High Speed Round Robin Queueing Networks

by

Roy D. Yates

Submitted to the Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1990

Author ....
Electrical Engineering and Computer Science
May 4, 1990

Certified by ....
Robert G. Gallager
Fujitsu Professor of Electrical Engineering
Thesis Supervisor

Accepted by ....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Acknowledgments

I wish to thank Professor Robert Gallager for all of the advice and assistance that he has provided during the past six years. I am particularly grateful for his encouragement and optimism while I looked for a thesis topic. My understanding of data networks is primarily a consequence of his willingness to share his remarkable insight.

I am grateful for the improvements suggested by my readers, Professors Dimitri Bertsekas and Pierre Humblet. I also want to thank Professor Al Drake for many laughs as well as some very useful advice.

I would like to acknowledge the efforts of the LIDS staff, particularly Nancy Young, to make LIDS an agreeable place to work.

A great number of fellow students helped make my hours at LIDS more enjoyable. Although their friendship probably delayed completion of this work, I thank them for that delay. My many discussions with Rajesh Pankaj, Abhay Parekh, and Dr. Richard Hughes were especially appreciated.

While pursuing this degree, much of my time has been occupied by my children, Tony and Brett. Their ability to distract me has made my life far more interesting. Lastly, I thank my wife Theresa for her love, support, and patience.

This work was supported by an AT&T Bell Laboratories fellowship and by the National Science Foundation under grant NSF-ECS-831698. I am thankful for the funding.

Contents

List of Figures

Chapter 1

Introduction

1.1 The Burstiness Problem

A data network consists of a set of interconnected data communication links. A network may be used for a single purpose or a variety of purposes. For example, a network may be used for voice conversation, electronic mail, file transfer, remote database lookup and the interactive use of remote computers. Each of these uses makes its own particular set of demands on a data network. Although a network that has a single use may be designed to be particularly efficient, there are benefits to building networks that serve a variety of uses.

In particular, the incremental cost of providing additional data link capacity for additional uses is usually small. Moreover, these incremental costs are rapidly getting smaller. Still, one may argue that a particular use may be in sufficient demand to merit a separate network optimized for that use. Eventually, we might have one network for voice, a second for electronic mail and a third for file transfer. The difficulty with this approach is simply that each subsequent new use may require a new network. Furthermore, as network speeds continue to increase, networks almost certainly will be used in ways that we cannot predict. Since it is difficult to identify the future uses of a network, it would be desirable to identify networks that would be suitable for a variety of uses.

However, different users can make conflicting demands on a network. An interactive user may send messages consisting of commands of just a few characters. For this user, small delay would be important. An electronic mail customer might send messages of at most a few thousand characters and would be able to tolerate significantly greater delays. Another user might use the network infrequently to transfer very large files such as digitized video images. This user would be called *bursty.* If a large file were sent over a link without interruption, the messages of other users of the link may suffer unacceptably long delays. That is, the bursty user can degrade the service received by the other users. In addition, as network speeds increase, users with greater burstiness can be accommodated and the problems associated with bursty users become more significant. Our intention is to show that the degradation in service caused by burstiness is avoidable. In particular, we will describe a network that provides satisfactory service for all users.

1.2 Reducing the Effect of Burstiness

We will examine a network model that treats a bursty source as a session that submits long messages. Our aim is to have the network provide service in such a way that long messages are unable to penalize short messages. Furthermore, we make the following claim:

Shorter messages should wait less.

We will say that a queue provides *length discrimination* if service is provided in such a way that shorter messages wait less. Without reference to a particular network or its uses, it is difficult to state this definition with greater precision. However, we can make several arguments supporting the claim that length discrimination is a desirable property for queueing systems with variable burstiness.

First, a long message can endure additional waiting time before that waiting time becomes a significant fraction of the system time. For a message with 10^5 packets, a waiting time of 10^3 slots would be inconsequential. Yet, for a one packet message, a wait of 10^3 slots would be very significant. Considering that a single digitized video image could require 10^8 bits, a message of 10^5 packets is not unrealistic.

A second argument favoring reduced delay for short messages is that a long message session should be willing to endure the delay that would occur if every other session submitted long messages. That is, a session cannot expect better service than that it would obtain if all sessions acted as it does.

Another reason to provide length discrimination is that it may be important for network control messages, which tend to be short, to receive fast service. Lastly, a final argument is even simpler. In many networks, short messages are often generated by impatient human customers.

The above arguments make no assumptions about the uses of a network. Hence, we believe that for future high speed networks, whose uses are difficult to determine, the property that short messages receive better service is particularly desirable. In this work, we identify a queue discipline that provides length discrimination. We will describe this service strategy, which we call *round robin*, in great detail. However, the basic method of round robin is quite simple. The backlogged messages are arranged in a ring and the server cycles around the ring providing each message one unit of service. When a new message arrives, it is inserted into the ring. When the service requirement of a message has been met, that message departs and is removed from the ring.

Given our intention to provide small delay for short messages, it is easy to see that many queueing disciplines are unsuitable for networks with different types of sources. Consider the following example of a single link with two message classes. We define a slot as the length of time required to send one packet. Suppose that at the end of each slot, a class 1 message arrives with probability 1/150, a class 2 message arrives with probability 3/50000 but no more than one message arrives in a slot. For both sessions, the number of packets in a message is geometrically distributed with an average length of 10 packets for a class 1 message and 10000 packets for a class 2 message.

Since the arrival process is memoryless, both short and long messages join the queue in a typical state. So, for the first come first served (FCFS) queue, the distribution of the waiting time (the time in the system before service) of the long message would be the same as that of the short message. In fact, the average waiting time for a message of either class is 18000 slots. Under round robin service, we will be able to verify that the average waiting time would be 18 slots for a class 1 message and 19998 slots for a class 2 message. Of course, providing better service for class 1 messages does increase delay for class 2 messages. However, the typical class 2 message has 10000 packets so that the increase in the average system time for class 2 messages is only about seven percent. This penalty is relatively modest especially if it were important to provide small delay for class 1 messages.

For the FCFS system, the long delay experienced by a short message results from both the service demands of long messages ahead of it in the queue as well as the (possibly long) residual requirement of the message in service. Even if the short message could bypass long messages in the queue, the possibility of arriving while a long message is in service can generate inordinately long delays. In short, burstiness is a problem for any queueing discipline that cannot preempt the message in service.

Note that when we speak of preemption of a message, the intention is to allow the server to interrupt the transmission of a message after one or more packets in order to transmit the packets of another message. In this context, a packet would contain perhaps several hundred bits so that the overhead required to associate a packet with a particular message would be relatively insignificant.

Even for reasonably simple choices of queue discipline, analyzing a single queue can be difficult. There may be many appropriate queueing disciplines that we are unable to effectively examine. We cannot hope to argue that a particular queue discipline is best. Instead, we will show that a queue that performs round robin on the messages in the system happens to have some desirable properties. In particular, given a fixed packet rate, a session will have its average message waiting time decrease as its message length decreases. That is, less bursty sessions will get better service. In addition, the service received by a session will depend only on that session's own message process and the overall packet arrival rate. In short, one session need not care if other sessions submit long messages infrequently or short messages at a high rate as long as the overall packet rate remains the same.

1.3 Related Work

The basic goals of this research have been to show that

- The round robin queue can be analyzed without approximation.
- The round robin queue has desirable properties for a network with bursty sources.

This thesis has been motivated by the flow control problems caused by bursty sources. However, our analysis of round robin is of interest, independent of its application to flow control. In this section, we will describe related work in the fields of queueing and flow control.

1.3.1 Queueing

The round robin service discipline has been studied directly as well as by a processor sharing approximation. We will consider these two types of analysis separately.

Direct Analysis of Round Robin

The term round robin is used to describe a wide variety of cyclic queueing systems. We choose to divide these queues into two basic types.

- *Message Round Robin* A single server cycles among different messages that are generated by a single arrival process. How the server moves from message to message must be specified.

- *Session Round Robin* A single queue is used by a fixed number of sessions. Each session has an independent message arrival process and a separate queue for its messages. Typically, the messages of a single session are given FCFS service. The server cycles among the sessions that have backlogged messages. The order of the sessions within this cycle may be fixed or may vary dynamically if a session loses its position whenever its queue becomes empty. How the server moves from one session to the next and how the sessions are ordered in the service cycle must be specified.

This thesis analyzes a particular type of message round robin queue. However, much of the queueing literature is directed toward session round robin. Session round robin systems are often called polling systems or cyclic service systems. Polling systems have been extensively studied under a wide variety of assumptions; see [Tak88] for a complete survey. Typically, analysis of these systems concentrates on the mean and distribution of the message waiting time at each queue. Among systems of this type, those that operate in discrete time for which the server provides a limited amount of service to each queue before switching are similar to the round robin queue we analyze. Takagi [Tak87] has analyzed a discrete time, limited service polling system in which each session queue can buffer a single message that consists of a geometric number of characters. During each visit of the server, one character of a message is served. In addition, the order of the sessions within the cycle is fixed and there is overhead associated with moving from one queue to the next. The transform of the message waiting time distribution is found. Despite the use of geometric message lengths, exact analysis of this queue is surprisingly difficult.

In [Dai87], a priority queue in which messages of each priority class are provided round robin service is studied. For each priority class i, the message arrivals form a Poisson process and message lengths (in bytes) are chosen from a general service distribution G_i. Arrival processes for different priority classes are independent. Messages are segmented into packets with overhead added. Between message arrivals and departures, the server cycles among the messages in the system that have highest priority. Upon the arrival of a message of priority higher than that of the message being served, transmission of the packet currently in service is completed before the new arrival goes into service. This work focuses on the mean system time of a message of a given priority class as a function of the message length and maximum packet size.

Doshi and Rege, [DoR85], examine a two stage queue in which the first queue is first come first served (FCFS) and the second queue is round robin. Messages in the FCFS queue are always given priority over messages in the round robin queue. The service strategy allows each message a maximum amount of service in the FCFS queue. A message that does not have its service requirement fulfilled in the FCFS queue moves to the round robin queue. We call such a system a FCFS-RR queue. The intention of FCFS-RR is to provide very fast service to the short messages that can complete service in the FCFS queue. Consideration is given to both a general service distribution as well as one in which a message belongs to one of two classes such that a message in the first class is certain to complete service in the FCFS queue while a message of the second class has an exponentially distributed length. The emphasis in this work is on the Laplace-Stieltjes transform of the system time distribution.

For the FCFS-RR queue, the special case in which the first packet of a message is served in the FCFS queue appears to be quite similar to the round robin queue studied in this thesis. This special case is also examined by Fraser and Morgan; see [FrM84]. In this work, successive packets from the queue are assembled into frames for transmission on trunks. Queueing and framing delays are studied, by analysis and simulation, for a network serving several types of traffic. The motivation for this work is to provide fast service for the single character messages of interactive users on a network that may also handle longer messages generated by file transfer.

Morgan continues this work in [Mor89]. In this work, it is stressed that 'well behaved users should be protected against the demands of the hogs.' Three systems, FCFS, Round Robin, and the two stage FCFS-RR queue, are considered. For a mixture of traffic, mean delay analysis is given. Heavy traffic approximations are also made. Discussion of this work in the context of flow control will follow in Subsection 1.3.2.

Processor Sharing

Often the round robin queue is approximated by the continuous time processor sharing model in which, given n jobs in the system, all jobs are served simultaneously each with rate $1/n$. It is argued [Kle76] that the processor sharing system is obtained as the limiting case of a round robin queue in which the unit of work becomes infinitesimal. That is, as the unit of work approaches zero, the round robin server can be thought of as serving all customers simultaneously.

When all service times have the same exponential distribution, the analysis of the processor sharing system is the same as that for the M/M/1 queue. With general service requirements and multiple classes, the stationary properties of the processor sharing queue can be analyzed through examination of the reverse time queue [Ros83] or by verification of the Chapman-Kolmogorov equations [Yas83]. However, this analysis requires the use of an uncountable state space. Alternatively, Kelly examines the reverse time queue while approximating general service requirements by sums of exponential random variables [Kel79]. This approximation of general service times by sums of exponential random variables seems particularly questionable when the service times are deterministic.

By reversibility, the stationary distribution of the processor sharing queue can be found. This allows the mean system time as a function of message length to be determined. However, reversibility does not yield any additional information about the moments or distribution of the message system time. Both Yashkov and Ott [Ott84] study the distribution of the message system time by a decomposition of the system time into a sum of independent random variables. Yashkov focuses on the case in which a message arrives to find the state of the queue described by the stationary distribution. Ott considers the case in which a message arrives to find a particular state and averages over this set of states, verifying the results of Yashkov.

The advantage of the processor sharing queue is that it is relatively tractable, particularly when the message lengths are exponentially distributed. However, one should keep in mind that the processor sharing queue is usually a mathematical abstraction. In real systems, the server usually provides discrete units of service, one at a time. It is difficult to determine the validity of the approximation of round robin by processor sharing without actually analyzing round robin directly. Clearly, one can put more faith in the approximation when the unit of work is small in comparison to the service requirements of the customers.

Priority Queues

How a network can provide service to customers with very different requirements has been examined by Régnier, using priority queues; see [Reg86]. Given a set of session rates and priority assignments, it is possible to find the average delay for each session. Sessions requiring low delay are given high priority. However, identifying the collection of feasible session rates and delays as well as an appropriate criterion for optimality is very complicated. The basic difficulty is that a session cannot receive higher priority without penalty to lower priority sessions.

Thorough analysis requires that message lengths be chosen from an exponential distribution common to all sessions. Furthermore, for each session, the message arrivals must form a Poisson process. Even with these restrictions, analysis of networks of these priority queues still requires the assumption that the queue departure processes are Poisson. Although this assumption is technically incorrect, it is widely used since a Poisson process tends to be a good approximation for the departure process in a network of queues. Differences in analysis aside, the Régnier's model does not characterize networks with sources of variable burstiness.

1.3.2 Flow Control

Flow control is the regulation of the movement of messages or packets into as well as within the network. Thorough discussion of flow control issues and methods can be found in [BeG87]. Flow control serves a variety of purposes including preventing buffer overflow, limiting cross network delay, enforcing session packet rates and reducing the effects of burstiness. Of these objectives, message based round robin service attempts only to restrict the ability of a bursty session to increase delay for other sessions. Round robin does not regulate the ability of sources to dump packets into the network nor does it provide a mechanism to assign or limit session packet rates. As we shall see, it will remain important to limit the average data rate of each session simply because the performance of any queue is inextricably tied to the overall load. However, we believe that a flow control strategy designed to severely restrict burstiness is not generally desirable. For example, when the network is not busy, limiting the ability of an active bursty session to submit packets to the network wastes network resources and results in that session enduring unnecessary delay.

For most of our analysis, we assume there exists infinite buffer space allowing arbitrarily long delays at each link. However, we demonstrate that it is possible to analyze a round robin queue in which the number of messages in the system is bounded. That is, a new arrival that would cause the number of messages in the queue to exceed the bound is blocked. Note that in our model, a message can be arbitrarily long. As a result, a bound on the number of backlogged messages is not a bound on the number of backlogged packets. However, we will show that the bound on messages does guarantee a bounded system time for any message that is not blocked. In short, round robin is not a flow control scheme, although it does achieve some of the basic goals of flow control.

The use of round robin service as part of a flow control strategy has been considered. Much of this effort has focused on the *fairness* properties of round robin. We now examine some of this work.

Max-Min Fair Flow Control

A network in which session round robin service is provided at each link in conjunction with window flow control has been analyzed by Hahne; see [Hah86]. At each link, a separate queue is maintained for each session. During each slot at each link, the sessions are polled in round robin order until a session that has a packet to send is found. After this packet is sent, polling begins at the next session in the round robin ring. At each link, a finite length queue (a *window*) is allocated for each session using the link. A session can send a packet over a link only if its window at the next link is not full.

Hahne's analysis of session round robin focused on the ability of the network to enforce session throughput rates that have the property that each session rate is as large as possible subject to the contraint that every smaller session rate is as large as possible. Such a set of rates is known as *max-min fair*.

Hahne examined session round robin under heavy load conditions. That is, each session offered a packet to the network at every opportunity. It was shown that the session throughput rates approached the max-min fair rates as the window sizes increased. In addition, it was verified that sessions that used small windows would be guaranteed both small delays and a lower bound on service rate.

Both Hahne's approach and the message round robin queue described in this thesis allow sessions that require small network delay to coexist with other sessions that may try to maximize their use of the network capacity. Moreover, we believe that the typical operation of the two round robin strategies often may be quite similar. When no session has more than one message at a link waiting for service, session and message round robin operate identically. For message round robin on high speed networks in which the rate of any one session is small, we show that it is very unlikely for a session to have more than one message waiting at a link. Consequently, the two round robin systems should behave similarly much of the time.

Hahne's heavy load analysis of session round robin permitted cross network delay guarantees to be made but did not allow the examination of the average properties of the network. In comparison, we will be able to analyze the typical behavior of message round robin but we will not be able to bound worst case cross network delay. In fact, the desire to characterize the typical operation of session round robin prompted the study of message round robin.

Other Forms of Fairness

For networks with sources of varying degrees of burstiness, alternative notions of fairness have been proposed. In [Mor89], Morgan defines fairness as 'the protection of light users insofar as possible.' Furthermore, he examines the use of round robin as a method of passive flow control. In particular, he concludes that round robin unlike packet FCFS, protects the well-behaved, that is non-bursty, user from the effects of ill-behaved or bursty sources.

Network fairness is also addressed in [DKS89]. Although fairness is not explicitly defined, the emphasis is on using the queueing discipline to protect light users, such as interactive computer sessions, from bursty traffic such as file transfers. Only networks in which packets have variable lengths are considered. With this restriction, it is concluded that packet by packet session round robin would be unfair but that bit by bit session round robin would be a fair but impractical strategy. A queue discipline that approximates bit by bit session round robin is then proposed. Upon arrival, a packet is given a time stamp corresponding to the time at which it would complete service under bit by bit session round robin. After a service completion (or message arrival in the preemptive version), service is then provided to the packet with the earliest time stamp. It is argued that the preemptive version of this method is impractical. For the nonpreemptive version, it is verified by simulation that round robin service protects an interactive user submitting 40 byte packets from a collection of file transfer sources submitting 1000 byte packets. The simulations were performed using a variety of flow control algorithms in conjunction with the queue discipline. In comparison, FCFS service failed to protect the interactive user despite the use of flow control.

In [Zha89], Zhang proposes a queue discipline similar to that in [DKS89]. The arriving packets of a session are stamped according to a *Virtual Clock* maintained for that session. In each slot, the packet with the earliest stamp is transmitted. After a packet arrival of session i, virtual clock i is advanced by $1/r_i$, where r_i is a predetermined average packet rate for session i. In addition, a virtual clock is never allowed to fall behind the real time. While the queue has a backlog, the packets of a session are not transmitted faster than the average rate. It is difficult to determine the extent to which this type of service is similar to round robin. Compared to round robin, the advantage of Zhang's approach is that the server is not constrained to treat all sessions identically.

1.4 Thesis Overview

This work can be divided into two parts. The first five chapters consider only an isolated round robin queue used by a single session. The remaining chapters consider networks used by many sessions in which round robin service is provided at each link.

In Chapter 2, we will outline the basic results that we will need from the theory of Markov chains. In addition, we will describe our data network model and consider our modeling assumptions. Lastly, we will describe the simplest form of the round robin queue and find its stationary distribution.

Chapter 3 considers a more complicated round robin queue in which messages are assigned to classes and the service requirement of a message depends on its class. We derive the stationary distribution for this queue and the marginal distribution for the number of queued messages belonging to a subset of the message classes. Further, we use classes to examine the average message delay as a function of the message length.

In Chapter 4, we describe a way to decompose the system time of a message into a collection of independent random variables. We then use this decomposition to find the first and second moments of the system time of a message as a function of the message length. An upper bound to the message system time variance is also found.

In Chapter 5, we find the stationary distribution for a last come first served (LCFS) queue. We do this by essentially the same method as we employed for the round robin queue. We find that the stationary distributions for the round robin and the LCFS queues are the same. This implies that the mean system time of a message is the same under the two strategies. This prompts us to compare the variance of the system time under the two service disciplines. We find that for long messages, the system time variance is roughly the same for both disciplines. However, we prove that round robin always provides a smaller system time variance than LCFS. In addition, we identify a class of systems, called permutation queues, all of which have the same stationary distribution. We verify that round robin and LCFS are types of permutation queues.

In Chapter 6, we examine a round robin queue that permits messages to be blocked if the number of messages in the system exceeds a threshold. The stationary distribution and the blocking probability is found. However, we find that a network of such queues is not amenable to analysis.

Chapter 7 considers a round robin queue that is used by multiple sessions. This queue provides service in such a way that we can reduce this queue to one that has multiple message classes. We also verify that the performance penalty associated with this reduction is small.

Chapter 8 proves that a network of round robin queues used by multiple sessions has a product form stationary distribution. That is, the equilibrium behavior of a link is the same as it would be in isolation with each session that had used that network link submitting new messages directly without passage through a network. We construct a proof that is valid for two different models. In the first model, when a message enters the network, its length is chosen from a distribution that may be unique to the session that submitted the message. This message has the same length at every link in the network. In the second model, each time a message arrives at a link, its length is an independent random variable that is described by a distribution that may be unique to the session that submitted the message. Effectively, a message chooses a new length each time it arrives at a new link. This second model is not a particularly appropriate choice in the context of data networks. However, with respect to more general queueing networks, the independence result for the second model is of some interest.

The effect of round robin service on routing and flow control is discussed in Chapter 9. We argue that routing based on average session data rates is an appropriate strategy when the links provide round robin service. In addition, we argue that the use of round robin service simplifies several other difficult data network problems. In particular, it becomes possible to argue that network costs or usage charges should have a very simple form.

We briefly summarize this work in Chapter 10. In addition, we consider some unresolved issues.

1.5 New Results

Much of the queueing analysis in this thesis appears to be new. To a certain extent, this is a consequence of the view that the discrete time M/G/1 queue is less insightful as well as less tractable than its continuous time counterpart. We believe that this view may be correct for FCFS systems but not for round robin queues. Although processor sharing is mathematically well understood, it is not physically realizable. Previously, the merits of the processor sharing approximation have been validated by simulation of round robin systems. Our analysis verifies that a particular discrete time round robin queue shares the following known properties of the processor sharing queue.

- The number in the queue is geometrically distributed.
- The service requirement of a departing message is independent of the state of the queue the instant after departure.
- The service already received by a given message is independent of the number of other messages in the queue as well as the service already received by those other messages.
- The stationary distribution of a network of queues has a product form solution.
- The mean delay of a message is proportional to the length of the message.

We are unaware of any previously known results of this type for a round robin queue,

The delay analysis of Chapter 4 is also new, partly because of the specificity of the round robin queue model. Although the message system time variance results are new, they are analagous to the known results for the processor sharing queue

The application of round robin to flow control has been studied more extensively. The desirability of round robin service has been recognized in a variety of contexts. In this work, we argue that length discrimination is a desirable queue property. We then verify that round robin service provides length discrimination. Morever, a session's mean message delay is found to be independent of the burstiness of the other sessions. In addition, this work argues that the burst length is a valuable way to characterize the burstiness of a source. Previously, this conclusion has been recognized (see [Mor89] for example) but stated less explicitly.

Chapter 2

Preliminaries

In this chapter, we describe the data network model that we will use throughout the remainder of this work. Furthermore, in Section 2.2, we summarize some results about Markov chains and reversibility that we will need in our subsequent analysis. In the last section, we will describe and analyze the simplest form of the round robin queue.

2.1 The Data Network Model

In our network model, each communication link provides an errorless point to point communication channel between a pair of nodes. Nodes can receive, store and transmit fixed length *packets* of data. A packet will take precisely one unit of time, a *slot*, to be sent over a link. We will assume that the nodes are completely synchronized such that all nodes begin packet transmissions at the same instant.

A packet originates at an external source and passes through a sequence of nodes and links to a destination that is also outside of the network. A source-destination pair is called a *session* and the sequence of nodes and links used by a session is called a *route*. At the end of each slot, we will allow each source to submit a collection of packets, a *message*, to the first node on its route. That is, during a slot, a node may receive an arbitrarily large number of packets from each of its external sources. However, a node can send only one packet per slot over a link.

At the end of each slot, a session submits a message with some fixed probability independent of all other message arrivals. We will call this probability the *message rate* of the session. For each session, the number of packets in a message will be a random variable that is independent of all arrival times and of all other message lengths. The product of a session's message rate and average message length will be called the *packet rate* of the session. A message that arrives at the end of slot s can begin service at the start of slot $s+1$. If the last packet of this message is sent during slot t, we will say that the *system time* of the message is $t - s$. We will define the *waiting time* to be the difference of the system time and the length of the message. Note that the waiting time equals the number of slots a message is in the system during which it does not receive service.

The requirements of a fixed packet length and synchronized slot times have been made for analytic convenience. The assumption of fixed packet lengths is quite reasonable since it is commonly made in the design of fast packet networks. The assumption of synchronized slot times is less reasonable. We shall see that requiring all links to operate at the same speed plays a crucial role in our analysis. However, the requirement that all packet transmissions begin at the same instant is of little importance and could be discarded.

Communication links are not errorless in practice, but ensuring that messages are received correctly can be separated from the issue of how the network should provide service to its messages. In addition, as links become increasingly reliable, errors become rare events that should have relatively little effect on the overall network throughput or delay. Since, it is our intention to characterize the performance of networks with bursty sources, we will ignore the small effect of errors.

As we shall see, the model we have chosen will be reasonably simple to analyze. However, one can argue that the memoryless arrival process is an unrealistically simple choice, particularly for bursty sessions. Perhaps a better model would be one in which a session is silent for some length of time followed by an active period during which the session submits packets at some high rate. Such an arrival process is not memoryless. Yet, if we call the collection of packets submitted during an active period a message, then the message length characterizes the burstiness of the session. If desired, we could add to the model an extra access link that feeds the packets into the network proper at the appropriate rate. Since the packets pass through each link one by one, it makes little difference whether all of the packets enter the network at once or in a high rate stream. In either case, we sacrifice little by taking advantage of the tractability of our simpler memoryless model.

2.2 Reversibility and Markov Chains

In this work, we will derive the stationary distributions of several queueing systems by examination of the queue under time reversal. Before doing so, we need to describe the properties of the time reversed Markov chain. A thorough treatment of this subject can be found in either [Kel79] or [Ros83].

Let $U_0, \ldots, U_k$ represent the sequence of states of a discrete time, aperiodic Markov chain with transition probabilities P_{ij} and stationary distribution π_i. Suppose that the system was started so that $P\{U_0 = i\} = \pi_i$. In this case, $P\{U_t = i\} = \pi_i$ for all t. We can generate a new process whose sample paths consist of sample paths of the Markov chain in reversed order. The sequence of states of this new system would be $U_k, \ldots, U_0$. We call this new process the reverse time system since it can be viewed as a process that starts at time k (with the stationary initial distribution) and runs backwards until time 0.

We can show that the sequence of state transitions of the reverse time system is also a Markov chain. If we let Y represent the sequence of states $U_{t+2}, \ldots, U_k$, we can write

$$\begin{aligned}
P\{U_t = i | U_{t+1} = j, Y = y\} &= \frac{P\{U_t = i, U_{t+1} = j, Y = y\}}{P\{U_{t+1} = j, Y = y\}} \\
&= \frac{P\{U_t = i\} P\{U_{t+1} = j | U_t = i\} P\{Y = y | U_t = i, U_{t+1} = j\}}{P\{U_{t+1} = j\} P\{Y = y | U_{t+1} = j\}} \\
&= \frac{\pi_i P_{ij} P\{Y = y | U_{t+1} = j\}}{\pi_j P\{Y = y | U_{t+1} = j\}} \\
&= \frac{\pi_i P_{ij}}{\pi_j}
\end{aligned}$$

The distribution of the state U_t given the states $U_{t+1}, \ldots, U_k$ depends only on the state U_{t+1}. So, the reverse time process is a Markov chain with transition probabilities

$$P^*_{ji} = \frac{\pi_i P_{ij}}{\pi_j}$$

We will make use of the following result.

Theorem 1 *Consider an irreducible Markov chain with transition probabilities P_{ij}. If one can find nonnegative numbers π_i, $i \geq 0$, summing to unity and $[P^*_{ij}]$, a transition probability matrix, such that*

$$\pi_i P_{ij} = \pi_j P^*_{ji} \tag{2.1}$$

then the π_i are the stationary probabilities for both chains and the P^*_{ij} are the transition probabilities for the reversed chain.

Proof Equation 2.1 directly implies that the P^*_{ji} are the transition probabilities for the reverse time chain. If we sum (2.1), over all i, we have

$$\begin{aligned}\sum_{i=1}^{\infty} \pi_i P_{ij} &= \pi_j \sum_{i=1}^{\infty} P^*_{ji} \\ &= \pi_j\end{aligned}$$

Summing (2.1) over all j yields

$$\sum_{j=0}^{\infty} \pi_j P^*_{ji} = \pi_i$$

Hence, the π_i represent the unique stationary distribution of both the forward and reverse time chains.

We call the chain *reversible* if for all states i and j,

$$P^*_{ij} = P_{ij}$$

When a queue is reversible, the forward and reverse time Markov chains are the same. That is, the forward time system is indistinguishable from the reverse time system. As an example, it can be verified that the Markov chain for any birth death process is reversible.

In the context of queues, there is a set of correspondences between the forward and reverse time systems. Consider a single link of the discrete time data network described in Section 2.1 Let X_t represent the state of the system at the start of slot t. In this case, $X_1, \ldots, X_k$ would be a sample path of the forward time system. The corresponding sample path in the reverse time system would be $X_k, \ldots, X_1$. That is, in reverse time, the system starts at slot k and proceeds through slots $k-1, k-2 \ldots$ In reverse time, X_t represents the state of the queue at the conclusion of slot t.

Suppose a message completes service in the forward time system at the end of slot s. At the start of slot $s+1$, the state X_{s+1} would reflect the fact that the system contains one less message. For the reverse time system, the fact that X_{s+1} has one more message than X_s would indicate that a message arrived at the start of slot s. In short, the forward time departure corresponds precisely to the reverse time arrival. Similarly, a forward time arrival is equivalent to a reverse time departure. We summarize these correspondences in the following table.

Forward Time	**Reverse Time**
Start of Slot s	End of Slot s
End of Slot s	Start of Slot s
Slot s Arrival	Slot s Departure
Slot s Departure	Slot s Arrival
Arrivals Before Slot s	Departures After Slot s
Departures After Slot s	Arrivals Before Slot s

The queues that we will consider will not be reversible. Instead, these queues will have a property defined by Kelly as quasi-reversibility. We will call a discrete time queue *quasi-reversible* if its state X_t is a stationary Markov chain such that X_t is independent of

- Arrival times of customers after time t
- Departure times of customers before time t

For the model we have chosen, the forward time arrival process is memoryless and X_t is independent of the arrival process in slots $t, t+1, \ldots$. We will examine queues for which we hypothesize that in reverse time, the state X_t is independent of the arrival process in slots $t-1, t-2, \ldots$. In this case, we can conclude that in forward time, X_t is independent of the departure process in slots $t-1, t-2, \ldots$ because the reverse time arrival process corresponds to the forward time departure process. In short, we will consider queues for which forward and reverse time are essentially indistinguishable as long as only the arrivals and departures of the system can be observed.

Our method will be to construct a system for which it will be relatively easy to guess the behavior of the reverse time system. This guess will specify a set of transition probabilities P^*_{ji}. We then will find π_i satisfying (2.1) to verify our guess.

2.3 One Link – One Session

Consider a data network with a single session using an isolated link. During each unit of time, a new message arrives with probability λ. With probability $1-\lambda$, there is no arrival. Each message has an independent integer packet length X described by $g(x) = P\{X = x\}$ and $\overline{G}(x) = P\{X > x\}$ such that $\overline{G}(0) = 1$. Given that a message has already had w units of service, we will need to know the conditional probability that the message's service requirement will be fulfilled after its next packet is sent. We denote this probability by $r(w) = P\{X = w + 1 | X > w\}$. It's easily seen that

$$r(w) = \frac{g(w+1)}{\overline{G}(w)}$$

In the transition from time t to time $t+1$, the following sequence of events occurs:

1. The message at the front of the queue has one packet sent.
2. Following service, this message will depart if all of its packets have been sent. Otherwise, the message will be rotated to the back of the queue.
3. If a new message arrives at the end of slot t, it is placed at the front of the queue to begin service at time $t + 1$.

Let $n, w_1, \ldots, w_n$ represent the state of the queue, where n is the number of messages in the system and w_i is the number of packets already sent of the message in position i. We will call this queue RR-1. Note that if the system state at the start of slot s has $w_1 = 0$, then a new message arrived and was inserted at the head of the queue at the end of slot $s - 1$. For all $i > 1$, $w_i \geq 1$ since a message cannot be rotated to the back of the queue until it has had one packet sent.

Conjecture 2 *The reverse time process is also a round robin queueing system with Bernoulli arrivals of rate λ and independent packet lengths distributed according to G. At time t, the following sequence of events occurs:*

1. *The message at the front of the queue departs if all of its packets have been sent.*
2. *If a message arrives at the start of slot t, it is inserted at the front of the queue. Otherwise, the message at the back of the queue is rotated to the front.*
3. *The message at the front of the queue has one packet sent.*

The system is still a discrete time Markov process with state $n, w_1, \ldots, w_n$ where n is the number of messages in the system and w_i is the number of packets of the message in position i that remain to be sent .

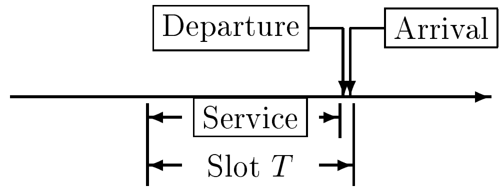

Figure 2-1: The Forward Time Sequence of Events Within a Slot

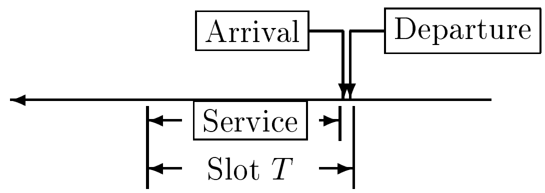

Figure 2-2: The Conjectured Reverse Time Sequence of Events Within a Slot

Note that we have chosen the round robin queue precisely so that it is possible to describe the reverse time system easily. Moreover, there is no mystery in how the reverse time queue was conjectured. In forward time, the sequence of events within a slot, as depicted in Figure 2-1, is

1. **Service** The front message is served.

2. **Departure or Rotation**. The front message departs or is rotated to the back.

3. **Arrival** If a new message arrives it is inserted at the front of the queue.

For the reverse time queue, the sequence of corresponding events occurs in the reverse order. That is, the reverse time sequence of events, as shown in Figure 2-2, is simply

1. **Departure**

2. **Arrival or (Reverse) Rotation**

3. **Service**

Note that it is vital that the new arrival in forward time receive the next unit of service since the corresponding event in reverse time consists of a message completing service and departing.

2.3.1 Proof of Conjecture 2

To verify Conjecture 2, we will use Theorem 1. Conjecture 2 specifies the reverse time transition probabilities P^*_{ij}. We will find limiting state probabilities π_i such that $\pi_i P_{ij} = \pi_j P^*_{ji}$ to prove the conjecture.

Given an arbitrary state u, we need to examine the set of transitions to neighboring states u' in the the forward time chain and the corresponding transitions from u' to u in the reverse time chain. We will call u and u' a state transition pair and will write

$$u \quad \underset{P^*_{u',u}}{\overset{P_{u,u'}}{\rightleftarrows}} \quad u'$$

The four basic transitions that we must consider are:

- The front message is served and departs. No new message arrives.

  In this instance, our state transition pair is

$$n, w_1, \ldots, w_n \quad \underset{\lambda g(w_1+1)}{\overset{(1-\lambda)r(w_1)}{\rightleftarrows}} \quad n-1, w_2, \ldots, w_n$$

  since in forward time, the probability of no new message is $1-\lambda$ and the probability of a service completion is $r(w_1)$. In reverse time, we need a new message to arrive requiring w_1+1 units of service to return to the starting state. This transition pair implies that $\pi_{n,w_1,\ldots,w_n}$ must satisfy

$$\pi_{n,w_1,\ldots,w_n}(1-\lambda)\frac{g(w_1+1)}{\overline{G}(w_1)} = \lambda g(w_1+1)\pi_{n-1,w_2,\ldots,w_n}$$

  Rewriting yields

$$\pi_{n,w_1,\ldots,w_n} = \frac{\lambda}{1-\lambda}\overline{G}(w_1)\pi_{n-1,w_2,\ldots,w_n}$$

  Repeating this process for the departure of the remaining messages in the system implies that

$$\pi_{n,w_1,\ldots,w_n} = \pi_\phi \prod_{i=1}^{n} \frac{\lambda}{1-\lambda}\overline{G}(w_i) \tag{2.2}$$

  where π_ϕ is the empty state probability. However, we must verify that this particular choice for $\pi_{n,w_1,\ldots,w_n}$ will satisfy (2.1) for the other transition pairs.

- The front message is served but does not depart. No new message arrives.

  For the reverse transition, we must not have a message arrive to ensure that the message at the rear of the queue is rotated and served. This state transition pair is

$$n, w_1, \ldots, w_n \quad \underset{1-\lambda}{\overset{(1-\lambda)[1-r(w_1)]}{\rightleftarrows}} \quad n, w_2, \ldots, w_n, w_1+1$$

  since $1-\lambda$ is the probability of no new arrival, and $1-r(w_1)$ is the conditional probability of a message not departing after w_1+1 units of service given that the message did not depart after w_1 units of service. In reverse time, the system makes the reverse transition when there is no arrival. We must check that

$$\pi_{n,w_1,\ldots,w_n}(1-\lambda)[1-r(w_1)] = (1-\lambda)\pi_{n,w_2,\ldots,w_n,w_1+1}$$

  Since $1-r(w_1)$ can be rewritten as $\overline{G}(w_1+1)/\overline{G}(w_1)$, we must have

$$\frac{\overline{G}(w_1+1)}{\overline{G}(w_1)}\left(\pi_\phi \prod_{i=1}^{n} \frac{\lambda}{1-\lambda}\overline{G}(w_i)\right) = \left(\pi_\phi \prod_{i=2}^{n} \frac{\lambda}{1-\lambda}\overline{G}(w_i)\right)\frac{\lambda}{1-\lambda}\overline{G}(w_1+1)$$

  which holds by cancellation.

- The front message is served and departs. A new message arrives.

  In reverse time, the front message departs immediately. To return to the original state, there must be a new arrival requiring exactly w_1+1 units of service. The corresponding transition pair is

$$n, w_1, \ldots, w_n \quad \underset{\lambda g(w_1+1)}{\overset{\lambda r(w_1)}{\rightleftarrows}} \quad n, 0, w_2, \ldots, w_n$$

  We must verify that

$$\lambda\frac{g(w_1+1)}{\overline{G}(w_1)}\left(\pi_\phi \prod_{i=1}^{n} \frac{\lambda}{1-\lambda}\overline{G}(w_i)\right) = \lambda g(w_1+1)\frac{\lambda}{1-\lambda}\overline{G}(0)\left(\pi_\phi \prod_{i=2}^{n} \frac{\lambda}{1-\lambda}\overline{G}(w_i)\right)$$

  which holds since $\overline{G}(0) = 1$.

- The front message is served but does not depart. A new message arrives.

For the reverse transition, the front message will immediately depart and the rear message will be rotated and served iff no new message arrives. Hence, the transitions are

$$n, w_1, \ldots, w_n \; \underset{1-\lambda}{\overset{\lambda[1-r(w_1)]}{\rightleftarrows}} \; n+1, 0, w_2, \ldots, w_n, w_1+1$$

Substituting $\overline{G}(w_1+1)/\overline{G}(w_1)$ for $1-r(w_1)$, we find that

$$\begin{aligned}
\pi_{n,w_1,\ldots,w_n} \frac{\lambda \overline{G}(w_1+1)}{\overline{G}(w_1)} &= (1-\lambda)\left(\frac{\lambda}{1-\lambda}\right)^2 \overline{G}(w_1+1)\left(\pi_\phi \prod_{i=2}^{n} \frac{\lambda}{1-\lambda}\overline{G}(w_i)\right) \\
&= (1-\lambda)\pi_{n+1,0,w_2,\ldots,w_n,w_1+1}
\end{aligned}$$

since $\overline{G}(0) = 1$.

This proves Conjecture 2.

2.3.2 The Distribution of the Number in the Queue

We can use (2.2) to find the distribution for the number in the queue. Note that

$$\overline{X} = \sum_{i=0}^{\infty} \overline{G}(i)$$

and that $\overline{G}(0) = 1$ so

$$\overline{X} - 1 = \sum_{i=1}^{\infty} \overline{G}(i)$$

Let N represent the number in the queue and $P_N(n)$ be the probability mass function for N. We note that if a message is not at the front of the queue, then at least one of its packets must have been sent. By summing over all possible states with $N \geq 1$ messages we find

$$\begin{aligned}
P_N(n) &= \sum_{w_1=0}^{\infty} \sum_{w_2=1}^{\infty} \cdots \sum_{w_n=1}^{\infty} \pi_{n,w_1,\ldots,w_n} \\
&= \pi_\phi \left(\frac{\lambda}{1-\lambda}\right)^n \left(\sum_{w_1=0}^{\infty} \overline{G}(w_1)\right)\left(\sum_{w_2=1}^{\infty} \overline{G}(w_2)\right) \cdots \left(\sum_{w_n=1}^{\infty} \overline{G}(w_n)\right) \\
&= \pi_\phi \frac{\lambda \overline{X}}{1-\lambda}\left(\frac{\lambda(\overline{X}-1)}{1-\lambda}\right)^{n-1}
\end{aligned}$$

We note that π_ϕ, the probability that the system is empty, is equal to $P_N(0)$. Applying $\sum_{n=0}^{\infty} P_N(n) = 1$ yields $\pi_\phi = 1 - \lambda\overline{X}$ and

$$P_N(n) = \begin{cases} 1 - \lambda\overline{X} & (n = 0) \\ (\lambda\overline{X})\left(1 - \dfrac{\lambda(\overline{X}-1)}{1-\lambda}\right)\left(\dfrac{\lambda(\overline{X}-1)}{1-\lambda}\right)^{n-1} & (n > 0) \end{cases} \tag{2.3}$$

We see that the queue is stable iff

$$\frac{\lambda(\overline{X}-1)}{1-\lambda} < 1$$

We note that this condition holds iff the usual condition for stability of a work conserving queue, $\lambda\overline{X} < 1$, holds.

2.3.3 Model Sensitivity

Certain aspects of the round robin queue model can be altered without changing the results in any great way. However, changing other aspects of the model destroys the reversible properties of the queue. We now will address these possibilities.

The operation of a round robin system from slot to slot requires several basic steps. We will describe these steps in the following general terms.

Service A message is served.

Departure After service, a message may depart.

Rotation A message may be rotated.

Arrival If a new message arrives, it is inserted in the queue.

The reverse time properties of the queue are very sensitive to the particular operations involved in each of these steps as well as the ordering of these steps. For many permutations of these steps, the reversibility analysis fails.

The choice of state description is also important. There are two basic ways to represent the state of the queue. We can choose the state $n, w_1, \ldots, w_n$, where w_i represents the number of packets already sent for the message in position i. The alternative is to have w_i represent the number of packets remaining to be sent.

For the queue RR-1, we chose w_i to be the service already received by message i. Had we chosen w_i to be the residual service requirement of message i, the analysis by reversibility would not have worked. We emphasize that this is difficult to see without actually writing down the possible state transitions..

One round robin queue that we can analyze orders the necessary steps in the following way.

1. **Arrival** If a new message arrives, insert it at the front of the queue.

2. **Service** The front message is served.

3. **Departure** The front message departs or is rotated.

We represent the state of the queue by $n, w_1, \ldots, w_n$, where w_i is the number of packets already sent of the message in position i. We will call this queue RR-2. For both RR-1 and RR-2 the stationary distribution is described by (2.2). However, RR-2 may be considered a more natural choice than RR-1 for a round robin queue in that a message has its first packet sent in the same slot in which the message arrives. For the corresponding event in reverse time, a message departs in the same slot in which its last packet is sent. In comparison, for RR-1 in reverse time, a message departs at the start of the slot after its last packet is sent. As a result, $w_i \geq 1$ for all i. This fact simplifies the distribution for the number in the queue. In particular, for this queue,

$$P_N(n) = \left(1 - \frac{\lambda(\overline{X} - 1)}{1 - \lambda}\right) \left(\frac{\lambda(\overline{X} - 1)}{1 - \lambda}\right)^n \qquad (n \geq 0)$$

If our intention is to examine only a single queue, RR-2 is the more straightforward one to consider.

The queue RR-1 allows the possibility that $w_1 = 0$. As we shall see, this will complicate later work. However, for the purpose of analyzing a network of queues, it will be vital that we preserve the (Service, Departure, Arrival) sequence of operations used by RR-1. Consider a network of RR-2 queues and suppose at the end of slot s, a message completes service at link l on its way to link l'. Under the (Arrival, Service, Departure) sequence of operations of RR-2, the message does not arrive at link l' until slot $s+1$. At the very start of slot $s+1$, neither the state of link l nor l' reflects the fact that a message will arrive at l' at the end of the slot. The message is effectively in limbo. In comparison, under the (Service, Departure, Arrival) approach used by RR-1, when the message leaves link l it arrives instantly at l' and the state of link l' at the start of slot $s+1$ includes the new arrival. For this reason, we will focus our attention on the (Service, Departure, Arrival) round robin queue.

Chapter 3

Multiple Message Classes

We now will analyze a somewhat more complicated system in which each message belongs to a class. The operation of this queue is the same as that of the queue RR-1 of Section 2.3. That is, the class of a message does not alter the service received by that message. If our only interest were the distribution of the number of messages in the queue, classes would be unnecessary. Message classes permit us to examine additional properties of the round robin queue.

We will denote the set of classes by $\mathcal{C}$ and we will assume that $\mathcal{C}$ is countable. During each slot, a message of class c arrives with probability λ_c. No more than one message can arrive in any slot and the probability of no message arriving during a slot is $1 - \lambda$ where $\lambda = \sum_{c \in \mathcal{C}} \lambda_c$. Each class c message has an independent integer packet length X_c described by $g_c(x) = \Pr\{X_c = x\}$ and $\overline{G}_c(x) = P\{X_c > x\}$ such that $\overline{G}_c(0) = 1$. As before, given that a class c message has already had w packets sent, the conditional probability that the message's service requirement will be fulfilled after its next packet is sent will be denoted by

$$\begin{aligned} r_c(w) &= \Pr\{X_c = w + 1 | X_c > w\} \\ &= \frac{g_c(w+1)}{\overline{G}_c(w)} \end{aligned}$$

At time t, the following sequence of events occurs:

1. The message at the front of the queue has one packet sent.

2. Following service, this message will depart if all of its packets have been sent. Otherwise, the message will be rotated to the back of the queue.

3. If a new message arrives at the end of slot t, it is placed at the front of the queue to begin service at time $t+1$.

Let $n, y_1, \ldots, y_n$ represent the state of the queue, where n is the number of messages in the system and $y_i = (w_i, c_i)$ such that for the message in position i, w_i is the number of packets already transmitted and c_i is the class of the message. Once again, we conjecture that the reverse time system is a round robin system of the same type as the forward system.

Conjecture 3 *The reverse time process is a round robin queueing system. At the start of slot t, a message of class c arrives with probability λ_c. No more than one message can arrive in any slot and the probability of no message arriving during a slot is $1-\lambda$ where $\lambda = \sum_{c=1}^{C} \lambda_c$. Each class c message has an independent service time distributed according to G_c. During slot t, the following sequence of events occurs:*

1. *The message at the front of the queue departs if all of its packets have been sent.*
2. *If a new message arrives, it is inserted at the front of the queue. Otherwise, the message at the back of the queue is rotated to the front.*
3. *The message at the front of the queue has one packet sent.*

The system is still a discrete time Markov process with the state $n, y_1, \ldots, y_n$ representing the remaining integer service demands (as well as the classes) of the messages in the system.

3.1 Proof of Conjecture 3

Of course, we will use Theorem 1 to verify Conjecture 3. This proof is virtually the same as before except we must consider the class of each message.

The four basic transitions that we must consider are:

- The front message is served and departs. No new message arrives.

  For the reverse transition, a new message of class c_1 and length w_1+1 must arrive to return to the original state. In this instance, the transition pair is

$$n, y_1, \ldots, y_n \quad \underset{\lambda_{c_1} g_{c_1}(w_1+1)}{\overset{(1-\lambda) r_{c_1}(w_1)}{\rightleftarrows}} \quad n-1, y_2, \ldots, y_n$$

This transition pair implies that $\pi_{n,y_1,\ldots,y_n}$ must satisfy

$$\pi_{n,y_1,\ldots,y_n}(1-\lambda)\frac{g_{c_1}(w_1+1)}{\overline{G}c_1(w_1)} = \lambda_{c_1} g_{c_1}(w_1+1)\pi_{n-1,y_2,\ldots,y_n}$$

Rewriting yields

$$\pi_{n,y_1,\ldots,y_n} = \frac{\lambda_{c_1}}{1-\lambda}\overline{G}_{c_1}(w_1)\pi_{n-1,y_2,\ldots,y_n}$$

Repeating this process for the departure of the remaining messages in the system implies that $\pi_{n,y_1,\ldots,y_n}$ must satisfy

$$\pi_{n,y_1,\ldots,y_n} = \pi_\phi \prod_{i=1}^{n} \frac{\lambda_{c_i}}{1-\lambda}\overline{G}_{c_i}(w_i) \tag{3.1}$$

where π_ϕ is the empty state probability. Once again, we must verify that this particular choice for $\pi_{n,y_1,\ldots,y_n}$ will satisfy (2.1) for the other transition pairs. The other transitions are:

- The front message is served but does not depart. No new message arrives.

  For the reverse transition, we must have no arrival to ensure that the message at the rear of the queue is rotated and served. This state transition pair corresponds to

$$n, y_1, \ldots, y_n \underset{1-\lambda}{\overset{(1-\lambda)[1-r_{c_1}(w_1)]}{\rightleftarrows}} n, y_2, \ldots, y_n, (w_1+1, c_1)$$

  We must check that

$$\pi_{n,y_1,\ldots,y_n}(1-\lambda)[1-r_{c_1}(w_1)] = (1-\lambda)\pi_{n,y_2,\ldots,y_n,(w_1+1,c_1)}$$

  Since $(1-r_{c_1}(w_1)$ can be rewritten as $\overline{G}_{c_1}(w_1+1)/\overline{G}_{c_1}(w_1)$, we must have

$$\frac{\overline{G}_{c_1}(w_1+1)}{\overline{G}_{c_1}(w_1)}\left(\pi_\phi \prod_{i=1}^{n} \frac{\lambda_{c_i}}{1-\lambda}\overline{G}_{c_i}(w_i)\right) = \left(\pi_\phi \prod_{i=2}^{n} \frac{\lambda_{c_i}}{1-\lambda}\overline{G}_{c_i}(w_i)\right)\frac{\lambda_{c_1}}{1-\lambda}\overline{G}_{c_1}(w_1+1)$$

  which holds by cancellation.

- The front message is served and departs. A new class c message arrives.

  In reverse time, the front message departs immediately. To return to the original state, there must be a new class c_1 arrival requiring exactly $w_1 + 1$ units of service. The corresponding transition pair is

$$n, y_1, \ldots, y_n \quad \underset{\lambda_{c_1} g_{c_1}(w_1+1)}{\overset{\lambda_c r_{c_1}(w_1)}{\rightleftarrows}} \quad n, (0, c), y_2, \ldots, y_n$$

Since $\overline{G}_c(0) = 1$, we can verify that

$$\begin{aligned} \pi_{n,y_1,\ldots,y_n} \frac{\lambda_c g_{c_1}(w_1+1)}{\overline{G}_{c_1}(w_1)} &= \lambda_{c_1} g_{c_1}(w_1+1) \frac{\lambda_c}{1-\lambda} \overline{G}_c(0) \left(\pi_\phi \prod_{i=2}^{n} \frac{\lambda_{c_i}}{1-\lambda} \overline{G}_{c_i}(w_i) \right) \\ &= \lambda_{c_1} g_{c_1}(w_1+1) \pi_{n,(0,c),y_2,\ldots,y_n} \end{aligned}$$

- The front message is served but does not depart. A new class c message arrives.

  For the reverse transition, the front message will immediately depart and the rear message will be rotated and served iff no new message arrives. Hence, the transitions are

$$n, y_1, \ldots, y_n \quad \underset{1-\lambda}{\overset{\lambda_c[1 - r_{c_1}(w_1)]}{\rightleftarrows}} \quad n+1, (0, c), y_2, \ldots, y_n, (w_1+1, c_1)$$

  Substituting $\overline{G}_{c_1}(w_1+1)/\overline{G}_{c_1}(w_1)$ for $1 - r_{c_1}(w_1)$, we observe that

$$\begin{aligned} \pi_{n,y_1,\ldots,y_n} \lambda_c[1 - r_{c_1}(w_1)] &= \lambda_c \left(\pi_\phi \prod_{i=2}^{n} \frac{\lambda_{c_i}}{1-\lambda} \overline{G}_{c_i}(w_i) \right) \frac{\lambda_{c_1}}{1-\lambda} \overline{G}_{c_1}(w_1+1) \\ &= (1-\lambda) \pi_{n+1,(0,c),y_2,\ldots,y_n,(w_1+1,c_1)} \end{aligned}$$

  since $\overline{G}_c(0) = 1$. This proves our conjecture for the multiple class case.

3.2 Properties of the Stationary Distribution

In this section, we will examine the stationary distribution of the round robin queue with classes. For some slot for which the state of the queue is described by the stationary distribution, let N represent the number of customers in the system and let $Y_i = (W_i, C_i)$ such that W_i is the number of packets already sent and C_i is the class of the message in position i. The joint distribution for $N, Y_1, \ldots, Y_N$ can be written

$$P_{N,Y_1,\ldots,Y_N}(n, y_1, \ldots, y_n) = \pi_\phi \prod_{i=1}^{n} \frac{\lambda_{c_i}}{1-\lambda} \overline{G}_{c_i}(w_i)$$

We now can derive the marginal distribution for N. Let $P_{N,C_1,\ldots,C_N}(n, c_1, \ldots, c_n)$ be the probability that the system contains n messages and the message in position i is of class c_i. Recalling that $\sum_{w=0}^{\infty} \overline{G}_c(w) = \overline{X}_c$, we find that

$$\begin{aligned}
P_{N,C_1,\ldots,C_N}(n, c_1, \ldots, c_n) &= \sum_{w_1=0}^{\infty} \sum_{w_2=1}^{\infty} \cdots \sum_{w_n=1}^{\infty} \pi_\phi \prod_{i=1}^{n} \frac{\lambda_{c_i}}{1-\lambda} \overline{G}_{c_i}(w_i) \\
&= \pi_\phi \frac{\lambda_{c_1}}{1-\lambda} \sum_{w_1=0}^{\infty} \overline{G}_{c_1}(w_1) \prod_{i=2}^{n} \left(\frac{\lambda_{c_i}}{1-\lambda} \sum_{w_i=1}^{n} \overline{G}_{c_i}(w_i) \right) \\
&= \pi_\phi \frac{\lambda_{c_1} \overline{X}_{c_1}}{1-\lambda} \prod_{i=2}^{n} \left(\frac{\lambda_{c_i}(\overline{X}_{c_i} - 1)}{1-\lambda} \right)
\end{aligned}$$

Let

$$\begin{aligned}
\rho_c &= \frac{\lambda_c(\overline{X}_c - 1)}{1-\lambda} \\
\hat{\rho}_c &= \frac{\lambda_c \overline{X}_c}{1-\lambda}
\end{aligned}$$

so that

$$P_{N,C_1,\ldots,C_N}(n, c_1, \ldots, c_n) = \pi_\phi \hat{\rho}_{c_1} \prod_{i=2}^{n} \rho_{c_i}$$

Define ρ and $\hat{\rho}$ by

$$\begin{aligned}
\rho &= \sum_{c \in \mathcal{C}} \rho_c \\
\hat{\rho} &= \sum_{c \in \mathcal{C}} \hat{\rho}_c
\end{aligned}$$

so that for $n \geq 1$,

$$\begin{aligned}
P_N(n) &= \sum_{c_1 \in \mathcal{C}} \cdots \sum_{c_n \in \mathcal{C}} P_{N,C_1,\ldots,C_N}(n, c_1, \ldots, c_n) \\
&= \pi_\phi \left(\sum_{c_1 \in \mathcal{C}} \hat{\rho}_{c_1} \right) \prod_{i=2}^{n} \sum_{c_i \in \mathcal{C}} \rho_{c_i} \\
&= \pi_\phi \hat{\rho} \rho^{n-1}
\end{aligned}$$

We can rewrite $\hat{\rho}$ as

$$\begin{aligned}
\hat{\rho} &= \sum_{c\in\mathcal{C}} \hat{\rho}_c \\
&= \sum_{c\in\mathcal{C}} \frac{\lambda_c \overline{X}_c}{1-\lambda} \\
&= \sum_{c\in\mathcal{C}} \left(\rho_c + \frac{\lambda_c}{1-\lambda}\right) \\
&= \rho + \frac{\lambda}{1-\lambda}
\end{aligned}$$

Since $\pi_\phi = P_N(0)$, applying $\sum_{n=0}^{\infty} P_N(n) = 1$ yields

$$P_N(n) = \begin{cases} (1-\rho)(1-\lambda) & (n=0) \\ (1-(1-\rho)(1-\lambda))(1-\rho)\rho^{n-1} & (n \geq 1) \end{cases} \tag{3.2}$$

Since the number in the queue is unaffected by the message classes, the distribution (3.2) should be identical to (2.3), the distribution for the number in the queue without classes. To verify this claim, we observe that

$$\begin{aligned}
\rho &= \sum_{c\in\mathcal{C}} \left(\frac{\lambda(\overline{X}-1)}{1-\lambda}\right) \\
&= \frac{\sum_{c\in\mathcal{C}} \lambda_c \overline{X}_c}{1-\lambda} - \frac{\lambda}{1-\lambda}
\end{aligned}$$

Since λ_c/λ is the conditional probability of a class c message arrival given there is an arrival, we see that

$$\overline{X} = \sum_{c\in\mathcal{C}} \frac{\lambda_c}{\lambda} \overline{X}_c$$

This implies that

$$\rho = \frac{\lambda(\overline{X}-1)}{1-\lambda}$$

just as in Section 2.3 and that

$$(1-\rho)(1-\lambda) = 1 - \lambda\overline{X}$$

We can conclude that the two distributions (2.3) and (3.2) are identical.

Now we will identify the probability that the message in position i is of class c_i.

For convenience, we let E represent the event $\{N = n, Y_j = y_j, 1 \leq j \leq n, j \neq i\}$. In this case,

$$P_{C_i,W_i|E}(c_i, w_i) = \frac{P_{N,Y_1,\ldots,Y_N}(n, y_1, \ldots, y_n)}{P\{E\}}$$

where

$$P\{E\} = \pi_\phi \left(\prod_{j \neq i} \frac{\lambda_{c_j}}{1-\lambda} \overline{G}_{c_j}(w_j) \right) \sum_{c_i \in \mathcal{C}} \frac{\lambda_{c_i}}{1-\lambda} \sum_{w_i} \overline{G}_{c_i}(w_i)$$

As always, the summation over w_i depends on i. When $i = 1$ the sum starts at $w_i = 0$ while for $i > 1$, the sum starts at $w_i = 1$. Consequently,

$$P\{E\} = \begin{cases} \pi_\phi \left(\prod_{j \neq i} \frac{\lambda_{c_j}}{1-\lambda} \overline{G}_{c_j}(w_j) \right) \hat{\rho} & i = 1 \\ \pi_\phi \left(\prod_{j \neq i} \frac{\lambda_{c_j}}{1-\lambda} \overline{G}_{c_j}(w_j) \right) \rho & i > 1 \end{cases}$$

This implies that

$$P_{C_i,W_i|E}(c_i, w_i) \begin{cases} \frac{1}{\hat{\rho}} \frac{\lambda_{c_i}}{1-\lambda} \overline{G}_{c_i}(w_i) & i = 1 \\ \frac{1}{\rho} \frac{\lambda_{c_i}}{1-\lambda} \overline{G}_{c_i}(w_i) & i > 1 \end{cases} \tag{3.3}$$

The event that $C_i = c_i$ and $W_i = w_i$ is independent of the event E. That is, given $N \geq i$, the probability that $C_i = c_i$ and $W_i = w_i$ is independent of N and the set $\{Y_j | j \neq i\}$. By summing (3.3) over all w_i, we find that

$$P_{C_i}(c_i) = \begin{cases} \hat{\rho}_{c_i}/\hat{\rho} & i = 1 \\ \rho_{c_i}/\rho & i > 1 \end{cases}$$

We can also find the marginal distribution for the number in some subset of the set of classes. Let $\mathcal{C}'$ be some subset of the set of classes $\mathcal{C}$. Our objective will be to identify the distribution of N', the number of messages with class $c \in \mathcal{C}'$. As a function of the state $N, Y_1, \ldots, Y_N$, define

$$T_i = \begin{cases} 1 & \text{if } C_i \in \mathcal{C}' \\ 0 & \text{otherwise} \end{cases}$$

for $1 \leq i \leq n$.

From (3.3), $T_1, \ldots, T_N$ are independent, conditional on N, and the distribution of T_i is independent of N for $N \geq i$. Furthermore, $T_2, \ldots, T_N$ are identically distributed. That is,

$$\begin{aligned} P\{T_i = 1\} &= \sum_{c_i \in \mathcal{C}'} P_{C_i}(c_i) \\ &= \begin{cases} \dfrac{\sum_{c_i \in \mathcal{C}'} \hat{\rho}_{c_i}}{\hat{\rho}} & i = 1 \\ \dfrac{\sum_{c_i \in \mathcal{C}'} \rho_{c_i}}{\rho} & i > 1 \end{cases} \end{aligned}$$

Designating

$$\begin{aligned} \rho' &= \sum_{c \in \mathcal{C}'} \rho_c \\ \hat{\rho}' &= \sum_{c \in \mathcal{C}'} \hat{\rho}_c \end{aligned}$$

allows us to write

$$P\{T_i = 1\} = \begin{cases} \hat{\rho}'/\hat{\rho} & i = 1 \\ \rho'/\rho & i > 1 \end{cases}$$

In the space where $N \geq 1$, define

$$\hat{N} = N - 1$$

$$\hat{M} = T_2 + \cdots + T_N$$

Conditional on $N \geq 1$, $\hat{M}$ equals the number of messages belonging to classes in $\mathcal{C}'$ excluding the message in position 1. The distribution for $\hat{N}$ is

$$\begin{aligned} P_{\hat{N}}(n) &= P_{N|N\geq 1}(n+1) \\ &= (1-\rho)\rho^n \qquad (n \geq 0) \end{aligned}$$

For a nonnegative integer valued random variable X, we define the z transform of X by $X(z) = E[z^X]$. In this case, for $i > 1$, T_i has the z transform

$$T(z) = 1 - \frac{\rho'}{\rho} + \left(\frac{\rho'}{\rho}\right) z$$

In addition

$$\hat{N}(z) = \frac{1-\rho}{1-\rho z}$$

Since $\hat{M}$ is the sum of $\hat{N}$ independent identically distributed random variables each described by $T(z)$, we have that

$$\hat{M}(z) = \hat{N}(T(z)) = \frac{1 - \frac{\rho'}{1-\rho+\rho'}}{1 - \frac{\rho'}{1-\rho+\rho'} z}$$

Comparison of $\hat{M}(z)$ with $\hat{N}(z)$ allows to conclude that $\hat{M}$ is geometrically distributed and

$$P_{\hat{M}}(m) = \left(1 - \frac{\rho'}{1-\rho+\rho'}\right)\left(\frac{\rho'}{1-\rho+\rho'}\right)^m \qquad (m \geq 0)$$

We recall that N' is the number of messages in the system with class $c \in \mathcal{C}'$. Note that N' differs from $\hat{M}$ when the message in position 1 belongs to $\mathcal{C}'$. Furthermore, $\hat{M}$ is defined only in the space conditioned by the event $N \geq 1$. We will be able to easily derive $P_{N'}(n)$. Given $N \geq 1$, T_1 and $\hat{M}$ are independent. This implies

$$\begin{aligned}
P_{N'|N\geq 1}(n) &= P\{T_1 = 1 | N \geq 1\} P_{\hat{M}}(n-1) + P\{T_1 = 0 | N \geq 1\} P_{\hat{M}}(n) \\
&= \frac{\hat{\rho}'}{\hat{\rho}} P_{\hat{M}}(n-1) + \left(1 - \frac{\hat{\rho}'}{\hat{\rho}}\right) P_{\hat{M}}(n) \\
&= \left(\frac{\hat{\rho}'(1-\rho) + \rho'\hat{\rho}}{\hat{\rho}(1-\rho+\rho')}\right)\left(1 - \frac{\rho'}{1-\rho+\rho'}\right)\left(\frac{\rho'}{1-\rho+\rho'}\right)^{n-1} \qquad (3.4)
\end{aligned}$$

We observe that

$$\begin{aligned}
\hat{\rho}' &= \sum_{c\in\mathcal{C}'} \hat{\rho}_c \\
&= \sum_{c\in\mathcal{C}'} \left(\rho_c + \frac{\lambda_c}{1-\lambda}\right) \\
&= \rho' + \frac{\lambda'}{1-\lambda}
\end{aligned}$$

where

$$\lambda' = \sum_{c\in\mathcal{C}'} \lambda_c$$

Substitution of $\hat{\rho}$ from (3.2) and $\hat{\rho}'$ into (3.4) yields for $n \geq 0$

$$P_{N'|N\geq 1}(n) = \frac{\rho' + \lambda' - \rho\lambda'}{(\rho(1-\lambda)+\lambda)\,(1-\rho+\rho')}\left(1 - \frac{\rho'}{1-\rho+\rho'}\right)\left(\frac{\rho'}{1-\rho+\rho'}\right)^{n-1}$$

Consequently, for $n \geq 1$, we find that

$$\begin{aligned} P_{N'}(n) &= P\{N \geq 1\} P_{N'/N\geq 1}(n) \\ &= \left(1 - \frac{(1-\rho)(1-\lambda')}{1-\rho+\rho'}\right)\left(1 - \frac{\rho'}{1-\rho+\rho'}\right)\left(\frac{\rho'}{1-\rho+\rho'}\right)^{n-1} \end{aligned}$$

Applying $\sum_{n=0}^{\infty} P_{N'}(n) = 1$ yields

$$P_{N'}(n) = \begin{cases} \dfrac{(1-\rho)(1-\lambda')}{1-\rho+\rho'} & (n = 0) \\ \left(1 - \dfrac{(1-\rho)(1-\lambda')}{1-\rho+\rho'}\right)\left(1 - \dfrac{\rho'}{1-\rho+\rho'}\right)\left(\dfrac{\rho'}{1-\rho+\rho'}\right)^{n-1} & (n \geq 1) \end{cases} \tag{3.5}$$

Ignoring the peculiarity associated with the message in position 1, we see that the distribution of N' is essentially the same as that for N with ρ replaced by $\frac{\rho'}{1-\rho+\rho'}$. This is simply a consequence of the fact that a geometric sum of Bernoulli random variables is still geometric. We will use the distribution of N' when we consider networks of round robin queues as well as in the next section.

3.3 System Time and Message Length

We can use message classes to examine the average system time of a message as a function of the message length. In particular, for $\tau = 1, 2, \ldots$ we can define class τ to be the set of length τ messages and N_τ to be the number of class τ messages in the system. If we let $\mathcal{C}'$ equal the set of messages of length τ, then N' equals N_τ.

From (3.5), we can compute $E[N']$ which happens to be

$$E[N'] = \frac{\rho'}{1-\rho} + \lambda'$$

We observe that λ' is the probability of length τ message arrival, which we denote by

λ_τ so that

$$\rho' = \frac{\lambda_\tau(\tau - 1)}{1 - \lambda}$$

As always,

$$\rho = \frac{\lambda(\overline{X} - 1)}{1 - \lambda}$$

Since $E[N_\tau] = E[N']$, we can conclude that

$$E[N_\tau] = \frac{\lambda_\tau(\tau - 1)}{1 - \lambda\overline{X}} + \lambda_\tau$$

We denote the system time of a length τ message by $V(\tau)$. From Little's Law, we know that $E[V(\tau)]$ satisfies $E[V(\tau)] = E[N_\tau]/\lambda_\tau$. Thus,

$$E[V(\tau)] = \frac{\tau - 1}{1 - \lambda\overline{X}} + 1$$

We see that the system time of a message grows linearly with the message length. This is very desirable in that short messages will have small delay. This verifies that the round robin queue does provide superior service to short messages.

Chapter 4

Delay Analysis for the Round Robin Queue

We now will analyze the system time of a message as a function of the message length. We find that the system time can be reduced to a sum of independent random variables. This work is based on a 'decomposition into independent delay elements' described by Yashkov in an analysis of the system time under processor sharing; see [Yas83]. Yashkov found an expression for the system time distribution for a typical customer. We will restrict our attention to the first and second moments of system time under round robin.

We will consider a single round robin queue without classes as in Chapter 2. For our purposes, it will be useful to consider what we have previously called the reverse time system in which the state describes the residual work requirements of the queued messages. The system time of a message is the same in forward or reverse time so that it makes no difference which model we use.

To review the model, we will examine the following queue. During each unit of time, a new message arrives with probability λ; otherwise, no new message arrives. The length in packets of the message is an independent integer random variable X described by $P\{X = x\} = g(x)$ and $P\{X > x\} = \overline{G}(x)$. In the transition from time t to time $t + 1$, the following sequence of events occurs:

1. The message at the front of the queue departs if all of its packets have been sent.

2. If a message arrives at the start of slot t, it is inserted at the front of the queue. Otherwise, the message at the back of the queue is rotated to the front.

3. The message at the front of the queue has one packet sent.

Let $n, w_1, \ldots, w_n$ represent the state of the queue where n is the number of messages in the system and w_i is the number of packets of the message in position i that remain to be sent.

Note that we do not need to consider the more complicated queue in which the messages are assigned to classes. Since the round robin service discipline ignores message classes, the system time of a message of a given class is unaffected by its class. Moreover, given a queue with classes $c_1, c_2, \ldots$, we can consider the system time distribution of this queue by examining a queue without classes in which $\lambda = \sum_i \lambda_{c_i}$ and $g(x) = \sum_i \frac{\lambda_{c_i}}{\lambda} g_{c_i}(x)$. Hence, for the purpose of analyzing delay, it is unnecessary to consider the more complicated system with classes.

We will examine $V(\tau, n, w_1, \ldots, w_n)$, the system time of a length τ message that upon arrival finds the system state to be $n, w_1, \ldots, w_n$. In addition, we will consider $V(\tau)$, the system time of a length τ message that upon arrival finds the system state described by the stationary distribution.

4.1 Decomposition of the System Time

Suppose that during some slot s for which the state of the queue is $n, w_1, \ldots, w_n$, we mark the arrival of a message of length τ. This new message immediately has its first packet transmitted. At the start of slot $s+1$, the state of the system then is $n+1, \tau-1, w_1, \ldots, w_n$, assuming that $w_1 > 0$. Suppose that the remaining packets of the marked message are transmitted in slots $s_{\tau-1} < s_{\tau-2} < \ldots < s_1$. Define cycle $\tau - 1$ as the interval of slots $[s+1, s_{\tau-1}]$. Similarly, for $1 \leq i < \tau - 1$, let cycle i be the slot interval $[s_{i+1}+1, s_i]$. Cycles $\tau-1, \tau-2, \ldots, 1$ will occur before the marked message departs. The last packet of the marked message departs at the end of cycle 1. Moreover, the system time of the marked message will be equal to one plus the total number of packets transmitted during the $\tau - 1$ cycles.

If a packet is to be transmitted during cycle t, we say that the packet belongs to cycle t or that it is a cycle t packet. In addition, we will say that an individual packet is on deck when that packet is the next backlogged packet to be transmitted. The on deck packet will be the next packet transmitted unless there is a new arrival that preempts its turn. Even if the on deck packet is preempted however, it will remain the next backlogged packet to be sent.

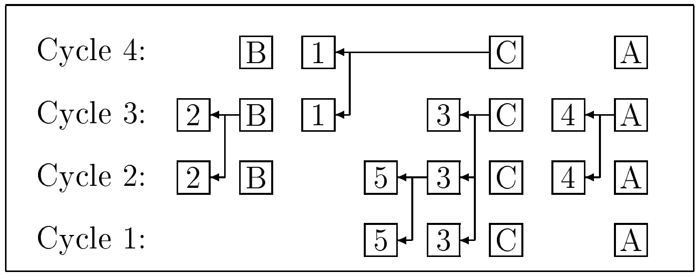

Figure 4-1: The System Time of a Marked Message

We represent packets by boxes, messages by columns of boxes, and cycles by rows. Within each cycle, successive packet transmissions occur from left to right. The 5 packet marked message **A** had its first packet transmitted and has 4 packets remaining. Messages **B** and **C** were in the queue at the time of the arrival of **A**. Message **C** may have more than 4 packets left. Subsequent message arrivals are labelled $1, \ldots, 5$. Arrows represent the descendency among packets. For example, both packets of message **1** are descendents of the cycle 4 packet of message **C** while the first three packets of message **3** and the first two packets of message **5** are descendents of cycle 3 packet of message **C**.

Suppose that when a cycle t packet is on deck, it is preempted by a new arrival of length y. The new arrival has its first packet transmitted immediately and its remaining $y-1$ packets inserted in the queue. In this case, we will say that each of the arriving packets that is to be sent before the marked message departs is a *descendent* of the preempted packet. Furthermore if packet p'' is a descendent of packet p' and p' is a descendent of packet p, then we will consider p'' to be a descendent of p as well. Clearly, the first packet of the new arrival becomes a descendent of the preempted packet. If $y < t$, then $y-1$ additional descendents are generated; otherwise, $t-1$ more descendents are created. Each of these descendents may generate further descendents, each of which causes a one slot delay for the marked message.

We define D_t, the delay generated by a cycle t packet, as the number of slots necessary to transmit the cycle t packet itself as well as all of its descendent packets. Given the cycle number t, the number of descendents created does not depend on τ, the length of the original marked message. This permits us to discuss the delay generated by a cycle t packet without reference to a particular message that is subject to this delay. For this reason, we have chosen to number the cycles $\tau-1, \tau-2, \ldots, 1$ as opposed to $1, 2, \ldots, \tau-1$.

Consider two packets p and $\hat{p}$, neither a descendent of the other, belonging to cycles t and $\hat{t}$. These packets generate delays D_t and $\hat{D}_{\hat{t}}$. We argue that D_t and $\hat{D}_{\hat{t}}$ are independent. To see this, let S_c represent the set of slots in cycle c during which descendents of p arrived. Likewise, suppose that the descendents of $\hat{p}$ that arrive during cycle $\hat{c}$ arrive during the set of slots $\hat{S}_{\hat{c}}$. The sets S_c and $\hat{S}_{\hat{c}}$ are disjoint since p and $\hat{p}$ can only have descendents in common if one is a descendent of the other. If we define $A(s, c)$ as the number of descendents that arrive during slot s of cycle c, then

$$\begin{aligned} D_t &= 1 + \sum_{c=0}^{t-1} \sum_{s \in S_{t-c}} A(s, t-c) \\ \hat{D}_{\hat{t}} &= 1 + \sum_{c=0}^{\hat{t}-1} \sum_{\hat{s} \in \hat{S}_{\hat{t}-c}} A(\hat{s}, \hat{t}-c) \end{aligned}$$

For $s \in S_c$ and $\hat{s} \in \hat{S}_{\hat{c}}$, $A(s, c)$ and $A(\hat{s}, \hat{c})$ are independent because the number of descendents created during a slot depends only on the arrival process and the current cycle number. This implies that D_t and $\hat{D}_{\hat{t}}$ must also be independent.

Upon the arrival of the marked message, each packet currently in the system initiates an independent process similar to a branching process. In a branching process however, the n immediate descendents of a parent all occur in the same generation. In comparison, for the D_t process, a packet that has n immediate descendents will have one descendent in each of the next n generations. This difference will greatly complicate the analysis.

Now we will describe the system time of the marked message. Consider a message that has w packets remaining to be sent when the marked message arrives. This message will have a packet to transmit in cycles $\tau - 1$, $\tau - 2$ and so on until either all of its packets have been sent or until the marked message completes service. The total contribution of this message to the system time of the marked message equals

$$D_\tau(w) = \sum_{j=1}^{\min(w, \tau-1)} D_{\tau-j} \tag{4.1}$$

Consequently, the delay contribution of the message in position i is $D_\tau(w_i)$. It should be apparent that $D_\tau(w_i)$ and $D_\tau(w_j)$ are independent since they are sums of random variables that are independent of one another. Furthermore, the system time of the marked message equals

$$V(\tau, n, w_1, \dots, w_n) = 1 + D_\tau(\tau - 1) + \sum_{i=1}^{n} D_\tau(w_i) \tag{4.2}$$

The first term corresponds to the slot it takes to transmit the first packet of the marked message. The second term is the delay contribution of the remaining packets of the marked message. Lastly, term i of the summation corresponds to the delay contributed by the message in position i. It will become apparent that we will not be able to easily derive the distribution of $V(\tau, n, w_1, \dots, w_n)$. However, this decomposition of $V(\tau, n, w_1, \dots, w_n)$ into a sum of independent random variables will allow us to examine its mean and variance in terms of the first and second moments of D_t. We now will describe a method to evaluate the necessary moments of D_t before continuing with our analysis of $V(\tau, n, w_1, \dots, w_n)$.

4.2 The first and second moments of D_t

In this section, we will identify $\overline{D_t}$ and $\overline{D_t^2}$ for all t. To do so, we will need to define

$$I_t(j) = \begin{cases} 1 & (0 \le j \le t) \\ 0 & \text{otherwise} \end{cases}$$

and for any random variable Y,

$$\overline{F}_Y(y) = P\{Y \ge y\}$$

We will make great use of of certain properties of $I_t(j)$. In particular,

$$E[I_Y(j)] = \overline{F}_Y(j) \tag{4.3}$$

$$E[I_Y(j)I_Y(k)] = \begin{cases} \overline{F}_Y(j) & k \le j \\ \overline{F}_Y(k) & k > j \end{cases} \tag{4.4}$$

4.2.1 Properties of $\overline{D}_t$

Consider a packet p that is at the front of the queue at the start of slot s of cycle t. With probability $1-\lambda$, no arrival occurs in slot s and packet p is sent. Otherwise, an arrival of length y occurs with probability

$$\lambda_y = \lambda g(y)$$

In this case, one packet of the new message is sent and $\min(y-1, t-1)$ additional descendents of p are generated. These descendent packets will belong to cycles $t-1, t-2, \ldots$ and will generate delays described by $D_{t-1}, D_{t-2}, \ldots$ In addition, packet p stays at the front of the queue and will generate another D_t units of delay in the next slot. If we let Y represent the number of packets that arrive in the slot, then we can write

$$D_t = 1 + \sum_{y=0}^{t-1} I_Y(y+1) D_{t-y} \tag{4.5}$$

Note that $Y = 0$ when no message arrives; otherwise, Y is the length of the arriving message. As a result,

$$P_Y(y) = \begin{cases} 1-\lambda & y = 0 \\ \lambda g(y) & y > 0 \end{cases}$$

In addition, for $y \geq 0$,

$$\begin{aligned} E[I_Y(y+1)] &= \overline{F}_Y(y+1) \\ &= \sum_{j=y+1}^{\infty} \lambda g(j) \\ &= \lambda \overline{G}(y) \end{aligned} \tag{4.6}$$

Since Y, the number of packets that arrives in a slot, is independent of the amount of additional work generated by any one of those arriving packets in a future slot, we can conclude that

$$\begin{aligned} \overline{D}_t &= 1 + \sum_{y=0}^{t-1} E[I_Y(y+1)] E[D_{t-y}] \\ &= 1 + \lambda \overline{G}(0) \overline{D}_t + \sum_{y=1}^{t-1} \lambda \overline{G}(y) \overline{D}_{t-y} \end{aligned} \tag{4.7}$$

Since $\overline{G}(0) = 1$, solving for $\overline{D}_t$ yields

$$\overline{D}_t = \frac{1}{1-\lambda}\left(1 + \lambda \sum_{y=1}^{t-1} \overline{G}(y)\overline{D}_{t-y}\right) \tag{4.8}$$

This allows us to easily compute $\overline{D}_t$ for all t.

Recall that D_t represents the work generated by a cycle t packet and all of its descendents in the next t cycles. Suppose a cycle t packet is preempted by the arrival of of a new message. For sufficiently large t, all of the packets of the new arrival become descendents of the cycle t packet. Furthermore, as t increases, it becomes increasingly likely that the descendent process generated by the cycle t packet will end in less than t cycles. The latter statement is true simply because the busy period terminates with probability one. Consequently, as t increases, D_t should converge to a random variable D. We merely verify that the mean of D_t converges to a quantity that we call $\overline{D}$.

Theorem 4 *Let*

$$\overline{D} = \frac{1}{1-\lambda\overline{\overline{X}}}$$

then $\overline{D}_1, \overline{D}_2, \ldots$ *is monotonically increasing sequence that converges to* $\overline{D}$.

Proof: Note that $\overline{D}_1 = 1/(1-\lambda)$. This implies

$$\begin{aligned} \overline{D}_2 - \overline{D}_1 &= \frac{\lambda\overline{G}(1)\overline{D}_1}{1-\lambda} \\ &\geq 0 \end{aligned}$$

Suppose that $\overline{D}_1 \leq \overline{D}_2 \leq \cdots \leq \overline{D}_t$. In this case,

$$\begin{aligned} \overline{D}_{t+1} - \overline{D}_t &= \frac{\lambda}{1-\lambda}\left(\overline{G}(t)\overline{D}_1 + \sum_{y=1}^{t-1}\overline{G}(y)(\overline{D}_{t+1-y} - \overline{D}_{t-y})\right) \\ &\geq 0 \end{aligned}$$

Hence, the sequence is increasing. Now we will verify that the sequnce is bounded above by $\overline{D}$. Since $\overline{X} \geq 1$, we see that $\overline{D}_1 \leq 1/(1-\lambda\overline{X})$. Suppose that for all $j < t$, we have $\overline{D}_j \leq 1/(1-\lambda\overline{X})$. In this case,

$$\overline{D}_t = \frac{1}{1-\lambda}\left(1 + \lambda\sum_{j=1}^{t-1}\overline{G}(j)\overline{D}_{t-j}\right)$$

$$\begin{aligned} &\leq \frac{1}{1-\lambda}\left(1+\frac{\lambda}{1-\lambda\overline{X}}\sum_{j=1}^{t-1}\overline{G}(j)\right) \\ &\leq \frac{1}{1-\lambda}\left(1+\frac{\lambda(\overline{X}-1)}{1-\lambda\overline{X}}\right) \\ &= \frac{1}{1-\lambda\overline{X}} \end{aligned}$$

Consequently, the sequence $\overline{D}_t$ converges. We now verify that $\overline{D}_t$ converges to $\overline{D}$. Define $\epsilon_t = \overline{D} - \overline{D}_t$. We observe that the sequence ϵ_t is nonnegative and monotonically decreasing. In addition, from (4.8), we can write

$$\epsilon_t = \frac{\lambda}{1-\lambda}\left(\frac{\overline{X} - \sum_{y=0}^{t-1}\overline{G}(y)}{1-\lambda\overline{X}} - \sum_{y=1}^{t-1}\overline{G}(y)\epsilon_{t-y}\right)$$

Since $\epsilon_t \geq 0$ for all t,

$$\epsilon_t \leq \frac{\lambda(\overline{X} - \sum_{y=0}^{t-1}\overline{G}(y)}{(1-\lambda)(1-\lambda\overline{X})}$$

Recalling that $\sum_{y=0}^{\infty}\overline{G}(y) = \overline{X}$, we see that $\lim_{t\to\infty}\epsilon_t \leq 0$. Since $\epsilon_t \geq 0$, it must be that $\lim_{t\to\infty}\epsilon_t = 0$. Hence,

$$\lim_{t\to\infty}\overline{D}_t = \overline{D}$$

This completes the proof of Theorem 4. We will make use of this result in Subsection 4.4.3.

4.2.2 The second moment of D_t

We can find the second moment of D_t by a similar approach. From (4.5),

$$\begin{aligned} \overline{D_t^2} &= E\left[\left(1+\sum_{j=0}^{t-1}I_Y(j+1)D_{t-j}\right)^2\right] \\ &= E\left[\left(1+I_Y(1)D_t+\sum_{j=1}^{t-1}I_Y(j+1)D_{t-j}\right)^2\right] \\ &= E\left[(1+I_Y(1)D_t)^2+2(1+I_Y(1)D_t)\sum_{j=1}^{t-1}I_Y(j+1)D_{t-j}+\left(\sum_{j=1}^{t-1}I_Y(j+1)D_{t-j}\right)^2\right] \end{aligned}$$

We apply (4.4) and (4.6) to the summation yielding

$$\overline{D_t^2} = 1 + 2\lambda\overline{D}_t + \lambda\overline{D_t^2} + 2(1+\overline{D}_t)\sum_{j=1}^{t-1}\lambda\overline{G}(j)\overline{D}_{t-j} + E\left[\left(\sum_{j=1}^{t-1} I_Y(j+1)D_{t-j}\right)^2\right]$$

From (4.8), we note that

$$\lambda\sum_{k=1}^{t-1}\overline{G}(k)\overline{D}_{t-k} = (1-\lambda)\overline{D}_t - 1$$

This implies that

$$\overline{D_t^2} = 2(\overline{D}_t)^2 - \frac{1}{1-\lambda} + \frac{1}{1-\lambda}E\left[\left(\sum_{j=1}^{t-1} I_Y(j+1)D_{t-j}\right)^2\right]$$

From (4.4) and (4.6), we observe that

$$\begin{aligned}
&E\left[\left(\sum_{j=1}^{t-1} I_Y(j+1)D_{t-j}\right)^2\right] \\
&= E\left[\left(\sum_{j=1}^{t-1} I_Y(j+1)D_{t-j}\right)\left(\sum_{k=1}^{t-1} I_Y(k+1)D_{t-k}\right)\right] \\
&= \sum_{j=1}^{t-1}\sum_{k=1}^{t-1} E[I_Y(j+1)I_Y(k+1)]E[D_{t-j}D_{t-k}] \\
&= \sum_{j=1}^{t-1}\left(\overline{G}(j)\overline{D}_{t-j}\sum_{k=1}^{j-1}\overline{D}_{t-k} + \overline{G}(j)\overline{D_{t-j}^2} + \overline{D}_{t-j}\sum_{k=j+1}^{t-1}\overline{G}(k)\overline{D}_{t-k}\right)
\end{aligned}$$

We can conclude that

$$\begin{aligned}
\overline{D_t^2} &= 2(\overline{D}_t)^2 - \frac{1}{1-\lambda} \\
&\quad + \frac{1}{1-\lambda}\sum_{j=1}^{t-1}\left(\overline{G}(j)\overline{D}_{t-j}\sum_{k=1}^{j-1}\overline{D}_{t-k} + \overline{G}(j)\overline{D_{t-j}^2} + \overline{D}_{t-j}\sum_{k=j+1}^{t-1}\overline{G}(k)\overline{D}_{t-k}\right) \quad (4.9)
\end{aligned}$$

This permits us to compute recursively $\overline{D_t^2}$ for all t. Henceforth, we will consider $\overline{D}_t$ and $\overline{D_t^2}$ to be known for all necessary values of t.

4.3 The Mean and Variance of $V(\tau, n, w_1, \ldots, w_n)$

We now continue with our examination of $V(\tau, n, w_1, \ldots, w_n)$, the system time of a length τ message that arrives to find the state of the queue to be $n, w_1, \ldots, w_n$. Taking the expectation over (4.2), we see that

$$\overline{V(\tau, n, w_1, \ldots, w_n)} = 1 + \overline{D_\tau(\tau - 1)} + \sum_{i=1}^{n} \overline{D_\tau(w_i)} \tag{4.10}$$

Using (4.1), we can restate $D_t(y)$ as

$$D_t(y) = \sum_{j=1}^{t-1} I_y(j) D_{t-j} \tag{4.11}$$

This implies that

$$\overline{D_\tau(w_i)} = \sum_{j=1}^{\tau-1} I_{w_i}(j) \overline{D}_{\tau-j}$$

Furthermore, from (4.10), we see that

$$\begin{aligned}
\overline{V(\tau, n, w_1, \ldots, w_n)} &= 1 + \sum_{j=1}^{\tau-1} \overline{D}_{\tau-j} + \sum_{i=1}^{n} \sum_{j=1}^{\tau-1} I_{w_i}(j) \overline{D}_{\tau-j} \\
&= 1 + \sum_{j=1}^{\tau-1} \overline{D}_{\tau-j} + \sum_{j=1}^{\tau-1} \overline{D}_{\tau-j} \sum_{i=1}^{n} I_{w_i}(j) \\
&= 1 + \sum_{j=1}^{\tau-1} \left(1 + \sum_{i=1}^{n} I_{w_i}(j)\right) \overline{D}_{\tau-j} \qquad (4.12)
\end{aligned}$$

Since $V(\tau, n, w_1, \ldots, w_n)$ is a sum of independent random variables, we can write

$$\sigma^2_{V(\tau,n,w_1,\ldots,w_n)} = \sigma^2_{D_\tau(\tau-1)} + \sum_{i=1}^{n} \sigma^2_{D_\tau(w_i)}$$

By (4.11), we see that $D_t(y)$ is a sum of independent random variables so that

$$\sigma^2_{D_t(y)} = \sum_{j=1}^{t-1} I_y(j) \sigma^2_{D_{t-j}}$$

As a result,

$$\sigma^2_{V(\tau,n,w_1,\ldots,w_n)} = \sum_{j=1}^{\tau-1} \sigma^2_{D_{\tau-j}} + \sum_{i=1}^{n} \sum_{j=1}^{\tau-1} I_{w_i}(j) \sigma^2_{D_{\tau-j}}$$

$$= \sum_{j=1}^{\tau-1} \left(1 + \sum_{i=1}^{n} I_{w_i}(j)\right) \sigma^2_{D_{\tau-j}} \tag{4.13}$$

Note that $\sum_{i=1}^{n} I_{w_i}(j)$ is simply the number of messages in the system, at the time of the arrival of the marked message, with residual length of j or more packets. Since we have shown how to compute the moments of D_t, given the arrival rate and the message length distribution, it is possible to evaluate the mean and variance of the system time of a message that arrives to find the queue in a particular state.

We emphasize that these expressions for the system time moments rely on the fact that the state $w_1, \ldots, w_n$ represents the residual system times of the queued messages. If w_i represented the number of packets already sent of the message in position i, as in the forward time queue of Chapter 2, then (4.12) and (4.13) would not describe the mean and variance of the system time of a message that arrives to find a state $n, w_1, \ldots, w_n$.

4.4 The System Time of a Typical Message of Length τ

It perhaps would be of greater interest to examine $V(\tau)$, the system time of a length τ message that upon arrival finds the state of the system described by the stationary distribution. We note that the results we derive for $V(\tau)$ would be correct in either the forward and reverse time system since the arrival process in either system is independent of the state of the queue.

Let $N, W_1, \ldots, W_N$ represent the state of the queue at the start of the slot in which the new message arrives. The joint distribution for $N, W_1, \ldots, W_N$ is the stationary distribution for the system which we recall from Section 2.3 to be

$$P_{N,W_1,\ldots,W_N}(n, w_1, \ldots, w_n) = (1 - \lambda\overline{X}) \left(\frac{\lambda}{1-\lambda}\right)^n \prod_{i=1}^{n} \overline{G}(w_i) \tag{4.14}$$

The set of feasible states for the stationary distribution allows $w_1 = 0$, yet, for all $i > 1$, we must have $w_i \geq 1$. Suppose $N, W_1, \ldots, W_N$ has the form $n+1, 0, w_1, \ldots, w_n$. The front message which has zero residual service requirement departs at the beginning of the slot, the instant before the new message arrives. The arriving message actually finds only the remaining n messages with residual requirements $w_1, \ldots, w_n$. It is only these n messages that contribute to the delay of the new arrival. Consequently, we will need to consider only the state of the queue the instant after the possible departure. We denote the state of the queue at this instant by $\tilde{N}, \tilde{W}_1, \ldots, \tilde{W}_{\tilde{N}}$. and we observe that we must have $\tilde{W}_i > 0$ for all i. The state $\tilde{N}, \tilde{W}_1, \ldots, \tilde{W}_{\tilde{N}}$ will be $\tilde{n}, \tilde{w}_1, \ldots, \tilde{w}_{\tilde{n}}$ iff $N, W_1, \ldots, W_N$ is either $\tilde{n}, \tilde{w}_1, \ldots, \tilde{w}_{\tilde{n}}$ or $\tilde{n}+1, 0, \tilde{w}_1, \ldots, \tilde{w}_{\tilde{n}}$. This implies that

$$\begin{aligned}
P_{\tilde{N},\tilde{W}_1,\ldots,\tilde{W}_{\tilde{N}}}(\tilde{n}, \tilde{w}_1, \ldots, \tilde{w}_{\tilde{n}}) &= P_{N,W_1,\ldots,W_N}(\tilde{n}, \tilde{w}_1, \ldots, \tilde{w}_{\tilde{n}}) \\
&\quad + P_{N,W_1,\ldots,W_N}(\tilde{n}+1, 0, \tilde{w}_1, \ldots, \tilde{w}_{\tilde{n}})
\end{aligned}$$

Applying (4.14), we find that

$$P_{\tilde{N},\tilde{W}_1,\ldots,\tilde{W}_{\tilde{N}}}(\tilde{n}, \tilde{w}_1, \ldots, \tilde{w}_{\tilde{n}}) = \left(\frac{1-\lambda\overline{X}}{1-\lambda}\right)\left(\frac{\lambda}{1-\lambda}\right)^{\tilde{n}} \prod_{i=1}^{\tilde{n}} \overline{G}(\tilde{w}_i)$$

We can derive the marginal distribution for $\tilde{N}$ as follows.

$$\begin{aligned}
P_{\tilde{N}}(\tilde{n}) &= \sum_{\tilde{w}_1=1}^{\infty} \cdots \sum_{\tilde{w}_{\tilde{n}}=1}^{\infty} P_{\tilde{N},\tilde{W}_1,\ldots,\tilde{W}_{\tilde{N}}}(\tilde{n}, \tilde{w}_1, \ldots, \tilde{w}_{\tilde{n}}) \\
&= \left(\frac{1-\lambda\overline{X}}{1-\lambda}\right)\left(\frac{\lambda}{1-\lambda}\right)^{\tilde{n}} \prod_{i=1}^{\tilde{n}} \sum_{\tilde{w}_i=1}^{\infty} \overline{G}(\tilde{w}_i) \\
&= \left(\frac{1-\lambda\overline{X}}{1-\lambda}\right)\left(\frac{\lambda(\overline{X}-1)}{1-\lambda}\right)^{\tilde{n}} \\
&= (1-\rho)\rho^{\tilde{n}}
\end{aligned}$$

The mean and variance of $\tilde{N}$ are

$$E\left[\tilde{N}\right] = \frac{\rho}{1-\rho}$$

and

$$\sigma^2_{\tilde{N}} = \frac{\rho}{(1-\rho)^2}$$

For the remainder of this section, we will call $\tilde{N}$ the number of messages in the queue and $\tilde{W}_i$ the residual service requirement of message i.

Let E_i be the event $\{\tilde{N} = \tilde{n}, \tilde{W}_j = \tilde{w}_j, j = 1 \ldots \tilde{n}, j \neq i\}$. In this case, we see that

$$P\{\tilde{W}_i = \tilde{w}_i | E_i\} = \frac{P_{\tilde{N},\tilde{W}_1,\ldots,\tilde{W}_{\tilde{N}}}(\tilde{n}, \tilde{w}_1, \ldots, \tilde{w}_{\tilde{n}})}{\sum_{\tilde{w}_i=1}^{\infty} P_{\tilde{N},\tilde{W}_1,\ldots,\tilde{W}_{\tilde{N}}}(\tilde{n}, \tilde{w}_1, \ldots, \tilde{w}_{\tilde{n}})}$$

$$= \frac{\overline{G}(\tilde{w}_i)}{\overline{X} - 1} \quad (\tilde{w}_i \geq 1) \tag{4.15}$$

Given that there are at least i messages, the residual length of the message in position i is independent of the rest of the state. That is, $\tilde{W}_1, \ldots, \tilde{W}_{\tilde{n}}$ are independent and identically distributed, conditional on $\tilde{N} \geq \tilde{n}$.

Let $M_\tau(i)$ denote the delay contribution of the message that is in position i at the time of the arrival of the marked message. Message i will have a packet to transmit in cycles $\tau - 1$, $\tau - 2$ and so on until either all of its packets have been sent or until the marked message completes service. Since message i has $\tilde{W}_i$ packets in the queue at the time of the arrival of the marked message, $\min(\tau - 1, \tilde{W}_i)$ packets of message i will be sent before the marked message completes service. This permits us to write

$$M_\tau(i) = \sum_{j=1}^{\tau-1} I_{\tilde{W}_i}(j) D_{\tau-j} \tag{4.16}$$

The distribution of $M_\tau(i)$ depends only on τ and $\tilde{W}_i$. As a consequence, given τ and $\tilde{N} = \tilde{n}$, the random variables $M_\tau(1), \ldots, M_\tau(\tilde{n})$ are independent and identically distributed. For convenience, we let $\tilde{W}$ and M_τ denote random variables that are distributed identically to $\tilde{W}_i$ and M_τ. The distribution of $\tilde{W}$ is

$$P_{\tilde{W}}(\tilde{w}) = \begin{cases} \dfrac{\overline{G}(\tilde{w})}{\overline{X} - 1} & \tilde{w} \geq 1 \\ 0 & \text{otherwise} \end{cases} \tag{4.17}$$

Furthermore,

$$M_\tau = \sum_{j=1}^{\tau-1} I_{\tilde{W}}(j) D_{\tau-j} \tag{4.18}$$

We can write the system time $V(\tau)$ as

$$V(\tau) = 1 + D_\tau(\tau - 1) + \sum_{i=1}^{\tilde{N}} M_\tau(i) \tag{4.19}$$

In the following subsections, we will use $\tilde{W}$ and M_τ to find the mean and variance of $V(\tau)$ as well as an upper bound to the variance of $V(\tau)$.

4.4.1 The Mean of $V(\tau)$

In Section 3.3, we found that the mean of $V(\tau)$ equaled

$$\overline{V(\tau)} = \frac{\tau - 1}{1 - \lambda\overline{X}} + 1$$

In this subsection, we will verify this fact. In doing so, we will identify properties of M_τ that will be needed subsequently.

Equation 4.19 permits us to write

$$\begin{aligned}
\overline{V(\tau)} &= 1 + \overline{D_\tau(\tau - 1)} + E\left[\sum_{i=1}^{\tilde{N}} M_\tau(i)\right] \\
&= 1 + \overline{D_\tau(\tau - 1)} + E\left[\tilde{N}\right]E[M_\tau] \\
&= 1 + \overline{D_\tau(\tau - 1)} + \frac{\rho\overline{M_\tau}}{1 - \rho} \qquad (4.20)
\end{aligned}$$

Taking the expectation over (4.18) yields

$$\overline{M_\tau} = \sum_{j=1}^{\tau-1} \overline{F}_{\tilde{W}}(j)\overline{D}_{\tau-j} \qquad (4.21)$$

Given $\overline{D}_1, \ldots, \overline{D}_\tau$, we can find a recursive computation for $\overline{M_\tau}$. In particular, from (4.21), we can write

$$\begin{aligned}
\overline{M_{\tau+1}} &= \sum_{j=1}^{\tau} \overline{F}_{\tilde{W}}(j)\overline{D}_{\tau+1-j} \\
&= \overline{D}_\tau + \sum_{j=1}^{\tau-1} \overline{F}_{\tilde{W}}(j+1)\overline{D}_{\tau-j} \qquad (4.22)
\end{aligned}$$

Since

$$\overline{F}_{\tilde{W}}(j+1) = \overline{F}_{\tilde{W}}(j) - \frac{\overline{G}(j)}{\overline{X} - 1}$$

we can restate (4.22) as

$$\begin{aligned}
\overline{M_{\tau+1}} &= \overline{D}_\tau + \sum_{j=1}^{\tau-1} \overline{F}_{\tilde{W}}(j)\overline{D}_{\tau-j} - \frac{1}{\overline{X} - 1}\sum_{j=1}^{\tau-1} \overline{G}(j)\overline{D}_{\tau-j} \\
&= \overline{D}_\tau + \overline{M_\tau} - \frac{1}{\overline{X} - 1}\sum_{j=1}^{\tau-1} \overline{G}(j)\overline{D}_{\tau-j}
\end{aligned}$$

From (4.8), we see that

$$\sum_{j=1}^{\tau-1} \overline{G}(j)\overline{D}_{\tau-j} = \frac{(1-\lambda)\overline{D}_\tau - 1}{\lambda}$$

We can conclude that

$$\overline{M_{\tau+1}} = \overline{M_\tau} + \frac{1}{\lambda(\overline{X}-1)} - \left(\frac{1-\lambda\overline{X}}{\lambda(\overline{X}-1)}\right)\overline{D}_\tau \tag{4.23}$$

This recursion enables us to simplify the expression (4.20) for $\overline{V(\tau)}$. From (4.11),

$$\overline{D_\tau(\tau-1)} = \sum_{j=1}^{\tau-1} \overline{D}_{\tau-j} \tag{4.24}$$

We can use (4.20) and (4.24) to write

$$\overline{V(\tau+1)} - \overline{V(\tau)} = \overline{D}_\tau + \frac{\rho}{1-\rho}\left(\overline{M_{\tau+1}} - \overline{M_\tau}\right)$$

We note that

$$\frac{\rho}{1-\rho} = \frac{\lambda(\overline{X}-1)}{1-\lambda\overline{X}}$$

Combining this with (4.23), we see that

$$\overline{V(\tau+1)} = \overline{V(\tau)} + \frac{1}{1-\lambda\overline{\overline{X}}}$$

Since $\overline{V(1)} = 1$, it must be that

$$\overline{V(\tau)} = \frac{\tau-1}{1-\lambda\overline{\overline{X}}} + 1$$

This is the result that we have already found by the reversibility argument. Since (4.8) for $\overline{D}_t$ cannot usually be simplified, it seems somewhat remarkable that the simple form of $\overline{V(\tau)}$ can be verified by this method.

4.4.2 The variance of $V(\tau)$

For the variance of $V(\tau)$, we see from (4.19) that

$$
\begin{aligned}
\sigma^2_{V(\tau)} &= \sigma^2_{D_\tau(\tau-1)} + \operatorname{var}\left[\sum_{i=1}^{\tilde{N}} M_\tau(i)\right] \\
&= \sigma^2_{D_\tau(\tau-1)} + E\left[\tilde{N}\right]\sigma^2_{M_\tau} + \sigma^2_{\tilde{N}}\overline{M_\tau}^2 \\
&= \sigma^2_{D_\tau(\tau-1)} + \left(\frac{\rho}{1-\rho}\right)\sigma^2_{M_\tau} + \left(\frac{\rho}{(1-\rho)^2}\right)\overline{M_\tau}^2 \\
&= \sigma^2_{D_\tau(\tau-1)} + \frac{\rho\overline{M^2_\tau}}{1-\rho} + \left(\frac{\rho\overline{M_\tau}}{1-\rho}\right)^2
\end{aligned} \tag{4.25}
$$

From (4.18), we can write

$$
\begin{aligned}
\overline{M^2_\tau} &= E\left[\left(\sum_{j=1}^{\tau-1} I_{\tilde{W}}(j)D_{\tau-j}\right)\left(\sum_{k=1}^{\tau-1} I_{\tilde{W}}(k)D_{\tau-k}\right)\right] \\
&= \sum_{j=1}^{\tau-1}\sum_{k=1}^{\tau-1} E[I_{\tilde{W}}(j)I_{\tilde{W}}(k)]E[D_{\tau-j}D_{\tau-k}]
\end{aligned}
$$

Applying (4.4) and (4.6) to the above expression yields

$$
\begin{aligned}
\overline{M^2_\tau} &= \sum_{j=1}^{\tau-1}\left(\sum_{k=1}^{j-1}\overline{F}_{\tilde{W}}(j)\overline{D}_{\tau-j}\overline{D}_{\tau-k} + \overline{F}_{\tilde{W}}(j)\overline{D^2_{\tau-j}} + \sum_{k=j+1}^{\tau-1}\overline{F}_{\tilde{W}}(k)\overline{D}_{\tau-j}\overline{D}_{\tau-k}\right) \\
&= \sum_{j=1}^{\tau-1}\left(\overline{D}_{\tau-j}\overline{F}_{\tilde{W}}(j)\sum_{k=1}^{j-1}\overline{D}_{\tau-k} + \overline{F}_{\tilde{W}}(j)\overline{D^2_{\tau-j}} + \overline{D}_{\tau-j}\sum_{k=j+1}^{\tau-1}\overline{F}_{\tilde{W}}(k)\overline{D}_{\tau-k}\right)
\end{aligned} \tag{4.26}
$$

We can simplify the computation of $\overline{M^2_\tau}$ as well. From (4.18), we can write

$$
\begin{aligned}
\overline{M^2_{\tau+1}} &= E\left[\left(\sum_{j=1}^{\tau} I_{\tilde{W}}(j)D_{\tau+1-j}\right)^2\right] \\
&= E\left[\left(D_\tau + \sum_{j=1}^{\tau-1} I_{\tilde{W}}(j+1)D_{\tau-j}\right)^2\right] \\
&= E\left[D^2_\tau + 2D_\tau\sum_{j=1}^{\tau-1} I_{\tilde{W}}(j+1)D_{\tau-j} + \left(\sum_{j=1}^{\tau-1} I_{\tilde{W}}(j+1)D_{\tau-j}\right)^2\right]
\end{aligned}
$$

$$= \overline{D^2_\tau} + 2\overline{D}_\tau \sum_{j=1}^{\tau-1} \overline{F}_{\tilde{W}}(j+1)\overline{D}_{\tau-j} + E\left[\left(\sum_{j=1}^{\tau-1} I_{\tilde{W}}(j+1)D_{\tau-j}\right)^2\right] \quad (4.27)$$

Equations (4.22) and (4.23) imply that

$$\begin{aligned}\sum_{j=1}^{\tau-1} \overline{F}_{\tilde{W}}(j+1)\overline{D}_{\tau-j} &= \overline{M_{\tau+1}} - \overline{D}_\tau \\ &= \overline{M_\tau} - \frac{\overline{D}_\tau(1-\lambda)-1}{\lambda(\overline{X}-1)} \quad (4.28)\end{aligned}$$

Furthermore, we can write

$$\begin{aligned}E\left[\left(\sum_{j=1}^{\tau-1} I_{\tilde{W}}(j+1)D_{\tau-j}\right)^2\right] &= E\left[\left(\sum_{j=1}^{\tau-1} I_{\tilde{W}}(j+1)D_{\tau-j}\right)\left(\sum_{k=1}^{\tau-1} I_{\tilde{W}}(k+1)D_{\tau-k}\right)\right] \\ &= \sum_{j=1}^{\tau-1}\sum_{k=1}^{\tau-1} E[I_{\tilde{W}}(j+1)I_{\tilde{W}}(k+1)]E[D_{\tau-j}D_{\tau-k}]\end{aligned}$$

Applying (4.4) to the above expression, we see that

$$E\left[\left(\sum_{j=1}^{\tau-1} I_{\tilde{W}}(j+1)D_{\tau-j}\right)^2\right] =$$
$$\sum_{j=1}^{\tau-1}\left(\overline{F}_{\tilde{W}}(j+1)\overline{D}_{\tau-j}\sum_{k=1}^{j-1}\overline{D}_{\tau-k} + \overline{F}_{\tilde{W}}(j+1)\overline{D^2_{\tau-j}} + \overline{D}_{\tau-j}\sum_{k=j+1}^{\tau-1}\overline{F}_{\tilde{W}}(k+1)\overline{D}_{\tau-k}\right)$$

Since

$$\overline{F}_{\tilde{W}}(j+1) = \overline{F}_{\tilde{W}}(j) - \frac{\overline{G}(j)}{\overline{X}-1}$$

we see from (4.26) that

$$E\left[\left(\sum_{j=1}^{\tau-1} I_{\tilde{W}}(j+1)D_{\tau-j}\right)^2\right] =$$
$$\overline{M^2_\tau} - \frac{1}{\overline{X}-1}\sum_{j=1}^{\tau-1}\left(\overline{G}(j)\overline{D}_{\tau-j}\sum_{k=1}^{j-1}\overline{D}_{\tau-k} + \overline{G}(j)\overline{D^2_{\tau-j}} + \overline{D}_{\tau-j}\sum_{k=j+1}^{\tau-1}\overline{G}(k)\overline{D}_{\tau-k}\right)$$

From (4.9), the recursive solution for $\overline{D_t^2}$, we observe that

$$E\left[\left(\sum_{j=1}^{\tau-1} I_{\tilde{W}}(j+1)D_{\tau-1}\right)^2\right] = \overline{M_\tau^2} - \frac{(1-\lambda)\left(\overline{D_\tau^2} - 2(\overline{D}_\tau)^2\right)+1}{\lambda(\overline{X}-1)} \tag{4.29}$$

Combining (4.27), (4.28) and (4.29) yields

$$\overline{M_{\tau+1}^2} = \overline{M_\tau^2} + 2\overline{D}_\tau\overline{M}_\tau + \frac{2\overline{D}_\tau - 1 - (1-\lambda\overline{X})\overline{D_\tau^2}}{\lambda(\overline{X}-1)} \tag{4.30}$$

Since $D_\tau(\tau-1) = D_{\tau-1} + \cdots + D_1$, a sum of independent random variables, we can write

$$\sigma^2_{D_\tau(\tau-1)} = \sigma^2_{D_{\tau-1}} + \cdots + \sigma^2_{D_1}$$

From (4.25), we can write

$$\sigma^2_{V(\tau+1)} - \sigma^2_{V(\tau)} = \sigma^2_{D_\tau} + \frac{\rho\left(\overline{M_{\tau+1}^2} - \overline{M_\tau^2}\right)}{1-\rho} + \frac{\rho^2\left(\overline{M_{\tau+1}}^2 - \overline{M_\tau}^2\right)}{(1-\rho)^2} \tag{4.31}$$

Equation 4.23 permits us to write

$$\overline{M_{\tau+1}}^2 = \overline{M_\tau}^2 + 2\overline{M_\tau}\left(\frac{1-(1-\lambda\overline{X})\overline{D}_\tau}{\lambda(\overline{X}-1)}\right) + \left(\frac{1-(1-\lambda\overline{X})\overline{D}_\tau}{\lambda(\overline{X}-1)}\right)^2 \tag{4.32}$$

In addition, we restate (4.30) as

$$\overline{M_{\tau+1}^2} - \overline{M_\tau^2} = 2\overline{D}_\tau\overline{M}_\tau + \frac{2\overline{D}_\tau - 1 - (1-\lambda\overline{X})\overline{D_\tau^2}}{\lambda(\overline{X}-1)} \tag{4.33}$$

Substituting (4.32) and (4.33) into (4.31) yields

$$\sigma^2_{V(\tau+1)} - \sigma^2_{V(\tau)} = \frac{2\overline{M_\tau}\lambda(\overline{X}-1) + \lambda\overline{X}}{(1-\lambda\overline{X})^2} \tag{4.34}$$

Given $\overline{D}_1, \ldots, \overline{D}_{\tau-1}$, we can use (4.23) to compute $\overline{M_\tau}$. Consequently, $\overline{D}_1, \ldots, \overline{D}_{\tau-1}$ is also sufficient to calculate $\sigma^2_{V(\tau)}$ efficiently.

4.4.3 An Upper Bound for $\sigma^2_{V(\tau)}$

We now will upper bound $\sigma^2_{V(\tau)}$, the variance of the system time of a typical message of length τ. From (4.21) and Theorem 4, we see that

$$\begin{aligned}
\overline{M_\tau} &\leq \overline{D} \sum_{j=1}^{\tau-1} \overline{F}_{\tilde{W}}(j) \\
&\leq \overline{D} \sum_{j=1}^{\infty} \overline{F}_{\tilde{W}}(j) \\
&= \overline{D} E\left[\tilde{W}\right]
\end{aligned} \tag{4.35}$$

Note that we can find $E\left[\tilde{W}\right]$ in terms of the moments of X. In particular, using the distribution of $\tilde{W}$ defined in (4.17),

$$\begin{aligned}
E\left[\tilde{W}\right] &= \frac{1}{\overline{X}-1} \sum_{w=1}^{\infty} w\overline{G}(w) \\
&= \frac{1}{\overline{X}-1} \sum_{w=0}^{\infty} w \sum_{j=w+1}^{\infty} g(j) \\
&= \frac{1}{\overline{X}-1} \sum_{j=1}^{\infty} g(j) \sum_{w=0}^{j-1} w \\
&= \frac{1}{\overline{X}-1} \sum_{j=1}^{\infty} g(j) \frac{j(j-1)}{2} \\
&= \frac{\overline{X^2}-\overline{X}}{2(\overline{X}-1)}
\end{aligned}$$

Recalling that $\overline{D} = 1/(1-\lambda\overline{X})$, it follows from (4.35) that

$$\overline{M_\tau} \leq \frac{\overline{X^2}-\overline{X}}{2(1-\lambda\overline{X})(\overline{X}-1)}$$

As a result, (4.37) implies that

$$\sigma^2_{V(\tau+1)} - \sigma^2_{V(\tau)} \leq \frac{\lambda\left(\sigma^2_X + (1-\lambda)\overline{X}^2\right)}{(1-\lambda\overline{X})^3} \tag{4.36}$$

Since $V(1) = 1$, we know that $\sigma^2_{V(1)} = 0$. We can conclude that

$$\sigma^2_{V(\tau)} \leq \frac{\lambda\left(\sigma^2_X + (1-\lambda)\overline{X}^2\right)}{(1-\lambda\overline{X})^3}(\tau-1) \tag{4.37}$$

Equation 4.37 demonstrates that the variance of the system time of a typical message is bounded by a linear function of the message length. Moreover, (4.36) implies that the difference between the upper bound (4.37) for $\sigma^2_{V(\tau)}$ and $\sigma^2_{V(\tau)}$ increases monotonically.

In the next section, we will examine $\sigma^2_{V(\tau)}$ through some examples. We will consider the upper bound for $\sigma^2_{V(\tau)}$ in Chapter 5 when we compare round robin and last come first served service disciplines.

4.5 Delay Examples

We have verified that $\overline{V(\tau)}$, the average system time for a typical length τ message is described by

$$\overline{V(\tau)} = \frac{\tau - 1}{1 - \lambda\overline{X}} + 1$$

However, we have found that the higher moments of $V(\tau)$ are quite difficult to characterize. In this section we will compute $\sigma^2_{V(\tau)}$ for some simple examples. In particular, it would be of interest to see how the system time variance for short messages changes when the load of the queue becomes increasingly dominated by long messages.

4.5.1 Geometric Message Lengths

Suppose that the message length X has the distribution

$$g(x) = (1 - \mu)\mu^{x-1} \quad (x \geq 1)$$

Note that

$$\overline{X} = \frac{1}{1 - \mu}$$

Increasing $\overline{X}$ corresponds to a more bursty arrival process. In Figure 4-2, we plot $\sigma^2_{V(\tau)}$ as a function of τ for three values of $\overline{X}$. Of course, smaller $\sigma^2_{V(\tau)}$ implies a more predictable system time which is a measure of better service.

In terms of the system time variance, Figure 4-2 indicates that all messages receive worse service as the arrival process becomes more bursty. In addition, this degradation is more severe for the long messages that are the cause of burstiness. However, one should keep in mind that $\overline{V(\tau)}$, the mean system time of a length τ message, is independent of the burstiness of the arrival process.

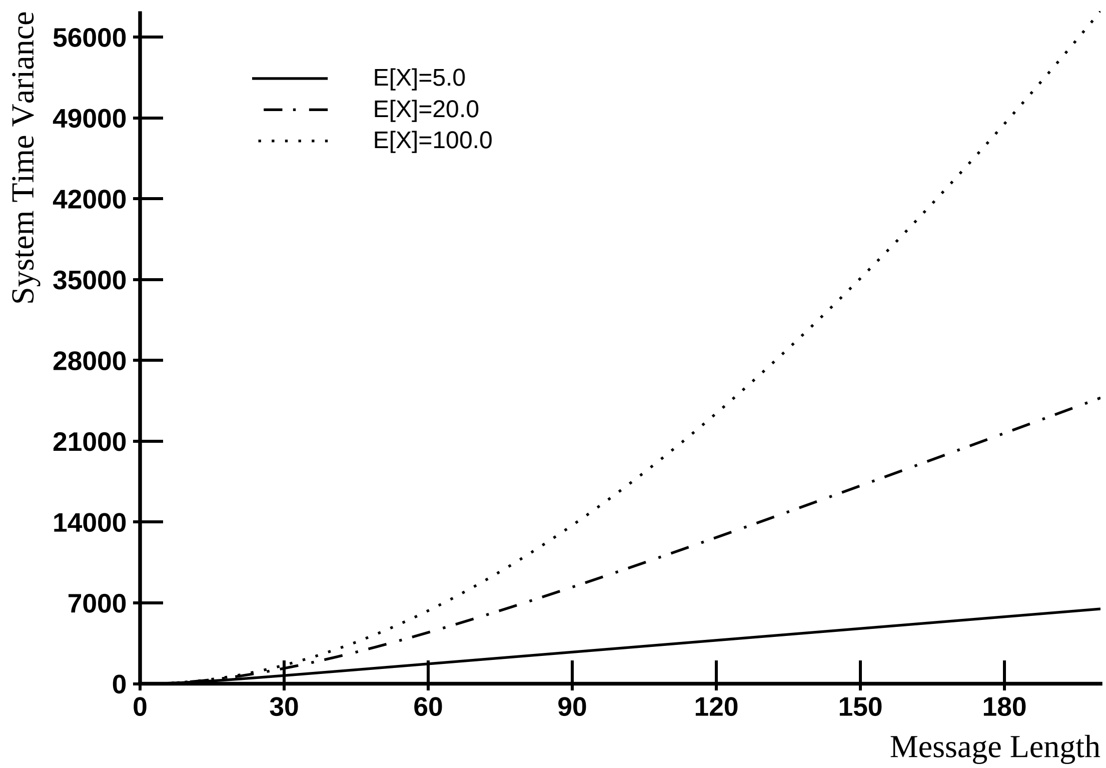

Figure 4-2: X: Geometric, $\lambda\overline{X} = 1/2$

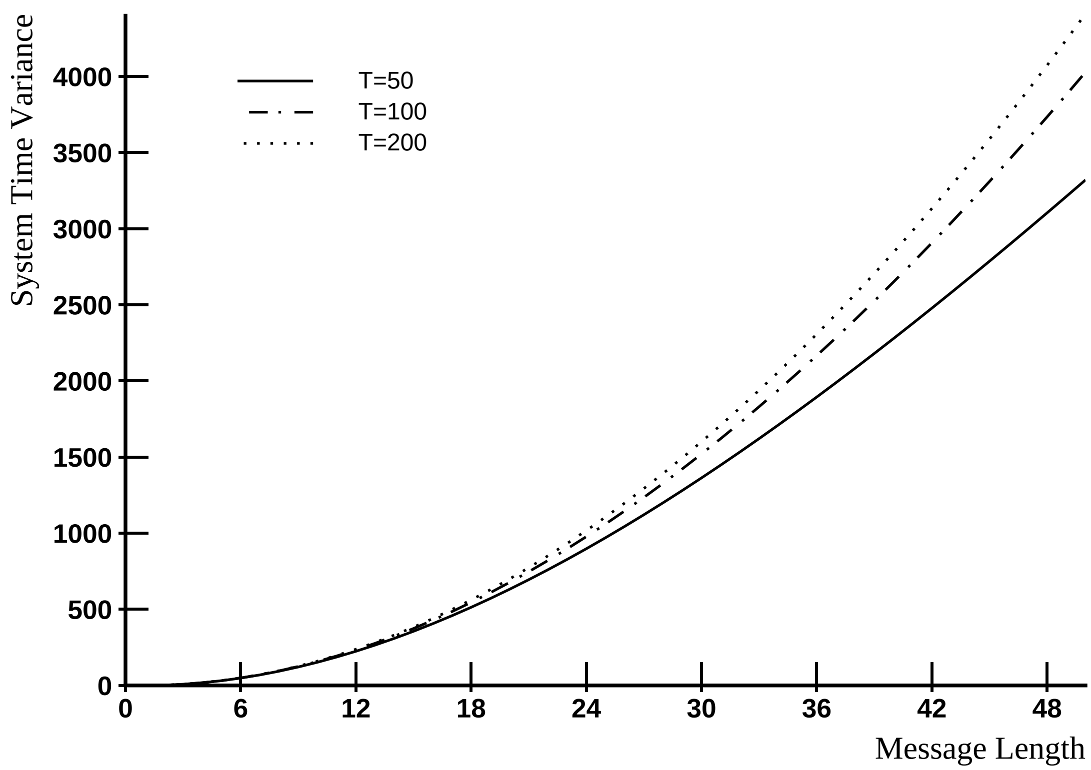

Figure 4-3: X: Uniform $[1, T]$, $\lambda\overline{X} = 1/2$

4.5.2 Uniform Message Lengths

Let the message length X be distributed as

$$g(x) = \frac{1}{T} \quad (1 \leq x \leq T)$$

As T increases, short messages become increasingly rare. Moreover, increasing T corresponds to increasingly bursty traffic. Once again, the interesting question is to what extent service for a short message is degraded as the burstiness increases. Figure 4-3 plots $\sigma^2_{V(\tau)}$ as a function of τ for three values of T. From Figure 4-3, we see that the service time standard deviation of a 20 packet message is essentially unchanged whether T equals 50, 100 or 200.

4.6 Summary of Results

We will conclude this chapter with a brief summary of results. For a message of length τ, we have found that we can compute the first two moments of $V(\tau, n, w_1, \ldots, w_n)$, the system time of the message given that the state upon arrival is $n, w_1, \ldots, w_n$, and $V(\tau)$, the system time given that the state upon arrival is described by the stationary distribution. In this section, we will summarize the procedure. We found that we can compute the mean of D_t for all $t < \tau$ via

$$\overline{D}_t = \frac{1}{1-\lambda}\left(1 + \lambda \sum_{y=1}^{t-1} \overline{G}(y)\overline{D}_{t-y}\right)$$

The second moment of D_t can be found from

$$\begin{aligned}
\overline{D_t^2} &= 2(\overline{D}_t)^2 - \frac{1}{1-\lambda} \\
&\quad + \frac{1}{1-\lambda}\sum_{j=1}^{t-1}\left(\overline{G}(j)\overline{D}_{t-j}\sum_{k=1}^{j-1}\overline{D}_{t-k} + \overline{G}(j)\overline{D^2_{t-j}} + \overline{D}_{t-j}\sum_{k=j+1}^{t-1}\overline{G}(k)\overline{D}_{t-k}\right)
\end{aligned}$$

We then showed that

$$\overline{V(\tau, n, w_1, \ldots, w_n)} = 1 + \sum_{j=1}^{\tau-1}\left(1 + \sum_{i=1}^{n} I_{w_i}(j)\right)\overline{D}_{\tau-j}$$

and

$$\sigma^2_{V(\tau,n,w_1,\ldots,w_n)} = \sum_{j=1}^{\tau-1}\left(1 + \sum_{i=1}^{n} I_{w_i}(j)\right)\sigma^2_{D_{\tau-j}}$$

We can find $\overline{M_\tau}$ by

$$\overline{M_{\tau+1}} = \begin{cases} 0 & \tau = 0 \\ \overline{M_\tau} + \dfrac{1}{\lambda(\overline{X}-1)} - \left(\dfrac{1-\lambda\overline{X}}{\lambda(\overline{X}-1)}\right)\overline{D_\tau} & \tau > 0 \end{cases}$$

The mean system time of a length τ message was found to be

$$\overline{V(\tau)} = \frac{\tau - 1}{1 - \lambda\overline{X}} + 1$$

The variance of $V(\tau)$ can be found from

$$\sigma^2_{V(\tau+1)} = \begin{cases} 0 & \tau = 0 \\ \sigma^2_{V(\tau)} + \dfrac{2\overline{M_\tau}\lambda(\overline{X}-1) + \lambda\overline{X}}{(1-\lambda\overline{X})^2} & \tau > 0 \end{cases}$$

We also found that

$$\sigma^2_{V(\tau)} \leq \frac{\lambda(\sigma_x^2 + (1-\lambda)\overline{X}^2)}{(1-\lambda\overline{X})^3}(\tau - 1)$$

We have already computed $\overline{V(\tau)}$ by reversibility. However, the reversibility argument only allows us to examine the mean delay. In the next chapter, we will use $\sigma^2_{V(\tau)}$ to differentiate round robin from last come first served service which we verify to provide identical first order delay behavior.

Chapter 5

LCFS Service with Multiple Message Classes

It is also possible to analyze a Last Come First Served (LCFS) queue using a reversibility argument. We will verify that the LCFS queue has many properties in common with the round robin system. In particular, for the appropriately defined LCFS system, the stationary distribution is the same as that of the round robin queue. As a consequence, the average system time of a message is the same for both service disciplines.

LCFS service provides length discrimination because a short message is less likely to be preempted by a new arrival. However, the benefits of length discrimination are achieved in a way that we feel is essentially haphazard. We believe that it is inappropriate for a message to receive service just because it happened to arrive last.

5.1 The LCFS Stationary Distribution

The basic model for the LCFS queue is the same as that for the multiple class round robin queue. These assumptions will now be repeated.

During each slot, a message of class c arrives with probability λ_c. No more than one message can arrive in any slot and the probability of no message arriving during a slot is $1 - \lambda$ where $\lambda = \sum_{c=1}^{C} \lambda_c$. Each class c message has an independent integer packet length X_c described by $g_c(x) = \Pr\{X_c = x\}$ and $\overline{G}_c(x) = P\{X_c > x\}$ such that $\overline{G}_c(0) = 1$. As before, given that a class c message has already had w packets sent, the conditional probability that the message's service requirement will be fulfilled after its next packet is sent will be denoted by

$$\begin{aligned} r_c(w) &= P\{X_c = w+1 | X_c > w\} \\ &= \frac{g_c(w+1)}{\overline{G}_c(w)} \end{aligned}$$

The basic sequence of service, departure and arrival is carried over from the round robin system. However, the LCFS system does not rotate the messages in the queue. In particular, at time t, the following sequence of events occurs:

1. The message at the front of the queue has one packet sent.

2. Following service, this message will depart if all of its packets have been sent.

3. If a new message arrives at the end of slot t, it is placed at the front of the queue to begin service at time $t+1$.

As before, let $n, y_1, \ldots, y_n$ represent the state of the queue, where n is the number of messages in the system and $y_i = (w_i, c_i)$ such that for the message in position i, w_i is the number of packets already transmitted and c_i is the class of the message. Once again, we conjecture that the reverse time system is a LCFS system of the same type as the forward system.

Conjecture 5 *The reverse time process is a LCFS queueing system. At the start of slot t, a message of class c arrives with probability λ_c. No more than one message can arrive in any slot and the probability of no message arriving during a slot is $1-\lambda$ where $\lambda = \sum_{c=1}^{C} \lambda_c$. Each class c message has an independent service time distributed according to G_c. During slot t, the following sequence of events occurs:*

1. *The message at the front of the queue departs if all of its packets have been sent.*

2. *If a new message arrives, it is inserted at the front of the queue.*

3. *The message at the front of the queue has one packet sent.*

The system is still a discrete time Markov process with the state $n, y_1, \ldots, y_n$ representing the remaining integer service demands (as well as the classes) of the messages in the system. Furthermore, the stationary distribution of the queue has the form

$$\pi_{n,y_1,\ldots,y_n} = \pi_\phi \prod_{i=1}^{n} \frac{\lambda_{c_i}}{1-\lambda} \overline{G}_{c_i}(w_i) \tag{5.1}$$

where π_ϕ is the empty state probability.

We prove Conjecture 5 in Subsection 5.1.1. The argument is essentially identical to that used for the multiple class round robin queue. Although we include the proof for completeness, the reader may wish to pass over it. Before we prove Conjecture 5, we will consider its implications.

We observe that the stationary distribution (5.1) for the LCFS queue is identical to that for the round robin system. Moreover, the state space for both systems is the same. That is, $w_1 \geq 0$ and $w_i \geq 1$ for all $i > 1$. Consequently, all of the marginal properties of the round robin system apply to the LCFS system as well. In particular, the marginal distribution for the number of messages of a certain class and the average delay for a message of a certain class must be the same. That is, the system time of a typical message of length τ must be the same under LCFS and round robin service. In short, if $U(\tau)$ represents the LCFS system time of a typical message of length τ, then $\overline{U(\tau)} = \overline{V(\tau)}$. One can conclude that short messages receive better service than long messages under LCFS as well. This is not surprising since the shorter a message is, the more likely it is to complete service before another message can arrive and preempt it.

However, the two systems are very different. In particular, under LCFS, a short message can be preempted by a long message that is completely served before the short message can reenter service. Consequently, with some small probability, a short message may wait a very long time. Under round robin however, the arrival of the long message merely degrades the service received by the short message. Consequently, one might expect that the delay variance for the LCFS system would be larger than that for round robin, particularly for short messages. To characterize the LCFS system time, we will examine the busy period of the LCFS queue in Section 5.2.

5.1.1 Proof of Conjecture 5

As always, we will use Theorem 1 to prove Conjecture 5. That is, for all neighboring states u and v, we will verify that the distributiion (5.1) satisfies

$$\pi_u P_{uv} = \pi_v P^*_{vu} \tag{5.2}$$

The four basic transitions that we must consider are:

- The front message is served and departs. No new message arrives.

For the reverse transition, a new message of class c_1 and length $w_1 + 1$ must arrive to return to the original state. In this instance, the transition pair is

$$n, y_1, \ldots, y_n \quad \underset{\lambda_{c_1} g_{c_1}(w_1+1)}{\overset{(1-\lambda) r_{c_1}(w_1)}{\rightleftarrows}} \quad n-1, y_2, \ldots, y_n$$

This transition pair implies that $\pi_{n,y_1,\ldots,y_n}$ must satisfy

$$\pi_{n,y_1,\ldots,y_n}(1-\lambda) g_{c_1}(w_1+1)/\overline{G}_{c_1}(w_1) = \lambda_{c_1} g_{c_1}(w_1+1) \pi_{n-1,y_2,\ldots,y_n}$$

Rewriting yields

$$\pi_{n,y_1,\ldots,y_n} = \frac{\lambda_{c_1}}{1-\lambda} \overline{G}_{c_1}(w_1) \pi_{n-1,y_2,\ldots,y_n}$$

It can be directly verified that this requirement is satisfied by the distribution (5.10).

- The front message is served but does not depart. No new message arrives.

  For the reverse transition, we must not have an arrival to ensure that the front message is served. This state transition pair corresponds to

$$n, y_1, \ldots, y_n \quad \underset{1-\lambda}{\overset{(1-\lambda)[1-r_{c_1}(w_1)]}{\rightleftarrows}} \quad n, (w_1+1, c_1), y_2, \ldots, y_n$$

  We must check that

$$\pi_{n,y_1,\ldots,y_n}(1-\lambda)[1-r_{c_1}(w_1)] = (1-\lambda) \pi_{n,y_2,\ldots,y_n,(w_1+1,c_1)}$$

  Since $1 - r_{c_1}(w_1)$ can be rewritten as $\overline{G}_{c_1}(w_1+1)/\overline{G}_{c_1}(w_1)$, we must have

$$\frac{\overline{G}_{c_1}(w_1+1)}{\overline{G}_{c_1}(w_1)} \left(\pi_\phi \prod_{i=1}^{n} \frac{\lambda_{c_i}}{1-\lambda} \overline{G}_{c_i}(w_i) \right) = \left(\pi_\phi \prod_{i=2}^{n} \frac{\lambda_{c_i}}{1-\lambda} \overline{G}_{c_i}(w_i) \right) \frac{\lambda_{c_1}}{1-\lambda} \overline{G}_{c_1}(w_1+1)$$

  which holds by cancellation.

- The front message is served and departs. A new class c message arrives.

  In reverse time, the front message departs immediately. To return to the original state, there must be a new class c_1 arrival requiring exactly $w_1 + 1$ units of service. The corresponding transition pair is

$$n, y_1, \ldots, y_n \quad \underset{\lambda_{c_1} g_{c_1}(w_1+1)}{\overset{\lambda_c r_{c_1}(w_1)}{\rightleftarrows}} \quad n, (0,c), y_2, \ldots, y_n$$

Since $\overline{G}_c(0) = 1$, we see that

$$\begin{aligned} \lambda_c \frac{g_{c_1}(w_1+1)}{\overline{G}_{c_1}(w_1)} \pi_{n,y_1,\ldots,y_n} &= \lambda_{c_1} g_{c_1}(w_1+1) \frac{\lambda_c}{1-\lambda} \left(\pi_\phi \prod_{i=2}^{n} \frac{\lambda_{c_i}}{1-\lambda} \overline{G}_{c_i}(w_i) \right) \\ &= \lambda_{c_1} g_{c_1}(w_1+1) \pi_{n,(0,c),y_2,\ldots,y_n} \end{aligned}$$

- The front message is served but does not depart. A new class c message arrives.

  For the reverse transition, the front message will immediately depart and the second message will served iff no new message arrives. Hence, the transitions are

$$n, y_1, \ldots, y_n \quad \underset{1-\lambda}{\overset{\lambda_c[1 - r_{c_1}(w_1)]}{\rightleftarrows}} \quad n+1, (0,c), (w_1+1, c_1), y_2, \ldots, y_n$$

  Substituting $\overline{G}_{c_1}(w_1+1)/\overline{G}_{c_1}(w_1)$ for $1 - r_{c_1}(w_1)$, we see that

$$\begin{aligned} \pi_{n,y_1,\ldots,y_n} \lambda_c [1 - r_{c_1}(w_1)] &= \lambda_c \left(\pi_\phi \prod_{i=2}^{n} \frac{\lambda_{c_i}}{1-\lambda} \overline{G}_{c_i}(w_i) \right) \frac{\lambda_{c_1}}{1-\lambda} \overline{G}_{c_1}(w_1+1) \\ &= (1-\lambda) \pi_{n+1,(0,c),y_2,\ldots,y_n,(w_1+1,c_1)} \end{aligned}$$

  since $\overline{G}_c(0) = 1$. This proves our conjecture for the multiple class LCFS queue.

5.2 The Busy Period

In this section, we will study the busy period of a LCFS queue. We note that this is also the busy period of the round robin queue as well as that for every other work conserving service discipline.

Consider the arrival of a new message. None of the messages that are already in the system will receive any service until the new arrival leaves the system. Consequently, the system time of the new message is independent of the state of the queue at the time of arrival. This new arrival might just as well arrive at an empty system. In this case, the arrival would mark the beginning of a busy period for the queue. If this new arrival departs at the end of slot t, then we will define the busy period to end during slot t. We state this as a definition since it is possible for a new message to arrive at the end of slot t, the instant after the departure of the original message. In this case, the system would be nonempty during slots t and $t+1$. However, by our definition, slot t marks the end of a busy period and a new busy period starts with slot $t+1$. We define the busy period in this way so that the system time of a typical message, that is a message with length described by $g(x)$, has the same distribution as the length of the busy period.

Suppose that we mark the arrival of a message at the end of slot s. We will denote the length of this message by X. This message has its first packet transmitted during slot $s+1$. Suppose $X = x$ and that the packets of this message are transmitted during slots $s_1 = s+1, s_2, \ldots, s_x$. With probability $1-\lambda$, there is no new arrival at the end of slot $s+1$ so that $s_2 = s+2$. However, with probability λ, there is a new arrival at the end of slot $s+1$. In this case, this new arrival will be completely served before the marked message can reenter service. Suppose that this new message completes service during slot t'. Since the system time of this new arrival has the same distribution as the busy period, we will say that the new arrival initiates a *sub-busy period* that ends during slot t'. The marked message will have its second packet transmitted during slot $t'+1$ iff there is no new arrival at the end of slot t'. If another message does arrive at the end of slot t', this message will initiate another sub-busy period. If we denote by K_j the number of sub-busy periods that occur between slots s_j and s_{j+1}, then it should be apparent that K_j has the following distribution.

$$P_{K_j}(k) = (1-\lambda)\lambda^k \quad (k \geq 0)$$

The z transform of K_j equals

$$\begin{aligned} K_j(z) &= E\left[z^{K_j}\right] \\ &= \frac{1-\lambda}{1-\lambda z} \end{aligned}$$

We define K as the total number of sub-busy periods that occur before the departure of the marked message so that

$$K = K_1 + K_2 + \cdots + K_{X-1}$$

Given $X = x$, the z transform of K is

$$E\Big[z^K | X = x\Big] = \left(\frac{1-\lambda}{1-\lambda z}\right)^{x-1}$$

If we denote the length of the busy period by Y and the length of sub-busy period j by Y_j, then

$$Y = X + Y_1 + \cdots + Y_K$$

Consequently,

$$\begin{aligned}
E\Big[z^Y | X = x, K = k\Big] &= E\Big[z^{x+Y_1+\cdots+Y_k}\Big] \\
&= z^x \left(E\Big[z^Y\Big]\right)^k \\
&= z^x \left(Y(z)\right)^k
\end{aligned}$$

Removing the conditioning on K yields

$$\begin{aligned}
E\Big[z^Y | X = x\Big] &= \sum_{k=0}^{\infty} P_{K|X}(k|x) z^x \left(Y(z)\right)^k \\
&= z^x E\Big[(Y(z))^K | X = x\Big] \\
&= z^x \left(\frac{1-\lambda}{1-\lambda Y(z)}\right)^{x-1}
\end{aligned}$$

Finally, taking the expectation over X, we find that

$$\begin{aligned}
Y(z) &= \sum_{x=1}^{\infty} g(x) z^x \left(\frac{1-\lambda}{1-\lambda Y(z)}\right)^{x-1} \\
&= \frac{X\left(\frac{(1-\lambda)z}{1-\lambda Y(z)}\right)}{\frac{1-\lambda}{1-\lambda Y(z)}}
\end{aligned}$$

By defining

$$\gamma(z) = \frac{1-\lambda}{1-\lambda Y(z)}$$

and

$$\beta(z) = X(z\gamma(z))$$

we can write

$$Y(z) = \frac{\beta(z)}{\gamma(z)} \tag{5.3}$$

We will take derivatives of $Y(z)$ to find the first two moments of Y. We recall that for any random variable Y,

$$Y'(z) \mid_{z=1} = \overline{Y}$$

and

$$Y''(z) \mid_{z=1} = \overline{Y^2} - \overline{Y}$$

We will identify the derivatives of $\gamma(z)$ and $\beta(z)$ in terms of those of $Y(z)$. In particular, taking the first two derivatives of $\gamma(z)$, we find that

$$\gamma'(1) = \frac{\lambda}{1-\lambda}\overline{Y} \tag{5.4}$$

and that

$$\gamma''(1) = \frac{\lambda}{1-\lambda}\left(\overline{Y^2} - \overline{Y}\right) + 2\left(\frac{\lambda}{1-\lambda}\right)^2 \overline{Y}^2 \tag{5.5}$$

In addition, for $\beta(z)$, we have

$$\beta'(1) = \overline{X}\left(1 + \frac{\lambda}{1-\lambda}\overline{Y}\right) \tag{5.6}$$

and

$$\beta''(1) = \overline{X^2}\left(1 + \frac{\lambda}{1-\lambda}\overline{Y}\right)^2 + \overline{X}\left(\frac{\lambda}{1-\lambda}\overline{Y^2} - \frac{\lambda}{1-\lambda}\overline{Y} + \left(\frac{\lambda}{1-\lambda}\overline{Y}\right)^2 - 1\right) \tag{5.7}$$

From (5.3), we see that

$$Y'(z) = \frac{\gamma(z)\beta'(z) - \beta(z)\gamma'(z)}{(\gamma(z))^2}$$

Noting that $\beta(1) = 1$ and $\gamma(1) = 1$, evaluation of $Y'(z)$ at $z = 1$ yields

$$\begin{aligned}\overline{Y} &= \overline{X} + \frac{\lambda(\overline{X} - 1)}{1-\lambda}\overline{Y} \\ &= \frac{(1-\lambda)\overline{X}}{1-\lambda\overline{X}} \qquad (5.8)\end{aligned}$$

The second derivative of Y(z) is

$$Y''(z) \quad = \quad \frac{\gamma(z)\beta''(z) - \beta(z)\gamma''(z)}{(\gamma(z))^2} - \frac{2\gamma'(z)(\gamma(z)\beta'(z) - \beta(z)\gamma'(z))}{(\gamma(z))^3}$$

Evaluating $Y''(z)$ at $z = 1$ yields

$$\overline{Y^2} - \overline{Y} = \beta''(1) - \gamma''(1) - 2\gamma'(1)(\beta'(1) - \gamma'(1))$$

Using (5.4), (5.5), (5.6), (5.7) and (5.8), we find that

$$\overline{Y^2} = \frac{(1-\lambda)(\overline{X^2} + \lambda^2\overline{X}^3 - 2\lambda\overline{X}^2)}{(1-\lambda\overline{X})^3}$$

From (5.8), we see that

$$\sigma_Y^2 = \frac{(1-\lambda)(\overline{X^2} - \lambda\overline{X}^2 - \overline{X}^2 + \lambda\overline{X}^3)}{(1-\lambda\overline{X})^3}$$

We are now ready to examine the system time of a message as a function of the message length.

5.3 The LCFS System Time

In this section, we will examine the system time of a message under LCFS service as a function of the length of the message. In the subsequent section, we will make a comparison with round robin service for several choices of message length distribution and arrival rate.

Consider the arrival of a message of length τ. The system time $U(\tau)$ of this message is

$$U(\tau) = \tau + Y_1 + \cdots + Y_K$$

where K is the number of sub-busy periods initiated during the lifetime of the message. The transform of K is

$$K(z) = \left(\frac{1-\lambda}{1-\lambda z}\right)^{\tau - 1}$$

It is easily shown that

$$\overline{K} = \frac{\lambda(\tau - 1)}{1 - \lambda}$$

and that

$$\sigma_K^2 = \frac{\lambda(\tau - 1)}{(1 - \lambda)^2}$$

The mean of $U(\tau)$ equals

$$\begin{aligned}
\overline{U(\tau)} &= \tau + \overline{K}\,\overline{Y} \\
&= \frac{\tau - 1}{1 - \lambda\overline{X}} + 1
\end{aligned}$$

The variance of $U(\tau)$ can be found as

$$\begin{aligned}
\sigma_{U(\tau)}^2 &= \overline{K}\sigma_Y^2 + \overline{Y}^2\sigma_K^2 \\
&= \frac{\lambda(\sigma_X^2 + (1 - \lambda)\overline{X}^2)}{(1 - \lambda\overline{X})^3}(\tau - 1)
\end{aligned} \tag{5.9}$$

Recalling that $V(\tau)$is the system time of a typical length τ message under round robin service, we see that $\overline{U(\tau)} = \overline{V(\tau)}$, as expected. In addition, we see that both the mean and variance of the system time of a length τ message grows linearly with τ. As a result, shorter messages receive better service with respect to the mean and variance of the system time.

5.4 A Comparison of LCFS and Round Robin

We now can compare Round Robin and LCFS. Since the mean system time for a length τ message is the same for both systems, our comparison of the two disciplines will focus on the variance of the system time. Of course, for a message of some given length, smaller system time variance corresponds to better service. Moreover, in the context of our desire to provide better service to short messages, we would consider a discipline to be superior if it provided smaller system time variance to short messages, even at the expense of greater variance for long messages.

In Chapter 4, we found a procedure that allows us to compute the round robin system time variance. We also found an upper bound for the round robin system time variance. In fact, (4.37) simply says that

$$\sigma^2_{V(\tau)} \leq \sigma^2_{U(\tau)}$$

Moreover, (4.36) implies that

$$\sigma^2_{U(\tau+1)} - \sigma^2_{V(\tau+1)} \geq \sigma^2_{U(\tau)} - \sigma^2_{V(\tau)}$$

That is, the round robin system time variance is always smaller than that for LCFS and the difference between the two is nondecreasing in the message length. It is still of some interest to identify the extent to which round robin does better than LCFS. Of course, this examination also indicates the merit of our upper bound for $\sigma^2_{V(\tau)}$. We will consider geometric and uniform message length distributions.

5.4.1 Geometric Message Lengths

Suppose that the message length X has the distribution

$$g(x) = (1-\mu)\mu^{x-1} \quad (x \geq 1)$$

Note that

$$\overline{X} = \frac{1}{1-\mu}$$

As we have said, increasing $\overline{X}$ corresponds to a more bursty arrival process. In Figure 5-1, we plot $\sigma^2_{V(\tau)}$ and $\sigma^2_{U(\tau)}$ as a function of τ. In this case, we have chosen a relatively short average message length to emphasize how long messages are treated similarly under the two disciplines. We observe that round robin is somewhat better than LCFS at every message length.

In Figure 5-2, we focus our attention on the service received by short messages when the average message is very long. In this instance, we see that for messages that are much shorter than average, round robin provides significantly reduced system variance. This is perhaps not so surprising since under LCFS service, a short message will have a very long system time on those occasions that it is preempted by a long message.

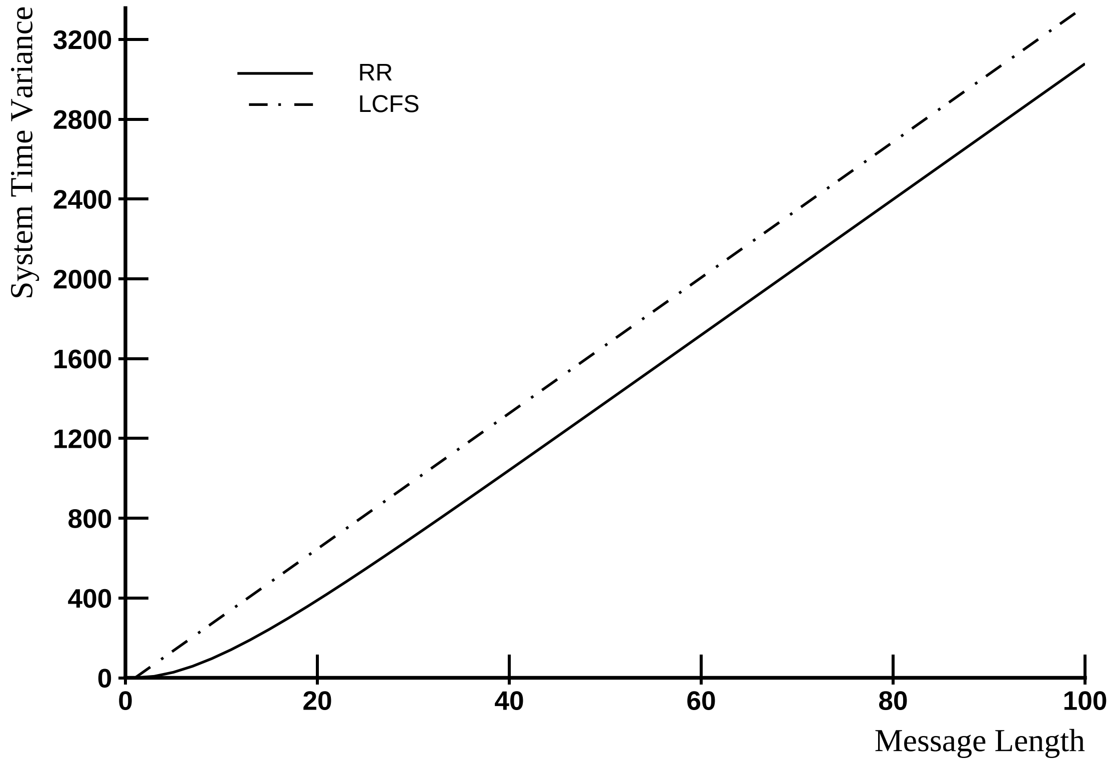

Figure 5-1: X: Geometric, $\overline{X} = 5$, $\lambda\overline{X} = 1/2$

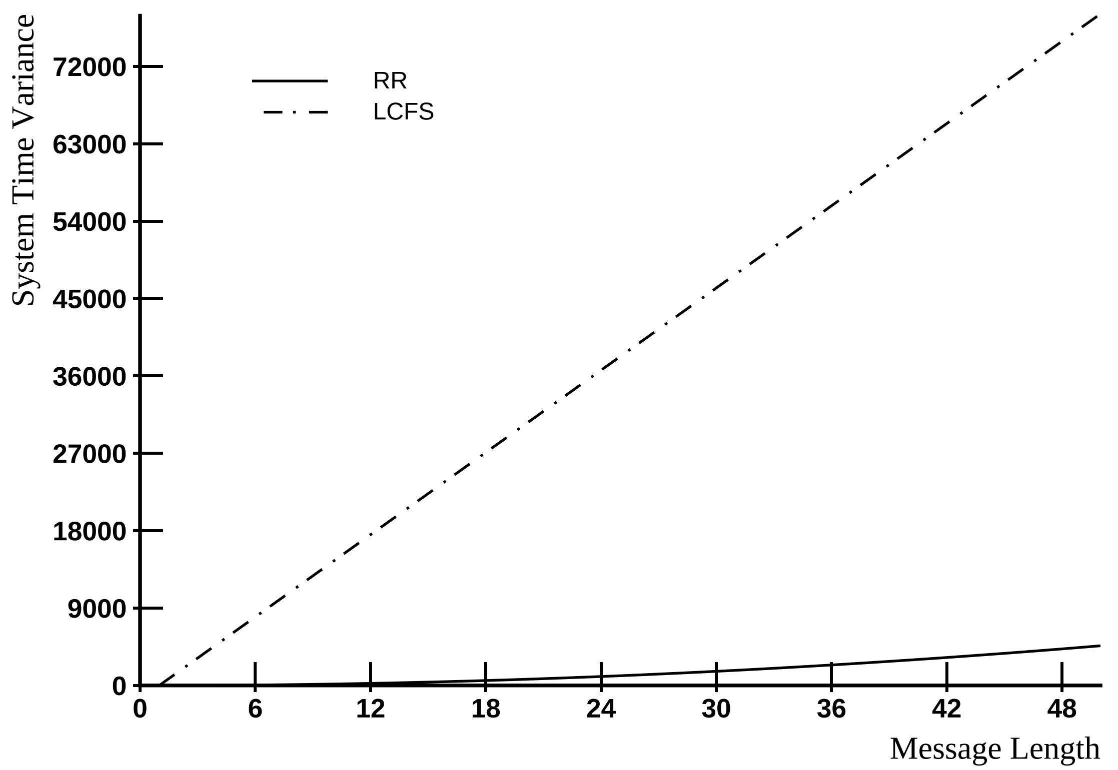

Figure 5-2: X: Geometric, $\overline{X} = 200$, $\lambda\overline{X} = 1/2$

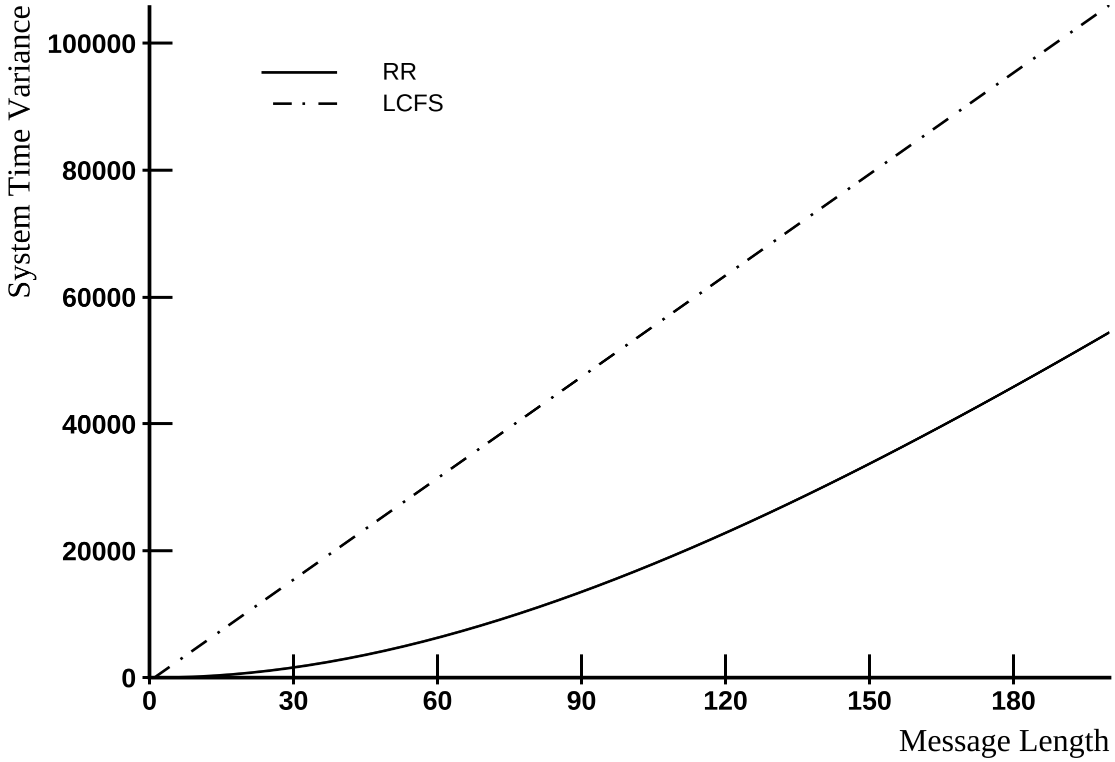

Figure 5-3: X: Uniform [1,200], $\lambda\overline{X} = 1/2$

5.4.2 Uniform Message Lengths

Let the message length X be distributed as

$$g(x) = \frac{1}{T} \quad (1 \leq x \leq T)$$

As T increases, short messages become rarer and the traffic becomes more bursty. As always, the interesting question is to what extent service for a short message is degraded as the burstiness increases. Figure 5-3 compares $\sigma^2_{V(\tau)}$ and $\sigma^2_{U(\tau)}$ as a function of τ. In this case, we see that round robin provides significantly better service than LCFS, especially to short messages.

5.5 Permutation Service

As we have shown, two very different service strategies, round robin and LCFS, have the same stationary distribution. This suggests there may be other service disciplines with the same stationary distribution. In this section, we identify a class of queues, including both round robin and LCFS, that have the same stationary distribution. For reasons that will become apparent, we will say queues of this class provide *permutation service.*

The basic model for the permutation queue is the same as that for the multiple class LCFS queue. These assumptions will now be repeated. During each slot, a message of class c arrives with probability λ_c. No more than one message can arrive in any slot and the probability of no message arriving during a slot is $1-\lambda$ where $\lambda = \sum_{c=1}^{C} \lambda_c$. Each class c message has an independent integer packet length X_c described by $g_c(x) = \Pr\{X_c = x\}$ and $\overline{G}_c(x) = P\{X_c > x\}$ such that $\overline{G}_c(0) = 1$. As before, given that a class c message has already had w packets sent, the conditional probability that the message's service requirement will be fulfilled after its next packet is sent will be denoted by

$$\begin{aligned} r_c(w) &= P\{X_c = w+1 | X_c > w\} \\ &= \frac{g_c(w+1)}{\overline{G}_c(w)} \end{aligned}$$

We define $Q_1, Q_2, \ldots$ such that Q_j is a j element permutation. Similarly, we define $Q_1^{-1}, Q_2^{-1}, \ldots$ as the set of inverse permutations. That is, given $y_1, \ldots, y_j$, we have

$$Q_j^{-1}(Q_j(y_1, \ldots, y_j)) = y_1, \ldots, y_j$$

Given Q_j, it should be apparent that Q_j^{-1} is well defined.

The basic sequence of service, departure and arrival is carried over from the round robin and LCFS systems. However, the permutation queue permutes the order of the backlogged messages. As before, let $n, y_1, \ldots, y_n$ represent the state of the queue, where n is the number of messages in the system and $y_i = (w_i, c_i)$ such that for the message in position i, w_i is the number of packets already transmitted and c_i is the class of the message. Suppose the state of the queue at the start of slot t is $n, y_1, \ldots, y_n$. In the transition to slot $t+1$, the following sequence of events occurs:

1. The message at the front of the queue has one packet sent.

2. Following service, this message will depart if all of its packets have been sent; otherwise, the elements of the queue are reordered as $Q_n((w_1+1, c_1), y_2, \ldots, y_n)$.

3. If a new message arrives at the end of slot t, it is placed at the front of the queue to begin service at time $t+1$.

We conjecture that the reverse time system is a permutation system that uses the inverse permutations $Q_1^{-1}, Q_2^{-1}, \ldots$

Conjecture 6 *The reverse time process is a permutation system. At the start of slot t, a message of class c arrives with probability λ_c. No more than one message can arrive in any slot and the probability of no message arriving during a slot is $1 - \lambda$ where $\lambda = \sum_{c=1}^{C} \lambda_c$. Each class c message has an independent service time distributed according to G_c. The state of the queue can be represented by $n, y_1, \ldots, y_n$ where n is the number of messages in the system and $y_i = (w_i, c_i)$ such that w_i is the residual service requirement in packets and c_i is the class of the message in position i. Suppose the state of the queue at the end (in reverse time) of slot $t+1$ equals $n, y_1, \ldots, y_n$. During slot t, the following sequence of events occurs:*

1. *The message at the front of the queue departs if $w_1 = 0$.*

2. *If a new message arrives, it is inserted at the front of the queue. Otherwise, if the messages in the queue are reordered in the following way. If $w_1 = 0$, implying the front message has already departed, the remaining messages in the queue are reordered as $Q_{n-1}^{-1}(y_2, \ldots, y_n)$. If the front message did not depart, that is $w_1 > 0$, then the messages are reordered as $Q_n^{-1}(y_1, \ldots, y_n)$.*

3. *The message at the front of the queue has one packet sent.*

The stationary distribution for the permutation queue is

$$\pi_{n,y_1,\ldots,y_n} = \pi_\phi \prod_{i=1}^{n} \frac{\lambda_{c_i}}{1-\lambda} \overline{G}_{c_i}(w_i) \tag{5.10}$$

where π_ϕ is the empty state probability.

The proof of Conjecture 6 is very similar to those for the round robin and LCFS queues. We include the proof in the following subsection, although the reader may wish to skip it.

5.5.1 Proof of Conjecture 6

Of course, we will use Theorem 1 to prove Conjecture 6. That is, for all neighboring states u and v, we will verify that the staionary distribution (5.10) satisfies

$$\pi_u P_{uv} = \pi_v P_{vu}^* \tag{5.11}$$

The four basic transitions that we must consider are:

- The front message is served and departs. No new message arrives.

  For the reverse transition, a new message of class c_1 and length $w_1 + 1$ must arrive to return to the state $n, y_1, \ldots, y_n$. In this instance, the transition pair is

$$n, y_1, \ldots, y_n \;\underset{\lambda_{c_1} g_{c_1}(w_1+1)}{\overset{(1-\lambda) r_{c_1}(w_1)}{\rightleftarrows}}\; n-1, y_2, \ldots, y_n$$

  This transition pair implies that $\pi_{n,y_1,\ldots,y_n}$ must satisfy

$$\pi_{n,y_1,\ldots,y_n}(1-\lambda)g_{c_1}(w_1+1)/\overline{G}c_1(w_1) = \lambda_{c_1} g_{c_1}(w_1+1)\pi_{n-1,y_2,\ldots,y_n}$$

  Rewriting yields

$$\pi_{n,y_1,\ldots,y_n} = \frac{\lambda_{c_1}}{1-\lambda}\overline{G}_{c_1}(w_1)\pi_{n-1,y_2,\ldots,y_n} \tag{5.12}$$

  It is easily verified that the distribution (5.10) satisfies (5.12).

- The front message is served but does not depart. No new message arrives.

  For the reverse transition, we must not have an arrival to ensure that the inverse permutation occurs and the correct message is served. This state transition pair corresponds to

$$n, y_1, \ldots, y_n \;\underset{1-\lambda}{\overset{(1-\lambda)[1-r_{c_1}(w_1)]}{\rightleftarrows}}\; n, Q_n((w_1+1, c_1), y_2, \ldots, y_n)$$

  We must check that

$$\pi_{n,y_1,\ldots,y_n}(1-\lambda)[1-r_{c_1}(w_1)] = (1-\lambda)\pi_{n,Q_n((w_1+1,c_1),seqy2n)}$$

  Since $1 - r_{c_1}(w_1)$ can be rewritten as $\overline{G}_{c_1}(w_1+1)/\overline{G}_{c_1}(w_1)$, we must have

$$\frac{\overline{G}_{c_1}(w_1+1)}{\overline{G}_{c_1}(w_1)}\left(\pi_\phi \prod_{i=1}^{n} \frac{\lambda_{c_i}}{1-\lambda}\overline{G}_{c_i}(w_i)\right) = \left(\pi_\phi \prod_{i=2}^{n} \frac{\lambda_{c_i}}{1-\lambda}\overline{G}_{c_i}(w_i)\right)\frac{\lambda_{c_1}}{1-\lambda}\overline{G}_{c_1}(w_1+1)$$

  which holds by cancellation.

- The front message is served and departs. A new class c message arrives.

  In reverse time, the front message departs immediately. To return to the state

$n, y_1, \ldots, y_n$, there must be a new class c_1 arrival requiring exactly $w_1 + 1$ units of service. The corresponding transition pair is

$$n, y_1, \ldots, y_n \quad \underset{\lambda_{c_1} g_{c_1}(w_1+1)}{\overset{\lambda_c r_{c_1}(w_1)}{\rightleftarrows}} \quad n, (0, c), y_2, \ldots, y_n$$

Since $\overline{G}_c(0) = 1$, we see that

$$\begin{aligned} \lambda_c \frac{g_{c_1}(w_1+1)}{\overline{G}_{c_1}(w_1)} \pi_{n,y_1,\ldots,y_n} &= \lambda_{c_1} g_{c_1}(w_1+1) \frac{\lambda_c}{1-\lambda} \left(\pi_\phi \prod_{i=2}^{n} \frac{\lambda_{c_i}}{1-\lambda} \overline{G}_{c_i}(w_i) \right) \\ &= \lambda_{c_1} g_{c_1}(w_1+1) \pi_{n,(0,c),y_2,\ldots,y_n} \end{aligned}$$

- The front message is served but does not depart. A new class c message arrives.

  For the reverse transition, the front message will immediately depart and the correct message will served iff no new message arrives. The transitions are

$$n, y_1, \ldots, y_n \quad \underset{1-\lambda}{\overset{\lambda_c[1 - r_{c_1}(w_1)]}{\rightleftarrows}} \quad n+1, (0, c), Q_n((w_1+1, c_1), y_2, \ldots, y_n)$$

  Substituting $\overline{G}_{c_1}(w_1+1)/\overline{G}_{c_1}(w_1)$ for $1 - r_{c_1}(w_1)$, we see that

$$\begin{aligned} \pi_{n,y_1,\ldots,y_n} \lambda_c [1 - r_{c_1}(w_1)] &= \lambda_c \left(\pi_\phi \prod_{i=2}^{n} \frac{\lambda_{c_i}}{1-\lambda} \overline{G}_{c_i}(w_i) \right) \frac{\lambda_{c_1}}{1-\lambda} \overline{G}_{c_1}(w_1+1) \\ &= (1-\lambda) \pi_{n+1,(0,c),Q_n((w_1+1,c_1),y_2,\ldots,y_n)} \end{aligned}$$

  since $\overline{G}_c(0) = 1$. This proves our conjecture for the multiple class permutation queue.

5.5.2 Properties of Permutation Queues

We have verified that the stationary distribution (5.10) for any permutation queue is identical to that for the round robin and LCFS systems. Moreover, by an appropriate choice of permutations $Q_1, Q_2, \ldots$, we can choose to represent either LCFS or round robin as a permutation queue. In particular, for round robin, we simply let Q_n represent the rotation of message $i > 1$ to position $i - 1$ and message 1 to position n. For LCFS, the permutation Q_n is simply the identity function. That is, under

LCFS, the messages are never reordered. Other disciplines can also be implemented. For example, we might choose to perform round robin among the messages in the first k positions. More unusual examples can also be constructed. We could choose $Q_1, Q_2, \ldots$ such that round robin is used when the number of messages in the system is even and LCFS is used when the number is odd.

We observe that the state space for every permutation queue is the same. That is, $w_1 \geq 0$ and $w_i \geq 1$ for all $i > 1$. Consequently, all of the marginal properties of the multiple class round robin system described in Section 3.2 hold for all permutation queues as well. In particular, the marginal distribution for the number of messages of a certain class and the average delay for a message of a certain class must be the same. In short, for any permutation queue, the system time of a typical message of length τ, $V_Q(\tau)$ satisifes

$$\overline{V_Q(\tau)} = \frac{\tau - 1}{1 - \lambda \overline{\overline{X}}} + 1$$

One can conclude that every permutation system provides length discrimination. When one pauses to consider arbitrary choices for $Q_1, Q_2, \ldots$, this result is very surprising.

5.6 Conclusions

At first glance, one would believe that LCFS would not be an appropriate service discipline for a link with bursty traffic since there is always the possibility that a short message will endure an extraordinarily long delay if it is preempted by a long message. However, we have verified that this possibility is more than offset by the likelihood that the short message will leave the system without ever being preempted. As a result, LCFS is a surprisingly good service discipline for bursty traffic. Of course, one should keep in mind that round robin always provides less variable service, particularly for short messages. That is, round robin is a better service discipline than LCFS.

We have also found that round robin and LCFS belong to a class of systems that we call permutation queues. All such systems have the same stationary distribution and the same state space. As a consequence, all such queues have the same marginal properties.

Chapter 6

A Round Robin Queue with Blocking

We can also impose a blocking rule on the round robin system. Let b be a positive integer and consider the multiple class round robin queue of Chapter 3 with the following additional condition:

- At the instant of an arrival, if the number of messages in the system equals b, do not let the arriving message enter the queue.

The blocked messages are effectively thrown away. We claim that the reverse time system is also unchanged except that new arrivals are blocked whenever the number of messages at the arrival instant equals b. Recall that during any slot, a message departs the queue the instant *before* a new message arrives. Consequently, even if there are b messages in the system at the start of slot s, it is possible to accept an arrival during slot s. This can occur when the front message departs the instant before the new message arrives. In this chapter, we will find the stationary distribution for the blocking queue as well as the probability that an arriving message is blocked.

6.1 The Blocking Queue Stationary Distribution

We will show that the stationary distribution is changed only by a new normalization constant. That is, the stationary distribution is

$$\pi_{n,y_1,\ldots,y_n} = \pi_\phi(b) \prod_{i=1}^{n} \frac{\lambda_{c_i}}{1-\lambda} \overline{G}_{c_i}(w_i) \tag{6.1}$$

where $\pi_\phi(b)$ is a normalization constant that depends on the blocking parameter b.

To prove this claim, we will use Theorem 1. That is, for every pair of states u and v, we must show that

$$\pi_u P_{uv} = \pi_v P^*_{vu} \tag{6.2}$$

For all states u and v that contain fewer than b messages, we have already shown that (6.2) is satisfied by the distribution in (6.1). Thus, we need to verify (6.2) when state u or v contains b messages. We will now describe these transitions.

The state transition from u with $b-1$ messages in the system to v with b messages can be depicted as

$$b-1, y_1, \ldots, y_{b-1} \quad \underset{1-\lambda}{\overset{\lambda_c(1-r_{c_1}(w_1))}{\rightleftarrows}} \quad b, (0,c), y_2, \ldots, y_{b-1}, (w_1+1, c_1)$$

since λ_c is the probability of a class c message arrival and $1-r_{c_1}(w_1)$ is the probability that the front message does not depart. In reverse time, we return to state u as long as there is no arrival. Note that

$$\begin{aligned} \pi_{b-1,y_1,\ldots,y_{b-1}} \lambda_c \frac{\overline{G}_{c_1}(w_1+1)}{\overline{G}_{c_1}(w_1)} &= (1-\lambda)\frac{\lambda_c}{1-\lambda}\overline{G}_c(0)\pi_{b-1,y_2,\ldots,y_{b-1},(w_1+1,c_1)} \\ &= (1-\lambda)\pi_{b,(0,c),y_2,\ldots,y_{b-1},(w_1+1,c_1)} \end{aligned}$$

There is a second type of transition in which state u has b messages and state v has $b-1$ messages. This transition occurs when the front message departs and there is no new arrival. This transition pair is

$$b, y_1, \ldots, y_b \quad \underset{\lambda_{c_1} g_{c_1}(w_1+1)}{\overset{(1-\lambda)r_{c_1}(w_1)}{\rightleftarrows}} \quad b-1, y_2, \ldots, y_b$$

since in reverse time, there must be an arrival of a class c_1 message of length w_1+1. We verify that

$$\pi_{b,y_1,\ldots,y_b}(1-\lambda)\frac{g_{c_1}(w_1+1)}{\overline{G}_{c_1}(w_1)} = \lambda_{c_1} g_{c_1}(w_1+1)\pi_{b-1,y_2,\ldots,y_b}$$

The two remaining transitions to consider occur when both states u and v have b messages. First, we examine the transition that arises when there is neither arrival nor departure. This state transition pair is

$$b, y_1, \ldots, y_b \underset{1}{\overset{1 - r_{c_1}(w_1)}{\rightleftarrows}} b, y_2, \ldots, y_b, (w_1 + 1, c_1)$$

For the forward transition, the front message does not depart with probability $1 - r_{c_1}(w_1)$. Given that the front message does not depart, a new arrival cannot enter the system. Note that $w_2 > 0$ since in forward time, every message always has one packet transmittted upon arriving. As a result, in the reverse time system, the front message, that is the message with residual length w_2, cannot depart. As a result, no new arrival can enter the queue so the rear message must be rotated and served. It is easily seen that

$$\pi_{b,y_1,\ldots,y_b} \frac{\overline{G}_{c_1}(w_1 + 1)}{\overline{G}_{c_1}(w_1)} = \pi_{b,y_2,\ldots,y_b,(w_1+1,c_1)}$$

The last transition pair to consider occurs when the queue has b messages and there is a departure and an arrival within a slot. The transition pair is

$$b, y_1, \ldots, y_b \underset{\lambda_{c_1} g_{c_1}(w_1 + 1)}{\overset{\lambda_c r_{c_1}(w_1)}{\rightleftarrows}} b, (0, c), y_2, \ldots, y_b$$

The probability that the front message departs is simply $r_{c_1}(w_1)$. Given that the front message departs, a new arrival is not blocked so that the probability of a class c message arrival is just λ_c. For the reverse transition, the front message immediately departs. The system returns to the original state if a class c_1 message of length $w_1 + 1$ arrives. We see that

$$\begin{aligned} \pi_{b,y_1,\ldots,y_b} \lambda_c r_{c_1}(w_1) &= \lambda_c g_{c_1}(w_1 + 1) \frac{\lambda_{c_1}}{1 - \lambda} \pi_{b-1,y_2,\ldots,y_b} \\ &= \lambda_{c_1} g_{c_1}(w_1 + 1) \frac{\lambda_c}{1 - \lambda} \overline{G}_c(0) \pi_{b-1,y_2,\ldots,y_b} \\ &= \lambda_{c_1} g_{c_1}(w_1 + 1) \pi_{b,(0,c),y_2,\ldots,y_b} \end{aligned}$$

This proves our claim for the stationary distribution. However, we still must find the new empty state probability $\pi_\phi(b)$. To do this, note that for $1 \leq n \leq b$,

$$P_N(n) = \sum_{c_1 \in \mathcal{C}} \sum_{w_1=0}^{\infty} \sum_{c_2 \in \mathcal{C}} \sum_{w_2=1}^{\infty} \cdots \sum_{c_n \in \mathcal{C}} \sum_{w_n=1}^{\infty} \pi_\phi(b) \prod_{i=1}^{n} \frac{\lambda_{c_i}}{1 - \lambda} \overline{G}_{c_i}(w_i)$$

$$
\begin{aligned}
&= \pi_\phi(b)\left(\sum_{c_1\in\mathcal{C}}\frac{\lambda_{c_1}}{1-\lambda}\sum_{w_1=0}^{\infty}\overline{G}_{c_1}(w_1)\right)\prod_{i=2}^{n}\left(\sum_{c_i\in\mathcal{C}}\frac{\lambda_{c_i}}{1-\lambda}\sum_{w_i=1}^{\infty}\overline{G}_{c_i}(w_i)\right)\\
&= \pi_\phi(b)\left(\sum_{c_1\in\mathcal{C}}\frac{\lambda_{c_1}\overline{X}_{c_1}}{1-\lambda}\right)\prod_{i=2}^{n}\left(\sum_{c_i\in\mathcal{C}}\frac{\lambda_{c_i}(\overline{X}_{c_i}-1)}{1-\lambda}\right)\\
&= \pi_\phi(b)\left(\rho+\frac{\lambda}{1-\lambda}\right)\rho^{n-1}
\end{aligned}
$$

Applying $P_N(0)=\pi_\phi(b)$ and $\sum_{n=0}^{b}P_N(n)=1$ yields

$$
P_N(n)=\begin{cases}\dfrac{(1-\rho)(1-\lambda)}{1-(1-(1-\rho)(1-\lambda))\,\rho^b} & n=0\\[2ex] \dfrac{(1-\rho)(1-\lambda)}{1-(1-(1-\rho)(1-\lambda))\,\rho^b}\left(\rho+\dfrac{\lambda}{1-\lambda}\right)\rho^{n-1} & 1\le n\le b\end{cases}
$$

At first glance, it would appear that the steady state blocking probability is simply $P_N(b)$. However, this is not the case since it is possible to accept a new arrival during a slot even if the system has b messages in the queue at the start of the slot. In the forward time system, an arriving message is blocked in some slot iff the system contained b messages at the start of the slot and the front customer does not depart. Of course, the blocking probability must be the same in the reverse time system. In reverse time, an arriving message is blocked iff the system contains b messages at the start of the slot and the front message does not depart. The front message does not depart iff $w_1>0$. Consequently, P_b, the blocking probability, is simply the probability that $N=b$ and $w_1>0$.

$$
\begin{aligned}
P_b &= \sum_{c_1\in\mathcal{C}}\sum_{w_1=1}^{\infty}\cdots\sum_{c_b\in\mathcal{C}}\sum_{w_b=1}^{\infty}\pi_\phi(b)\prod_{i=1}^{b}\frac{\lambda_{c_i}}{1-\lambda}\overline{G}_{c_i}(w_i)\\
&= \pi_\phi(b)\prod_{i=1}^{b}\left(\sum_{c_i\in\mathcal{C}}\frac{\lambda_{c_i}}{1-\lambda}\sum_{w_i=1}^{\infty}\overline{G}_{c_i}(w_i)\right)\\
&= \pi_\phi(b)\prod_{i=1}^{b}\left(\sum_{c_i\in\mathcal{C}}\frac{\lambda_{c_i}(\overline{X}_{c_i}-1)}{1-\lambda}\right)\\
&= \pi_\phi(b)\rho^b\\
&= \frac{(1-\rho)(1-\lambda)\rho^b}{1-(1-(1-\rho)(1-\lambda))\,\rho^b}
\end{aligned}
$$

Observing that $(1-\rho)(1-\lambda) = 1 - \lambda\overline{X}$, we can reduce P_b to

$$P_b = \frac{(1-\lambda\overline{X})\rho^b}{1-\lambda\overline{X}\rho^b}$$

6.2 Remarks

In our model, a message can be of arbitrary length. As a result, the blocking queue does not limit the number of packets in the system. However, for a message that is not blocked, there is an upper bound to the system time. In particular, it can be shown that for $V(\tau)$, the system time of an unblocked length τ message,

$$V(\tau) \leq b(\tau - 1) + 1$$

This is a significant advantage for round robin service. For the LCFS queue, blocking does not provide this benefit. That is, for a LCFS queue with blocking, the system time of a message cannot be upper bounded because that message may be preempted by an arbitrarily large number of other messages.

We have verified that the reverse time arrival process is memoryless and identical to the forward time arrival process. Consequently, the forward time departure process is identical to the forward time arrival process. One might think that the forward time departure process would be of a lower rate than the arrival process since some of the forward time arrivals get blocked. In fact, what happens is that a blocked arrival immediately departs as though all of its packets were instantly transmitted. The forward time departure process includes these departures of blocked messages.

For a single queue, this anomaly makes no difference. However, using this approach to analyze a network of round robin blocking queues results in an unrealistic model. In particular, if a message is blocked at a link, that message would immediately depart that link and arrive at the next link on its route. The blocking link would simply be bypassed. Although one could analyze such a system, it makes little real sense. Consequently, the results to be described for networks of round robin queues cannot be extended to queues with blocking.

Lastly, we note that this analysis easily extends to all permutation queues.

Chapter 7

Multiple Sessions – One Link

7.1 Multiple Message Arrivals

Suppose the queue is used by k sessions such that during each time unit, session j submits a message with probability θ_j, independent of all other events. For each session j, the number of packets in a message will be represented by L_j, a random variable that is independent of all arrival times and of all other message lengths. This arrival process allows the possibility of multiple message arrivals within a single time slot. The corresponding event in the reverse time system has multiple messages departing in the same slot. Since the queue can provide service to only one message in a slot, each of the additional messages in the reverse time multiple departure must have completed service in an earlier slot but not departed. In this circumstance, it is not clear what kind of service discipline will yield a reverse time system that is easy to describe. We will consider one type of service that does have desirable analytical properties.

We can include this system under the model of the previous section if we consider multiple messages arriving in the same slot to be a single arrival of a *message group*. Round robin service is provided to the message groups. In each slot, one packet of a message group is transmitted. If every packet of each message in that group has been transmitted then the message group departs. Otherwise, the message group is rotated to the back of the queue.

This approach has some disadvantages. Although a message may have had all of its packets sent, the message does not leave the system until every other message in the group has had its packets sent. This seems particularly unfair to a short message that is grouped together with a very long message. However, this would be a rare event whose importance we will be able to analyze precisely.

The set of message group classes is the set of all nonempty subsets of sessions and the class of a message group is uniquely determined by the set of sessions that contributed messages to that group. The service requirement of a message group has a distribution defined by the sum of lengths of the constituent messages. For our k sessions, we have defined $2^k - 1$ arrival classes. We could apply directly the results of the previous section to find the distribution of the number of message groups in the queue for each class. However, our primary interest is in the number of messages of each session in the queue. Define $\mathcal{C}_1$ as the set of classes that include a session 1 message and N_1 as the number of arrivals in the system that belong to one of the classes in $\mathcal{C}_1$. Each class c arrival with $c \in \mathcal{C}_1$ contains exactly one session 1 message and each session 1 message is contained in a message group of class c such that $c \in \mathcal{C}_1$. Hence, the number of session 1 messages in the system is equal to N_1. In Section 3.2, we found the distribution of N', the number of queued messages with class $c \in \mathcal{C}'$. With the following definitions

$$\begin{aligned}
\rho_c &= \frac{\lambda_c(\overline{X}_c - 1)}{1-\lambda}, \\
\rho_1 &= \sum_{c \in \mathcal{C}_1} \rho_c, \\
\rho &= \sum_{c \in \mathcal{C}} \rho_c,
\end{aligned}$$

the distribution for N_1 is precisely that found for N' with $\rho' = \rho_1$, $\mathcal{C}' = \mathcal{C}_1$ and $\lambda' = \theta_1$. Hence,

$$P_{N_1}(n) = \begin{cases} \dfrac{(1-\rho)(1-\theta_1)}{1-\rho+\rho_1} & (n=0) \\ \left(1 - \dfrac{(1-\rho)(1-\theta_1)}{1-\rho+\rho_1}\right)\left(1 - \dfrac{\rho_1}{1-\rho+\rho_1}\right)\left(\dfrac{\rho_1}{1-\rho+\rho_1}\right)^{n-1} & (n \geq 1) \end{cases}$$

We now will evaluate ρ_1. For an arbitrary slot, let S_j, $1 \leq j \leq k$, be an indicator random variable such that $S_j = 1$ if a session j message arrives in the slot and $S_j = 0$ otherwise. So

$$P_{S_j}(s) = \begin{cases} \theta_j & (s=1) \\ 1-\theta_j & (s=0) \end{cases}$$

Each class $c \in \mathcal{C}$ can be uniquely denoted by the binary k-tuple $(s_1, \ldots, s_k)$ where $s_j = 1$ if a class c message group contains a session j message and $s_j = 0$ otherwise. Thus,

$$\overline{X}_{(s_1,\ldots,s_k)} = \sum_{j=1}^{k} s_j \bar{L}_j$$

Note that $1 - \lambda$ is the probability of no arrival which occurs iff none of the k sessions submits a message. Thus

$$1 - \lambda = \prod_{j=1}^{k} (1 - \theta_j) \tag{7.1}$$

We also note that

$$\begin{aligned} \lambda_{(s_1,\ldots,s_k)} &= \prod_{j=1}^{k} P_{S_j}(s_j) \\ &= P_{S_1 \ldots S_k}(s_1, \ldots, s_k) \end{aligned}$$

Hence, we find that

$$\begin{aligned} \rho_{(s_1,\ldots,s_k)} &= \frac{\lambda_{(s_1,\ldots,s_k)}(\overline{X}_{(s_1,\ldots,s_k)} - 1)}{1 - \lambda} \\ &= \frac{1}{1-\lambda} P_{S_1 \ldots S_k}(s_1, \ldots, s_k) \left(\sum_{j=1}^{k} s_j \bar{L}_j - 1 \right) \end{aligned}$$

Since $\mathcal{C}_1$ is the set of all possible classes $(s_1, \ldots, s_k)$ such that $s_1 = 1$, we can write

$$\begin{aligned} \rho_1 &= \sum_{(s_1,\ldots,s_k) \in \mathcal{C}_1} \rho_{(s_1,\ldots,s_k)} \\ &= \frac{1}{1-\lambda} \sum_{s_2=0}^{1} \cdots \sum_{s_k=0}^{1} P_{S_1 \ldots S_k}(1, s_2, \ldots, s_k) \left(\bar{L}_1 + \sum_{j=2}^{k} s_j \bar{L}_j - 1 \right) \end{aligned}$$

Applying $P_{S_1}(1) = \theta_1$, we have

$$\begin{aligned} \rho_1 &= \frac{\theta_1}{1-\lambda} \sum_{s_2=0}^{1} \cdots \sum_{s_l=0}^{1} P_{S_2 \ldots S_k}(s_2, \ldots, s_k) \left(\bar{L}_1 - 1 + \sum_{j=2}^{k} s_j \bar{L}_j \right) \\ &= \frac{\theta_1}{1-\lambda} \left(\bar{L}_1 - 1 + \sum_{j=2}^{k} E[S_j] \bar{L}_j \right) \\ &= \frac{\theta_1}{1-\lambda} \left(\bar{L}_1 - 1 + \sum_{j=2}^{l} \theta_j \bar{L}_j \right) \end{aligned}$$

We can define the session j packet arrival rate (in packets/slot) as

$$\bar{R}_j = \theta_j \bar{L}_j$$

and

$$\bar{R} = \sum_{j=1}^{k} \bar{R}_j$$

which is the average load on the system. In this case,

$$\rho_1 = \frac{\bar{R}_1 - \theta_1 + \theta_1(\bar{R} - \bar{R}_1)}{1 - \lambda}$$

Also, note that $\bar{R} = \sum_{c \in \mathcal{C}} \lambda_c \overline{X}_c$ so

$$\begin{aligned} \rho &= \frac{1}{1-\lambda} \sum_{c \in \mathcal{C}} \lambda_c (\overline{X}_c - 1) \\ &= \frac{\bar{R} - \lambda}{1 - \lambda} \end{aligned}$$

The distribution for N_1 becomes

$$P_{N_1}(n) = \begin{cases} \beta_1 & (n = 0) \\ (1 - \beta_1)(1 - \alpha_1)\alpha_1^{n-1} & (n \geq 1) \end{cases}$$

where α_1 is defined by

$$\begin{aligned} \alpha_1 &= \frac{\rho_1}{1 - \rho + \rho_1} \\ &= \frac{1}{1 - \theta_1} \left[\frac{\bar{R}_1 - \theta_1 + \theta_1(\bar{R} - \bar{R}_1)}{1 - (\bar{R} - \bar{R}_1)} \right] \end{aligned}$$

and β_1 is defined by

$$\begin{aligned} \beta_1 &= \frac{(1 - \rho)(1 - \theta_1)}{1 - \rho + \rho_1} \\ &= \frac{1 - \bar{R}}{1 - (\bar{R} - \bar{R}_1)} \end{aligned}$$

Note that session 1 has no special properties. Any of the other sessions could be relabeled as session 1 and the distribution for N_1 would be correct for the corresponding new values of θ_1 and $\bar{R}_1$. Defining

$$\begin{aligned}
\alpha_j &= \frac{1}{1-\theta_j}\left[\frac{\bar{R}_j - \theta_j + \theta_j(\bar{R} - \bar{R}_j)}{1-(\bar{R}-\bar{R}_j)}\right] \\
\beta_j &= \frac{1-\bar{R}}{1-(\bar{R}-\bar{R}_j)}
\end{aligned}$$

we can say that N_j, the number of session j messages in the system, is distributed as

$$P_{N_j}(n) = \begin{cases} \beta_j & (n=0) \\ (1-\beta_j)(1-\alpha_j)\alpha_j^{n-1} & (n \geq 1) \end{cases}$$

The distribution of N_j depends only on α_j and β_j. This implies that session i can affect $P_{N_j}(n)$ only by changing $\bar{R}_i$, its contribution to $\bar{R}$. In particular, whether session i submits long messages infrequently or short messages with great frequency has no effect on N_j (although it would have an effect on the distribution of the system time for session j messages). This result is very surprising considering the complicated process of lumping together concurrent message arrivals.

We can easily compute $\overline{N}_j = E[N_j]$ which happens to be

$$\begin{aligned}
\overline{N}_j &= \frac{1-\beta_j}{1-\alpha_j} \\
&= \frac{(1-\theta_j)\bar{R}_j}{1-\bar{R}}
\end{aligned}$$

Let $\overline{T_s(j)}$ and $\overline{T_w(j)}$ equal the average system time and the average waiting time for a session j message. We can verify that Little's theorem holds for this system so that $\overline{T_s(j)} = \overline{N}_j/\theta_j$. Recalling that the waiting time is defined as difference between the system time and the service time, we can write

$$\begin{aligned}
\overline{T_w(j)} &= \overline{T_s(j)} - \bar{L}_j \\
&= \frac{(\bar{R}-\theta_j)\bar{R}_j}{\theta_j(1-\bar{R})} \\
&= \left(\frac{\bar{R}}{1-\bar{R}}\right)\bar{L}_j - \frac{\bar{R}_j}{1-\bar{R}}
\end{aligned}$$

We now can see the extent to which bursty sessions receive inferior service. To isolate the effect of long messages from the effect of increased load, consider the case in which $\bar{R}_j$, the load offered by session j, and $\bar{R}$, the overall load, are held constant. In this instance, the average waiting time of a session j message grows linearly with the average message length. More importantly, the service received by session j, as measured by $\overline{N}_j$ and $\overline{T_w(j)}$, is independent of the burstiness of the other sessions.

7.2 A Bound on Multiple Message Groups

We can show that groups containing multiple messages are relatively rare. Let $\mathcal{C}_m$ represent the set of message groups containing multiple messages and let $\mathcal{C}_s$ be the set of message groups containing a single message. Furthermore, let N_m equal the number of message groups in the system with class $c \in \mathcal{C}_m$.

Our intention is to upper bound the probability that N_m is greater than zero. Once again, we apply the result found in Section 3.2 for the distribution of the number of messages (or in this case message groups) in a subset of the classes. In particular, from (3.5), we have that

$$\begin{aligned} P\{N_m > 0\} &= 1 - P\{N_m = 0\} \\ &= \frac{\lambda_m(1-\rho) + \rho_m}{1 - \rho + \rho_m} \end{aligned} \tag{7.2}$$

where

$$\rho_m = \sum_{c \in \mathcal{C}_m} \rho_c \tag{7.3}$$

and

$$\lambda_m = \sum_{c \in \mathcal{C}_m} \lambda_c$$

Note that

$$\begin{aligned} \lambda_m &= 1 - P\{\text{no arrival}\} - P\{\text{arrival of a single message}\} \\ &= 1 - \prod_{j=1}^{k}(1-\theta_j) - \sum_{i=1}^{k} \theta_i \prod_{j \neq i}(1-\theta_j) \end{aligned}$$

From (7.1), we recall that $1 - \lambda = \prod_{j=1}^{k}(1-\theta_j)$. This implies

$$\begin{aligned} \lambda_m &\leq 1 - (1-\lambda) - (1-\lambda)\sum_{i=1}^{k} \theta_i \\ &= \lambda - (1-\lambda)\epsilon \end{aligned}$$

where we have defined

$$\epsilon = \sum_{i=1}^{k} \theta_i$$

Observing that $\epsilon \geq \lambda$, we see that

$$\lambda_m \leq \lambda^2$$

For high speed networks, λ, the probability of an arrival in a slot, is typically small. Consequently, λ_m, the probability of a multiple arrival, is very small. However, this isn't sufficient to say that multiple arrivals are not very important. An arrival with multiple messages may be more likely to have a large service requirement, causing that arrival to stay in the queue a long time. So, we will verify that $P\{N_m > 0\}$ tends to be small.

To identify ρ_m, we observe from (7.3) that

$$\begin{aligned}
\rho_m &= \sum_{c \in \mathcal{C}} \rho_c - \sum_{c \in \mathcal{C}_s} \rho_c \\
&= \rho - \sum_{c \in \mathcal{C}_s} \frac{\lambda_c(\overline{X}_c - 1)}{1 - \lambda} \\
&= \rho - \frac{1}{1-\lambda} \sum_{i=1}^{k} \theta_i \left(\prod_{j \neq i} (1 - \theta_j) \right) \left(\bar{L}_i - 1 \right) \\
&\leq \rho - \sum_{i=1}^{k} \theta_i \left(\bar{L}_i - 1 \right) \\
&= \rho - \bar{R} + \epsilon
\end{aligned}$$

Substitution of this upper bound for ρ_m into (7.2) yields

$$P\{N_m > 0\} \leq \frac{\lambda_m(1-\rho) + \rho - \bar{R} + \epsilon}{1 - \bar{R} + \epsilon}$$

Recalling that

$$\rho = \frac{\bar{R} - \lambda}{1 - \lambda}$$

we find that

$$\begin{aligned}
P\{N_m > 0\} &\leq \frac{\epsilon \bar{R}}{1 - \bar{R} - \epsilon} \\
&\leq \epsilon \left(\frac{\bar{R}}{1 - \bar{R}} \right)
\end{aligned}$$

For high speed networks, ϵ is usually small. Consider the following example of a 100 Mb/s link serving 1000 sessions each with a 64 Kb/s data rate. We will assume a packet size of 1000 bits and that the average message has 10 packets. In this instance, for each session i, $\theta_i = 6.4 \times 10^{-5}$ so that $\epsilon = 0.064$. Hence $P\{N_m > 0\} < 0.113$. In short, most of the time, the system does not have any multiple message groups in the queue.

7.3 A Bound on Messages of One Session

We next show that it is also unusual for a session to have more than a single message in service at a link. As before, we will let N_j represent the number of session j messages in the queue. Note that

$$\begin{aligned} P\{N_j > 1\} &= 1 - P\{N_j = 0\} - P\{N_j = 1\} \\ &= \alpha_j(1 - \beta_j) \end{aligned}$$

where

$$\begin{aligned} \beta_j &= \frac{1 - \bar{R}}{(1 - \bar{R}) + \bar{R}_j} \\ \alpha_j &= \frac{1}{1 - \theta_j} \frac{\bar{R}_j - \theta_j + \theta_j(\bar{R} - \bar{R}_j)}{(1 - \bar{R}) + \bar{R}_j} \end{aligned}$$

By substitution, we find that

$$\begin{aligned} P\{N_j > 1\} &= \left(\frac{\bar{R}_j}{(1 - \bar{R}) + \bar{R}_j} \right)^2 - \frac{\theta_j \bar{R}_j (1 - \bar{R})}{(1 - \theta_j)\left((1 - \bar{R}) + \bar{R}_j\right)^2} \\ &< \left(\frac{\bar{R}_j}{(1 - \bar{R}) + \bar{R}_j} \right)^2 \end{aligned}$$

As long as the packet rate of a session is only a small fraction of the residual link capacity, that session will be very unlikely to have more than one message in the queue. For example, consider a link with utilization $\bar{R} = .99$ to which session j offers a load $\bar{R}_j = 10^{-4}$. In this instance, $P\{N_j > 1\} < 10^{-4}$. We can conclude that message round robin behaves like session round robin much of the time.

Chapter 8

Networks of Round Robin Queues

In this chapter, we will verify that in a network of round robin queues, the queue at each link behaves as though it were independent of the other queues in the network. That is, the stationary distribution for the network is simply the product of the stationary distributions of the individual queues.

We will examine networks of round robin queues by generalizing the method used to analyze a single queue used by multiple sessions. We are interested in the following two cases.

- *Fixed Message Lengths* When a message enters the network, its length is chosen from a distribution that may be unique to the session that submitted the message. This message will have the same length at every link in the network.

- *Random Message Lengths* Each time a message arrives at a link, its length is an independent random variable that is described by a distribution that may be unique to the session that submitted the message.

Clearly, the second model is less appropriate for data networks; however, it is appropriate for a number of other queueing networks. We will construct a framework that will allow us to verify our independence claim for both models.

8.1 The Round Robin Network

Briefly, we review our basic data network model. Each communication link provides an errorless point to point communication channel between a pair of nodes. Nodes can receive, store, and transmit fixed length *packets* of data. A packet will take precisely one unit of time, a *slot*, to be sent over a link. We will assume that the nodes are completely synchronized such that all nodes begin packet transmissions at the same instant.

A packet originates at an external source and passes through a pre-assigned sequence of links to a destination that is also outside of the network. A source-destination pair is called a *session* and the sequence of nodes and links used by a session is called a *route*. We will assume that a route never contains a cycle.

At the end of each slot, each source might submit a message consisting of a random number of packets to the first link on its route. For each session, the length of a message is an independent random variable with a distribution that may be unique to that session. To be precise, let $G_1, G_2, \ldots$ be a countable set of service requirement distributions. We say that a message is of *type* k if its message length is described by G_k. At the end of each slot, session j submits a type k message to the network with probability $\theta_j(k)$, independent of all other message arrivals. However, a session never submits more than one message to the network in any slot.

When multiple sessions use the same link, it becomes possible to have more than one message arrive at a link within a single slot. The messages that arrive at the same link in the same slot form a *message group*. We will call the session that submits a message the *owner* of that message. A message group can contain as many messages as there are sessions using a link. However, a message group typically contains a single message since the simultaneous arrival of multiple messages is a rare event. The service requirement of a message group equals the sum of the lengths of the constituent messages. The class of a message group specifies the type and the owner of each constituent message.

Round robin service is provided to the message groups and a message completes service and departs only when all of the messages of its group have been sent. The instant after a message completes service, the message either arrives at the next link on its route or exits the network if it has arrived at its destination. To be precise, at time t, the following sequence of events occurs at each link l.

1. The message group at the front of the queue has one packet sent.

2. Following service, this message group will depart if all of its packets have been sent. Otherwise, the message group will be rotated to the back of the queue. If the message group departs, each message of that message group instantaneously either arrives at the next link of its route or exits the network.

3. At the end of slot t, if any new messages arrive, either from external sources or other links, they are lumped together as a single message group that is placed at the front of the queue to begin service at time $t+1$.

We will number the links 1 through L and the sessions 1 through J. The state of link l can be represented by $u(l) = n(l), y(l,1), \ldots, y(l, n(l))$ where $n(l)$ is the number of message groups in the queue and $y(l,i) = (w(l,i), c(l,i))$ such that for the message group in position i at link l, $w(l,i)$ is the number of packets already transmitted and $c(l,i)$ is the class of the message group.

We conjecture that in the reverse time network there is a round robin queue at each link of the same type as in the forward time sytem.

Conjecture 7 *The reverse time process is also a round robin queueing system. At the end of each slot t, a type k message of session j arrives with probability $\theta_j(k)$, independent of all other message arrivals, at the last link of route j. No more than one message of a single session can arrive in any slot. At time t, the following sequence of events occurs:*

1. *The message group at the front of the queue departs if all of its packets have been sent. If the message group departs, each message of that message group instantaneously either arrives at the previous link of its forward route or exits the network.*

2. *If any messages arrive at the start of slot t, they are lumped together as a single message group. This message group would be inserted at the front of the queue. Otherwise, the message group at the back of the queue is rotated to the front.*

3. *The message group at the front of the queue has one packet sent.*

The system is still a discrete time Markov process with state $u = u(1), \ldots, u(L)$ where $u(l) = y(l,1), \ldots, y(l, n(l))$ and $y(l,i) = (w(l,i), c(l,i))$ such that for the message group in position i at link l, $w(l,i)$ is the the remaining integer service requirement and $c(l,i)$ is the class of the message group.

During each slot at every link l, a message group arrival of class $c(l,i)$ occurs with probability $\lambda_{c(l,i)}$. No more than one message group can arrive at a link in a single slot. Let $\pi_\phi(l)$ be the probability that the queue at link l is empty. The stationary *distribution for the network of round robin queues is*

$$\pi_u = \prod_{l=1}^{L} \pi_{u(l)}$$

where

$$\pi_{u(l)} = \pi_\phi(l) \prod_{i=1}^{n(l)} \left(\frac{\lambda_{c(l,i)}}{1-\lambda_l} \right) \overline{G}_{c(l,i)}(w(l,i)) \tag{8.1}$$

For a single queue, we found that the message group arrival process was the same in both forward time and reverse time. That is, in reverse time, session j message arrivals formed a Bernoulli process of rate θ_j. Note that arrivals in reverse time correspond exactly to departures in forward time, so that in any slot, a session j message completes service on the link with probability θ_j. Conjecture 3 states that the stationary distribution at each link is the same as it would be if that link were in isolation receiving Bernoulli arrivals from each of the sessions that use the link.

For link l in isolation, we have already verified that $\pi_{u(l)}$ is the stationary distribution. To prove Conjecture 7, we will show that for all pairs of neighboring network states u and v,

$$\pi_u P_{uv} = \pi_v P^*_{vu}$$

where P^*_{vu} is the v to u transition probability in the reverse time system. When we considered an isolated queue, we verified the limiting state distribution by considering four separate cases. To make the same argument for a network, we must consider all four of these cases at once at all links of the network simultaneously. As a consequence, the necessary bookkeeping is much more complicated.

8.2 Proof of Conjecture 7

We will rewrite our expression for $\pi_{u(l)}$. Define S_l as the set of sessions using link l. In addition, let $s(l,i,j)$ be the type of message that session j has contributed to the message group in position i at link l. If this message group does not contain a session j message, then we adopt the convention that $s(l,i,j) = 0$.

For each session j using a link, let $\theta_j(k)$ be the (conjectured) probability of an arrival of a session j message of type k at that link. In addition, we will designate

$$\theta_j(0) = 1 - \sum_{k=1}^{\infty} \theta_j(k)$$

to be the probability of no session j message arriving. Since the routes do not contain cycles, only one message of a session can arrive at a link in a single slot so that, by the conjecture,

$$\lambda_{c(l,i)} = \prod_{j \in S_l} \theta_j(s(l,i,j))$$

The overall message group arrival rate at link l, λ_l, satisfies

$$1 - \lambda_l = \prod_{j \in S_l} \theta_j(0)$$

since $1 - \lambda_l$ is the probability of no messages arriving. If we define

$$\hat{\theta}_j(k) = \frac{\theta_j(k)}{\theta_j(0)}$$

then we can write

$$\frac{\lambda_{c(l,i)}}{1 - \lambda_l} = \prod_{j \in S_l} \hat{\theta}_j(s(l,i,j))$$

As a result, from (8.1), we see that

$$\pi_{u(l)} = \pi_\phi(l) \prod_{i=1}^{n(l)} \left(\prod_{j \in S_l} \hat{\theta}_j(s(l,i,j)) \right) \overline{G}_{c(l,i)}(w(l,i))$$

We can define

$$\begin{aligned} \beta_{u(l)} &= \prod_{i=1}^{n(l)} \prod_{j \in S_l} \hat{\theta}_j(s(l,i,j)) \\ \gamma_{u(l)} &= \prod_{i=1}^{n(l)} \overline{G}_{c(l,i)}(w(l,i)) \end{aligned}$$

so that

$$\pi_{u(l)} = \pi_\phi(l) \beta_{u(l)} \gamma_{u(l)}$$

The conjectured limiting state distribution π_u for the network becomes

$$\begin{aligned} \pi_u &= \prod_{l=1}^{L} \pi_{u(l)} \\ &= \pi_\phi \prod_{l=1}^{L} \beta_{u(l)} \gamma_{u(l)} \end{aligned}$$

where π_ϕ, the probability that the system is empty, is defined as

$$\pi_\phi = \prod_{l=1}^{L} \pi_\phi(l)$$

To describe a transition from a given state u to an adjacent state v, we define the following random variables:

- A_j: If a new session j message enters the network, A_j is the type of that message; otherwise, $A_j = 0$.

- D_j: If a session j message exits the network, D_j is the type of that message; otherwise, $D_j = 0$.

- $A(l, j)$: If a session j message arrives at link l, $A(l, j)$ is the type of that message; otherwise, $A(l, j) = 0$.

- $D(l, j)$: If a session j message departs from link l, $D(l, j)$ is the type of that message; otherwise, $D(l, j) = 0$.

Note that if link l is the first link on the route of session j, then $A_j = A(l, j)$. Similarly, when link l is the last link on the route of session j, $D_j = D(l, j)$. In addition, we define the sets of links

$$\begin{aligned} A &= \{l| \text{ a message group arrives at link } l\} \\ D &= \{l| \text{ a message group departs from link } l\} \end{aligned}$$

The probability of a new message group arriving at a link depends only on the set of external message arrivals along with the message group departures from each link. Given that a message group departs from a link, it must be the group at the front of that queue. Consequently, the initial state u, the set of external message arrivals $\{A_j | j = 1 \ldots J\}$ and the set of links with departing message groups D is sufficient to specify the new state v as well as the sets A, $\{D_j | j = 1 \ldots J\}$, and $\{A(l, j) | l = 1 \ldots L, j = 1 \ldots J\}$. Suppose a transition occurred from u to a new state $v = v(1), \ldots, v(L)$ such that A_j, D_j, $A(l, j)$ and $D(l, j)$ took on the values a_j, d_j, $a(l, j)$ and $d(l, j)$ and the sets A and D were specified. Furthermore, for links $l \in A$, let $\hat{c}(l)$ equal the class of the arriving message group. Given this information, we can write down $v(l)$ for each l. In fact,

$$v(l) = \begin{cases} n(l), (0, \hat{c}(l)), y(l,2), \ldots, y(l, n(l)) & l \in A \cap D \\ n(l) - 1, y(l,2), \ldots, y(l, n(l)) & l \in A^c \cap D \\ n(l) + 1, (0, \hat{c}(l)), y(l,2), \ldots, y(l, n(l)), (w(l,1) + 1, c(l,1)) & l \in A \cap D^c \\ n(l), y(l,2), \ldots, y(l, n(l)), (w(l,1) + 1, c(l,1)) & l \in A^c \cap D^c \end{cases}$$

In this case,

$$\beta_{v(l)} = \begin{cases} \left(\prod_{j \in S_l} \hat{\theta}_j(a(l,j))\right) \prod_{i=2}^{n(l)} \prod_{j \in S_l} \hat{\theta}_j(s(l,i,j)) & l \in A \cap D \\ \prod_{i=2}^{n(l)} \prod_{j \in S_l} \hat{\theta}_j(s(l,i,j)) & l \in A^c \cap D \\ \left(\prod_{j \in S_l} \hat{\theta}_j(a(l,j))\right) \prod_{i=1}^{n(l)} \prod_{j \in S_l} \hat{\theta}_j(s(l,i,j)) & l \in A \cap D^c \\ \prod_{i=1}^{n(l)} \prod_{j \in S_l} \hat{\theta}_j(s(l,i,j)) & l \in A^c \cap D^c \end{cases}$$

Suppose $l \in A^c$, that is no new message group arrives at link l. In that instance, $a(l,j) = 0$ for all j. Consequently, $\hat{\theta}_j(a(l,j)) = 1$ for all j. This implies that for all $l \in A^c$,

$$\prod_{j \in S_l} \hat{\theta}_j(a(l,j)) = 1$$

This result allows us to write the following

$$\beta_{v(l)} = \begin{cases} \beta_{u(l)} \left(\dfrac{\prod_{j \in S_l} \hat{\theta}_j(a(l,j))}{\prod_{j \in S_l} \hat{\theta}_j(s(l,1,j))} \right) & l \in D \\ \beta_{u(l)} \left(\prod_{j \in S_l} \hat{\theta}_j(a(l,j)) \right) & l \in D^c \end{cases}$$

We must also identify $\gamma_{v(l)}$. Given the state v, we find that

$$\gamma_{v(l)} = \begin{cases} \overline{G}_{\hat{c}(l)}(0)\left(\prod_{i=2}^{n(l)} \overline{G}_{c(l,i)}(w(l,i))\right) & l \in A \cap D \\ \prod_{i=2}^{n(l)} \overline{G}_{c(l,i)}(w(l,i)) & l \in A^c \cap D \\ \overline{G}_{\hat{c}(l)}(0)\left(\prod_{i=2}^{n(l)} \overline{G}_{c(l,i)}(w(l,i))\right) \overline{G}_{c(l,1)}(w(l,1)+1) & l \in A \cap D^c \\ \left(\prod_{i=2}^{n(l)} \overline{G}_{c(l,i)}(w(l,i))\right) \overline{G}_{c(l,1)}(w(l,1)+1) & l \in A^c \cap D^c \end{cases}$$

For each class c, $\overline{G}_c(0) = 1$, so we can write

$$\gamma_{v(l)} = \begin{cases} \gamma_{u(l)}\left(\dfrac{1}{\overline{G}_{c(l,1)}(w(l,1))}\right) & l \in D \\ \gamma_{u(l)}\left(\dfrac{\overline{G}_{c(l,1)}(w(l,1)+1)}{\overline{G}_{c(l,1)}(w(l,1))}\right) & l \in D^c \end{cases}$$

As a result,

$$\begin{aligned} \pi_v &= \left(\prod_{l=1}^{L} \beta_{v(l)}\right)\left(\prod_{l=1}^{L} \gamma_{v(l)}\right) \\ &= \pi_u \left(\frac{\prod_{l=1}^{L}\prod_{j \in S_l} \hat{\theta}_j(a(l,j))}{\prod_{l \in D}\prod_{j \in S_l} \hat{\theta}_j(s(l,1,j)}\right)\left(\frac{\prod_{l \in D^c} \overline{G}_{c(l,1)}(w(l,1)+1)}{\prod_{l=1}^{L} \overline{G}_{c(l,1)}(w(l,1))}\right) \end{aligned}$$

Suppose that a session j message departs from link l and enters service at link l'. In this case, $l \in D$ and $s(l,1,j) = a(l',j)$. This implies that for every message that departs from a link but does not exit the network, the above expression for π_v contains cancelling factors $\hat{\theta}_j(a(l',j))$ and $\hat{\theta}_j(s(l,1,j))$. The only noncancelling factors are contributed by messages that enter or exit the network. As a consequence,

$$\pi_v = \pi_u \left(\frac{\prod_{j=1}^{J} \hat{\theta}_j(a_j)}{\prod_{j=1}^{J} \hat{\theta}_j(d_j)}\right)\left(\frac{\prod_{l \in D^c} \overline{G}_{c(l,1)}(w(l,1)+1)}{\prod_{l=1}^{L} \overline{G}_{c(l,1)}(w(l,1))}\right)$$

Recalling that $\hat{\theta}_j(k) = \theta_j(k)/\theta_j(0)$, we observe that

$$\pi_v = \pi_u \left(\frac{\prod_{j=1}^{J} \theta_j(a_j)}{\prod_{j=1}^{J} \theta_j(d_j)} \right) \left(\frac{\prod_{l \in D^c} \overline{G}_{c(l,1)}(w(l,1)+1)}{\prod_{l=1}^{L} \overline{G}_{c(l,1)}(w(l,1))} \right) \tag{8.2}$$

Now we will describe the transition probability P_{uv}. Note that a session j message exits the system iff a message group containing a session j message completes service at the last link of route j. Also, an arrival of class $\hat{c}(l)$ occurs at link l iff each session j message in the arrival either just entered the network or was contained in a message group that just completed service at the previous link of route j. So a transition is completely specified by the set of message groups that complete service at each link along with the external message arrivals. As a result,

$$P_{uv} = \left(\prod_{j=1}^{J} P\{A_j = a_j\} \right) P\{D\}$$

where $P\{D\}$ is the probability that the set of links at which departures take place is precisely the set D. We can write

$$P\{A_j = a_j\} = \theta_j(a_j)$$

To find $P\{D\}$, we recall that the probability that a departure occurs at link l is the conditional probability that the front message group of class $c(l,1)$ will depart after its next unit of service, given that it has already received $w(l,1)$ units of service. As always, we denote this probability by

$$r_{c(l,1)}(w(l,1)) = \frac{g_{c(l,1)}(w(l,1)+1)}{\overline{G}_{c(l,1)}(w(l,1))}$$

As a result,

$$P\{D\} \;=\; \left(\prod_{l \in D} r_{c(l,1)}(w(l,1)) \right) \left(\prod_{l \in D^c} (1 - r_{c(l,1)}(w(l,1))) \right)$$

$$= \frac{\left(\prod_{l\in D} g_{c(l,1)}(w(l,1)+1)\right)\left(\prod_{l\in D^c} \overline{G}_{c(l,1)}(w(l,1)+1)\right)}{\prod_{l=1}^{L} \overline{G}_{c(l,1)}(w(l,1))}$$

This implies

$$P_{uv} = \frac{\left(\prod_{j=1}^{J} \theta_j(a_j)\right)\left(\prod_{l\in D} g_{c(l,1)}(w(l,1)+1)\right)\left(\prod_{l\in D^c} \overline{G}_{c(l,1)}(w(l,1)+1)\right)}{\prod_{l=1}^{L} \overline{G}_{c(l,1)}(w(l,1))} \tag{8.3}$$

In the state v at each link $l \in A$, $w(l,1) = 0$. In the forward time system, this corresponds to a new message group that has received no service at the front of the queue for each link $l \in A$. However, consider the reverse time system. At the front of the queue for each link $l \in A$, we have a message group that has no residual service requirement. Also, departures occur at the beginning of a slot. Consequently, in the reverse time system, the front message group at each link $l \in A$ immediately departs. Each message of such a message group immediately returns to its previous link. We return to the state u if

- For each session j, a reverse time session j message of type d_j enters the network.

  This ensures that for each $l \in D$, a message group of class $c(l,1)$ arrives at l in the reverse time system. Recall that by convention, if $d_j = 0$, then no session j message enters the network.

- For each $l \in D$, the reverse time *message group* arrival requires service $w(l,1)+1$.

  Note that this new arrival immediately receives a unit of service so that at the end of the slot, the remaining work is $w(l,1)$.

These two requirements imply that the reverse time transition probability is

$$P^*_{vu} = \left(\prod_{j=1}^{J} \theta_j(d_j)\right) \prod_{l\in D} g_{c(l,1)}(w(l,1)+1) \tag{8.4}$$

From Equations 8.4 and 8.3, we see that

$$\frac{P_{uv}}{P^*_{vu}} = \left(\frac{\prod_{j=1}^{J} \theta_j(a_j)}{\prod_{j=1}^{J} \theta_j(d_j)} \right) \left(\frac{\prod_{l \in D^c} \overline{G}_{c(l,1)}(w(l,1)+1)}{\prod_{l=1}^{L} \overline{G}_{c(l,1)}(w(l,1))} \right)$$

From (8.2), we find that

$$\frac{P_{uv}}{P^*_{vu}} = \frac{\pi_v}{\pi_u}$$

This is the requirement of Theorem 1. This verifies that π_u is the correct limiting state distribution. That is, we have verified that each queue acts as though it were in isolation with independent Bernoulli message arrivals from each session using the link.

8.3 Remarks

We have yet to verify that our framework includes both the fixed and random message length cases. For fixed message lengths, let

$$G_k(x) = \begin{cases} 1 & x \leq k \\ 0 & x > k \end{cases}$$

This choice implies that the length of a type k message is always equal to k.

Alternatively, for random message lengths, let every session j message be of type j. That is,

$$\theta_j(k) = \begin{cases} 1 & k = j \\ 0 & \text{otherwise} \end{cases}$$

In this case, whenever a session j message arrives at a link, its length is chosen from G_j.

In addition, Conjecture 7 can be proven for a network of permutation queues.

Chapter 9

Routing and Flow Control

In this chapter, we will examine some of the other issues involving the operation of a network of round robin queues. First, we will describe how the ordinary round robin queue can be modified to prevent a bursty session from cheating and receiving better service than that to which it is entitled. Second, we will consider the typical behavior of this modified round robin queue to show that it essentially operates like a session round robin queue. Third, we will explain how round robin service justifies simple routing and less restrictive flow control. Lastly, we describe how round robin service simplifies the issue of network costs.

9.1 Typical Operation of the Round Robin Queue

Although we have identified a tractable model of a network of round robin queues with certain desirable properties, it should be apparent that one would not build the exact model system. We now consider some of these alterations.

One can see that forming message groups is not a very good idea. Clearly, whenever multiple messages arrive simultaneously, it would be much more reasonable to queue the messages so that they enter service in consecutive slots. We are not able to analyze such a system precisely. However, for our model, we have been able to say that a link typically has no multiple message groups in service. Furthermore, the occurrence of multiple simultaneous message arrivals is a rare event. Consequently, neither network operation nor performance should be appreciably different with this modification.

As we have said, the purpose of round robin service is to provide better service to short messages. Clearly, it would be possible for a bursty session to thwart this intention by breaking up long messages into short messages that are submitted in consecutive slots. However, this cheating session could be easily identified. We have verified that an ordinary session very rarely has more than one message waiting at a link, whereas the cheating session would regularly have several messages in service at the same time. We can discourage this behavior by adoption of the following rule:

- Whenever a session submits a message while it has another message in service, the new message is treated as a continuation of the previous message.

This rule would make little difference to honest sessions, however it would prevent bursty sessions from cheating. By making this modification, we have changed the message round robin system into a session round robin. [See page 11 for a description of session round robin.] We believe that the message round robin provides an analytically tractable close approximation to session round robin, but if one were to implement a system with round robin service, one would use session round robin.

In our network model, a message arrives and enters service at a link precisely after it has completed service at its previous link. That is, a message does not begin service at a link until all of its packets have been queued at that link. This creates additional delay, especially for long messages. To avoid this circumstance, it would be logical to allow a message to enter service as soon as its first packet arrives. If the next packet arrives before the next round robin turn for that message, that packet would be transmitted in the appropriate slot. Otherwise, the message would be considered to have completed service and the late arriving packet would be treated as the start of a new message.

If this change is done for a single message, it can be seen that the message never gets a round robin turn any earlier than it would have had none of its packets arrived late. However, allowing this message to enter service earlier would create additional delay for other messages that were already in service. If this change were made for all sessions, the resulting queue would be very difficult to analyze. The message lengths and interarrival times of a session at a link would have a complicated dependence on the state of the queues of the previous links of that session. Therefore, it is difficult to argue that this change always improves delay performance for all sessions.

9.2 Routing under Round Robin

The functions of a network routing algorithm include

- Redirection of traffic when links or nodes fail.
- Adjustment of routes to avoid congestion and reduce delay.

We will not address the problem of node and link failures because this problem is not significantly altered under round robin service. However, we will consider briefly the effect of round robin on the performance aspects of routing algorithms.

We will focus our attention on the static routing problem. That is, given a network and a set of sessions, how do we assign routes to these sessions. Most approaches to optimal routing are based on the average packet rates of the sessions. The delay on a link is modeled as a convex function of the overall load on the link. Typically, the link delay is approximated by the delay on an FCFS M/M/1 queue with the same overall load. Routes are assigned to minimize a global delay measure such as the average waiting time on the most heavily loaded link or the sum of the average session delays.

This approach has some difficulties. In particular, the link delay approximation can be poor if some of the sessions are bursty. In particular, for the FCFS M/G/1 discrete time queue, the appropriate form of the Pollaczek-Khinchin formula for the average waiting time would be

$$\overline{W}_G = \frac{\frac{1}{2}\lambda(\overline{X^2} - \overline{X})}{1 - \lambda\overline{X}}$$

When the service requirement is geometric, $\overline{X^2} = 2\overline{X}^2 - \overline{X}$ so that the waiting time for the M/M/1 discrete time system is

$$\overline{W}_M = \frac{\lambda(\overline{X}^2 - \overline{X})}{1 - \lambda\overline{X}}$$

For a sufficiently bursty arrival process, the M/M/1 approximation will underestimate the link delay. Consider the following example. Let the probability of an arrival, λ equal 1/40. With probability 10^{-5}, a message will have 10^6 packets, otherwise a message will have 10 packets. The M/M/1 delay approximation has $\overline{W}_M$ equal to 19 slots. The actual mean waiting time, $\overline{W}_G$, equals 250,000 slots. For long messages, the additional delay resulting from burstiness will make relatively little difference. For short messages however, this unexpected delay may be critical. Consequently, on a network with very different types of traffic and FCFS service, rate based routing algorithms can be inadequate.

The choice of service discipline has a substantial effect on the set of feasible routing assignments. A simple example would be a network that provides a data service and a voice service. Voice sessions may have constraints on the average and maximum delay but would submit relatively short messages at a low packet rate. Data sessions may be of much higher rate but may not have any delay requirements. If first come first served service were used and data messages were sufficiently long, it would be impossible to meet the delay requirements for voice traffic. In this case, it would be desirable to use a service discipline that provides smaller delay to the voice traffic. If FCFS service were unavoidable however, it may be possible to assign routes in such a way as to use one set of links for voice traffic and a second set of links for data traffic.

We can conclude that routing is closely tied to service discipline and that for FCFS service, routing based on average packet rates may be not be sufficient. However, under round robin service, these routing decisions become much simpler. We have demonstrated that the distribution for the number of messages of a session queued at a link depends only on

- The average message arrival rate of the session.
- The average message length of the session.
- The overall load on the link.

A session is affected by the other sessions with which it shares a link only through the overall load. The service received by a session is independent of the burstiness of all other sessions. The best route for a session can be easily determined by looking at the loads on the various links. Consequently, the overall load on a link is an appropriate measure of congestion.

This fact suggests several reasonable choices for routing algorithms including

- Minimization of the load on the network's most heavily loaded link.
- Minimization of the average number of messages in the network.

Either of these problems is a reasonably simple optimization problem whose solution depends only on the respective session loads. More importantly, both of these approaches are based on session packet rates but do not require a possibly poor approximation of the link delay. The link load provides an accurate description of the link delay. The use of round robin service permits the use of simple rate based routing even when different types of traffic share a link. In short, under round robin service, we can use the same routing as before but with greater justification.

9.3 Flow Control

Traditionally, the effects of burstiness have been mitigated by the use of flow control and priorities. Flow control restricts the ability of the bursty session to dump packets into the network, allowing low rate sessions, which would not be restricted by flow control, to receive better service. However, assuming that a network has sufficient buffering capacity, it is not desirable to control burstiness. When a bursty session. submits a long message to an empty link, it would be beneficial to send that message as quickly as possible. Consequently, we do not want to enforce maximum packet rates on sessions. In short, sessions should be able to offer any instantaneous load to the network with minimal restrictions.

Of course, minimal restriction does not mean without restriction. In real networks, the buffer space is always finite and the buffer overflow problem cannot be avoided. For any queue discipline, an appropriate flow control strategy must do the following:

- Provide effective recovery from buffer overflow.
- Assign and enforce session packet rates to provide acceptable service and avoid buffer overflow.

The queueing analysis in this thesis has assumed that the buffer space is large enough to ensure that buffer overflow is a rare event. In this case, we do not need to consider the operation of the queue when buffers are nearly full. However, if the buffer space were relatively small, it would be important to characterize the operation of the congested round robin queue. This would require that a particular flow control method be specified. The analysis of a flow control strategy tends to be difficult. For an example, see [Hah86], in which session round robin with link by link window flow control is examined.

The issue of assignment of average packet rates can be considered within our analytical framework. Assuming that session packet rates have been chosen to ensure that buffer overflow is a rare event, we have shown that the quality of service received by a session is a function of the arrival process of that session as well as the overall packet arrival rate. For a session to receive acceptable service, the network must regulate the average packet rates of all sessions. The advantage of round robin over other service strategies such as FCFS, is that it is not necessary to regulate the burstiness of the sessions.

9.4 Priorities and Network Costs

Priority queues provide a general and powerful method for meeting delay constraints for networks with varying kinds of traffic. The basic approach of all priority queues is to assign high priority to messages that require small delay. When a message enters the queue, it is guaranteed to enter service ahead of any message of lower priority.

The difficulty with this approach is that the level of service received by sessions of a certain priority depends strongly on the offered load of sessions of higher priority. Consequently, if the network is to guarantee a certain quality of service to a session of some priority, it must ensure that all sessions of higher priority do not have excessive loads. One would do this by requiring a session to request a packet rate and priority level before commencing operation. The network would then enforce that rate. This means that that the network is required to enforce rates on all sessions, excepting perhaps sessions of the lowest priority. Furthermore, it should be apparent that the network must also control the burstiness of all sessions.

How rates and priorities should be assigned is a difficult problem. First, the question of whether a set of rates and priority assignments is feasible with respect to a set of delay constraints is not simple. Second, how the network should charge users for higher priority service is not obvious. Consider a simple example of a network that offers two grades of service. If the additional charge for high priority service is too low, all users will request high priority service, spoiling the intent of priorities. The issue of appropriate charges for priority service is a difficult one involving, first, the user's willingness to pay for improved service, second, the effect of improved service for one session on the service of other sessions, and third, the cost of providing better service.

The appeal of round robin is its relative simplicity. A session submits its packets at essentially any instantaneous rate, subject to a constraint on the average rate. The service received by the session simply depends on its own burstiness as well as the resulting loads on the links used by that session. Since a session affects the network performance only through its average packet rate, each session can be charged by its packet rate. Essentially, sessions that send the same number of bits are charged the same amount, but non-bursty sessions get better service at the same price.

Of course, simplicity has its disadvantages. In particular, a session may desire a level of service that is unattainable because of the loads of the other sessions. If a session *requires* a certain level of service, then the network must assign and enforce session packet rates to meet the requirement. Alternatively, a session may be willing to endure additional delay for a reduced price. This lower cost, higher delay service would also be unavailable.

Chapter 10

Conclusion

This research can be divided into an examination of the problem of burstiness and an analysis of the round robin queue. It appears that it is difficult to characterize the problems associated with networks of bursty sources. As a result, we simply have argued that an appropriate network service strategy should have certain basic properties. Subsequently, we verified that a round robin service discipline has these properties. We consider these two subjects separately.

10.1 The Burstiness Problem

At the start of this thesis, we described how burstiness can become a significant problem as network speeds increase. To do this, it was necessary to choose an appropriate network model. We chose to model a data network with bursty sources as a discrete time queueing system in which a burst of data is represented by a message whose length corresponds to that of the burst. We believe that this model captures the essential features of a network with sources of differing degrees of burstiness.

The basic question we then address is how should service be provided to users whose requirements are both diverse and ill-defined. Our contention is simply that for any traffic mix, short messages should wait less. This approach provides fast service to those for whom fast service is a possibility. That is, the system time of a long message must be large simply because that message is long. However, it is often possible for long messages to endure a modest additional delay in order to provide significantly better service to short messages. Furthermore, we argue that the service received by a user should be no better than that it would receive if every other user were just as bursty.

One can argue that we have sidestepped the real issue of how networks will be used in the future. To answer this question, one must examine both the uses for which future networks will be suited as well as the ways in which people would like to use networks. Whether networks will be appropriate for a certain use will depend on the costs and capabilities of future networks. In addition, as new networks come into being, users will find new ways to use those networks. We have taken the point of view that, rather than trying to second guess the detailed usage patterns of new networks, one should attempt to make the performance seen by a user relatively independent of the behavior of other users.

10.2 The Round Robin Queue

We have found that a round robin service discipline provides those properties that we deemed to be desirable for networks with bursty sources. To be somewhat more precise, we have verified that under round robin service, the mean system time of a message is a linear function of the message length. Moreover, the mean system time is solely a function of the message length and the overall load and does not depend on the burstiness of the arrival process. Similarly, we found that the system time variance of a message is upper bounded by a linear function of the message length. In addition, for short messages, the system time variance was less sensitive to the arrival process burstiness. These facts imply that short messages receive faster and more predictable service.

We believe that it is somewhat surprising that a discrete time round robin queue can be analyzed directly without appeal to a processor sharing approximation. Moreover, the examination of the discrete time round robin increases our understanding of processor sharing and of the robustness of the processor sharing model. In addition, we feel that there are other queueing problems for which the examination of the reverse time queue in discrete time may provide some additional insight.

With respect to networks of round robin queues, there are several unresolved issues. First, it would be most desirable to have a more realistic network model that remains analytically tractable. In particular, it would be most desirable to find a model in which a message is not queued in its entirety before beginning transmission on an outgoing link. However, we believe it is unlikely that such a model exists.

Typically, network models of this sort lose most of the nice properties that permit analysis by reversibility.

Second, round robin treats all users the same. For the round robin queues we have described, users can circumvent the intention of round robin service to provide better service to non-bursty sessions. Under message round robin, the circumvention is accomplished by allowing a session to break up long messages into multiple short messages. Similarly, under session round robin, a customer could inititiate more than one session. Either one of these disciplines would be difficult to analyze because message arrivals in nearby slots would no longer be independent. Consider a bursty session that is permitted to break a long message into many short messages. Suppose this session submits a short message in slot s. With high probability, this short message is simply the first part of a long message. Consequently, the arrival of the short message increases the probability that this session will submit additional short messages in subsequent slots. This implies that arrival processes in disjoint slots would no longer be independent. Once again, such a model does not appear to be amenable to analysis by reversibility.

Bibliography

[BeG87] Bertsekas, D. P., and Gallager, R. G. 1987. *Data Networks.* Englewood Cliffs, NJ: Prentice-Hall.

[Dai87] Daigle, J. N. 1987. Message Delays with Prioritized HOLP and Round-Robin Packet Servicing. *IEEE Trans. Commun. COM-35.* No. 6, pp. 609–619.

[DKS89] Demers, A., Keshav, S., and Shenker, S. Analysis and Simulation of a Fair Queueing Algorithm. To appear in *SIGCOMM '89 Proceedings.*

[DoR85] Doshi, B. T., and Rege, K. M. 1985. Analysis of a Multistage Queue. *AT&T Tech. J.* **64**, No. 2, Part 1, pp. 369–390.

[Mor89] Morgan, S. P. 1989. Queueing Disciplines and Passive Congestion Control in Byte-Stream Networks. In *Proceedings of IEEE INFOCOM '89.* IEEE, New York, pp. 711-720.

[FrM84] Fraser, A. G., and Morgan, S. P. 1984. Queueing and Framing Disciplines for a Mixture of Data Traffic Types. *AT&T Tech. J.* **63**, No. 6, pp. 1061–1087.

[Hah86] Hahne, E. L. 1986. *Round Robin Scheduling for Fair Flow Control in Data Communication Networks.* PhD thesis. MIT, Dept. of Electrical Engineering and Computer Science.

[Kel79] Kelly, F. P. 1979. *Reversibility and Stochastic Networks.* New York: John Wiley & Sons.

[Kle76] Kleinrock, L. 1976. *Queueing Systems, Vol. 2.* New York: John Wiley & Sons.

[Ott84] Ott, T. J. 1984. The sojourn-time distribution in the M/G/1 queue with processor sharing. *J. Appl. Prob.* **21**, No. 2, pp. 360–378.

[Reg86] Régnier, J. M. 1986. *Priority Assignment in Integrated Services Networks.* PhD thesis. MIT, Dept. of Electrical Engineering and Computer Science.

[Ros83] Ross, S. M. 1983. *Stochastic Processes.* New York: John Wiley & Sons.

[Tak87] Takagi, H. 1987. Exact Analysis of round-robin scheduling of services. *IBM J. Res. Dev.* **31**, No. 4, (July), pp. 484–488.

[Tak88] Takagi, H. 1988. Queuing Analysis of Polling Models. *ACM Computing Surveys.* **20**, No. 1, pp. 5–28.

[Yas83] Yashkov, S. F. 1983. A derivation of the response time distribution for a M/G/1 processor-sharing queue. *Problems of Control and Information Theory.* **12**, No. 2, pp. 133–148.

[Zha89] Zhang, L. *A New Architecture for Packet Switching Network Protocols.* PhD thesis. MIT. Dept. of Electrical Engineering and Computer Science.