYPOOL PUBLISHERS

ABSTRACT

This book is an introduction to Markov chain modeling with applications to communication networks. It begins with a general introduction to performance modeling in Chapter 1 where we introduce different performance models. We then introduce basic ideas of Markov chain modeling: Markov property, discrete time Markov chain (DTMC) and continuous time Markov chain (CTMC). We also discuss how to find the steady state distributions from these Markov chains and how they can be used to compute the system performance metric. The solution methodologies include a balance equation technique, limiting probability technique, and the uniformization. We try to minimize the theoretical aspects of the Markov chain so that the book is easily accessible to readers without deep mathematical backgrounds. We then introduce how to develop a Markov chain model with simple applications: a forwarding system, a cellular system blocking, slotted ALOHA, Wi-Fi model, and multichannel based LAN model. The examples cover CTMC, DTMC, birth-death process and non birth-death process. We then introduce more difficult examples in Chapter 4, which are related to wireless LAN networks: the Bianchi model and Multi-Channel MAC model with fixed duration. These models are more advanced than those introduced in Chapter 3 because they require more advanced concepts such as renewal-reward theorem and the queueing network model. We introduce these concepts in the appendix as needed so that readers can follow them without difficulty. We hope that this textbook will be helpful to students, researchers, and network practitioners who want to understand and use mathematical modeling techniques.

KEYWORDS

Markov Chain modeling, continuous time Markov chain, discrete time Markov chain, performance modeling, communication networks, Markov property, queueing theory, queueing network, balance equation, steady state distribution, uniformization, limiting probability, production form solution, Jackson network, BCMP network, wireless LAN, Wi-Fi, blocking probability, slotted ALOHA, Bianchi model, CSMA Markov chain, Multi-channel MAC

Contents

Preface ix

1 Performance Modeling 1

1.1 System, Model and Modeling 1

1.1.1 What Are System, Model and Modeling? 1

1.1.2 Why Modeling? 1

1.1.3 Classifications of Models 2

1.2 Performance Models 3

1.2.1 Simulation Models 3

1.2.2 Mathematical (Analytical) Models 5

1.3 Performance Study Steps 8

1.4 Towards a Good Performance Modeler 11

1.5 Summary 11

2 Markov Chain Modeling 13

2.1 What is Markov Chain Modeling? 13

2.2 Discrete Time Markov Chains 14

2.3 Defining States 18

2.4 Solving Discrete Time Markov Chains (DTMC) 19

2.4.1 Steady State Distribution π of DTMC 19

2.4.2 Power of Transition Probability Matrix 21

2.4.3 Solving the Balance Equations 22

2.5 Calculation of Performance Value 23

2.6 Continuous Time Markov Chain (CTMC) 23

2.6.1 Definition 23

2.6.2 CTMC Model: Rate Matrix 25

2.6.3 Birth-and-Death Process 26

2.7 Solving CTMC $(\mathcal{S}, \mathbf{Q})$ . . . 28

2.7.1 Solving Balance Equations of $(\mathcal{S}, \mathbf{Q})$ 28

2.7.2 Uniformization: $(\mathcal{S}, \mathbf{Q}) \to (\mathcal{S}, \mathbf{P})$ 30

2.8 Summary . . . 31

3 Developing Markov Chain Performance Models . . . 33

3.1 Performance Modeling Steps . . . 33

3.2 A Simple Forwarding System . . . 35

3.3 Cellular System with Blocking . . . 38

3.4 Slotted ALOHA . . . 42

3.5 Wi-Fi Network - CSMA Markov Chain . . . 46

3.6 A Multi-Channel MAC Protocol Model . . . 50

3.7 Summary . . . 55

4 Advanced Markov Chain Models . . . 57

4.1 The Bianchi Model . . . 57

4.1.1 Overview 57

4.1.2 The DTMC Model of a Backoff Timer 57

4.1.3 Network Model 60

4.2 A Multichannel MAC Protocol with a Fixed Duration - A Queueing Network Model . . . 62

4.2.1 Overview 63

4.2.2 Model Description 64

4.2.3 Numerical Example 68

4.3 Summary . . . 69

A Exercises . . . 71

Bibliography . . . 75

Author's Biography . . . 77

Index . . . 79

Preface

A Markov chain is a very powerful and widely used tool in various fields including physics, economics, engineering and so on. It is popularly used because of its simplicity, flexibility and ease of computing key metric. The Markov property, which a Markov chain must satisfy, makes the chain very simple to be described. Though it is simple, it is flexible enough to model various systems with arbitrary number of states and its transition matrix. Furthermore, the beauty of the Markov chain in the performance modeling is that it provides a simple numerical method to compute performance metric.

Even with the wide acceptance of the Markov chain modeling, many students who lack in knowledge of mathematics and stochastic processes find it very difficult to do the Markov chain modeling. Especially, the deep mathematical backgrounds of the Markov process make it hard for students to use the Markov chain modeling. Even students with mathematical backgrounds do not know how to develop a model, since the model development process is more of "art."

The purpose of this lecture is to provide concise, self-sufficient and easy to read materials for advanced undergraduate and graduate level students to understand the performance modeling methodologies with a Markov chain. The book is written as an introductory guide to Markov chain modeling for beginners. We try to answer how to define states and transition probabilities in Chapter 2 without detailing the theoretical aspects of a Markov chain. We provide many examples of modeling in Chapter 3 so that students can learn modeling by follow the examples. Examples in Chapter 4 are more advanced because they require knowledge of queueing networks and renewal reward theorem. They are good examples for advanced students who are interested in performance research with the Markov chain. We would like to provide students with various backgrounds how to develop a Markov chain model using various examples in communication networks.

I am grateful to many reviewers for their valuable comments with earlier versions of the draft. In particular, I thank Jiwoong Lee, Daehan Kwak, Moonsoo Kang, Jin Guang and two anonymous reviewers for providing constructive comments. I thank Mike Morgan of Morgan & Claypool for his encouragement and his help in getting this text reviewed and published. I thank professor Jean Walrand for guiding me and encouraging me to start writing this lecture note. Without his guidance, this book would not be possible. Most importantly, I would like to dedicate this book to my wife, Mijung Park for her unwavering support and my two young kids, Sangwoo and Sunwoo.

Jeonghoon Mo
Sinchon, Seoul
April 2010

CHAPTER 1

Performance Modeling

1.1 SYSTEM, MODEL AND MODELING

1.1.1 WHAT ARE SYSTEM, MODEL AND MODELING?

A *system*, which is a target of modeling, is a collection of interacting entities. More formally, it is "a set of components that are related by some forms of interaction, and which act together to achieve some objective or purpose." (25). For example, a computer system consists of hardware, operating system (OS), application software, and a networking system. These entities interact with each other to serve a common goal, which is to provide computing service to the computer user in this case. Another example of a system can be a wireless LAN system that consists of Wi-Fi devices and an access point. Wi-Fi devices are equipped with wireless LAN cards and send or receive packets to or from the access point. They collectively form a network system.

A *model* is a simplified representation of a system. Models are used to improve the understanding of systems. The are different types of models: physical, computer programs, and mathematical equations.

- **Test Dummy.** An example of a physical model is a crash test dummy. It is an anthropometric test device that represents a human being.
- **DNA Model.** Another example is the double-helix DNA model of Watson and Crick, which represents the structure of the DNA in a cell. The two scientists developed the model to visualize the structure and improve the understanding of the DNA.
- **Newton's Law.** The third model example is Newton's law of motion:

  $$F = ma,$$

  which states that the force (F) is equal to mass (m) times acceleration (a). The equation is very simple, but it enables to study the motion of objects.

Modeling is the process of developing a model from a target system. Figure 1.1 shows two steps involved in modeling.

1.1.2 WHY MODELING?

Models, simplified representations of systems, are used for various ways. Sometimes, it helps deepen understanding of the system. It also can be used to compare alternative inventions or policies. For

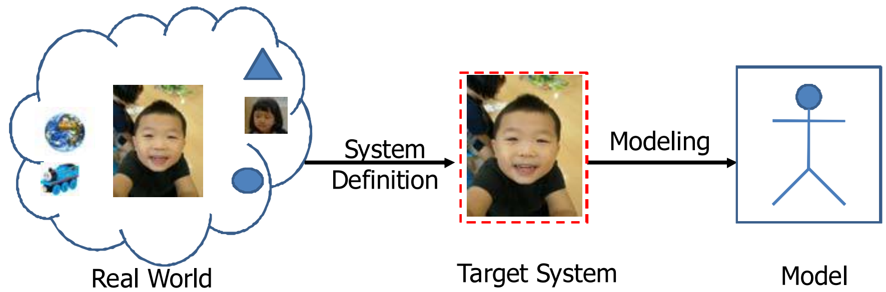


Figure 1.1: Modeling.

example, the basic understanding from the DNA model made it possible to complete the human GENOME project (1). The test dummies are used to improve the safety of vehicles. Mathematical models of Wi-Fi networks can be used to compare performance of different MAC protocols or to optimize system parameters.

Possible reasons for using a model instead of the target system are as follows:

- *A target system may not exist.* This situation usually happens in engineering fields when a company plans to develop a new product. For example, to understand the performance of its next generation microprocessor, Intel uses a model such as a computer simulation.

- *It can be dangerous to experiment with the target system.* In the case of the car crash test, experimenting with human beings is too dangerous, and it can be fatal. So dummies are used instead.

- *It can be expensive to experiment with the target system.* The flight simulator model is popularly used for training purposes since the cost of flying an airplane is so high that training with realistic simulation environments is preferred in airline companies.

1.1.3 CLASSIFICATIONS OF MODELS

There are many different models and various ways of classifying them.

- Models can be classified based on the *academic fields*. Examples include economic models, biological models, chemical models, molecular models, queueing models, and on.

- Another way of classifying models is based on *how they are represented*. Physical models use physical objects; mathematical models use mathematical languages; computer models use computer languages. A simulation model is a kind of computer model because most simulations are performed using computer and computer languages.

- A model can be *stochastic* or *deterministic*. Stochastic models are used to represent uncertainty in the real world. They are concerned with phenomena that vary as time advances, and where the variation has a significant chance component. For instance, consider the availability of a fax machine in the department office or the stock price of Samsung Electronics that varies over time. Deterministic models do not have the chance component that the stochastic models have. Their outcomes are precisely determined through combinations of events and states. Therefore, deterministic models behave the same when the initial conditions are the same. An example of a deterministic model is Newton's law of motion.

- A model is *static* if it does not account for the time element; otherwise, it is *dynamic*. A Markov chain model, which we will explain in Chapter 2, is dynamic because it models the change of states over time. An example of static model is Newton's model $F = ma$. A static model can be considered as a snapshot. Dynamic models can be further classified into *continuous time models* or *discrete time models* based on how time advances. In a discrete time model, time advances only at discrete points while a continuous time model advances continuously. A Markov chain can be either a discrete or a continuous time model.

- A *simulation model* is a computer program that attempts to simulate an abstract model of a certain system while an *analytical model* or a *mathematical model* uses mathematics to represent a particular system.

1.2 PERFORMANCE MODELS

A *performance model* is a model that is used to assess the performance of a system. It is a model whose objective is to evaluate the performance of a system.

Figure 1.2 shows different methods of studying the performance of a system.

1.2.1 SIMULATION MODELS

Simulation is the most popular technique for studying system performance by mimicking the behavior of the target system. The development of fast computers and flexible programming languages enabled wide acceptance of simulation methods as a performance study tool. These days, simulation studies are used to model from very small scale atomic systems to very large scale communication networks (9).

A *simulation model* is a computer program that mimics the behavior of a target system. With a flexible computer language, a modeler can describe the behavior of a target system in various details. The right part of Figure 1.3 shows a simulation model written in Tcl[1] with the NS-2 simulator (2). It models a computer network as shown on the left that consists of seven nodes and five links.

The simulation language can be either a specialized language such as SIMAN (21) and SIMULA (7) or a general purpose language such as C/C++, Java, or FORTRAN. Simulation

[1]Tcl is a popular script language created by John Ousterhout.

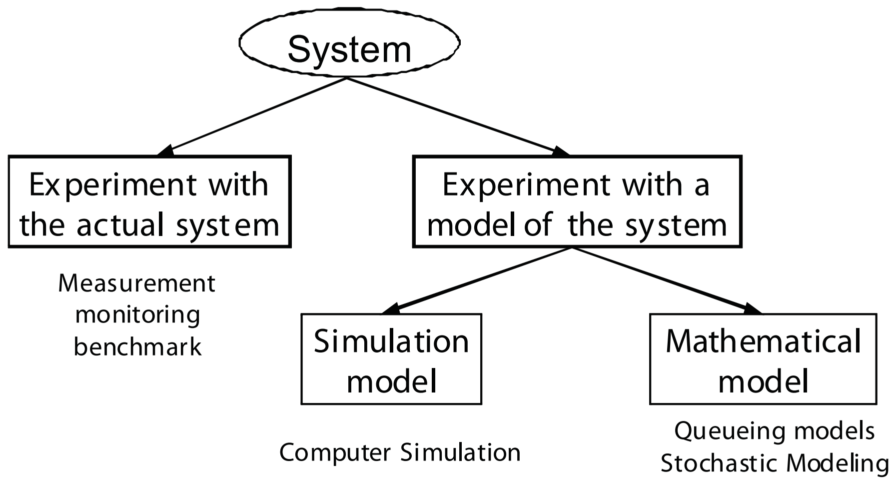


Figure 1.2: Different ways of studying a system.

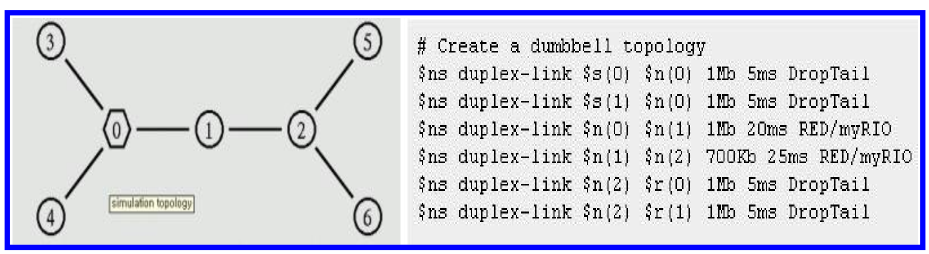


Figure 1.3: An example of a simulation model: The computer code is shown on the right. The left part of the figure shows the corresponding conceptual network topology.

languages save much time because they already include built-in tools for simulation event handling, time advancing, random number generation, data analysis and more. A modeler may choose a general purpose language such as C/C++ because he/she is familiar with it. The advantage of a general purpose language is its flexibility.

Most simulators are based on the *discrete event simulation* (DES) technique, in which the operations of a system are represented as a chronological sequence of events. In DES, each event occurs when the state of the system changes. For performance evaluation purposes, we mostly use DES. The *Monte Carlo simulation*, named after the Monte Carlo casino in Monaco, is another popular method to study static systems. Unlike DES, it is used to model a system that does not change status with time. That method is useful when a deterministic algorithm cannot provide an exact solution.

It uses repeated random inputs to calculate the desired output. For example, calculation of definite integrals is a common application of Monte Carlo simulation.

1.2.2 MATHEMATICAL (ANALYTICAL) MODELS

A mathematical model uses mathematical language to describe a system. Mathematical models can take various forms and are used in many different fields such as engineering, natural science and social science. A mathematical model can take the form of differential equations, e.g.

$$m\frac{d^2}{dt^2}x(t) = -\mathrm{grad}(V(x(t))), \tag{1.1}$$

where $x(t)$ is the position of a particle in a potential field in physics. It can be an optimization model of the following form,

$$\begin{aligned} &\text{maximize} \quad f(x) && (1.2)\\ &\text{subject to} \quad x \in \mathcal{C}, && (1.3) \end{aligned}$$

where f is an objective function and $\mathcal{C}$ is a constraint set.

In performance modeling, popular models include queueing models and stochastic models.

- *Queueing models* are used to analyze systems that provide services to customers whose arrival times and service durations are random. They can be used to model diverse systems including telecommunication, manufacturing, and more. Figure 1.4 shows a queueing model that mimics

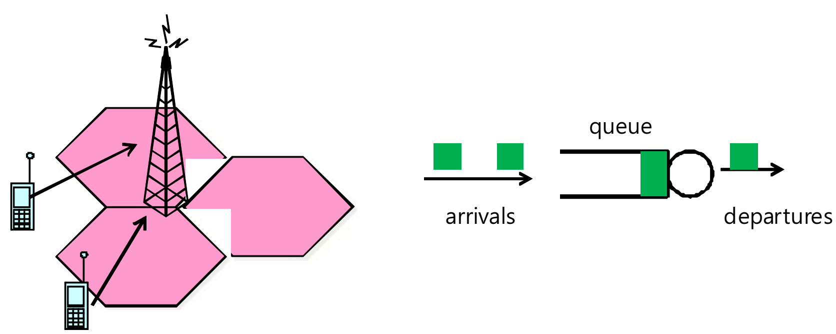


Figure 1.4: A Queueing Model Example (right) of Uplink Transmission of a Cellular Network System (left).

the behavior of a cellular data system. The system consists of a base station and two transmitters. The mobile transmitters send packets to the base station, which form an input traffic. Upon

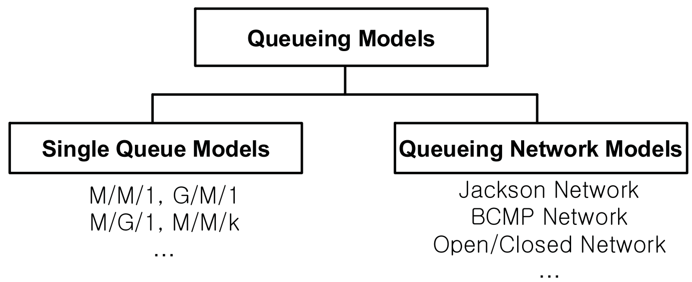


Figure 1.5: Classifications of Queueing Models.

reception of packets, the base station stores them, processes and forwards them in a first come, first serve manner. The buffer in the base station corresponds to a queue, as shown on the right side, and the CPU in the base station corresponds to the server of the queue.

A queueing model can be a single queue model or a queueing network model based on the number of queues in the model as shown in Figure 1.5. Single queue models are further classified by arrival process, service distribution, number of servers and so on. For example, M/M/1 [2] is a queueing model in which the input traffic is Poisson process; the service distribution is exponential and the number of server is 1. The *M* stands for memoryless, which we will explain in 2.6. The basic queueing models are combined to construct a *queueing network model,* which is a network of queues. The queueing network models can be used to analyze more complicated systems that consists of multiple queues. For more on models, refer to (14; 18; 15; 26).

The popularity of the queueing model comes from the fact that it provides useful steady state measures such as the average number in the queue, average time in the queue, the server utilization, and more. These service measures are related to user dissatisfaction and a queueing model provide a way to analyze the system.

- *Stochastic models* are popularly used in performance studies because they mimic the behavior of dynamic systems with uncertainties that evolves over time. The stochastic models use time sequenced collections of random variables to represent the system. The random variables are employed to model uncertainties in real world systems and "collections of them" are needed to mimic the behaviors over time. Consider the packet arrivals in the Figure 1.4. Packet generation times are uncertain because they depend on many unknown factors such as user behavior and

[2] A queueing model is usually specified by a formalism called *Kendall Notation*

$$A/B/C,$$

where A: arrival process; B: service distribution; and C: the number of servers.

more. Due to the uncertainties, we use stochastic input traffic model with a few parameters. The stochastic input traffic model can be used to construct a queueing model of the system.

The Markov chain or a renewal process are examples of the stochastic model. The stochastic model can be Markovian or Non-Markovian as shown in Figure 1.6. The Markov model is subclassified into discrete time Markov chain and continuous time Markov chain. Popular stochastic models include the birth-death process and Poisson processes. (Refer to 2.6.3).

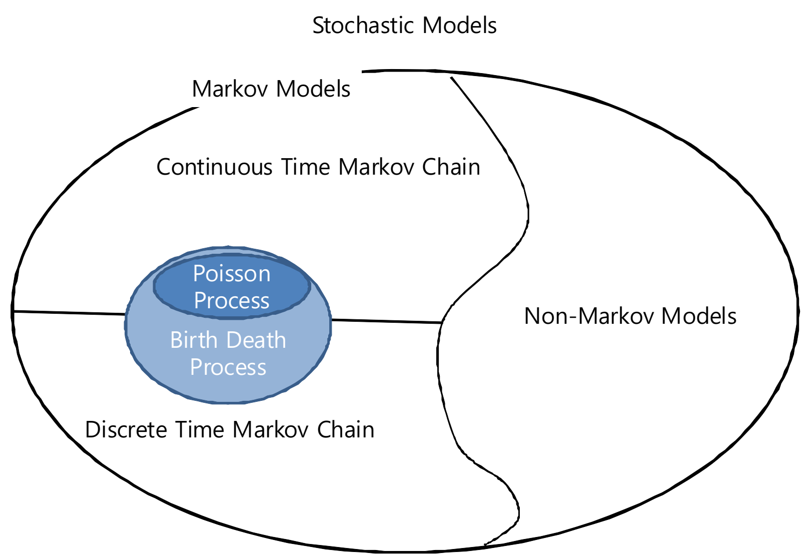


Figure 1.6: Stochastic Models.

- Other analytical performance models include regression models, reliability models, and so on. The regression model shows relationship between a dependent variable and one or more independent variables. It helps us to understand how the independent variables impact the dependent variable. For example, $y = ax + b$ is a linear regression model where x is an independent variable, y is a dependent variable, and [a, b] are constants. The reliability model is a probability model that help us to understand failure rate or availability of systems. Prevention of failure is very important in system such as nuclear plants, satellite or airplane as one failure can cause tragic outcomes.

1.3 PERFORMANCE STUDY STEPS

A performance study follows well structured steps as shown in Figure 1.7, which starts with *system definition* and ends up with *results interpretation*.

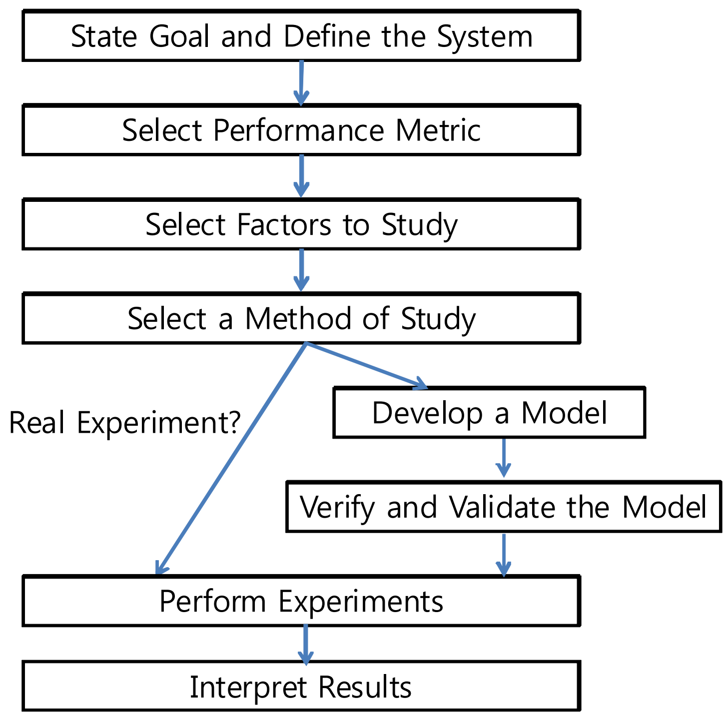


Figure 1.7: Performance Study Steps.

1. **State Goals and Define the System.** When we start a performance evaluation project, the first question to answer is *"What do you want to know or understand from this study?"* The system definition comes next and basically answers the question "What constitutes the system or what are components of the system?" by delineating the boundaries.

   The goal selection is very important since not only can it change a system definition, but also it impacts the modeling step. Assume that you are a network administrator of Yonsei University. Given the campus network, the goal may be to decide whether or not to upgrade the access router. In this case, the system would consist of the access router of which performance depends heavily on the input traffic. On the other hand, if the goal is to decide additional memory should be install in the access router, the system may be limited to memory unit in the access routers and other components that interact with the memory.

2. **Select Performance Metric and Factors.** A metric is the criteria used to compare system performances. The most popular performance metrics in network systems include throughput, delay, and blocking probability. Availability is another popular metric used in many system performance assessments. In the above example of Yonsei University, we can pick the delay and the throughput per user as the system metrics.

   There are many parameters that can affect the system performance. To perform a thorough study, we need to list all of those parameters that can affect performance. Among those, some are more important than others, or some are controllable and some are not. Those controllable parameters, that we would like to vary to see their impact, are called factors.

3. **Select a Method of Study.** Performance study can be done in many different ways as shown in Figure 1.2. One can experiment with a real system or can play with a mathematical or a simulation model. Table 1.1 shows characteristics of different evaluation methods. Regarding

Table 1.1: Comparison of Evaluation Techniques

	real system	simulation	mathematical
cost	high	medium	low
accuracy	high	medium	low
required time	depends	medium	short

   the accuracy of results, the real system is the best; simulation model is the second; the mathematical model is the last. This is because modeling typically includes simplifying assumptions in the model development process. The mathematical model is more restrictive than the simulation model as it requires more simplifying assumptions in general. However, the analytical model is, in general, cheaper and takes less time than other methods. The real system is more expensive than the mathematical model.

   Therefore, the methodology selection is typically carried out in consideration of *cost*, required *accuracy of output*, and *available time*. If you need an answer in a few days, a simple analytical model can be considered at the sacrifice of accuracy. However, if accuracy of results is important, real measurement can be selected.

   Sometimes, the real system may not exist. In such cases, model development is a must. This often happens in the new product development process. When a company develops a next generation product, it must develop a model to predict the performance.

4. **Develop a Model.** The process of model development is an art since it involves subjective handling. A modeler's intention is reflected when (s)he decides the scope, timescale, and complexity of a model.

   The modeler needs to figure out important aspects of the system in relation to the goal of a given study. More important parts are modeled in details while less important parts are simplified or left out. For example, if the goal is to understand the impact of a transport protocol, the details of the protocol should be kept while the network topology can be simplified to a two node network with a source and a destination. However, if the goal is to understand the performance of an ethernet based network, the transport protocol can be simplified, but the details of ethernet based protocol should be kept in the model.

   The timescale of a model should be selected appropriately also given the goal. A CPU chip designer would like to mimic the cycle accurate behavior of chip on the scale of nanoseconds while software designer can consider the CPU as a blackbox with a timescale of microseconds. Depending on the goal, appropriate timescale can be different.

   A model can be very accurate and close to a target system or very coarse and simplified. A modeler decides the complexity of model with consideration of the goals of study. When the goal is to assess the performance of a computer, important subsystem of a computer should be modeled. However, when the goal is to understand the performance of computer network with 100 computers, each computer is considered as a blackbox with the details of computers being neglected.

   The budget and time constraints also impact the decision on the complexity. The less the available time and cost, the coarser the model. When accuracy is important, detailed model can be a better choice. In general, as there are trade offs between accuracy and cost, a good modeler balance the two aspects to achieve the goals of the study.

   Development of a simulation model is a writing a computer code that mimics behavior of a target system. An analytical model uses mathematical equations or stochastic processes for representing a performance model. An analytical model is in general more concise and simpler than the computer simulation code. However, it is more restrictive with stronger assumptions than a simulation model. On the other hand, simulation models are more flexible and can be more realistic. Another advantage of a mathematical model is that it provides valuable insight on system behaviors while developing the model.

   We discuss the details of Markov chain model development in Section 2.

5. **Verify and Validate the Model.** Once a model is developed, the modeler needs to verify and validate the model. Verification and validation, often used interchangeably, mean different things. Verification is a process of making sure the model does what it is intended to do while the validation is checking the developed model for a good representation of the real world target system. Suppose that you develop a conceptual model of your target system before

writing a simulation code or a simulation model. The computer code that you have written may include logical errors or bugs. Checking whether your computer code does not have these bugs is the process of verification. Validation is checking the model against a real target system. Even though your code may not have any logical error, it still may not be a good representation of the target system. Verification and validation and are often blended in practice.

Oftentimes, we can observe cross validation between simulation and analytical models in research papers. People use its counter-part to validate a model. A simulation model is used to validate an analytical one or vice versa.

6. **Perform Experiments and Interpret Results.** With a developed model or a real testbed, you can perform experiments to see the impact of the selected factors. You can design your experiments to assess the impact of factors. Once you collect the results, you try to interpret them to meet the goals of the study.

1.4 TOWARDS A GOOD PERFORMANCE MODELER

Becoming a good performance modeler requires expertise in both target systems and performance models; getting expertise takes time and effort. As we mentioned in Section 1.1, modeling is a process of developing a replica of a target system. Since the replica uses a mathematical language, expertise in mathematical language is a must to be a good modeler. In addition, the modeler must have expertise in the field of the target system. For example, he/she needs to have a good knowledge of the communication network area to model a communication network. Without full understanding of the system, it may not be an easy task to grasp the important aspects. Since these important aspects of the system should be reflected in a good model, the modeler's judgement based on solid knowledge is essential.

In performance modeling, there are a great number of models, each of which can be mathematically deep. They include queueing models such as M/M/1, M/M/k, queueing network models, Markov models, renewal models and more. Getting familiar with these models takes some time and effort. Furthermore, it is the job of a modeler to select an appropriate model to use for a given system. For instance, as a professional golfer selects one of his irons from his bag based on his judgement, a good modeler needs to select a suitable model for a given system. A professional golfer has prior knowledge on the distance of each iron for given conditions. Therefore, a good modeler should have some knowledge on models to be able to pick an appropriate model.

1.5 SUMMARY

- A *system*, a collection of related entities, is a target of modeling process, and it has a boundary.
- A *model* is a simplified representation of a system with a purpose, and it is used to gain a deeper understanding of its target system. We call the process of developing a model of a system *modeling*.

- A performance model is a model with an objective to evaluate the performance of its target system. It can be classified into a simulation model and a mathematical model. Popular mathematical models include queueing models, stochastic models, and Markov chain models.
- Developing a performance model is a challenging task because modelers need expertise in both mathematical languages and their fields of specialty.

CHAPTER 2

Markov Chain Modeling

Figure 2.1: Andrey Markov (1856-1922), A Russian Mathematician.

A *Markov chain,* named after a Russian mathematician Andrey Markov (1856-1922), is one of the most popular mathematical tools that is used to model a dynamic system that changes its state over time. It is used in various fields including engineering, economics, genetics, social sciences and more. Its popularity stems from various reasons including its simplicity, flexibility and ease of computation. In this chapter, we introduce backgrounds on the Markov chain in order to execute performance modeling and evaluation. For more information on this subject, interested readers may refer to the textbook (22).

2.1 WHAT IS MARKOV CHAIN MODELING?

A Markov chain enables us to model a *dynamical system,* which is defined as one that changes its state over time. Consider a number $x(t)$ of customers waiting in a line at time t. It is a dynamical system since its state $x(t)$, the number of customers, changes over time. Another example can be the stock price $p(n)$ of Samsung Electronics at the end of day n for $n = 1, 2, \cdots$. Note that the price varies over time and the stock system can be considered as a dynamical system.

If the state changes at predefined instants as in the stock price of day n, it can be modeled by a discrete time model; otherwise, a continuous time model is more suitable. For instance, the

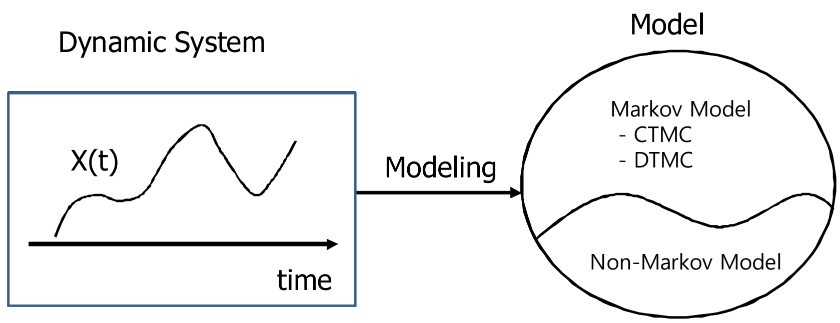


Figure 2.2: A dynamical system can be modeled into a Markov model or a non-Markov model.

number $x(t)$ of customers is better represented by a continuous time model. Markov models are also classified into discrete time Markov chains (DTMC) or continuous time Markov chains (CTMC), depending on when the state changes.

Not all characteristics of dynamical systems can be represented by a Markov chain, only those that satisfy the so called "Markov property." Basically, the Markov property says that the transition rule is simple enough to be represented by a matrix. [1] Characteristics without the Markov property are modeled with different methods such as simulation or non-Markov models.

With the help of the property, Markov models are specified by two components: the state space $\mathcal{S}$ of the system and the state transition rules. If the Markov model is a DTMC, the transition rules are represented by a transition probability matrix $\mathbf{P}$; if the model is a CTMC, the rules are represented by a transition rate matrix $\mathbf{Q}$.

Accordingly, the Markov chain is specified by $(\mathcal{S}, \mathbf{P})$ if the model is a DTMC and by $(\mathcal{S}, \mathbf{Q})$ if it is a CTMC.

2.2 DISCRETE TIME MARKOV CHAINS

- A discrete time Markov chain (DTMC) is a sequence of random variables called *states* $\{X(n), n = 1, 2, 3, \cdots\}$ with the Markov property which states that the future states are independent of the past states given the current state.

- More formally, the *Markov property* can be defined as follows: A discrete-time stochastic process $\{X(n), n = 0, 1, 2, \cdots\}$ has a Markov property if

$$\begin{aligned} P(X(n+1) = j | X(n) = i, X(n-1) = i_{n-1}, \cdots, X(0) = i_0) \\ = P(X(n+1) = j | X(n) = i), \forall n. \end{aligned}$$

[1]We will discuss that property in more details in 2.2 and 2.6.

This definition specifies that, given the current state $X(n)$, the probability distribution of the future state $X(n+1)$ depends does depend on the previous $\{X(m), m < n\}$.. In other words, the current state provides enough information to predict the future state.

- A Markov chain can be used to model a *dynamical system* whose state changes over time. More formally, if we let $X(n)$ be the state of a system at time n, the sequence $\{X(n), n = 1, 2, 3, \cdots, \}$ of states represent a dynamical system.

- The *state space* $\mathcal{S}$ is the set of all possible states. At any instance, the state of a Markov chain belongs to the state space $\mathcal{S}$. The set $\mathcal{S}$ can be finite or countably infinite, as we explain in examples below.

  Example: Consider a weather dynamical system $\{X(n), n = 1, 2, 3, \cdots\}$ whose state $X(n)$ represents the weather in Seoul at day n. The sequence forms a dynamical system. Assume that the weather can be either sunny or rainy, so that the weather system can be in one of two states: Sunny or Rainy. Thus, the state space $\mathcal{S}$ of the weather system is

  $$\mathcal{S} = \{\text{Sunny, Rainy}\}.$$ [2]

- In the DTMC, the transition rule between states is specified by *transition probabilities.* For the purpose of discussing a simple model, we make the simplifying assumption that sequence of characteristics Sunny or Rainy of the weather in Seoul on successive days is a DTMC.

  The transition probability p_{ij} from state $i \in \mathcal{S}$ to $j \in \mathcal{S}$ can be written as follows:

  $$p_{ij} = P\{X(n+1) = j | X(n) = i\}, i, j \in \mathcal{S}, \tag{2.1}$$

  where $X(n)$ denotes the state of a Markov chain at time n. It is a conditional probability that the process is in state j at time $n+1$ given that it is in state i at time n. The next state is probabilistically determined by the transition probabilities.

  In the case of the weather example, there are four possible transitions: Sunny → Sunny, Sunny → Rainy, Rainy → Sunny and Rainy → Rainy. Let $p_{SS} = a$, $p_{SR} = b$, $p_{RS} = c$ and $p_{RR} = d$ be the transition probabilities for the four cases. The transition probabilities can be shown in a matrix form, which is called "*transition probability matrix*" as follows:

  $$\mathbf{P} = \begin{bmatrix} p_{SS} & p_{SR} \\ p_{RS} & p_{RR} \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}. \tag{2.2}$$

  Since the conditional probability must satisfy the *axioms of the probability,*

  1. $a, b, c, d \geq 0$;

[2]We are making simplifying assumptions that weather does not change during a day and that there are only two types of weather, sunny and rainy.

2. $a + b = 1$;
3. $c + d = 1$.

So the Markov chain model can be rewritten:

$$\mathcal{S} = \{S, R\}, \ \mathbf{P} = \begin{bmatrix} a & 1-a \\ 1-d & d \end{bmatrix}. \tag{2.3}$$

- In general, a transition probability matrix can be represented by a square matrix $\mathbf{P}$:

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1j} & \cdots & p_{1m} \\ p_{21} & p_{22} & \cdots & p_{2j} & \cdots & p_{2m} \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ p_{i1} & p_{i2} & \cdots & p_{ij} & \cdots & p_{im} \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ p_{m1} & p_{m2} & \cdots & p_{mj} & \cdots & p_{mm} \end{bmatrix}, \tag{2.4}$$

where $p_{ij} := P(X(n+1) = j | X(n) = i), \forall i, j \in \mathcal{S}$ and m is the size of $\mathcal{S}$. For a matrix to be a transition probability matrix, its elements satisfy

$$p_{ij} \geq 0, \sum_j p_{ij} = 1, \forall i \in \mathcal{S}.$$

- A *state transition diagram* can be used to visualize a Markov chain. Figure 2.3 shows the state transition diagram for the Markov chain of (2.3). Each node corresponds to a state $\in \mathcal{S}$ and a directed link from node i to node j corresponds to transition probability p_{ij}. There exists a link from node i to node j iff $p_{ij} > 0$.

  In the figure, we can see that the chance of tomorrow being sunny on the condition that today is sunny is p and that the chance of tomorrow being rainy is $1 - a$. Similarly, the chance of tomorrow being rainy (sunny) given that today is rainy is d $(1 - d)$.

- A Markov chain model is a probability model because its transition rule is a probabilistic one described by a "transition probability matrix." When a chain is in state i at time n, its next state is probabilistically determined by the i-th row $[p_{ij}, \forall j \in \mathcal{S}]$ of the transition probability matrix. In the weather example, if today's weather is 'Sunny', it will be 'Sunny' tomorrow with probability of a or 'Rainy' with probability of $1 - a$.

- Note that in a Markov chain model, the probability of future state $X(n+1)$ depends on the current state $X(n)$ but not on the history of state changes $\{X(m), m < n\}$ when n is the current. If it is 'Sunny' today, the probability of being 'Rainy' tomorrow is $(1 - a)$ in the example. Whether or not it was 'Rainy' or 'Sunny' yesterday, the probability is the same as

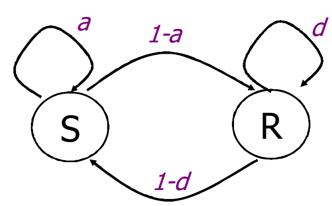


Figure 2.3: A State Transition Diagram.

long as it is 'Sunny' today. In other words, the current state provides enough information to predict the future state. Note also the dependence of the future state on the current state. The probability of being 'Rainy' is different if is it 'Sunny' or 'Rainy' today.

For this example, the Markov property does not really hold, i.e., the current state does not fully characterize the future state of a process. In such a case, the transition probability matrix is not enough to specify the evolution of states. Tomorrow's weather depends on today's weather but also on the weather of the previous days. Since the transition probability matrix does not provide enough information, we cannot use the Markov chain model in this case. We should understand that the 'Markov property' is key to a Markov chain model.

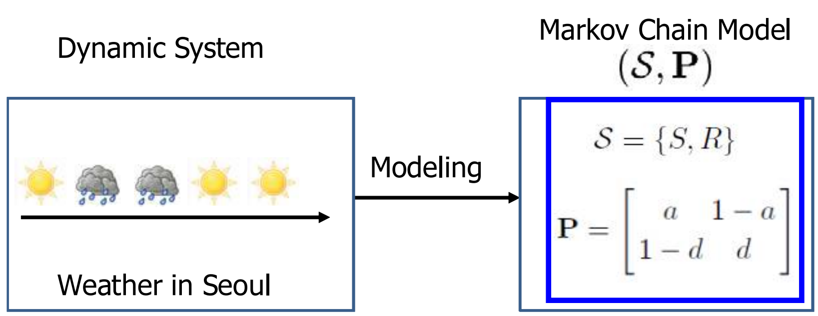


Figure 2.4: A Markov chain models a dynamic system by specifying $(\mathcal{S}, \mathbf{P})$ where $\mathcal{S}$ is the set of states and $\mathbf{P}$ represent transition rules. A dynamical system changes its states over time.

2.3 DEFINING STATES

When creating a Markov chain model of a system, we need to define a set of appropriate states. In the weather example, we defined two states {Sunny, Rainy} and a corresponding transition probability matrix. By assuming the Markov property, we created the Markov chain model of the weather system. Are the two states appropriate?

To be an appropriate set of states, the states of a Markov chain must satisfy two criteria:

- The information contained in the definition of each state must be sufficient to allow you to achieve the goals of the study.

- The states must contain enough information to allow you to construct the single step transition probability matrix P or the transition rate matrix Q.

The first criterion specifies that we should be able to compute a required performance metric from the steady state solution of the Markov chain. Thus, the state definition is appropriate if it can serve the purposes of the study. Regarding the weather model, we cannot answer the appropriateness of state definition, because the goals are not specified.

The second criterion concerns the Markov property, which states that the current state provides sufficient information to determine the future. We should define states so that the state transition rules satisfy the Markov property. In the weather model; then, remember that we assumed that today's weather is sufficient to predict that of tomorrow in a probabilistic manner. With the assumption, we come up with a Markov chain model $\{X(n), n = 1, 2, 3, \cdots\}$. However, if the assumption does not hold, the model may not be a good one. Suppose that a careful study reveals that the weather of tomorrow depends on that of today and also that of yesterday. The modeler thought that it is important to reflect the new study in the weather model, then we need to define a new states. Note that if we define a new state $Y(n) := (X(n+1), X(n))$, $\{Y(n), n = 1, 2, 3, \cdots\}$ holds the Markov property because knowing $Y(n-1)$ provides sufficient information to predict $Y(n+1)$, Hence, $\{(X(n+1), X(n)), n = 1, 2, 3, \cdots\}$ is a Markov chain. The state space is $\mathcal{S} = \{(S, S), (S, R), (R, S), (R, R)\}$, and the transition probabilities need to be changed appropriately.

In general, we have multiple options on how to define states that satisfy the two criteria discussed above. The simplest possible model is preferable because it makes computations of the steady state solution easier.

As an example, suppose that we would like to understand the performance of a Wi-Fi system with two Wi-Fi devices, say D1 and D2, and two available channels, say C1 and C2. Each device works in a time slotted manner. In the beginning of the slot, a device either decides to transmit a packet in one of the channel or remains idle. Upon decision of transmission, it checks the availability of a randomly selected channel before transmission. If the channel is idle, then it starts to transmit a packet. Otherwise, it waits for a random duration.

One possible definition of states that satisfy the two criteria is $X_1 = (x_1, x_2)$ where x_i is the location of D_i for $i = 1, 2$. This definition forms state space S_1 given by:

$$\mathcal{S}_1 = \{(\text{Idle, Idle}), (\text{C1, Idle}), (\text{Idle, C1}), (\text{C2, Idle}), (\text{Idle, C2}), (\text{C1,C2}), (\text{C2,C1})\}.$$

However, we can reduce the number of states by using the fact that the devices behave similarly. For the purpose of analyzing the transmission rate of the system, it is not necessary to keep track of the status of each device. Accordingly, we define the state to be the status of the two channels. Then the state space is

$$\mathcal{S}_2 = \{(Idle, Idle), (Busy, Idle), (Idle, Busy), (Busy, Busy)\}.$$

We can further reduce the number of states by defining the state to be the number of busy channels. Then the corresponding state space is

$$\mathcal{S}_3 = \{0, 1, 2\}.$$

With the three definitions of state above, we can compute the steady state throughput of the system from the steady state probabilities. The simplest model is the last one for which the system throughput is $1\pi_1 + 2\pi_2$ where π_i is the steady state probability of state i for $i = 1, 2$, respectively.

As another example, consider two models of voice traffic popularly used in a telephone performance test as shown in Figure 2.5. The first model, shown on the left of the figure, has two states whereas the second model shown on the right has six states. In the first model, a caller alternates between *talk* and *silence* states. The second model considers two speakers, A and B. The six states represent the state of the two speakers and the possible sequences of states. Since each speaker can be in either a talk or silence period, there are at least four states. The model further divides the mutual silence and double talk states into two. Hence, there are six states. The arrows in Figure 3.11 show appropriate events that trigger state changes.

2.4 SOLVING DISCRETE TIME MARKOV CHAINS (DTMC)

In the previous section, we learned that if a sequence of random variables $\{X(n), n \geq 0\}$ with a state space $\mathcal{S}$ has the Markov property, we can specify it simply by its state space and transition probability matrix tuple $(\mathcal{S}, \mathbf{P})$. In this section, we will see why Markov chain models are popular and useful by learning how to get steady state probabilities.

2.4.1 STEADY STATE DISTRIBUTION π OF DTMC

A Markov chain model is popularly used because it is easy to compute a steady state distribution, which can be understood as 'a long term fraction of time being in a state'. We call computing the steady state distribution π of a Markov chain model specified by $(\mathcal{S}, \mathbf{P})$ 'solving the Markov chain,' as shown in Figure 2.6. One can show that a DTMC with a finite state space has one and only one

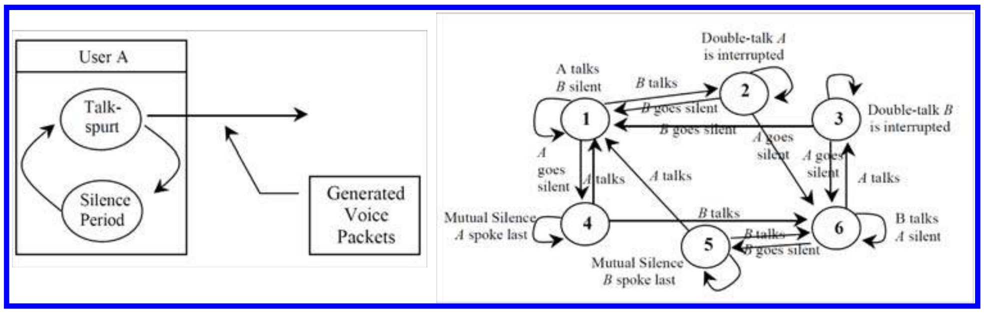


Figure 2.5: Two voice traffic models: two-state model (left) and six-state model (right).

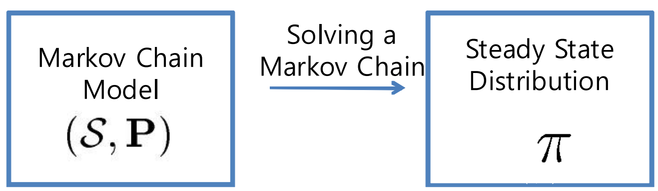


Figure 2.6: Solving a Markov Chain.

steady state distribution if the process X_n can go from any state to any other state (not necessarily in one step). Markov chains with this property are said to be *irreducible*.

From the obtained *steady state distribution* or π, we can easily compute a metric of our interests. The umbrella manufacturer can utilize the fraction of Rainy days to forecast its demands. The bank manager can estimate the fraction of idle bank tellers from the steady state probability that there are no customers. Similarly, the blocking probability of a cell in a cellular network can be calculated from the fraction of time that the number of on-going calls is equal to the capacity of the cell. As these examples show, the steady state distribution of a Markov chain enables to calculate useful performance metrics of a system when the model is appropriate.

There are two standard methods for calculating the steady state distribution π of a Markov chain model $(\mathcal{S}, \mathbf{P})$:

- Limit of the power of the transition probability matrix $\mathbf{P}^n$.

- Solving the balance equations $\pi = \mathbf{P}\pi$.

2.4.2 POWER OF TRANSITION PROBABILITY MATRIX

The easiest way of calculating a steady state distribution is through the powers of the transition probability matrix $\mathbf{P}$. Consider the weather example with $(a, d) = (0.6, 0.75)$ as shown below:

$$\mathcal{S} = \{S, R\}, \ \ \mathbf{P} = \begin{bmatrix} 0.6 & 0.4 \\ 0.25 & 0.75 \end{bmatrix}. \tag{2.5}$$

If we calculate powers of matrix $\mathbf{P}$,

$$\begin{aligned} \mathbf{P}^{(2)} &= \begin{bmatrix} 0.4600 & 0.5400 \\ 0.3375 & 0.6625 \end{bmatrix} \mathbf{P}^{(4)} = \begin{bmatrix} 0.3938 & 0.6061 \\ 0.3788 & 0.6212 \end{bmatrix} \\ \mathbf{P}^{(8)} &= \begin{bmatrix} 0.3848 & 0.6152 \\ 0.3845 & 0.6155 \end{bmatrix} \mathbf{P}^{(16)} = \begin{bmatrix} 0.3846 & 0.6154 \\ 0.3846 & 0.6154 \end{bmatrix}. \end{aligned}$$

- Note first that the entries of the columns of the matrices converge to the same value as the power of $\mathbf{P}$ increases. The difference of two values in the first column decreases from $0.35(0.6 - 0.25)$ in $\mathbf{P}$ to $0(0.3846 - 0.3846)$ in $\mathbf{P}^{(16)}$. After convergence, any row of the converged matrix, is a row vector equal to the steady state distribution. In this example, the steady state distribution $(\pi_s, \pi_r) = (0.3846, 0.6154)$. The steady-state distribution also provides the long-run probability of sunny weather (about 38%) and of rainy weather (about 62%).

- The theoretical result is as follows. Assume the existence of $\lim_{n\to\infty} \mathbf{P}^{(n)}$ and let

$$\Pi = \lim_{n\to\infty} \mathbf{P}^{(n)}. \tag{2.6}$$

  Then each row vector of Π is identical and taking the first row give a steady state distribution π of $\mathbf{P}$. In practice, multiply $\mathbf{P}$ until the maximum difference in all columns is less than a given threshold value, say 10^{-5}. This limit exists for irreducible Markov chains such that the process X_n can go from a state to itself in a number of steps that is not necessarily a multiple of a integer larger than one. Such a Markov chain is said to be *aperiodic*.

- The n-th power $\mathbf{P}^{(n)}$ of $\mathbf{P}$ is called "n-step transition probability matrix" because its element $p_{ij}^{(n)}$ corresponds to the probability that $X_{(m+n)} = j$ given that $X_m = i$ or $P[X_{(m+n)} = j|X_m = i]$. To see this, let us use the weather example again. We like to compute the probability that it will rain the day after tomorrow given that it rains today or $P[X_3 = R|X_1 = R]$. Since it can be 'Sunny' or 'Rainy' tomorrow, two possible paths are $R \to R \to R$ and $R \to S \to R$. $P(R \to R \to R) = \frac{3}{4} \cdot \frac{3}{4} = \frac{9}{16}$. $P(R \to S \to R) = \frac{1}{4} \cdot 0.4 = 0.1$ Since the two events are exclusive, $P[X_3 = R|X_1 = R] = \frac{9}{16} + 0.1 = 0.6625$. Note the value is the same as the value of $p_{RR}^{(2)}$ in $\mathbf{P}^{(2)}$.

- The above result follows from the well-known *Chapman-Kolmogorov* equation

$$P_{ij}^{(m+n)} = \sum_{k \in \mathcal{S}} P_{ik}^{m} P_{kj}^{n}, \forall n, m \geq 0, \forall i, j, \tag{2.7}$$

which provides a way to compute n-step transition probabilities. It says that probability of moving from state i to state j in $m+n$ steps is the sum of multiplications $p_{ik}^{(m)}$ and $p_{kj}^{(n)}$ over all possible k. This result follows from the Markov property and conditioning on k. In a matrix form, it can be

$$\mathbf{P}^{(m+n)} = \mathbf{P}^{(m)} \cdot \mathbf{P}^{(n)}. \tag{2.8}$$

That is, the n-step transition probability can be computed from simple matrix multiplications.

- The limit of the n-step transition probability $p_{ij}^{(n)}$ is π_j. This shows that, as n increases, the impact of the initial state i diminishes and the limit depends on only final state j not on the initial state i (if the Markov chain is irreducible and aperiodic).

2.4.3 SOLVING THE BALANCE EQUATIONS

The second way of calculating the steady-state distribution of a Markov chain $(\mathcal{S}, \mathbf{P})$ is by solving the following balance equations:

$$\pi = \pi \mathbf{P}, \quad \sum \pi_j = 1. \tag{2.9}$$

where π is a vector and $\mathbf{P}$ is a square matrix. Note that the number of equations is larger by one than that of variables. However, since there is a dependency in $\mathbf{P}$, we can solve the above equations after dropping one of the equations in $\pi = \pi \mathbf{P}$.

- In the weather example, we have the following balance equations

$$(\pi_s, \pi_r) = (\pi_s, \pi_r) \begin{bmatrix} 0.6 & 0.4 \\ 0.25 & 0.75 \end{bmatrix}, \quad \pi_s + \pi_r = 1. \tag{2.10}$$

There are three equations and two variables in this case. Note, however, that the first two equations are the same since they both reduce to $0.4\pi_s = 0.25\pi_r$. Consequently, we have two equations and two variables. If we solve the equation, we have $(\pi_s, \pi_r) = (\frac{5}{13}, \frac{8}{13})$, which is the same as the solution in the first approach.

- Rewriting the i-th balance equation gives

$$\pi_i = \sum_j \pi_j p_{ji} \text{ or } \pi_i (1 - p_{ii}) = \sum_{j \neq i} \pi_j p_{ji}. \tag{2.11}$$

The left term $\pi_i (1 - p_{ii})$ can be interpreted as the long term transition rate out of state i while the right term $\sum_{j \neq i} \pi_j p_{ji}$ is the long term transition rate into state i. The term "balance" in the balance equation comes from this observation that the rates in and out of state i are equal or balanced.

2.5 CALCULATION OF PERFORMANCE VALUE

Once you have the steady state distribution π of a Markov chain $(\mathcal{S}, \mathbf{P})$, you can use it to calculate the value of a performance metric of interest, which typically is the average value $E[F(X)]$ of a function $F(\cdot)$ of the state X. One has

$$E[F(X)] = \sum_{j \in \mathcal{S}} F(j) \cdot \pi_j. \tag{2.12}$$

- In Section 3.2, if X represents the number of packets in the buffer, and we are interested in knowing the average number of packets in the system, $F(X) = X$ and

$$E[F(X)] = E[X] = \sum_{j=1}^{\infty} j \cdot \pi_j \ .$$

- In Section 3.3, when a cellular system allows K simultaneous calls in a single cell, the blocking probability of the system is the probability that arriving calls sees the system is in state K. If we assume the Poisson arrival process, then π_K is the blocking probability.

Oftentimes, the reward at state $X = j$ is an independent random variable R_j with mean value $E[R_j]$. In this case, we can calculate

$$E[R] = \sum_{j \in \mathcal{S}} E[R_j] \cdot \pi_j. \tag{2.13}$$

Consider, for example, a carpenter who makes his living by producing wooden chairs. Assume that his productivity depends on the weather of the day. On sunny days, he produces five chairs, on average. However, on rainy days, he produces only three chairs, on average. We can compute the average number of chairs the carpenter produces per day by using the above formula and assuming that the weather follows the Markov chain in the previous section. The average number of chairs that he produces is

$$E[R] = E[R_s] \cdot \pi_s + E[R_r] \cdot \pi_r = 5 \cdot \frac{5}{13} + 3 \cdot \frac{8}{13} = \frac{49}{13} \approx 3.77 \text{ chairs/day.} \tag{2.14}$$

2.6 CONTINUOUS TIME MARKOV CHAIN (CTMC)

This section defines the *continuous time Markov chain* and discusses a number of examples. It also explains how to calculate the steady state distribution of such a process.

2.6.1 DEFINITION

A continuous time stochastic process $\{X_t, t \geq 0\}$ is a *continuous time Markov chain* (CTMC) if it has the following Markov property:

$$P[X_{t+s} = j | X_s = i, X_u = x_u, 0 \leq u < s] = P[X_{t+s} = j | X_s = i] \ \forall i, j \in \mathcal{S}, \forall s, \forall t, \tag{2.15}$$

where $\mathcal{S}$ is the state space. This property, as in discrete-time, says that, given the current state X_s, the future state X_{t+s} and past states $\{X_u, u < s\}$ are independent. That is, the current state X_s is sufficient to determine the evolution of the process.

Suppose that a CTMC X_t enters state i at time 0 and that it does not leave the state for a duration of s. What is the probability that this process stays in state i for the next t units of time? If we let Y_i denote the amount of time that the chain stays in state i before jumping to another state, then

$$P[Y_i \geq s + t | Y_i \geq s],$$

is the answer to the question. Due to the Markov property, given the current state $X_s = i$, the history of the process between $[0, s]$ is independent of the future evolution. Hence, we have

$$P[Y_i \geq s + t | Y_i \geq s] = P[Y_i \geq t]. \tag{2.16}$$

A random variable is said to be *memoryless* if it satisfies equation (2.16). The conditional probability that you wait another t (e.g., 10) time units for the event given that you have already waited s (i.e., 20) time units is not different from the probability that you wait t (i.e., 10) time units for the event from time 0. In other words, the distribution of remaining time until the next event is the same regardless of the duration since the past event. For example, assume that you have used your notebook computer for three years. If the lifetime of your notebook computer is memoryless, then the remaining life distribution of your notebook is the same as that of a brand new notebook computer of the same model. Is it the case that a three year old laptop has the same residual life time as a new one? Probably not. Thus, laptop computers do not have a memoryless life time.

The only continuous random variable Y_i with the memoryless property is the exponential random variable. This random variable has the probability density function $f(\cdot)$ given as follows:

$$f(y) = q_i e^{-q_i y}, \; y \geq 0, \tag{2.17}$$

where q_i is a parameter that represents an average rate of event occurrences. [3] You can check that the expected duration of the exponential random variable is $E[Y_i] = \frac{1}{q_i}$.

We learned that a CTMC stays in state $i \in \mathcal{S}$ for an exponential random duration with rate q_i before moving to another state. When it changes its state, where does it jumps to? In DTMC, transition probability p_{ij} determines the next state. Similarly, we can construct a CTMC, by specifying transition probability p_{ij} where $\sum_{j \neq i} p_{ij} = 1$. After staying in state i for an exponential duration with rate q_i, it moves to state j with probability p_{ij}. Hence, One way to construct a CTMC is by specifying $(\mathcal{S}, q_i, p_{ij}, \forall i, j)$.

Alternatively, we can specify a CTMC model by a state space $\mathcal{S}$ and transition rates q_{ij}, $i, j \in \mathcal{S}$ where

$$q_{ij} = q_i \cdot p_{ij} \quad \forall i, j \in \mathcal{S}.$$

[3] The geometric distribution is the discrete counterpart of the continuous exponential distribution and it also has the memoryless property, expressed for discrete values of time.

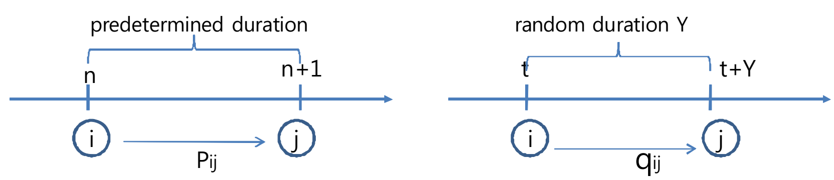


Figure 2.7: Discrete Time Markov Chain vs. Continuous Time Markov Chain: In the CTMC, the change of state can happen any time while it is limited in the DTMC. In DTMC, transition probability p_{ij} from state i to j is used while transition rate q_{ij} from state i to j is used in the CTMC.

Then q_{ij} is the rate at which the process moves to state j when it is in state i. We call q_{ij} a *transition rate* from i to j. As the rate q_i is equal to $\sum_{j \neq i} q_{ij}$ and p_{ij} is the same as $\frac{q_{ij}}{q_i}$, specifying $(\mathcal{S}, q_{ij})$ is equivalent to $(\mathcal{S}, q_i, p_{ij})$.

The memoryless property is very important in establishing the Markov property of the CTMC. Because the Markov chain spends an exponentially distributed random time in each state, the remaining lifetime is independent of the age. Accordingly, as long as the process is in state i at time t, the remaining lifetime distribution remains the same, independently of the past of its evolution.

2.6.2 CTMC MODEL: RATE MATRIX

In Section 2.2, we learned that the a DTMC model is represented by a state space and a transition probability matrix $(\mathcal{S}, \mathbf{P})$. Since intervals between state transitions are predetermined in the DTMC, the transition probability determines to which state the chain moves. In Figure 2.7, the times n and $n+1$ are when a state transition can happen. With probability p_{ij} the DTMC moves from state i to state j.

A CTMC model is represented by a state space $\mathcal{S}$ and a transition rate matrix $\mathbf{Q}$. The matrix $\mathbf{Q}$ represents the transition rules of the CTMC in the same was as the transition probability matrix $\mathbf{P}$ defines the transition rules in a DTMC. The difference between the two models is that each component of $\mathbf{Q}$ is a transition rate while those of $\mathbf{P}$ are transition probabilities. We explain the CTMC model with the following example.

- Consider a radio channel which can be either in a good state or bad state as in Figure 2.8[4]. In this case, the state space is

$$\mathcal{S} = \{\text{Good}, \text{Bad}\} .$$

[4]We simplified the channel by assuming that it has only two possible states.

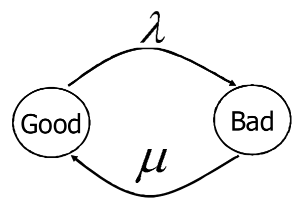


Figure 2.8: CTMC Example: Two-state channel model.

- Assume that the transition rate q_{gb} from 'Good' to 'Bad' state $= \lambda$ and $q_{bg} = \mu$. The transition rate matrix is

$$\mathbf{Q} = \begin{bmatrix} q_{gg} & q_{gb} \\ q_{bg} & q_{bb} \end{bmatrix} = \begin{bmatrix} -\lambda & \lambda \\ \mu & -\mu \end{bmatrix}. \tag{2.18}$$

  The diagonal entries q_{gg} and q_{bb} are determined so that the sum of the elements of each row is zero.

- In general, a CTMC model is represented by $(\mathcal{S}, \mathbf{Q})$ where $\mathcal{S} = \{1, 2, \ldots, m\}$ and

$$\mathbf{Q} = \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1j} & \cdots & q_{1m} \\ q_{21} & q_{22} & \cdots & q_{2j} & \cdots & q_{2m} \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ q_{i1} & q_{i2} & \cdots & q_{ij} & \cdots & q_{im} \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ q_{m1} & q_{m2} & \cdots & q_{mj} & \cdots & q_{mm} \end{bmatrix}. \tag{2.19}$$

  where $q_{ii} = -\sum_{j \neq i} q_{ij}$ and $0 \leq q_{ij} < \infty$ for all $i \neq j$. The diagonal entry q_{ii} does not provide any new information because it is a dependent term. It is selected to be the negative sum of all other entries in a row for the sake of computational convenience. As you shall see, the balance equation can be computed simply from $\pi Q = 0$, if we define $q_{ii} = -q_i = -\sum_{j \neq i} q_{ij}$. Here, we assumed that the number of states is m. The component q_{ij} represents the transition rate from i to j. Note that the sum of each row is zero.

2.6.3 BIRTH-AND-DEATH PROCESS

Engineers commonly use birth-and-death processes to model packet- or circuit-switched communication systems. In such applications, the state of the process typically represents the number of customers, packets, or busy channels in the system.

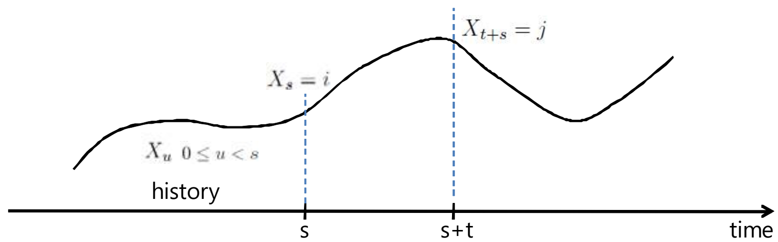


Figure 2.9: The Markov property says that the current state is sufficient to predict the future evolution. Knowing X_s is sufficient and the history of evolution $[X_u, u < s]$ does not help in predicting the future.

For a representative example, assume that there are k customers in some system. When a new arrival enters the system, the state changes to $k + 1$, whereas when a departure happens, the state becomes $k - 1(\geq 0)$. The arrival rate is λ_k and the departure rate is μ_k.

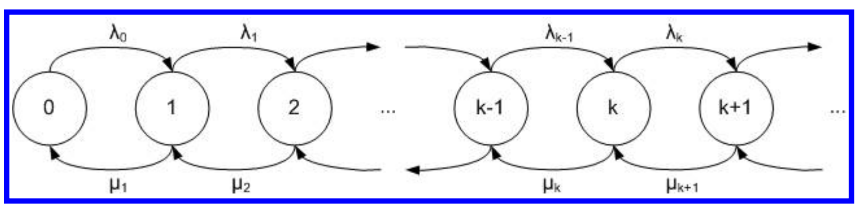


Figure 2.10: State Transition Diagram of a Birth-and-Death Process.

The state space of a birth-and-death process is

$$\mathcal{S} = \{0, 1, 2, \cdots, L\},$$

where L is the maximum number of customers allowed in the system. (L can be ∞.) The transition rates q_{kl} for $k \neq l$ are given by

$$q_{kl} = \begin{cases} \lambda_k, & l = k + 1, 0 \leq k < L; \\ \mu_k, & l = k - 1, k \geq 1; \\ 0, & \text{otherwise}. \end{cases} \tag{2.20}$$

The transition rate matrix $\mathbf{Q}$ for $L = 4$ is given by:

$$\mathbf{Q} = \begin{bmatrix} -\lambda_0 & \lambda_0 & 0 & 0 & 0 \\ \mu_1 & -(\lambda_1 + \mu_1) & \lambda_1 & 0 & 0 \\ 0 & \mu_2 & -(\lambda_2 + \mu_2) & \lambda_2 & 0 \\ 0 & 0 & \mu_3 & -(\lambda_3 + \mu_3) & \lambda_3 \\ 0 & 0 & 0 & \mu_4 & -\mu_4 \end{bmatrix}. \tag{2.21}$$

A birth-and-death process is a *pure death process* if $\lambda_k = 0$, for all $k \geq 0$. It is a *pure birth process* if $\mu_k = 0$ for all $k \geq 0$. A *Poisson process* is a pure birth process with $L = \infty$ and $\lambda_k = \lambda$ for all $k \geq 0$.

2.7 SOLVING CTMC $(\mathcal{S}, \mathbf{Q})$

We can define the transition probability function $p_{ij}(t)$ of a CTMC as follows:

$$p_{ij}(t) = P[X_{t+s} = j | X_s = i], \tag{2.22}$$

which denotes the probability that a process will be in state j after t units of time, given that it is presently in state i.

We extend the definition of the limiting probability in a DTMC to the case of the CTMC, assuming the existence of the limiting probability as follows:

$$\pi_j = \lim_{t \to \infty} p_{ij}(t). \tag{2.23}$$

One can show that a finite CTMC has a unique steady state distribution if it is irreducible, which means that the process X_t can go from every state to every other state. If it is infinite and irreducible, then the CMTC has at most one steady state distribution.

Getting a steady state probability π from a CTMC $(\mathcal{S}, \mathbf{Q})$ is called 'solving the CTMC.' We can solve a CTMC $(\mathcal{S}, \mathbf{Q})$ using two different approaches:

- Solving balance equations of $(\mathcal{S}, \mathbf{Q})$;
- Converting the CTMC $(\mathcal{S}, \mathbf{Q})$ to a DTMC $(\mathcal{S}, \mathbf{P})$ and then applying methods in Section 2.4.

2.7.1 SOLVING BALANCE EQUATIONS OF $(\mathcal{S}, \mathbf{Q})$

The balance equations of CTMC $(\mathcal{S}, \mathbf{Q})$ are

$$\pi \mathbf{Q} = 0, \tag{2.24}$$

where π is a steady state distribution vector. So π satisfies

$$\sum_j \pi_j = 1. \tag{2.25}$$

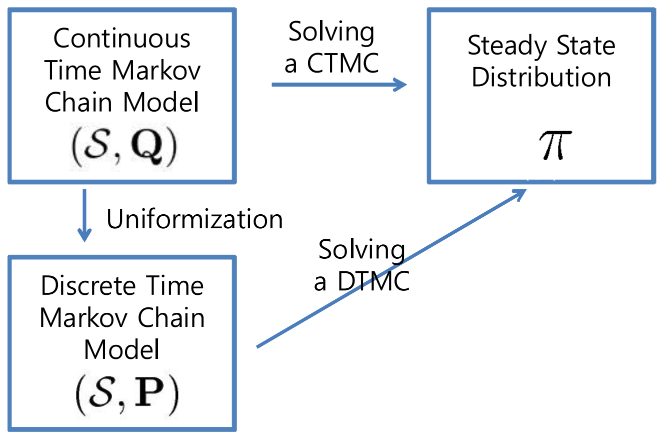


Figure 2.11: Solving a Markov Chain.

- In the channel example, the balance equations are written as follows:

$$\pi \mathbf{Q} = (\pi_g, \pi_b) \begin{bmatrix} -\lambda & \lambda \\ \mu & -\mu \end{bmatrix} = 0, \quad \pi_g + \pi_b = 1. \tag{2.26}$$

These equations can be rewritten as follows:

$$\pi_g \lambda = \pi_b \mu, \quad \pi_g + \pi_b = 1. \tag{2.27}$$

Solving the equations gives $(\pi_g, \pi_b) = (\frac{\mu}{\lambda+\mu}, \frac{\lambda}{\lambda+\mu})$.

- The balance equations of the birth-and-death process can be written as follows:.

$$\begin{aligned} -\pi_0(\lambda_0) + \pi_1\mu_1 &= 0 \\ \pi_{k-1}\lambda_{k-1} - \pi_k(\lambda_k + \mu_k) + \pi_{k+1}\mu_k &= 0, \quad k \geq 1. \end{aligned}$$

You can verify that the solution of these equations is

$$\pi_k = \pi_0 \Pi_{i=1}^{k} \frac{\lambda_{i-1}}{\mu_i}, \; 1 \leq k \leq L,$$

where π_0 is such that these probabilities add up to one. If $L = \infty$, such a value π_0 may not exist, in which case the Markov chain does not admit a steady state distribution. (Intuitively, the population grows too fast.)

- The i-th balance equation of a CTMC can be written as:

$$q_i \pi_i = \sum_{j \neq i} q_{ji} \pi_j$$

where $q_i = \sum_{j \neq i} q_{ij}$, The left-hand side is the rate at which the process leaves state i and the right-hand side is the rate at which the process enters state i. The naming of the "balance equations" comes from the fact that the two rates are equal or balanced. In the channel example, the balance equation is $\pi_g \lambda = \pi_b \mu$. The first term corresponds to the rate at which the process leaves the 'good' state while the second term corresponds to the rate at which the process enters the 'good' state.

2.7.2 UNIFORMIZATION: $(\mathcal{S}, \mathbf{Q}) \rightarrow (\mathcal{S}, \mathbf{P})$

In general, it is easier to work with a discrete time model than with a continuous time one. Uniformization is a technique to convert a CTMC into a corresponding DTMC. The uniformization procedure is as follows:

1. Pick a rate γ that it is larger than the maximum of $-q_{ii}$ over i, i.e.,

$$\gamma \geq \max_i \{-q_{ii}\} .$$

Recall that $-q_{ii}$ is the rate of transitions out of state i.

2. Generate the transition probability matrix $\mathbf{P}$ defined as follows:

$$p_{ij} = \begin{cases} \frac{q_{ij}}{\gamma}, & \text{if } i \neq j; \\ 1 + \frac{q_{ii}}{\gamma}, & \text{otherwise.} \end{cases} \tag{2.28}$$

It is not difficult to see that $p_{ij} \in [0, 1]$ for all i, j and $\sum_j p_{ij} = 1$ for all i.

3. Note that $P = I + \gamma^{-1} Q$ where I is the identity matrix. Consequently,

$$\pi P = \pi(I + \gamma^{-1} Q) = \pi + \gamma^{-1} \pi Q,$$

which shows that $\pi P = \pi$ if and only if $\pi Q = 0$. That is, the CMTC and the DTMC obtained by uniformization have the same steady state distribution.

- In the channel example,

$$\mathcal{S} = \{Good, Bad\}, \quad \mathbf{Q} = \begin{bmatrix} -\lambda & \lambda \\ \mu & -\mu \end{bmatrix}. \tag{2.29}$$

If we pick $\gamma = \lambda + \mu$, so that $\gamma \geq \max(\mu, \lambda)$, the corresponding DTMC is

$$\mathcal{S} = \{Good, Bad\}, \quad \mathbf{P} = \begin{bmatrix} \frac{\mu}{\lambda+\mu} & \frac{\lambda}{\lambda+\mu} \\ \frac{\mu}{\lambda+\mu} & \frac{\lambda}{\lambda+\mu} \end{bmatrix}. \tag{2.30}$$

If we solve $(\mathcal{S}, \mathbf{P})$, we have $(\pi_g, \pi_b) = (\frac{\mu}{\lambda+\mu}, \frac{\lambda}{\lambda+\mu})$, which is the same as the result of solving the CTMC. Since the row vectors are identical in $\mathbf{P}$, row vector itself is the steady state distribution.

2.8 SUMMARY

- A *Markov Chain* is used to model a dynamical system that changes its state over time. Markov chains are classified into continuous time Markov chains (CTMC) and discrete time Markov chains (DTMC). The state of a DTMC is allowed to change only at discrete instants while that of a CTMC can change at any time. The state evolution of a Markov chain depends only on the current state, not on the past states. This is the *Markov property*.

- Because of the Markov property, a *DTMC* is specified by $(\mathcal{S}, \mathbf{P})$, where $\mathcal{S}$ is the state space and $\mathbf{P}$ is the transition probability matrix of the DTMC. A *state transition diagram* helps visualize a Markov chain model. In such a diagram, states are represented by circles and arrows between circles correspond to transitions between states; arrows are usually marked with the corresponding transition probability.

- Steady state distribution π of DTMC $(\mathcal{S}, \mathbf{P})$ can be computed as

  1. any row vector of the limit $\Pi := \lim_{n\to\infty} \mathbf{P}^n$; or
  2. a solution to a balance equations $\pi = \pi\mathbf{P}$ and $\sum_{s\in\mathcal{S}} \pi_s = 1$.

- The steady state distribution π is used to calculate performance metrics of interest, such as average delay, average backlog, and blocking probability.

- A CTMC is specified by $(\mathcal{S}, \mathbf{Q})$, where $\mathcal{S}$ is the state space and $\mathbf{Q}$ is the transition rate matrix of the DTMC. As the state transition from i to j in a CTMC is exponentially distributed, element q_{ij} of matrix $\mathbf{Q}$ is the transition rate of an exponential random variable. Due to memoryless property of the exponential random variables, the Markov property holds for a CTMC.

- A birth-and-death process is one of the most popular CTMC models in performance modeling. Because it has a closed form steady state solution, it is widely used for many applications.

- The steady state distribution π of CTMC $(\mathcal{S}, \mathbf{Q})$ can be computed as

  1. a solution to a balance equation $0 = \pi\mathbf{Q}$ and $\sum_{s\in\mathcal{S}} \pi_s = 1$; or
  2. by transforming CTMC $(\mathcal{S}, \mathbf{Q})$ to DTMC $(\mathcal{S}, \mathbf{P})$ with a technique called *uniformization*. Once we have a DTMC, we can use the techniques for DTMC to find π.

CHAPTER 3

Developing Markov Chain Performance Models

One purpose of this chapter is to illustrate the process of modeling simple systems when they are analyzed using Markov chains. We use five examples: a simple forwarding system, a cellular network with blocking, slotted ALOHA, a Wi-Fi network, and a multi-channel MAC network. The first example uses a discrete time model to evaluate the buffer occupancy of the forwarding system. The second example is the popular Erlang model to calculate blocking probabilities. The slotted ALOHA model is introduced next using a DTMC. The last two examples are throughput models of Wi-Fi based local area networks.

3.1 PERFORMANCE MODELING STEPS

The development of the Markov chain model follows the following modeling steps: 1) understanding the system, (2) model construction, and (3) verification and validation as shown in Figure 3.1. The feedback loop indicates that it is a repetitive process.

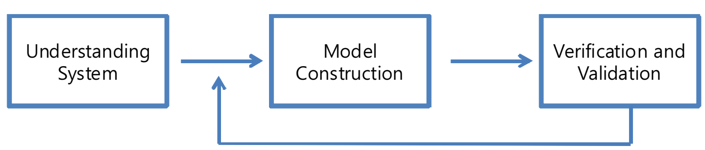


Figure 3.1: Modeling steps.

Understanding the System. The first and most important step in any modeling effort is to have a proper understanding of the system. Without it, the model could be misleading. Furthermore, it is important for a good modeler to have a sufficient understanding of the modeling objectives. In many cases, a team approach is appropriate. For example, a modeling expert would team up with a system expert to execute the modeling project.

In the development of performance models, it is necessary to identify the potential bottleneck components of the system. These are the components that impact the overall system performance more than others. Oftentimes, capturing the potential impact of bottleneck components is the key to the success of the model. However, the components that are bottleneck may not be obvious in the beginning of the investigation and modelers typically revise and improve the model after the validation step. To reduce the cost and risk of model development, many modelers start with a very simple high-level model. They use that model to confirm that the modeling direction is appropriate. Once the simple model provides the initial understanding, the modelers revise it by adding details that are important to the performance of the system.

Model Construction. After understanding the system, a modeler selects an appropriate model type. Since there are many model types, the models chooses one based on the nature of problem, taking into account the time and budget resources. For example, simulation models take a longer time and more resources in general. Analytical modeling takes less effort but requires more assumptions and approximations. In the discussion below, we suppose that the modeler selected a Markov chain model.

To develop a Markov chain model, the following questions need to be answered:

- What is the state of the system?
- How does the state change and what are the transition rules?
- Do the state have the Markov property?

As we discussed, a DTMC is defined by a state space $\mathcal{S}$ and a transition probability matrix $\mathbf{P}$. Similarly, a CTMC is defined by a state space $\mathcal{S}$ and a transition rate matrix $\mathbf{R}$.

Choosing a state space and its transition rules for a given system is an art because it requires identifying its key characteristics. The number of states can be as small as two or it can be many. For example, the weather system of Chapter 2 has two states. However, if a modeler wants the model to be more accurate, the number of states can be much larger. The modeler also needs to decide whether to use a CTMC or a DTMC. Another important question is to identify the timescale of the state changes. The WLAN examples in the next chapter reveals many different time scales for the same WLAN model.

The Markov chain model can be used once the Markovian property is justified. Otherwise, the model may produce incorrect results. One should understand that the possibility of misleading results due to incorrectly assuming the Markov property.

Verification and Validation. After the model development, we validate the model against real data or other types of model. Most commonly used is the cross validation of an analytical model against a simulation model or real field data. The validation step is important and it makes the model more precise. When we find noticeable differences, we revise the model to make it more accurate. The modeler must think carefully to understand the gap. The thought process is very helpful in revising the developed model.

The feedback arrows in Figure 3.1 shows the revision process.

3.2 A SIMPLE FORWARDING SYSTEM

System Description Consider a campus network that has a group of local area networks (LANs) and one access router as a gateway to the Internet. As students start complaining of slow network connections, the network manager decides to investigate the performance issue and finds that the access router is a potential bottleneck. To address the problem, the network manager is not sure whether to replace or upgrade the router. If the decision is to purchase a new router, the manager must determine its characteristics. To study these questions, the manager decides to develop a performance model to assess the different options.

Figure 3.2: A Switch that forwards packets uplink.

Modeling The manager asks you to help model the network. As an expert, you decide to simplify the model by focusing on the key aspects. Since the LANs are not bottleneck, you focus on the router as shown in Figure 3.2. You ignore the complexities of the LANs but model the total traffic they generate for the router. You study historical data and identify the input traffic patterns such as the number of packets. You also study the router CPU and memory.

You know that you can study this system either though computer simulation or with a mathematical model. You opt for a mathematical model as the system is not excessively complex and a Markov chain model can provide a detailed understanding of the router performance. The next question is whether to use a DTMC or CTMC. Both seem to be good candidates, and you decide to use a discrete-time model.

States In the DTMC, you need to define the state X_n at time n. What should the state be what should the time epoch n mean? As you are interested in the performance of the access router, to observe the behavior of buffer occupancy and CPU, you let X_n be the number of packets in the system at time n, then

$$X_n \in \mathcal{S} := \{0, 1, 2, 3, \cdots, L-1, L\}. \tag{3.1}$$

Here, n is the time slot long enough to transmit one packet and L is the size of the buffer, in number of packets.

After studying the historical data, you find out that packets arrive at the router with probability α in time slot n, independently of anything else. The packet that arrives at time slot n is available to be forwarded in the next time slot $n+1$. The CPU of the router is involved in many different tasks and allocates only a fraction β of its cycles to the forwarding task. As a consequence, the router is able to forward a packet with probability β in a given slot. With probability of $(1-\beta)$, the CPU is performing a different task. You further simplify the system by assuming that the arrivals and departures are independent.

Transition Probabilities With this understanding of the arrivals and of the router behavior, you can specify the transition probabilities. Note that state transitions happen when a packet arrives or departs. When a packet arrives at time n, the buffer occupancy X_n either increases or stays the same depending on whether departure happens or not. If there is a departure in the same slot, then $X_{n+1} = X_n$; otherwise, $X_{n+1} = X_n + 1$. Similarly, if there is no arrival in time slot n, X_n either stays the same or decreases by one. So, the transition probability matrix $\mathbf{P}$ is as follows:

$$p_{ij} = \begin{cases} p_1 = \alpha(1-\beta), & j = i+1, i = 1, 2, \cdots; \\ p_2 = (1-\alpha)\beta, & j = i-1, i = 1, 2, \cdots; \\ p_3 = \alpha\beta + (1-\alpha)(1-\beta), & j = i, i = 1, 2, \cdots; \\ \alpha, & i = 0, j = 1; \\ (1-\alpha), & i = 0, j = 0; \\ 0, & \text{otherwise} \end{cases} \tag{3.2}$$

The transition probability $p_{0,1}$ from state 0 to 1 is α because there can be no departure when the router buffer is empty.

Figure 3.3 shows the state transition diagram of the router buffer occupancy.

Solution of the model First, note that the Markov chain model that we have is a discrete time birth-and-death process. The balance equations of the process are:

$$\begin{aligned} \pi_0 \cdot \alpha &= \pi_1 \cdot p_2, & (3.3)\\ \pi_1 &= \pi_0 \cdot \alpha + \pi_1 \cdot p_3 + \pi_2 \cdot p_2, & (3.4)\\ \pi_n &= \pi_{n-1} \cdot p_1 + \pi_n \cdot p_3 + \pi_{n+1} \cdot p_2 \text{ for } n \geq 2. & (3.5) \end{aligned}$$

Summing up the first $n+1$ balance equations, we find that

$$\pi_n \cdot p_1 = \pi_{n+1} \cdot p_2 \text{ for } n \geq 1.$$

These equations are called the *local balance equations.*

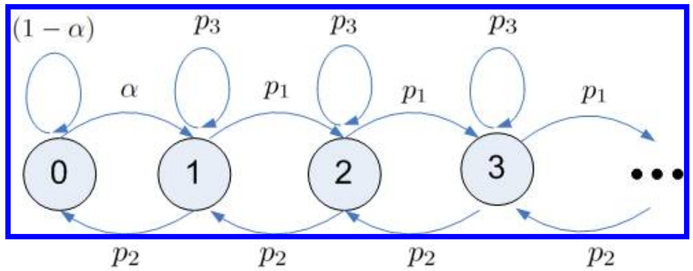


Figure 3.3: A Discrete Time Markov Chain Model of Router Buffer Occupancy.

From these equations and $\sum \pi_n = 1$, we find that the steady state distribution π_n is given by:

$$\pi_n = \frac{\alpha}{p_2}\left(\frac{p_1}{p_2}\right)^{n-1}\pi_0 \tag{3.6}$$

$$\pi_0 = \left(1 + \frac{\alpha}{p_2}\frac{p_2}{p_2 - p_1}\right)^{-1}. \tag{3.7}$$

This invariant distribution is similar to that of the continuous time birth-and-death process that we discussed earlier.

From the steady state distribution, the average backlog $E[X]$ can be computed as:

$$E[X] = \sum_{n}^{\infty} n \cdot \pi_n.$$

The average backlog is plotted in Figure 3.4 for different values of α and β. The x-axis correspond to the value of α and the y-axis corresponds to the average backlog. The three lines from left to right corresponds to different forwarding probabilities $\beta = 0.3$, 0.6, and 0.9, respectively. Note that as the arrival probability approaches β, the average buffer occupancy increases rapidly. We can calculate the average delay D from the average backlog L using Little's result that states that $L = \alpha D$.

Validation You now have your first DTMC model of the campus access router represented by (3.1) and (3.2) and its analytical solution can be used to compute metrics of interest. To determine whether the model is accurate enough or requires further modification, the validation work can take much more time and effort than the model development.

You can validate the model against benchmark cases. For example, you can measure the average delay through the router and the arrival rate of packets. By comparing these numbers with the prediction of the model, you can estimate the value of β. You can then repeat the measurements

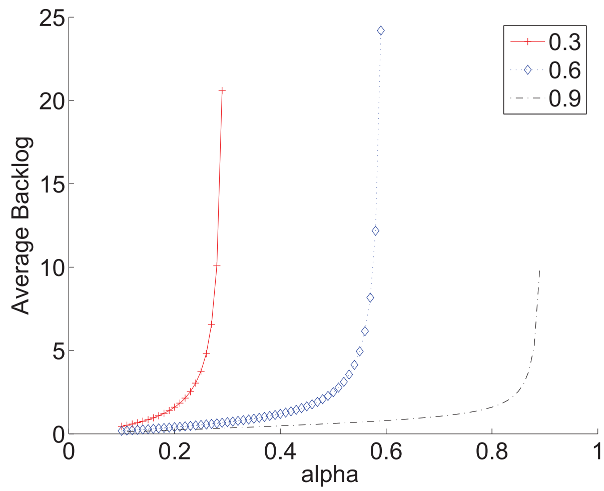


Figure 3.4: The number of backlogs vs. probability α of arrivals for different values of $\beta = \{0.3, 0.6, 0.9\}$.

at different times of the day that correspond to different average arrival rates and delays. If the model predictions are accurate enough, you can trust the model. Otherwise, you may need to revise it.

3.3 CELLULAR SYSTEM WITH BLOCKING

Our second example is a cellular communication system and we develop a CTMC model to understand the blocking probability, one of the most important system metrics in a cellular network.

System Description Assume that you are a network designer for a cellular network company. As there have been numerous complaints from customers on the call blocking, your boss asks you to initiate performance studies for the problematic area.

The network blocks calls because the number of channels is finite. Consider a cellular system that consists of single base station that serves incoming and outgoing phone calls. There are a limited number of wireless channels for these phone calls, and if all the channels are being used when a call is placed, the user receives a busy signal and must try to make a connection at later time. Once a connection is made, the channel is unavailable to other users until the current user terminates the connection by hanging up. Figure 3.5 shows the cellular system with two on-going phone calls. The number of channels is six in the figure.

The company has a policy of controlling the blocking probability below 1%. However, initial investigation of the system reveals that the customers in the areas have experienced more frequent call blocking than this threshold. It turns out that the population in the area has increased constantly as new developments were built.

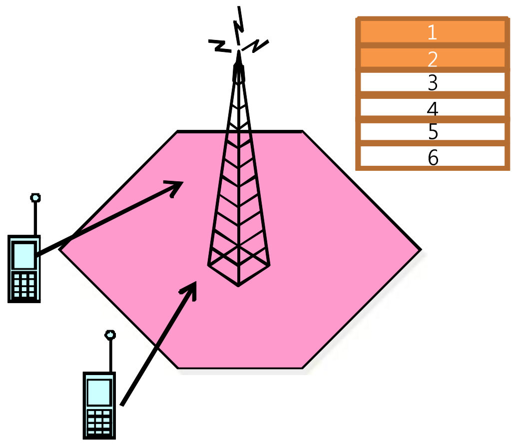


Figure 3.5: A Cellular System with Blocking: Two channels out of six are busy and the others are idle.

As an experienced engineer, you decide to reduce the size of cells in the area to lower the blocking probability. With a smaller cell size, the blocking probability goes down as the rate of incoming calls is reduced. But you are not sure how to select the reduced cell radius? To answer the question, you decide to develop an analytical model.

Modeling The important factors that affect the blocking probability are the call arrival rate, the average call duration, and the number of available channels. We can collect information on these factors from historical data.

The study can be done by simulation or by an analytical model. In order to develop an analytical model, many simplifying assumptions are needed.

Assumptions We make the following assumptions:

1. The duration of a phone conversation is exponentially distributed with mean $1/\mu$.

2. The time between phone calls to the system is distributed exponentially with mean $1/\lambda$, even when all the channels are busy.

The above assumptions seem very restrictive, but they are essential to develop a CTMC model. The mean value of the call durations can be obtained from historical data and we further assume that the call durations are exponentially distributed to make a CTMC model. The second assumption is another way of saying that call arrivals form a Poisson process with rate λ. When the number of potential users is large, the Poisson assumption is quite accurate.

States As discussed in the previous chapter, a CTMC model is determined by $(\mathcal{S}, \mathbf{Q})$. In the cellular system model, the state space is

$$\mathbf{S} = \{0, 1, 2, \cdots, K-1, K\}, \tag{3.8}$$

where K is the number of available channels. The state $X(t)$ represents the number of busy channels at time t and $X(t) \in \mathcal{S}$.

Transition Rates The state changes when a new call arrives and when a call terminates, which corresponds to a birth-and-death process. The arrival rate λ_i when there are i channels are busy is

$$\lambda_i = \lambda \;\; i \in 0, 1, \cdots, K-1. \tag{3.9}$$

The death rate μ_i when the number of busy channels is i is:

$$\mu_i = i\mu \;\; i \in 1, \cdots, K. \tag{3.10}$$

This is because each conversation terminates at rate μ, independently of the other conversations. Accordingly, the first conversation terminates at rate $i\mu$.

Figure 3.6 shows the state transition diagram of the Markov model.

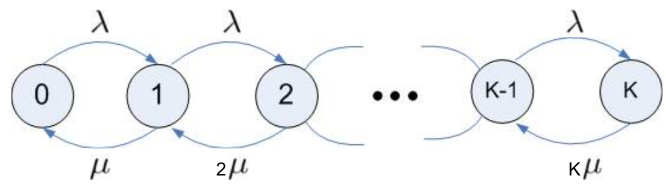


Figure 3.6: A Markov Chain Model of the Cellular System with K channels.

Solutions The balance equations $\pi\mathbf{Q} = 0$, of the CTMC model are as follows:

$$\begin{aligned} \pi_0 \cdot \lambda &= \pi_1 \cdot \mu, & (3.11)\\ \pi_i(\lambda + i\mu) &= \pi_{i-1} \cdot \lambda + \pi_{i+1} \cdot (i+1)\mu \text{ for } i = 1, 2, \cdots, K-1; & (3.12)\\ \pi_K \cdot K\mu &= \pi_{K-1} \cdot \lambda. & (3.13) \end{aligned}$$

Summing the first $(i+1)$ equations gives

$$\pi_i\lambda = \pi_{i+1} \cdot (i+1)\mu \text{ for } i = 1, 2, \cdots, K-1.$$

From the identity and $\sum_i \pi_i = 1$, we have the following closed-form solution for steady state distribution:

$$\pi_0 = \left[\sum_{k=0}^{K} \frac{\rho^i}{i!}\right]^{-1} \tag{3.14}$$

$$\pi_i = \pi_0 \frac{\rho^i}{i!}, \quad 1 \le i \le K, \tag{3.15}$$

where $\rho = \frac{\lambda}{\mu}$. The value of π_0 is obtained from $\sum_i \pi_i = 1$.

The blocking probability of the cellular system is π_K. Indeed, the arrival rate of calls does not depend on the state of the system. Consequently, the probability that a call arrives when all the channels are busy is the probability π_K that all channels are busy.

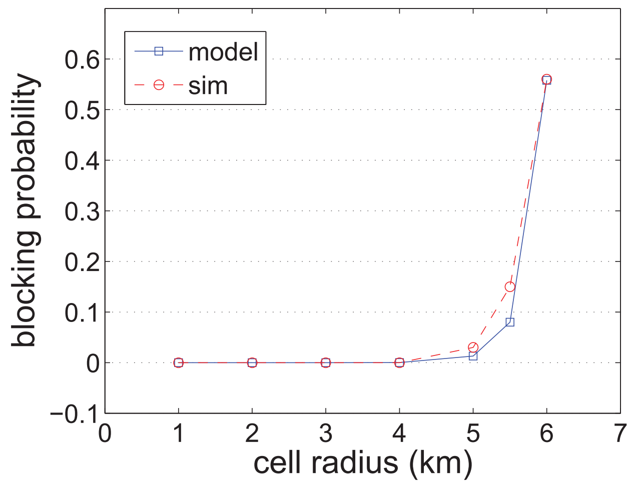


Figure 3.7: Blocking Probability with different cell radius.

Figure 3.7 shows the blocking probability π_K of the cellular system when the number K of channels is 20. We assume that the arrival rate per square kilometer is two calls per hour and that arrival rate $\lambda = 3.14 \times r^2 \times 2$ calls. The X-axis corresponds to the cell radius and the Y-axis to the blocking probability. The solid line with square tick are from the Erlang B formula (3.15) and the dotted line with circle tick is from the simulation. We can observe that the blocking probability is almost zero when the cell radius is smaller than $4Km$ and that it reaches 9% as the cell radius becomes $5.5Km$. We can conclude that the size of the cell is smaller than $5Km$ to satisfy the blocking probability threshold from the model.

Validation and Verification of the Model. The analytical models we developed seem restrictive with many assumptions such as exponential duration of phone calls and the Poisson arrival of calls. Because of these assumptions, validation and verification processes are needed. As an alternative to the mathematical model, oftentimes simulation models are developed to cross-validate the analytical model. An advantage of the simulation model is that we do not need to make the assumptions that we made in the Markov model. However, it is more complex to develop the simulation model.

The dotted line in Figure 3.7 shows the results from the simulation model. The results are very close except when the radius is $5.5Km$. The blocking probability 14% from the simulation is a bit higher than 9% from the Markov model. From the cross-check, we can assure the correctness of the model. When the simulation model does not exist, practitioners can rely on field data, and expert opinions for the validation efforts.

3.4 SLOTTED ALOHA

In this section, we describe the slotted ALOHA system as the third example of Markov chain modeling.

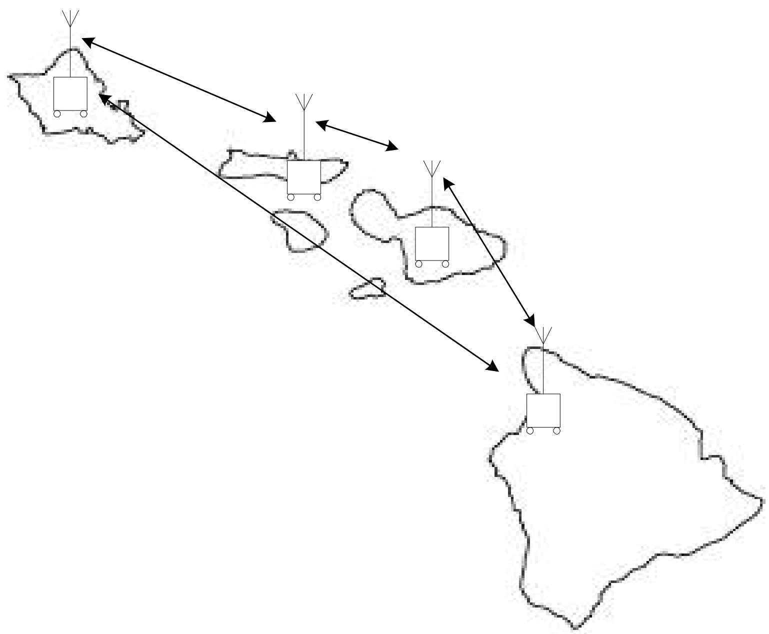


Figure 3.8: The ALOHA network.

ALOHA was a pioneering wireless networking system developed at the University of Hawaii in the early 1970s. To enable communication between separated campuses of the University of Hawaii, Professor Abramson developed the packet switched network with the first multiple access scheme as shown in Figure 3.8. As multiple transmitters share a communication channel, if two or more nodes transmit simultaneously, the transmissions fail. If no node transmits, the channel is unused. The challenge in the multiple access protocol is to coordinate multiple transmitters to achieve efficiency and fairness. Refer to (3; 27) for thorough coverage of the multiple access protocols and issues.

System Description There are two versions of the ALOHA protocol: pure and slotted. In the pure ALOHA protocol, each transmitter can send packet at any time, while packet transmission is restricted to the beginning of time slots in the slotted ALOHA protocol. The slotted protocol is more synchronized than the pure one and it is known that this restriction improves the throughput by a factor of two over the pure ALOHA protocol by reducing the number of collisions.

The multiple access protocol of the slotted ALOHA protocol works as follows. When a packet arrives at an unbacklogged node, the node simply transmits the packet in the first slot after the arrival, thus risking occasional collisions but achieving very small delays if collisions are rare. When a collision occurs, each node sending one of the colliding packets discovers the collision when it does not receive an acknowledgment for its packet and it becomes backlogged. The backlogged node waits for a random number of time slots and retransmits the packet. If the backlogged nodes were to retry transmission in the next slot, another collision would be inevitable. A random delay is included in the protocol to avoid repetitive collisions.

To understand the performance of slotted ALOHA, including the average delay or the maximum sustainable throughput, we develop a Markov chain model.

Modeling

As time is a sequence of slots in slotted ALOHA, it is natural to consider a discrete time model. We model the system with a DTMC model, which was proposed in (3).

Assumptions We make the following assumptions:

- There are N nodes in the system; let n be the number of backlogged nodes at the beginning of a given slot.
- Each backlogged node transmits a packet with probability of p, independently of other nodes. Each one of the $N - n$ unbacklogged nodes transmits a packet that arrived in the previous slot.
- Packets arrive to each node according to a Poisson process with rate λ/N. Since the number of such arrivals in one time unit is Poisson distributed, no packet arrives with probability $e^{-\lambda/N}$; thus the probability that the unbacklogged node transmits in a given slot is
$$q := 1 - e^{-\lambda/N}.$$

State Space. Let X_t be the number of backlogged nodes in the system at the beginning of time slot t. Then the state space $\mathcal{S}$ of the system is

$$\mathcal{S} = \{0, 1, 2, \cdots, N\},$$

where N is the number of nodes in the system. The possible events that change the number of backlogged nodes are: (1) packet arrivals to unbacklogged nodes, (2) successful transmission, and (3) failure of transmission. The number of backlogged nodes increases by the number of new arrivals to unbacklogged nodes. It decreases by one if a successful packet transmission happens in the time slot. A successful transmission can happen (1) if one arrival from unbacklogged nodes and no transmission attempts from the backlogged nodes or (2) if no arrivals to the backlogged nodes and one attempts from the backlogged nodes. Thus, we have the following transition probabilities.

Transition Probabilities. Let $r_u(i,n)$ be the probability that i unbacklogged nodes transmit packets in a given slot. Similarly, let $r_b(i,n)$ be the probability that i backlogged nodes transmit packets in a given slot. Then,

$$\begin{aligned} r_u(i,n) &= \binom{N-n}{i} q^i (1-q)^{N-n-i} && (3.16) \\ r_b(i,n) &= \binom{n}{i} p^i (1-p)^{n-i}. && (3.17) \end{aligned}$$

With these notations, the transition probability can be written by:

$$P_{n,n+i} = \begin{cases} r_u(0,n)r_b(1,n), & i=-1; \\ r_u(1,n)r_b(0,n) + r_u(0,n)[1-r_b(1,n)], & i=0; \\ r_u(i,n)[1-r_b(0,n)], & i=1; \\ r_u(i,n), & 2 \le i \le N-n. \end{cases} \tag{3.18}$$

Note that the state n goes to $n-1$ if only one of the backlogged nodes transmits a packet and no unbacklogged nodes transmit. Since both events are independent, the probability is $r_u(0,n)r_b(1,n)$. The state n remains the same either 1) when there is one attempt from the unbacklogged nodes and zero attempts from the backlogged nodes or (2) when there is zero attempts from the unbacklogged nodes and zero attempts or collision from the backlogged nodes. The state n goes to $n+1$ when one attempt from the unbacklogged nodes collides with attempts from the backlogged nodes. The state increases by $i (\ge 2)$ when there are i attempts from the unbacklogged nodes regardless of backlogged nodes.

Numerical Solution Note that $P_{ij} = 0$ for $j \le i+2$. The balance equations $\pi = \pi \mathbf{P}$ of the model are:

$$\begin{aligned} \pi_i &= \sum_{j=0}^{i+1} P_{ij}\pi_j, \text{ for } i = 0, 1, \cdots, N-1; && (3.19) \\ \pi_N &= \sum_{j=0}^{N} P_{Nj}\pi_j. && (3.20) \end{aligned}$$

Due to this structure, the steady state probabilities can be computed in an iterative way. Since π_1, π_2 $\cdots, \pi_N$ can be expressed in terms of π_0, the value of π_0 can be found numerically from $\sum_i \pi_i = 1$. Unlike the previous two examples, we do not have a closed form solution for this model, but it can be obtained numerically. With the steady state distribution, the average number of backlogged nodes can be found, from which we compute the average delay experienced by packets.

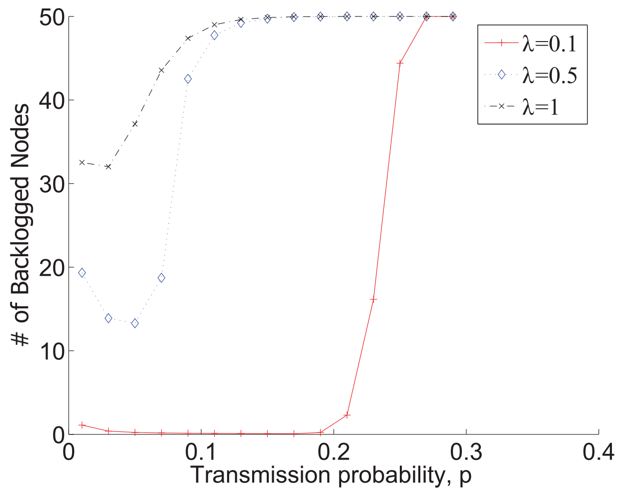


Figure 3.9: The number of backlogged nodes vs. the transmission probability p for different values of $\lambda = \{0.1, 0.2, 0.3\}$.

Figure 3.9 plots the average number of backlogged nodes for different values of p and λ when the number N of nodes is 50. When transmission probability is too small, the backlogged nodes are larger because the radio resource is rather idle. As p increases, the backlog reduces and when it is more than a certain value, i.e., 0.2 when $\lambda = 0.1$, it increases very rapidly. That is because the number of collision increases with a higher value of transmission probability p. The figure shows the importance of the transmission probability p to the performance of the ALOHA system.

As the transmission probability p is important to the stability of the ALOHA networks, people have tried to estimate an appropriate value of p. However, since the number of nodes n in the system is unknown to the nodes, proposed algorithms tried to estimate it. The essence of such algorithms is to increase p upon idle slots and decrease p upon collisions. For example, (10) proposed a recursive control algorithm of p that achieves stable operation of the slotted ALOHA networks.

Heuristic Analysis of slotted ALOHA It is known that the efficiency of the slotted ALOHA protocol is 36% under some idealized assumptions. This means that out of 100 time slots, only 36 slots can be used for data transmission and the remaining 64 slots are wasted. The waste comes from collisions, from postponing due to random delay and so on.

To understand the efficiency of 36%, consider the following throughput equation of the slotted ALOHA:

$$S = Ge^{-G}. \tag{3.21}$$

In this formula, S is the total throughput rate in packets per time slot and G is the total rate of transmission attempts in packets per time slot. Multiplying the total attempts rate G by the successful transmission probability e^{-G} gives the total throughput rate S in (3.21). To see why e^{-G} is the successful transmission rate, we argue as follows. When a large number of nodes attempt to transmit, each with a small probability, the transmission attempts form a Poisson process. A given node is successful if no other node attempts to transmit in a duration equal to 1. The probability of that event is the probability that the Poisson process with rate G does no jump in one unit of time. This is the probability that the time until the next transmission is larger than 1. Since that time is exponentially distributed with rate G, this probability is e^{-G}. Figure 3.10 shows the throughput formula Ge^{-G}. Note that the maximum throughput is e^{-1}, which is roughly 36%.

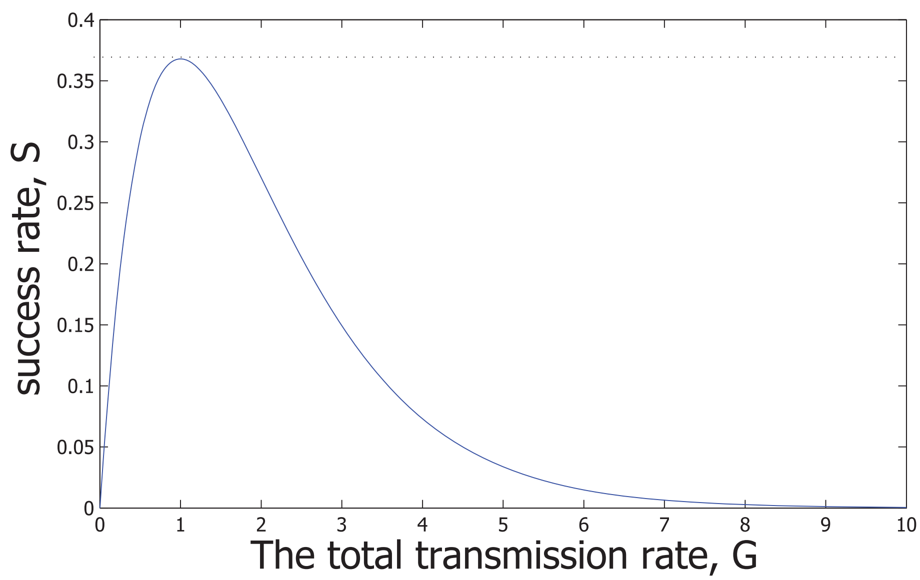


Figure 3.10: The throughput of the slotted ALOHA protocol.

3.5 WI-FI NETWORK - CSMA MARKOV CHAIN

This section introduces a CSMA model of Wi-Fi network to assess their system throughput. The model was originally introduced in (5) (see also (17)) and used in (13) to study distributed protocols for such networks.

System Description A group of Wi-Fi devices use the carrier sense multiple access with collision avoidance (CSMA/CA) protocol to share a radio channel. When using this protocol, before transmitting, each device checks whether the radio channel is busy or idle. If busy, the device waits until the channel becomes idle. If a collision happens, each device waits (backs off) for a random time. The device with the shortest random waiting time retransmits its packet. If two transmitters are close enough, they cannot transmit simultaneously under the CSMA/CA protocol because one senses that the radio channel is busy when the other transmits. We are interested in the average throughput of transmitters in the Wi-Fi network.

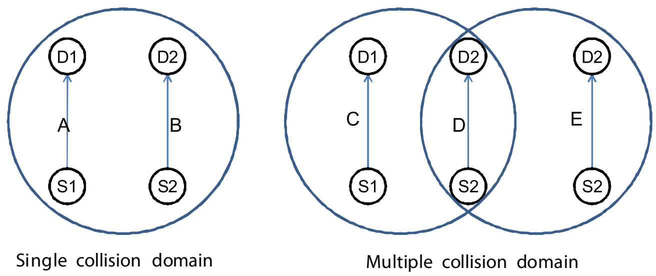


Figure 3.11: The Bianchi model can be applied only to the left case. In a single collision domain model, each link interferes with all other links. In a multiple collision domain model, since there are more than one collision domain, a link interferes with some of the links. In the left plot, link C interferes with link D, but not with link E.

Figure 3.11 shows two simple network examples. Links A and B cannot be used simultaneously as they are neighboring with each other. However, Links C and E can be used simultaneously.

Contention Graph. The CSMA Markov chain uses an interference model called a *contention graph*. The vertices of the contention graph are the links of the network and its edges connect the interfering links, i.e., links that cannot transmit simultaneously.

Figure 3.12 shows the contention graphs that correspond to the networks of Figure 3.11. For instance, the contention graph on the right of the figure shows that links C and D conflict and so do links D and E. The graph also indicates that links C and E do not conflict since they are not connected by an edge in that graph.

Figure 3.12: Contention Graphs: A node in a contention graph corresponds to a link in the network; a link in the contention graph represents contention relationship between two links of the network.

Modeling the CSMA Markov Chain

Assumptions:

- Each link has exponentially distributed waiting times;
- The transmission times are exponentially distributed with rate 1;
- Each node always has data to transmit;
- Conflicts between links are captured by a contention graph.

The exponential distribution in the first two assumptions is needed to model the link activities as a CTMC. The mean transmission time is set to one for simplicity. The third assumption indicates that we are interested in the saturated throughput when the sources are not a bottleneck. With these assumptions, we define a CTMC as follows.

State Space. The CSMA Markov chain represents the state of the links $\mathcal{L}$ in the network. Let $x_l(t)$ be the state of link l at time t, which can be either active (i.e., transmitting) or idle. When the link is idle, $x_l(t) = 0$; otherwise, $x_l(t) = 1$. Let $X(t) = [x_l(t), l \in \mathcal{L}]$ be the vector of states of all links in the network and $\mathcal{S}$ be the set of possible values of $X(t)$. Because of conflicts between links, $\mathcal{S}$ has fewer than 2^K elements.

Consider the network in the left of Figure 3.11. Since links A and B interfere with each other, the state (1, 1) is not possible. The state space for this example is

$$\mathcal{S} = \{(0, 0), (1, 0), (0, 1)\}.$$

Similarly, the state space for the network on the right of the figure is

$$\mathcal{S} = \{(0, 0, 0), (1, 0, 0), (0, 1, 0), (0, 0, 1), (1, 0, 1)\}.$$

The number of possible states in the CSMA Markov chain is the same as the number of *independent sets*. Two links A and B are said to be *independent* when they can be simultaneously active. In the conflict graph, an independent set is a set of vertices that are not connected by an edge.

Transition Probabilities. There are two types of transition in the CSMA Markov chain model: a start of transmission and the end of a transmission. When a transmission starts, one idle link becomes active and the state changes from x^i to $x^i + e_k$ with rate R_k. (Here, e_k is a K-dimensional vector whose k-th element is one and all others are zero.) This transition is possible if link k is not active in state x^i (i.e., $x^i_k = 0$) and if all the links that conflict with k are not active in x^i. When a transmission ends, the state $x^i + e_k$ becomes x^i with rate 1.

Note that the rate R_k is the rate of the exponential waiting time of link k and it represents the aggressiveness of that link. The higher the value of R_k, the more aggressive link k is in trying to capture the radio channel.

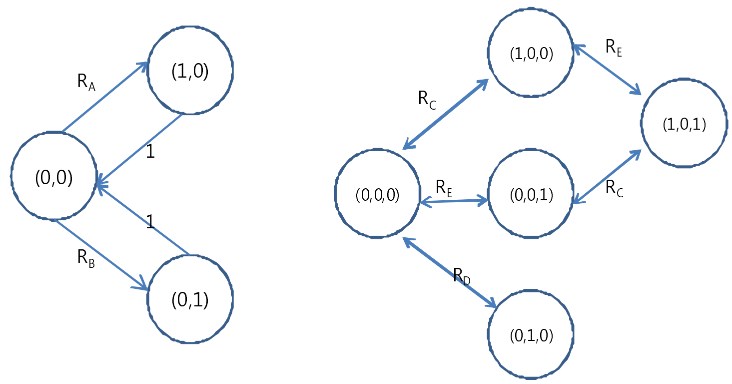


Figure 3.13: Example state transition diagrams for the networks of Figure 3.11.

This model focuses on the multiple collision domains represented by a general conflict graph. In contrast, the Bianchi model (4) focuses on the binary exponential backoff algorithm in a single collision model.

Figure 3.13 shows the state transition diagram of the CSMA Markov chain model for two networks shown in Figure 3.11. In the model on the left, there are only three states $\{(0,0), (1,0), (0,1)\}$, because two links are conflicting. State $(0,0)$ transitions to state $(1,0)$ with rate R_A and to $(0,1)$ with rate R_B. The model on the right corresponds to the three-link network of Figure 3.11. In this model, we do not show the transition rate 1 of transmission completions, for the sake of simplicity.

Solution. We can find the steady state probability π by solving the balance equations $\pi \mathbf{Q} = 0$. For the first example, the transition rate matrix Q is given by:

$$\mathbf{Q} = \begin{bmatrix} -(R_A + R_B) & R_A & R_B \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix}.$$

We find

$$\pi = \left(\frac{1}{1 + R_A + R_B}, \frac{R_A}{1 + R_A + R_B}, \frac{R_A}{1 + R_A + R_B} \right).$$

Similarly, the transition rate matrix for the second example is

$$\mathbf{Q} = \left[\begin{array}{c|ccccc} (0,0,0) & -(R_C + R_D + R_E) & R_E & R_D & R_C & 0 \\ (1,0,0) & 1 & -(R_C + 1) & 0 & 0 & R_C \\ (0,1,0) & 1 & 0 & -1 & 0 & 0 \\ (0,0,1) & 1 & 0 & 0 & -(R_E + 1) & R_E \\ (1,0,1) & 0 & 1 & 0 & 1 & -1 \end{array} \right].$$

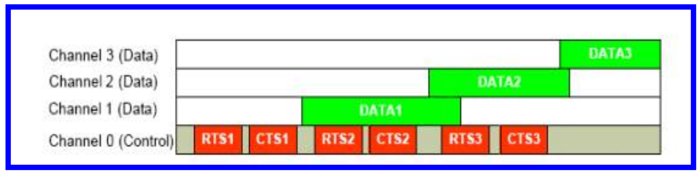


Figure 3.14: Operations of Dedicated Control Channel Multichannel MAC.

Solving the balance equations $\pi\mathbf{Q} = 0$ gives

$$\pi = \frac{(1, R_C, R_D, R_E, R_C \cdot R_E)}{1 + R_C + R_D + R_E + R_C \cdot R_E}.$$

We can see that the utilization of link is

$$(\rho_C, \rho_D, \rho_E) = \frac{(R_C + R_C \cdot R_E, R_D, R_E + R_C \cdot R_E)}{1 + R_C + R_D + R_E + R_C \cdot R_E}.$$

If $R_C = R_D = R_E = 5, (\rho_C, \rho_D, \rho_E) = (\frac{30}{41}, \frac{5}{41}, \frac{30}{41})$. Observe that the middle link D is much less utilized than the other two, which reveals the unfairness issue of the CSMA/CA protocol of Wi-Fi networks.

3.6 A MULTI-CHANNEL MAC PROTOCOL MODEL

There are three orthogonal channels in 802.11b/g and 12 in 802.11a. However, the standard 802.11a/b/g protocols use only one channel at a time and do not exploit the potential simultaneous transmissions on different channels. Multi-channel MAC protocols enable different devices to transmit in parallel on distinct channels. The parallelism increases the throughput and can potentially reduce the delay, provided that the channel access time is not excessive.

In this section, we present a multichannel MAC protocol model using a discrete time Markov chain (DTMC). The model was developed for a dedicated control channel MAC protocol, which is the most common approach in protocols that use multiple channels [1]. The model is focused on how the protocol enhanced throughput by utilizing parallelism and simplified the detailed RTS/CTS exchange.

Operations of a Dedicated Control Channel MAC In this dedicated control channel MAC, every device has two radios. One radio is tuned to a channel dedicated to control messages; the other radio

[1] Refer to (19) for other multichannel MAC protocols. They classified multichannel MAC protocols into four: dedicated control channel, split, common hopping, and multiple rendezvous approaches. They differ in how devices agree on the channel to be used for transmission and how they resolve potential contention for a channel.

can tune to any other channel. In principle, all devices can overhear all the agreements made by other devices, even during data exchange. This system's efficiency is limited only by the contention for the control channel and the number of available data channels. Fig. 3.14 illustrates the operations of Dedicated Control Channel MAC. Note that channel 0 is the control channel and that channels 1, 2, and 3 are for data transmission. When device A wants to send a packet to device B, it transmits an RTS (request-to- send) packet on the control channel. That RTS specifies the lowest-numbered free channel. Upon receiving the RTS, B responds with a CTS (clear-to-send) packet on the control channel, confirming the data channel suggested by A. The RTS and CTS packets also contain a Network Allocation Vector (NAV) field, as in 802.11, to inform other devices of the duration for which the sender, the receiver, and the chosen data channel are busy. Since all devices listen to the control channel at all times, they can keep track of the busy status of other devices and channels even during data exchange. Devices avoid busy channels when selecting a data channel.

The major advantage of Dedicated Control Channel Protocol is that it does not require time synchronization; rendezvous always happens on the same channel. The main disadvantage of this protocol is that it requires a separate dedicated control radio and a dedicated channel, thereby increasing cost and decreasing spectral efficiency when few channels are available.

The DTMC Multichannel MAC Model

The objective is to understand the capacity enhancement achieved by using a multiple channel MAC. We selected a DTMC model instead of a CTMC one because it is easier to handle. The first question is how to determine the duration of each period. In our model, we introduce the concept of slot, a short duration during which devices can exchange RTS/CTS to make an agreement. We assume that time is divided into small slots as shown in Figure 3.15, with perfect synchronization at slot boundaries. Each time slot is just long enough to exchange RTS/CTS frames to make an agreement.

State Space. Since the objective is to understand the impact of multichannel, we define a discrete time Markov Chain X_n to be the number of busy channels at time n. Then the state space for the model is

$$\mathbf{S} = \{0, 1, 2, \cdots, M\},$$

where M is the number of data channels of the system. The number M of data channels is two in the example of Figure 3.15. The evolution of X_n or the number of busy data channels is 0, 0, 0, 1, 2, 2, 2, 1, 1, 0, and 1 in the example.

Assumptions. To make the model fit into a DTMC, we make the following assumptions.

1. Upon making an agreement, the devices can transmit only one packet (one may think of a "packet" as the amount of data that can be transferred per channel agreement);

2. The packet lengths, which are integer multiples of slot durations, are independent and geometrically distributed with parameter q (i.e., packet duration has a mean of $1/q$ slots);

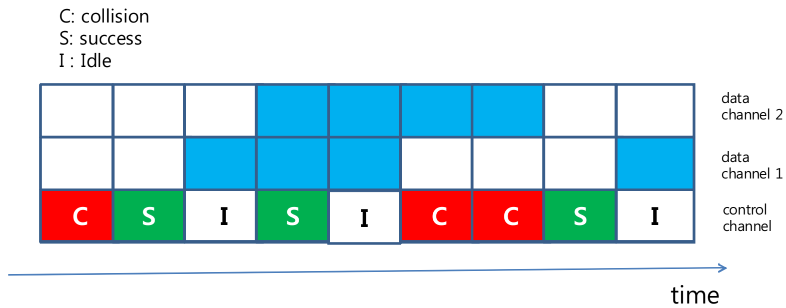


Figure 3.15: A DTMC Model illustration of a multichannel MAC.

3. Devices always have packets to send to all the other devices; in each time slot, an idle device attempts to transmit with probability p. The receiver of a sender is decided randomly with equal probability among possible candidates.

The second and third simplifications are essential to make the model Markovian. The second assumption, the geometric length of packet size, makes the termination events independent of the past of X_n because the geometric distribution is memoryless. Similarly, the third assumption makes the birth events Markovian because each device transmits with probability of p independently of the past evolution.

Transition Probabilities.

- The relationship between X_{N+1} and X_N can be written as follows:

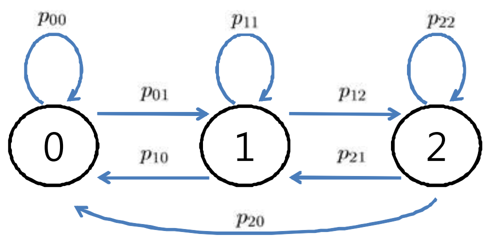
$$X_{n+1} = X_n + A_n - D_n, \quad n \geq 0, \tag{3.22}$$

where A_n is the number of new agreements at time n, and D_n is the number of terminations at time n. Note that $A_n = 1$ if a new agreement is made in time slot n and $A_n = 0$, otherwise. If $X_n = M$, which means that all channels are busy, then $A_n = 0$ with probability 1. The number of departures D_n at time n ranges from 0 to X_n. If $X_n = 0$, then $D_n = 0$ with probability 1.

- We assume the slotted ALOHA model [2] to model the exchange of RTS/CTS. When a device has a packet to transmit, it attempts to transmit a packet with probability p by sending an RTS. The agreement is made when only one device attempts to transmit an RTS. Hence, the success probability S_k in the next time slot, given that k pairs are communicating in the current slot, is given below:

$$S_k = (N - 2k)p(1 - p)^{(N-2k-1)}. \tag{3.23}$$

[2]The ALOHA model assumes the single collision domain as in the Bianchi model.



•

Figure 3.16: State Transition Diagram for the Multichannel Model.

Here, N is the number of devices in the network. Since k pairs are currently communicating, $N-2k$ devices are inactive. Also, $p(1-p)^{(N-2k-1)}$ is the probability that one specific device transmits an RTS with probability p in a given time slot while all other $N-2k-1$ devices do not. Since all $N-2k$ devices can try to transmit an RTS, we have the expression for S_k.

- The probability $T_k^{(j)}$ that j transfers finish when the system is in state k is given by the following expression:

$$\begin{aligned} T_k^{(j)} &= Pr[j \text{ transfers terminate at time } t | X_{t-1} = k] \\ &= \binom{k}{j} q^j (1-q)^{k-j}. \end{aligned} \tag{3.24}$$

Since one active device terminates transmission with probability p independently, the probability of j termination out of k active transmission is equal to $q^j(q-q)^{k-j}$. Since there are $\binom{k}{j}$ possible combinations, we have the probability shown in (3.24).

- The transition probability matrix for the example is as follows:

$$\begin{aligned} \mathbf{P} &= \begin{bmatrix} p_{00} & p_{01} & p_{02} \\ p_{10} & p_{11} & p_{12} \\ p_{20} & p_{21} & p_{22} \end{bmatrix} \\ &= \begin{bmatrix} (1-S_0) & S_0 & 0 \\ (1-S_1)T_1^{(1)} & (1-S_1)T_1^{(0)} + S_1 T_1^{(1)} & S_1 T_1^{(0)} \\ T_2^{(2)} & T_2^{(1)} & T_2^{(0)} \end{bmatrix}. \end{aligned}$$

When the current state is 0, i.e., when there is no active transmission, the state transition occurs when there is a new arrival. Therefore, $[p_{00}, p_{01}] = [1-S_0, S_0]$ depends only on $S_0 = Np(1-p)^{N-1}$. The probability $p_{02} = 0$ follows from the fact that there can be only one agreement in a time slot.

Figure 3.16 shows the state transition diagram of the multichannel example with three channels. The model is not a birth-and-death model since there is a possible transition from state 2 to state 0.

- The state transition probability is

$$p_{kl} = \begin{cases} (1 - S_0), & \text{if } k = 0, l = 0; \\ S_0, & \text{if } k = 0, l = 1; \\ T_k^{(k-l)}(1 - S_k) + T_k^{(k-l+1)} S_k, & \text{if } 1 \le k \le M - 1 \\ T_M^{(M-l)}, & \text{if } k = M; \\ 0, & \text{otherwise.} \end{cases} \tag{3.25}$$

where $T_k^{(j)} = 0$ when $j < 0$.

Numerical Solution. The average utilization ρ per channel can be obtained as

$$\rho = \frac{\sum_{i \in \mathcal{S}} i \cdot \pi_i}{M}, \tag{3.26}$$

where π_i is the limiting probability that the system is in state i and $\mathcal{S}$ is the state space of the Markov chain. One obtains π_i by solving the balance equations of the Markov chain. We obtain the total system throughput as

$$TH_{ded} = M \cdot C \cdot \rho, \tag{3.27}$$

where C is the channel transmission rate per channel.

We evaluated a system with following parameters shown in Table 3.1. We assume that the

Table 3.1: System Parameters

Parameter		Value
N	Number of Devices	20
M	Number of Data Channels	2
p	Transmission Probability	0.05
R	Channel Bit Rate	1Mbps
PHYhdr	Physical Layer Header	128 bits
RTS	RTS Frame Size	160 bits + PHYhdr
CTS	CTS Frame Size	112 bits + PHYhdr
q	Termination Probability	0.1

duration of each slot is the time to transmit one RTS and one CTS frame, which, according to the standard, is equal to RTS + SIFS + CTS + SIFS $= 288 + 10 + 240 + 10 \approx 550\mu$sec. The success probabilities S_k for $k = 0, 1$ are

$$\begin{aligned} S_0 &= 20 \cdot p(1-p)^{19} = 0.3774 \\ S_1 &= 18 \cdot p(1-p)^{17} = 0.3763. \end{aligned}$$

Since we assume that the geometric termination probability is $q = 0.1$, the average packet size is $E[L] = 10$ slot duration or $10 * 550(\mu\text{sec}) * 1(\text{Mbps})/8 = 687.5$bytes.

With these parameters, we can formulate the transition probability matrix

$$\mathbf{P} = \begin{bmatrix} (1-S_0) & S_0 & 0 \\ (1-S_1)T_1^{(1)} & (1-S_1)T_1^{(0)} + S_1T_1^{(1)} & S_1T_1^{(0)} \\ T_2^{(2)} & T_2^{(1)} & T_2^{(0)} \end{bmatrix} = \begin{bmatrix} .6226 & .3774 & 0 \\ .0623 & .5990 & .3387 \\ .0100 & .1800 & .8100 \end{bmatrix}.$$

Multiplying $\mathbf{P}$ by itself 100 times gives

$$\mathbf{P}^{(100)} = \begin{bmatrix} .0709 & .3339 & .5952 \\ .0709 & .3339 & .5952 \\ .0709 & .3339 & .5952 \end{bmatrix}.$$

Since each row is the steady state distribution π, the system utilization ρ is given by $\rho = (.3339 * 1 + .5952 * 2)/2 = 0.7622$. The capacity of the system TH_{ded} is given by:

$$TH_{ded} = 2(\text{channels}) \times 1(\text{Mbps/channel}) \times 0.7622 \approx 1.52\text{Mbps}.$$

3.7 SUMMARY

- A Markov chain model development follows these steps: (1) understanding the system, (2) model construction, and (3) verification and validation. It is an iterative process. Once a model is constructed through verification and validation, the model is revised until it seems appropriate.
- We used a simple forwarding system to illustrate a discrete-time, birth-and-death process. This process has an analytical solution, which can be found by solving the balance equations.
- The blocking model is a continuous-time, birth-and-death process whose blocking probability is known as the Erlang formula.
- The DTMC model of slotted ALOHA is not a birth-and-death process, and it does not have a closed form solution. However, we can compute the steady state distribution numerically.
- The CSMA Markov chain models the behavior of links in a wireless LAN as on-off processes with exponential durations. This model captures the interference relationship between links using a conflict graph. With appropriate selection of parameters, the model can be used to assess the performance of large-scale wireless LAN networks that are not limited to a single collision domain.
- The multichannel MAC model captures the behavior of a dedicated control channel protocol with a DTMC. Though the model is not as accurate as the Bianchi model in that the backoff timer is not modeled, it can be used to assess the throughput enhancements of a multiple channel MAC protocol.

CHAPTER 4

Advanced Markov Chain Models

In this chapter, we describe two more advanced Markov chain models. The first example is the Bianchi model of Wi-Fi networks. The second example models the protocol level behavior of a multi-channel MAC.

In the first example, we see that a Markov chain model is used as a part of the whole performance model. The performance model consists of two parts: the network part and the part. The dynamics of each device is modeled by the DTMC model and the network level dynamics are captured by the network model. In the second example, we use a closed queueing network model to study the multi-channel MAC protocol.

4.1 THE BIANCHI MODEL

4.1.1 OVERVIEW

Bianchi proposed a model of an 802.11-based wireless LAN system in his paper (4). The goal of the model is to estimate the throughput of such a network when there are N active nodes. Bianchi assumes a single collision domain to avoid the complexity of the interference relationship. As all devices are in a single collision domain, only one out of N can successfully transmit a packet at any one time.

Bianchi's model consists of two parts: the device and network parts, as shown in Figure 4.1. The device part models the detailed behavior the backoff timer of a CSMA/CA device. The network part uses key performance parameters from the DTMC model to calculate the saturated throughput of the system.

4.1.2 THE DTMC MODEL OF A BACKOFF TIMER

Wi-Fi devices adopt the *binary exponential backoff* algorithm to control their transmissions and to limit collisions. Instead of transmitting a packet immediately, a device waits until its backoff timer expires, even when the channel is idle. When an on-going transmission terminates, the value of timer is set to a uniform random number in [0,CW-1], where CW is the current contention window size. The backoff timer keeps decrementing as long as the channel is sensed idle. The device pauses its timer when it senses a busy channel and resumes the count-down when it senses that the channel is idle for at least DIFS seconds, where DIFS is specified by the standard. When the timer

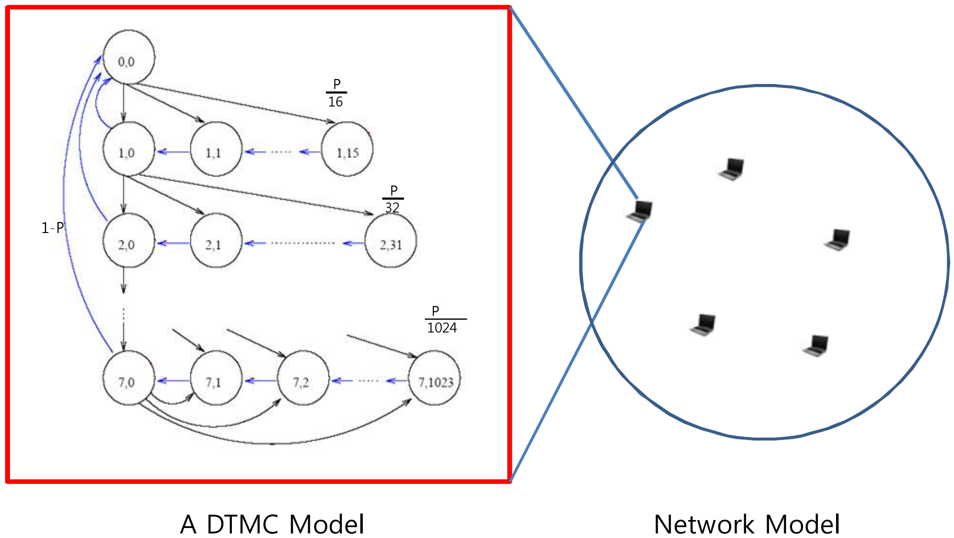


Figure 4.1: Bianchi model consists of two parts: a DTMC model and Network Model.

expires, the station attempts a transmission at the beginning of the next slot. If the transmission is successful, the device repeats the same steps for the next packet. In the event of a collision, the device doubles the value of CW, which explains the name "binary exponential backoff" given to this algorithm. The initial value of CW is set to CW_{min} (typically 32). When the value of CW reaches a predetermined CW_{max}, (typically 1024), it stops increasing. The device then drops the packet if three more transmission attempts again result in collision.

The motivation for the binary exponential backoff algorithm is that transmission failures tend to indicate that many device are attempting to transmit. Accordingly, to limit the chances of collision, the devices should reduce the probability that they try in a given time slot. The device reduces its probability of transmission by increasing CW.

It is important to understand the behavior of the backoff timer as it determines the average transmission attempt of Wi-Fi devices. Bianchi uses a two-dimensional Markov chain model of the backoff timer.

State Space. The backoff timer is modeled by a two-dimensional state vector $(s(t), b(t))$. In this vector, $b(t)$ is the value of the backoff timer at time t and $s(t)$ is the backoff stage that represents the number of previous unsuccessful transmission attempts of the current packet. The state space of the DTMC model $B(t) = (s(t), b(t))$ is given by:

$$\mathcal{S} = \{(s, b)|0 \leq s \leq m, 0 \leq b \leq CW_{max}(s)\},$$

where m is the maximum number of allowed transmissions. The value of m is 7 in the Figure 4.1. The maximum contention window size $CW_{max}(s)$ when the backoff stage is s is:

$$CW_{max} = \begin{cases} 1, & \text{s=0;} \\ 2^{3+s}, & s = 1, 2, \cdots, 7. \end{cases} \tag{4.1}$$

Note that CW_{max} starts from 16 and increases up to 1024. State $(0, 0)$ is the initial state after a successful transmission.

State Transition Probabilities.

- The backoff timer value $b(t)$ is decreased by one every clock tick as long as the channel is sensed to be idle. Therefore, the probability of transition from state (b, s) to state $(b-1, s)$ is one for $b \geq 1$. The horizontal arrows in the left plot of Figure 4.1 correspond to this.

- When the backoff timer value $b(t)$ reaches zero, the mobile device attempts transmission. The model assumed that the trial is a failure with probability p and a success with probability $(1-p)$, independent of anything else. Upon failure, $s(t+1) = s(t) + 1$ and the backoff timer value $b(t+1)$ is uniformly selected among $[0, CW_{max} - 1]$ where CW_{max} is the maximum contention window size. The value of CW_{max} is doubled upon failure of transmission attempt. The downward arrows in Figure 4.1 corresponds to the failure and probability $\frac{p}{CW_{max}}$ is equally assigned to all next level nodes, which models the random generation. The upward arrow from state $(s, 0)$ to $(0, 0)$ with probability $(1-p)$ happens when transmission attempt is a successful one.

Solving the DTMC. We can solve the DTMC with the methods explained in Section 2.4 to have a steady state distribution $\pi_{s,b}$ for a given p, the probability of collision given that there is a transmission in the slot. The objective of the DTMC model is to find the probability τ that a device transmits in a randomly chosen time slot. From the Markov chain, we can express the probability $\tau(p)$ as a function of p:

$$\tau(p) = \sum_{i=1}^{m} \pi_{i,0}, \tag{4.2}$$

which is the sum of probabilities that a backoff timer value is zero over backoff stage from 1 to m. Since it attempts transmission when the counter reaches zero, if we sum $\pi_{i,0}$ over i, it is the probability that a device attempts transmission.

Since the value of p is unknown, Bianchi uses another equation between p and τ:

$$p = 1 - (1-\tau)^{N-1}, \tag{4.3}$$

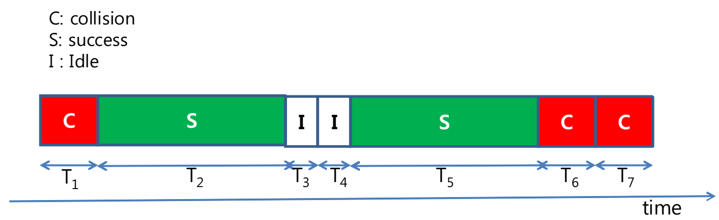


Figure 4.2: Example Scenario of the Bianchi Network Model.

which states that the probability p that a transmitted packet encounters a collision, is the probability that at least one of the $N-1$ remaining devices transmits, which is given by the right-hand side of (4.3). By using the two equations (4.2) and (4.3), we can find the value of τ and p, which turns out to be unique.

4.1.3 NETWORK MODEL

Bianchi assumes that each device transmits a packet with probability of τ in a slot, independently of other devices, where the value of τ is derived from the DTMC model. He considered the time slot for attempts as a renewal epoch and used the renewal-reward theorem (see Appendix) to find the average throughput.

In the beginning of each slot time, each device attempts transmitting with probability of τ, independently of the other devices. Three outcomes of such independent trials are possible: 1) no device attempts transmitting; 2) only one device attempts transmitting, which corresponds to a "success"; 3) more than one device attempt transmitting, which results in a "collision." Figure 4.2 demonstrates one possible evolution of the process, in which collision, success, idle, idle, success, collision, collision events occur in sequence. Note that the duration of each renewal period T_i differs depending on the type of event.

From the renewal reward theorem, the saturated throughput of the Wi-Fi network can be computed as follows:

$$TH_{802.11} = \frac{\text{E[packet size transmitted in a slot time]}}{\text{E[length of cycle time]}} = \frac{E[P]}{E[T]}, \tag{4.4}$$

where T and P represent the length of a cycle time and size of a packet transmitted in a cycle. Since there are three possible cases, the expected length $E[T]$ of a cycle is given by

$$E[T] = \sum_{e \in \{\text{idle, succ, coll}\}} p_e T_e, \tag{4.5}$$

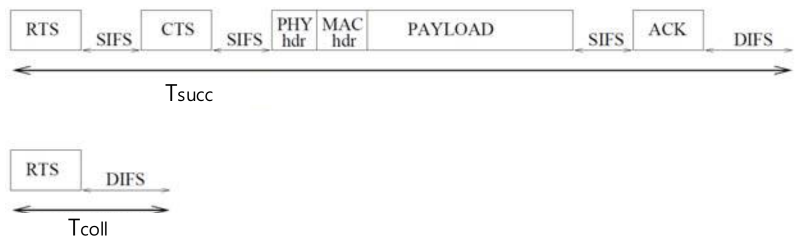


Figure 4.3: The durations T_{succ} and T_{coll}.

where T_e is the duration of event $e \in$ {idle, succ, coll}. Similarly, the expected reward $E[P]$ or the expected packet size during one cycle is

$$E[P] = p_{succ} \cdot E[L], \tag{4.6}$$

where L is the length of a packet. Since a packet is transmitted only in the case of "success," we have a zero reward in the two other cases.

The probabilities p_e of events can be expressed as follows:

$$\begin{aligned} p_{idle} &= (1-\tau)^n; & (4.7)\\ p_{succ} &= n\tau(1-\tau)^{n-1}; & (4.8)\\ p_{coll} &= 1 - p_{idle} - p_{succ}. & (4.9) \end{aligned}$$

Combining (4.5) and (4.6) gives

$$TH_{802.11} = \frac{p_{succ} \cdot E[L]}{p_{idle}T_{idle} + p_{succ}T_{succ} + p_{coll}T_{coll}}. \tag{4.10}$$

The duration of each event is protocol specific and those for the 802.11b standard are

$$\begin{aligned} T_{idle} &= \text{slot-time} = 20;\\ T_{succ} &= \text{RTS} + \text{SIFS} + \delta + \text{CTS} + \text{SIFS} + \delta\\ &\quad + \text{H} + \text{E[L]} + \text{SIFS} + \delta + \text{ACK} + \text{DIFS} + \delta;\\ T_{coll} &= \text{RTS} + \text{DIFS} + \delta\ , \end{aligned}$$

where δ is the propagation delay, H is the time to transmit PHY/MAC headers and $E[L]$ is the mean time to transmit a data payload. Figure 4.3 illustrates how T_{succ} and T_{coll} are derived. The RTS, CTS, and ACK are the MAC level control frames, and those in the above equation are the duration of those frames. Similarly, DIFS and SIFS is the duration of the interframe space between frames. Table 4.1 shows the parameter values for the 802.11b network.

Table 4.1: 802.11b System Parameters

Parameter	Value
PHYhdr	128 bits
MAChdr	272 bits
ACK	112 bits + PHYhdr
RTS	160 bits + PHYhdr
CTS	112 bits + PHYhdr
Channel Bit Rate	11Mbps
Propagation Delay	1 μ sec
Slot Time	20 μ sec
SIFS	10 μ sec
DIFS	50 μ sec

From the 802.11b parameters of Table 4.1, we have

$$\begin{aligned} T_{idle} &= 20\mu\text{sec}; \\ T_{coll} &= 280 + 50 + 1 = 331\mu\text{sec}; \\ T_{succ} &= 9412\mu\text{sec}. \end{aligned}$$

Figure 4.4 plots the probability τ of transmission (left) and the throughput (right) of the Wi-Fi network from the model with different numbers of devices when the channel capacity is 11Mbps. Note that the maximum throughput achieved is roughly 4.5Mbps out of 11Mbps when there are about 10 devices in the network. In the left plot, we can observe that the transmission probability decreases as the number of devices increases. Otherwise, the throughput would be much smaller due to excessive collisions.

4.2 A MULTICHANNEL MAC PROTOCOL WITH A FIXED DURATION - A QUEUEING NETWORK MODEL

Our second example deals with the same wireless LAN network as the first one, but it differs in that it uses a multichannel MAC protocol with three phases. We would like to understand the saturated throughput of multichannel MAC protocol. The model is more advanced in that it requires understanding of closed queueing networks and that we deal with fixed durations not an exponential ones. The fixed duration makes it difficult to use a Markov chain model. We briefly explain the protocol and then introduce two models.

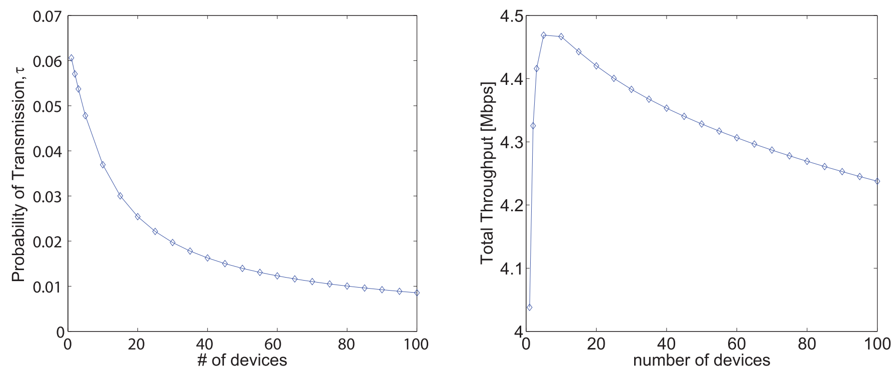


Figure 4.4: The probability (τ) of transmission (left) and the saturated throughput of Wi-Fi network from the Bianchi Model.

4.2.1 OVERVIEW

Assume that there are M data channels and one control channel, and that devices are equipped with a single radio interface, which is shared for both control and data transmissions. The control channel is used for making new agreements on data transmissions. As there is only one interface, after making an agreement over the control channel, the device jumps to an agreed data channel and transmits a packet. Once the transmission is done, the device returns to the control channel. As the device cannot overhear the control channel while it is away for data transmission, it cannot know the status of the data channels. If it starts transmission immediately, it is possible that the channel it selected is the busy channel.

One solution to the problem is to force devices to wait for a certain duration, i.e., T_{max} before making a new agreement. As long as T_{max} is longer than the maximum packet transmission time, the device can avoid collision because all on-going transmissions must be ended by then. This idea was used in multichannel protocols in (24; 16). We will discuss the model of the WiFlex introduced in (16).

The multichannel protocol consists of three phases: observation, contention and transmission as shown in Figure 4.5. The observation phase is the duration of waiting to avoid collision, which we just discussed. During the contention phase, devices try to make an agreement over the control channel. Devices agree upon the channel and duration. Once agreement is made, both devices move to the agreed channel and start transmission.

Assume that device A returns to the control channel after finishing packet transmission. Device A, after waiting for the duration of T_{max}, tries to send a packet to another device, say B. A contends for the transmission chance over the control channel and makes an agreement with B. Both devices jump to an agreed channel and exchange a packet. It returns back to the control channel and repeats the same process.

4.2.2 MODEL DESCRIPTION

Consider a network that consists of N single radio devices. The devices transmit packets using one of M data channels and exchange control packets in a separate control channel. The devices are in one of three phases: observation, contention, or transmission. The device stays in the observation phase for the fixed duration of T_{max}. It then contends for data transmission to make an agreement in the control channel. After making an agreement, the device jumps to the agreed data channel and transmits a packet. We consider two models: a three queue model and an (L+2) queue model. In the first one, we simply assume that the duration of the observation is exponential while in the second one, we approximate the observation phase with L queues of exponential duration with a mean of $\frac{T_{max}}{L}$. The first model is a special case of the second when $L = 1$. We first explain the case $L = 1$ to introduce a closed queueing network model.

4.2.2.1 3-Queue Model

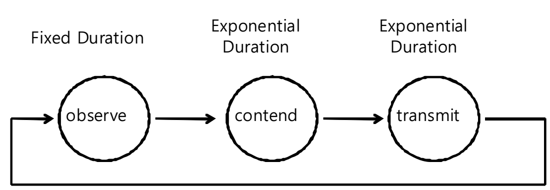


Figure 4.5: Each device goes through three states: observe, contend and transmit. The fixed length of the observation phase make the Markovian modeling difficult.

State Space. Let $n_i(t)$ be the number of devices in state i for $i \in \{o, c, t\}$ at time t. The three-dimensional vector (n_o, n_c, n_t) can be used to represent the state of the system. [1] So the state space **S** is given by:

$$\mathcal{S} = \{n = (n_o, n_c, n_t) | n_o + n_c + n_t = N, n_t \le M, n_i \ge 0, \forall i\}.$$

State Transition Rates. A typical device moves along the closed queueing network with three stages as shown in the Figure 4.5.

- The first circle models the observe phase of devices. After finishing transmission, the device takes a vacation at the first queue. Though it is of a fixed length, we assume that the first queue has exponentially distributed independent service times with rate $\mu_o = (T)^{-1}$. So its service rate is

$$\mu_o(i) := i\mu_o, \quad 0 \le i \le N. \tag{4.11}$$

[1] Since $n_o + n_c + n_t = N$, a two-dimensional vector is enough.

- The second circle models the contention for RTS/CTS [2] exchange in the (single) control channel. We assume that the queue has an exponential service rate μ_c. The rate μ_c is introduced to model the CSMA/CA contention success rate on the average. If RTS/CTS duration is d_{ts} and success probability of reservation is p_{succ} in each RTS/CTS duration, then μ_c can be approximated by:

$$\mu_c \approx \frac{p_{succ}}{d_{ts}}. \tag{4.12}$$

  Then the service rate $\mu_c(i)$ of the first queue with i devices in the queue is

$$\mu_c(i) := \min(i, 1)\mu_c, \quad 0 \le i \le N. \tag{4.13}$$

  In a highly contending slotted time CSMA network, $p_{succ} = e^{-1}$.

- Once a device succeeds in making an agreement by exchanging RTS/CTS message, it moves to the third queue, which models the transmit phase. We assume each transmission takes an exponentially distributed independent random time with mean $\mu_t := E[L]^{-1}$, where L is a length of transmission. Since the system has M data channels, the queue is modeled to have M servers. The service rate of this queue is load-dependent. So

$$\mu_t(i) := \min(i, M)\mu_t, \quad 0 \le i \le N. \tag{4.14}$$

  We further assume that all queues have an infinite buffer size.

- A state transition rate matrix of the Markov chain is as follows. Whenever a device succeeds in control channel access for RTS/CTS exchange, the transition from (n_o, n_c, n_t) to $(n_o, n_c - 1, n_t + 1)$ occurs at rate

$$q_{(n_o,n_c,n_t),(n_o,n_c-1,n_t+1)} = \mu_c, \tag{4.15}$$

  for $n_c > 0$. Also, a device finishing its transmission leaves queue t and enters vacation into queue v at rate

$$q_{(n_o,n_c,n_t),(n_o+1,n_c,n_t-1)} = \min(n_t, M) \cdot \mu_t, \tag{4.16}$$

  for $1 \le n_t \le N - n_c$. A device returns from the vacation to the control channel contention state with the following rate:

$$q_{(n_o,n_c,n_t),(n_o-1,n_c+1,n_t)} = n_o \cdot \mu_o. \tag{4.17}$$

  Therefore, the transition rate matrix for the continuous time Markov chain is

$$q_{n,n'} = \begin{cases} n_o \cdot \mu_o, & \text{if } n' = (n_o - 1, n_c + 1, n_t); \\ \mu_c & \text{if } n' = (n_o, n_c - 1, n_t + 1); \\ \min(n_t, K) \cdot \mu_t, & \text{if } n' = (n_o + 1, n_c, n_t - 1); \\ 0 & \text{otherwise.} \end{cases} \tag{4.18}$$

[2] RTS (Request to Send) and CTS (Clear to Send) are control messages exchanged between sender and receiver to make an agreement.

Solving a Queueing Network. The closed queueing network model is known to have a product form solution because it belongs to a BCMP network, a generalization of a Jackson network. A queueing network is a BCMP network if 1) all queues are one of four specific types discussed below and 2) the next queue that a customer enters is random. The four types of queue are 1) a FCFS queue with exponential service duration, 2) infinite server queue, 3) processor sharing queue, and 4) a single server queue with LCFS with pre-emptive resume.

As the observation queue belongs to the infinite server queue and the contention and the transmission queues belong to the FCFS queue, the model is a BCMP network, which has a productive form solution. The stationary distribution with respect to n can be found as a product form:

$$\begin{aligned}
\pi(n_o, n_c, n_t) &= \gamma f_o(n_o) f_c(n_c) f_t(n_t)\\
f_o(n_o) &= \frac{(\frac{\lambda}{\mu_o})^{n_o}}{n_o!}\\
f_c(n_c) &= (\frac{\lambda}{\mu_c})^{n_c}\\
f_t(n_t) &= \frac{(\frac{\lambda}{\mu_t})^{n_t}}{\min(n_t, M)! M^{(n_t - M)^+}},
\end{aligned}$$

where γ is a normalization constant

$$\gamma = \left(\sum_{n:n \in \mathcal{S}} f_c(n_c) f_t(n_t) f_v(n_v) \right)^{-1}, \tag{4.19}$$

and λ is a stationary state input rate to a queue to be chosen appropriately to compute the system performance.

The system throughput TH can be computed using the marginal distribution $\pi(n_t)$, which we can obtain by summing $\pi(n_o, n_c, n_t)$ over n_c from 0 to $N - n_t$. With the marginal distribution $\pi(n_t)$, the system throughput TH is:

$$TH = \sum_{n_t=1}^{\min(N,K)} n_t \cdot \pi(n_t) \cdot R, \tag{4.20}$$

where R is the transmission rate of a channel.

4.2.2.2 $(L + 2)$ Queue Model

Since the 3-Queue model was limited by assuming the exponential duration of the observation phase, we now extend the previous one by approximating the fixed duration with the sum of exponential random variables.

The fixed duration, which is a positive valued distribution, can also be approximated by a sum of exponential distribution. Let

$$Y(n) = \sum_{i=1}^{n} X_i(n) \ ,$$

where $X_i(n)'s$ are i.i.d. exponentially distributed random variables with mean $\frac{T}{n}$. Here, the number of phase is n, and after spending exponential duration with a mean of $\frac{T}{n}$, it jumps to the next phase. As $n \to \infty$ $Y(n)$ approaches a distribution of fixed length T. For more information on the phase type distribution, refer to the Chapter 2 of (20).

Figure 4.6 shows a schematic diagram of $L+2$ Queue model, in which the observation phase is represented by L independent tandem queues with infinite servers with a mean service time of $\frac{T}{L}$. With a large number L of queues, the observe phase approaches the fixed length of T.

State Space. The state space of the new model is similar to the three queue model but with more dimensions of $L+2$, which is given by

$$\mathcal{S} = \{n = [n_i, i = 1, 2, \cdots, l+2] | n_i \geq 0, \forall i, \sum_i n_i = N\}.$$

The first L queues model the observe phase and the last two queues model the contend and transmit phase, respectively.

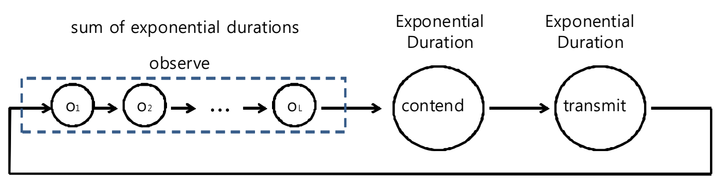


Figure 4.6: CTMC model of the multichannel model.

State Transition Rates. A state transition rate matrix of the Markov chain is as follows:

$$q_{n,n'=n-e_i+e_{i+1}} = \begin{cases} \frac{1}{L} n_i \cdot \mu_o, & \text{for } i = 1, 2, \cdots, L; \\ \mu_c & \text{if } i = L+1; \\ \min(n_t, K) \cdot \mu_t, & \text{if } i = L+2; \\ 0 & \text{otherwise.} \end{cases} \tag{4.21}$$

Note that transition rate is the same as that of 3-Queue model except the first L queues. The transition rate from n to $n - e_i + ei + 1$ slowed down by a factor of L and it is $\frac{1}{L} n_i \cdot \mu_o$ for $i = 1, 2, \cdots, L$.

Solving the Queueing Network. Note that the given CTMC model belongs to the BCMP queueing network because the first L queues are infinite server queues and others are exponential serviced FCFS queues. The steady state distribution has the product form solution as follows:

$$\pi(n) = \gamma \prod_{i=1}^{L+2} f_i(n_i), \tag{4.22}$$

where

$$f_i(n_i) = \begin{cases} \frac{(\frac{\lambda L}{\mu_c})^{n_i}}{n_i!}, & i = 1, 2, \cdots, L; \\ (\frac{\lambda}{\mu_o})^{n_o}, & i = L+1; \\ \frac{(\frac{\lambda}{\mu_t})^{n_t}}{\min(n_t, M)! M^{(n_t - M)^+}}, & i = L+2. \end{cases} \tag{4.23}$$

The system throughput can be computed the same way as the 3-Queue model. After computing the marginal distribution $\pi(n_t)$, we compute the throughput from equation (4.20).

4.2.3 NUMERICAL EXAMPLE

We consider a very simple example given by:

1. There are $N = 2$ identical sender-receiver pairs in a single collision domain and each of the senders has packets to transmit to a receiver.
2. There is one data channel ($M = 1$) and one control channel in the system.
3. A data channel has bandwidth $R = 1$Mbps.

The state of the system can be expressed as the 3-tuple number of devices at each stage: $n := (n_o, n_c, n_t) \in \mathcal{S}$, where the state space $\mathcal{S}$ is defined as

$$\mathcal{S} := \{(n_o, n_c, n_t) | n_o + n_c + n_t = 2, n_t \leq 1, n_o, n_c, n_t \in \mathbf{Z}^+\}, \tag{4.24}$$

where $\mathbf{Z}^+$ set of nonnegative integers. There are five possible states

$$\mathcal{S} = \{(2,0,0), (1,1,0), (0,2,0), (1,0,1)(0,1,1)\}\ .$$

The transition rate matrix of the above example is as follows:

$$\mathbf{Q} = \left(\begin{array}{c|ccccc} & (2,0,0) & (1,1,0) & (0,2,0) & (1,0,1) & (0,1,1) \\ \hline (2,0,0) & -\mu_o & \mu_o & 0 & 0 & 0 \\ (1,1,0) & 0 & -(\mu_o+\mu_c) & \mu_o & \mu_c & 0 \\ (0,2,0) & 0 & 0 & -\mu_c & 0 & \mu_c \\ (1,0,1) & \mu_t & 0 & 0 & -(\mu_t+\mu_o) & \mu_o \\ (0,1,1) & 0 & \mu_t & 0 & 0 & -\mu_t \end{array}\right). \tag{4.25}$$

Figure 4.7 shows the state transition diagram for the 3 Queue model. We assume that $\mu_o = \mu_t = \frac{1}{20 \text{ slot time}}$ and $\mu_c = \frac{1}{e} \approx 0.3679$.

$$\mathbf{Q} = \left(\begin{array}{c|ccccc} & (2,0,0) & (1,1,0) & (0,2,0) & (1,0,1) & (0,1,1) \\ \hline (2,0,0) & -0.05 & 0.05 & 0 & 0 & 0 \\ (1,1,0) & 0 & -0.4179 & 0.05 & 0.3679 & 0 \\ (0,2,0) & 0 & 0 & -.3679 & 0 & .3679 \\ (1,0,1) & 0.05 & 0 & 0 & -0.1 & 0.05 \\ (0,1,1) & 0 & 0.05 & 0 & 0 & -0.05 \end{array}\right). \tag{4.26}$$

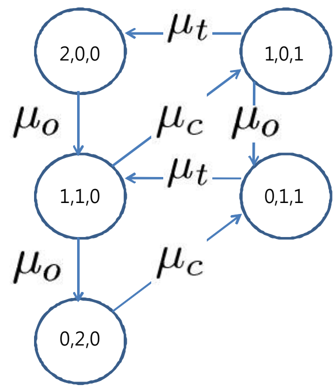


Figure 4.7: State Transition Diagram.

We can convert the transition rate matrix $\mathbf{Q}$ into a transition probability matrix $\mathbf{P}$ by dividing $\mathbf{Q}$ by $\mu_t + \mu_o$ and adding an identity matrix:

$$\mathbf{P} = I + \frac{\mathbf{Q}}{\mu_t + \mu_o} = \left(\begin{array}{c|ccccc} & (2,0,0) & (1,1,0) & (0,2,0) & (1,0,1) & (0,1,1) \\ \hline (2,0,0) & 0.8804 & 0.1196 & 0 & 0 & 0 \\ (1,1,0) & 0 & 0 & 0.1196 & 0.8804 & 0 \\ (0,2,0) & 0 & 0 & 0.1196 & 0 & .8804 \\ (1,0,1) & 0.1196 & 0 & 0 & 0.7607 & 0.1196 \\ (0,1,1) & 0 & 0.1196 & 0 & 0 & .8804 \end{array} \right). \tag{4.27}$$

The steady state distribution of the system is given by

$$\pi = (0.2793, 0.0759, 0.0103, 0.2793, 0.3552) .$$

The system throughput from equation (4.20)

$$TH = 1 \cdot (0.2793 + 0.3552) \cdot 1\text{Mbps} \approx .63\text{Mbps}.$$

4.3 SUMMARY

- Two advanced Markov chain models are introduced in the chapter: the Bianchi model and Multi-channel MAC model.

- The Bianchi model, the first throughput model of a wireless LAN network, is based on DTMC and Aloha model. It is very accurate to compute the saturated throughput but is limited to a single collision domain.
- The multi-channel MAC with fixed duration model is an application of a closed queueing network. Because it belongs to BCMP network, the closed form solution exists.

APPENDIX A

Exercises

RENEWAL PROCESS AND THE RENEWAL REWARD THEOREM

If an event is happening repeatedly, a *renewal process* can be used to model the repeating events. The *renewal process* is a counting process with inter-event times that are independent, identical, and random variables. For example, suppose that buses arrive at a station every five minutes, independently of anything else. The counting process $N(t)$ of bus arrivals is a renewal process.

Renewal Process. *Let $X_1, X_2, \cdots$ be a sequence of positive independent, identically distributed random variables that represent inter-event durations. Then, the counting process $N(t)$ of the events is a renewal process.* In the case of the Bianchi model, the beginning of time slots for transmission attempts are considered as a repeated events. The inter-event time varies depending on whether the slot is an idle one, a successful slot or a collision slot.

The renewal reward theorem is helpful tool to compute the time average rewards of a renewal process. We denote by R_n the reward earned at time of the n-th renewal. Assume that R_n's are independent and identically distributed. We can define a reward process $R(t)$ such that

$$R(t) = \sum_{i=1}^{N(t)} R_n,$$

which is a cumulative reward earned up to time t.

Renewal Reward Theorem. *If $E[R] < \infty$ and $E[X] < \infty$,*

$$\text{with probability 1}, \lim_{t\to\infty} \frac{R(t)}{t} = \frac{E[R]}{E[X]},$$

where $E[X_n] = E[X]$ and $E[R_n] = E[R]$ for all n.

The theorem states that the long run average of the reward process is given by the ratio of expected reward per cycle to the expected duration of a cycle. For instance, consider a machine that goes up and down repeatedly. The availability of the machine can be computed using the renewal-reward theorem. If we define a cycle to be one on duration and one off duration, then $E[X] = E[ON] + E[OFF]$. The reward per cycle is the duration of on period or $E[R] = E[ON]$. Therefore, the availability is give by:

$$\text{availability}=\frac{E[ON]}{E[ON]+E[OFF]}\ .$$

In the Bianchi model, they consider the reward as the amount of data transmission in each epoch to find average throughput. Refer to (28) or (22) for more information on the renewal process.

QUEUEING NETWORKS

A queueing network consists of a group of queues and customers to the queues and it is very popularly used to model a system with multiple resources. A computer system with hard-disk, CPU and processes can be modeled with a queueing network (15). So do manufacturing systems with multiple machines (6).

With more queues, analyzing a queueing network is more difficult than a single queue model in general. However, Jackson, in his seminal paper (11), discovered a class of queueing networks that are tractable analytically, which is named 'Jackson Networks" (11).

Jackson Network A group of interconnected queues is called 'Jackson Network' if

- The arrivals to queue i form a Poisson process with rate λ_i for all $i = 1, 2, \cdots, m$, where m is the number of queues and λ_i can be zero for a subset of queues.
- The service time is exponentially distributed with rate μ_i and customers are served in a FCFS manner.
- Upon completion of service at i, the customer moves to queue j with probability p_{ij} or leaves the system with probability $1 - \sum_j p_{ij}$.

Jackson showed that the steady state queue distribution π exists when the utilization ρ_i is less than one at every queue and the distribution for state $(k_1, k_2, \cdots, k_m)$ is given by :

$$\pi(k_1, k_2, \cdots, k_m) = \prod_i \rho_i^{k_i}(1 - \rho_i). \tag{A.1}$$

Note that it is product of individual queue equilibria and is known for "product form solution". The utilization ρ_i at queue i is the total arrival rate divided by the service rate or $\rho_i := \frac{r_i}{\mu_i}$. The total arrival rate r_i is the sum of external arrival rate and internal arrival rate to queue i, which is given by

$$r_i = \lambda_i + \sum_k p_{ki} r_k, \quad \text{for } i = 1, 2, \cdots, m\ .$$

Consider a computer system consisting of a CPU and a hard disk. The arrivals of processes to the CPU is Poisson with rate λ and the processing time at the CPU is exponentially distributed with rate μ_1. With probability of $\frac{1}{4}$, the process leaves the system or it moves to the hard disk with probability of $\frac{3}{4}$. The processing time at the hard disk is also exponential with rate μ_2. All process returns to the CPU upon completion at the hard disk.

The total arrival rates r_1 and r_2 satisfy $r_1 = \lambda_1 + r_2$ and $r_2 = \frac{1}{4}r_1$. Solving the equations, we have $r_1 = \frac{4}{3}\lambda$ and $r_2 = \frac{1}{3}\lambda$. Hence, the steady state distribution is

$$\pi(k_1, k_2) = \prod_{k=1}^{2} \rho_i^{k_i}(1 - \rho_i),$$

where $\rho_i = \frac{r_i}{\mu_i}$ for $i = 1, 2$.

Closed Queueing Network A queueing network is open if there are external arrivals to one or more queues and departures. However, if there are no external arrivals to and departures from the system, it is called *closed queueing network* . A Jackson network is a closed one, if $\lambda_i = 0$ for all i and if $\sum_j p_{ij} = 1$ for all i. Assume that there are K customers are moving along a closed Jackson network. As there being no arrivals and departures, the number of people in the system is fixed to K or

$$\sum_i k_i = K.$$

Let $\mathcal{K}$ be a set of states $\overrightarrow{k} = (k_1, k_2, \cdots, k_m)$ such that $\sum_i k_i = K$ or

$$\mathcal{K} := \{\overrightarrow{k} \mid \sum_i k_i = K\}.$$

The steady state distribution of the closed Jackson network is

$$\pi(k_1, k_2, \cdots, k_m) = C \prod_i \rho_i^{k_i}, \tag{A.2}$$

where C is normalizing constant to make $\sum \pi(k_1, k_2, \cdots, K) = 1$ or $C = \left[\sum_{\overrightarrow{k} \in \mathcal{K}} \prod_i \rho_i^{k_i}\right]^{-1}$. Note that the closed Jackson network also has the product form solution. However, determining the constant C can be a quite challenging task when the numbers m and K are large.

BCMP networks The simplicity of the product form solution attracted many researchers and they tried to find more general queueing networks with the property. A group of four scientists, Baskett, Chandy, Muntz and Palacios, found more general queueing networks that exhibit the product form solution, which is named *BCMP network* (8). A queueing network is a BCMP network if 1) all queues are one of the four types and 2) the next queue that a customer enters is random. The four types of the queue are 1) a FCFS queue with exponential service duration, 2) infinite server queue, 3) processor sharing queue, and 4) a single server queue with LCFS with pre-emptive resume. Refer to (26) for queueing network models.

Bibliography

[1] Human Genome Project Information. http://www.ornl.gov/sci/techresources/Human_Genome/home.shtml. 2

[2] The Network Simulator 2, http://www.isi.edu/nsnam/ns. 3

[3] D. P. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, 2nd edition, 1992. 42, 43

[4] G. Bianchi. "Performance Analysis of the IEEE 802.11 Distributed Coordination Function" *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, 2000. DOI: 10.1109/49.840210 49, 57

[5] R. R. Boorstyn, A. Kershenbaum, B. Maglaris, and V. Sahin, "Throughput Analysis in Multi-hop CSMA Packet Radio nNetworks," *IEEE Transactions on Communications*, 35(3):267–274, 1987. DOI: 10.1109/TCOM.1987.1096769 46

[6] J. Buzzacot and J. G. Shanthikumar, *Stochastic Models of Manufacturing Systems*, Prentice Hall, 1993. 72

[7] O. Dhal and K. Nygaard, "SIMULA: An Algol-based Simulation Language," *Communications of the ACM*, 9:671–678, 1966. DOI: 10.1145/365813.365819 3

[8] R. R. Muntz F. Baskett, K. M. Chandy and F. G. Palacios, "Open, Closed and Mixed Networks of Queues with Different Classes of Customers," *Journal of the ACM*, 22:248–260, 1975. DOI: 10.1145/321879.321887 73

[9] R. M. Fujimoto, K. Perumalla, A. Park, H. Wu, M. H. Ammar, and G. F. Riley, "Large-scale Network Simulation: How Big? How Fast?," In the *11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer Telecommunications Systems*, pages 116–123, 2003. DOI: 10.1109/MASCOT.2003.1240649 3

[10] B. Hajek and V. Loon, "Decentralized Dynamic Control of a Multiaccess Broadcast Channel," *IEEE Transactions on Automatic Control*, 27(3):559–569, 1982. DOI: 10.1109/TAC.1982.1102978 45

[11] J. R. Jackson, "Jobshop-like queueing systems," *Management Science*, X:131–142, October 1963. DOI: 10.1287/mnsc.10.1.131 72

[12] R. K. Jain, The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling," Wiley, 1991.

[13] L. Jiang and J. Walrand, "A Distributed Algorithm for Optimal Throughput and Fairness in Wireless Networks with a General Interference Model," In *Proc. of the Allerton Conference*, 2008. 46

[14] L. Kleinrock, *Queueing Systems. Volume 1: Theory*. Wiley-Interscience, 1975. 6

[15] E. D. Lazowska, J. Zahorjan, G. S. Graham, and K. C. Sevcik, *Quantitative System Performance, Computer System Analysis Using Queueing Network Models*, Prentice Hall, 1984. 6, 72

[16] J. Lee, J. Mo, T. Trung, J. Walrand, and H. So, "Wiflex: Multi-channel Cooperative MAC Protocol for Heterogeneous Wireless Devices," In *Proc. of IEEE WCNC*, 2008. 63

[17] S. C. Liew, C. Kai, J. Leung, and B. Wong, "Back-of-the-envelope Computation of Throughput Distributions in CSMA Wireless Networks," Technical report, Chinese University of Hong Kong, 2008. DOI: 10.1109/ICC.2009.5198774 46

[18] J. Medhi, *Stochastic Models in Queueing Theory*, Academic Press, 2nd edition, 2002. 6

[19] J. Mo, W. So, and J. Walrand, "Comparison of Multichannel MAC Protocols," *Transactions on Mobile Computing*, 7(1):50–65, 2008. DOI: 10.1109/TMC.2007.1075 50

[20] M. F. Neuts, *Matrix-Geometric Solutions in Stochastic Models: an Algorthmic Approach*, Dover Publications Inc., 1981. 67

[21] C. Dennis Pegden, R. E. Shannon, and R. P. Sadowski, *Introduction to Simulation Using SIMAN*, McGraw-Hill, 1995. 3

[22] S. M. Ross, *Introduction to Probability Models*, Academic Press, 7th edition, 2000. 13, 72

[23] S. M. Ross, *Stochastic Processes*, Wiley, New York, 2nd edition, 2005.

[24] J. Shi, T. Salonidis, and E. Knightly, "Starvation Mitigation Through Multi-channel Coordination in CSMA based Wireless Networks," In *Proc. of ACM MobiHoc 2006*, May 2006. DOI: 10.1145/1132905.1132929 63

[25] K. Case W. Turner, J. Mize and J. Nazemetz, *Introduction to Industrial and Systems Engineering*, Prentice Hall, New Jersey, 3rd edition, 1993. 1

[26] J. Walrand, *An Introduction to Queuing Networks*, Prentice Hall, 1988. 6, 73

[27] J. Walrand, *Communication Networks: A First Course*, McGraw-Hill, 1991. 42

[28] R. W. Wolff, *Stochastic Modeling and the Theory of Queues*, Prentice Hall, 1989. 72

Author's Biography

JEONGHOON MO

Jeonghoon Mo received the B.S. degree from Seoul National University, Korea, and the M.S. and Ph.D. degrees from the University of California, Berkeley. He is currently a professor in the department of Information and Industrial Engineering at Yonsei University, Korea. Before joining Yonsei, he has previously worked at AT&T Labs, Tera Blaze, and KAIST. His research interests include network economics, wireless communications and mobile services, optimization, game theory, and performance analysis.

Index

Analytical model, 3
Andrey Markov, 13
Aperiodic, 21
Axioms of the probability, 15

Backoff timer, 57
BCMP network, 66, 73
Bianchi model, 57
Binary exponential backoff, 57
Birth and death process, 26, 31

Closed queueing network, 64, 73
Contention graph, 47
Continuous time Markov chain, 23
Continuous time model, 3
CSMA Markov chain, 46, 48
CSMA/CA protocol, 46

Dedicated Control Channel MAC, 50
DES or Discrete event simulation, 4
Deterministic model, 3
Discrete time model, 3
Dynamic model, 3
Dynamical system, 13, 15

Independent set, 48
Irreducible, 20

Jackson network, 66, 72

Kendall notation, 6

Local balance equation, 36

Markov chain, 13
Markov property, 14
Mathematical model, 3
Memoryless property, 24
Model, 1
Model construction, 34
Model validation, 34
Model verification, 34
Modeling, 1
Modeling steps, 33
Monte Carlo simulation, 4
Multi-channel MAC protocol, 50

Performance model, 3
Poisson process, 28
Pure Aloha, 43
Pure birth process, 28
Pure death process, 28

Queueing model, 5
Queueing network, 6, 62

Renewal process, 71
Renewal-reward theorem, 60

Simulation model, 3
Slotted Aloha, 42, 43
State space, 15
State transition diagram, 16
Static model, 3
Steady state distribution, 20
Stochastic model, 3, 6

System, 1

Transition probability, 15
Transition probability matrix, 15
Transition rate, 25

Wi-Fi, 46