

ESC-TR-2004-079

**Technical Report  
1098**

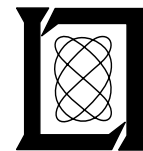
# **EHF Satellite Communications-on-the-Move Blockage Channel Modeling**

H. Yao

2 November 2004

---

**Lincoln Laboratory**  
MASSACHUSETTS INSTITUTE OF TECHNOLOGY  
*LEXINGTON, MASSACHUSETTS*



---

Prepared for the MILSATCOM Joint Program  
Office SMC/MCX under Air Force Contract  
F19628-00-C-0002.

Approved for public release; distribution is unlimited.

**This report is based on studies performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology. This work was sponsored by the MILSATCOM Joint Program Office SMC/MCX under Air Force Contract F19628-00-C-0002.**

**This report may be reproduced to satisfy needs of U.S. Government agencies.**

**The ESC Public Affairs Office has reviewed this report, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.**

**This technical report has been reviewed and is approved for publication.**

**FOR THE COMMANDER**

  
**Gary Tunungian**  
**Administrative Contracting Officer**  
**Plans and Programs Directorate**  
**Contracted Support Management**

**Non-Lincoln Recipients**  
**PLEASE DO NOT RETURN**  
**Permission has been granted to destroy this**  
**document when it is no needed.**

**Massachusetts Institute of Technology  
Lincoln Laboratory**

**EHF Satellite Communications-on-the-Move  
Blockage Channel Modeling**

*Huan Yao  
Group 64*

**Technical Report TR-1098**

2 November 2004

Approved for public release; distribution is unlimited.

**Lexington**

**Massachusetts**



## **ABSTRACT**

In the next-generation Transformational Satellite Communications System, one essential element is to provide satellite communication to on-the-move terminals, which suffer from intermittent channels due to blockages by objects in their environments. Presented is our approach for modeling these blockage channels for different environments, in order to allow accurate prediction of system behavior via numerical simulation or symbolic analysis. In particular, we focus on the scenario of link-layer ARQ (automatic repeat request) and the prediction of the averages and distributions of the queue length and delay experienced by the data packets. We investigate two models. The simple two-state Markov model is amenable to symbolic analysis, but is less accurate, while a more complex mixture model leads to better predictions, but can be studied only via numerical simulation. We also introduce a new memory-decay curve concept for the characterization of channel memory and the estimation of model parameters.



## **ACKNOWLEDGMENTS**

This work was supported by the MILSATCOM Joint Program Office (MJPO). Their support is greatly appreciated.

Throughout this study, I received help from many people around me, in particular, Ken Hetling, Tom Macdonald, Wayne Phoel, and Tom Seay. This project started as a simple study and took on a life of its own under their encouragement and guidance.

I would like to thank Jeff Schodorf for providing me with the experimental data used in this study. I also thank Andrew Worthen and Ron Bauer, who were involved in the original project, for their insightful discussions.

Finally, I thank Dean Kolba, Wayne Phoel, John Choi, Tom Macdonald, and Ken Hetling for reviewing this manuscript and providing valuable comments.



## TABLE OF CONTENTS

ABSTRACT	iii
ACKNOWLEDGMENTS	v
LIST OF ILLUSTRATIONS	ix
LIST OF TABLES	xi
1. INTRODUCTION	1
1.1 PROBLEM DESCRIPTION	1
1.2 THE APPROACH	2
1.3 OUTLINE	3
2. BACKGROUND	5
2.1 EXPERIMENTAL DATA	5
2.2 LINK-LAYER ARQ	7
3. STATISTICAL ANALYSIS METHODS	9
3.1 OPENING AND BLOCKAGE DURATIONS	9
3.2 NEW IDEA: MEMORY-DECAY CURVE	11
4. TWO-STATE MARKOV MODEL	13
4.1 MODEL DESCRIPTION	13
4.2 PARAMETER ESTIMATION USING MEMORY-DECAY CURVE	14
4.3 ARQ BEHAVIOR ANALYSIS	17
4.4 ARQ BEHAVIOR COMPARISON	19
5. HIGHER-ORDER MODELS	21
5.1 CHOOSING A HIGHER-ORDER MODEL	21
5.2 PARAMETER ESTIMATION USING MEMORY-DECAY CURVE	25
5.3 ARQ BEHAVIOR COMPARISON	27
6. SUMMARY AND FUTURE WORK	29
APPENDIX A – PROOF OF MEMORY-DECAY CURVE ON-OFF SYMMETRY	31
APPENDIX B – DERIVATION OF ARQ BEHAVIOR FOR THE TWO-STATE MARKOV MODEL	33
B.1 QUEUE-LENGTH RESULTS	33
B.2 DELAY RESULTS	37
REFERENCES	41



## LIST OF ILLUSTRATIONS

Figure No.		Page
1	A generic block diagram of a portion of a system containing a COTM channel.	1
2	Our modeling approach which involves not only building a channel model based on experimental data, but also validating the model using a specific application, in particular, ARQ.	2
3	Example of experimental data and initial processing: A 30-second segment of experimental data from the urban environment (left) is converted to a binary sequence (right) using a fade threshold of 10 dB. The nature of the COTM channel is blockage with memory.	5
4	All available channel connectivity data (excluding repetition) from open, rural, and urban environments. Because the rural data are highly atypical, they are not used.	6
5	ARQ model at the transmitter in which packet arrivals are modeled by a stationary Bernoulli process and packet departures are governed by the blockage channel.	7
6	Demonstration of measurement of opening and blockage durations from channel connectivity sequence.	9
7	Opening duration and blockage duration histograms for the open environment. The log scale emphasizes the exponential distributions and the long openings.	10
8	Example of shuffling opening and blockage durations to demonstrate that changing the order of the segments may affect ARQ queue behavior.	10
9	Experimentally measured memory-decay curves for the open and urban environments. They both decay monotonically, but have different time scales.	12
10	The simple two-state Markov model. The two states are labeled $U$ for unblocked and $B$ for blocked.	13
11	Physical world scenario associated with the two-state Markov model. The model is exact if both the widths of the buildings and the gaps between them are exponentially distributed and the vehicle velocity is constant.	14
12	Estimation of two-state Markov model parameters by least-squares fit of the memory-decay curve for both the open and urban environments.	16
13	A two-layer semi-infinite Markov chain for analyzing the behavior of ARQ over a blockage channel modeled by the two-state Markov model, assuming perfect and immediate acknowledgments and Bernoulli arrivals.	17

**LIST OF ILLUSTRATIONS**  
**(Continued)**

<b>Figure No.</b>		<b>Page</b>
14	Comparison of the queue-length distributions and averages obtained from the experimental data and the two-state Markov model. There are large gaps for the open environment.	20
15	Demonstration of a deficiency of the two-state Markov model. In the open environment, there are particularly long openings that are more frequent than what the model predicts.	22
16	The multi-open-state Markov model and its corresponding physical world scenario. The model is obtained by adding a third long-open state to the original two-state Markov model.	23
17	The mixture model and its corresponding physical world scenario. The model is obtained by treating the original two-state Markov model as one joint state and adding a long-open state.	23
18	An equivalent representation of the mixture model as two parallel two-state Markov chains.	24
19	Estimation of the mixture model parameters by least-squares fits of memory-decay curves for both open and urban environments. Excellent fits are achieved.	26
20	Comparison of the queue-length distributions and averages obtained from the experimental data and the mixture model. The gaps are smaller than for the two-state Markov model case.	28
B-1	Repeat of Figure 13: A two-layer semi-infinite Markov chain for analyzing the behavior of ARQ over a blockage channel modeled by the two-state Markov model, assuming perfect and immediate acknowledgments and Bernoulli arrival.	33

## LIST OF TABLES

<b>Table No.</b>		<b>Page</b>
1	The Effect of Shuffling Opening and Blockage Segments	11
2	Estimated Two-State Markov Model Parameters	16
3	Summary of Two-State Markov Model Performance	20
4	Comparison of the Ease of Analyzing Model Statistics	24
5	Estimated Mixture Model Parameters	27
6	Estimated Mixture Model Parameters for Implementation	27
7	Summary of Mixture Model Performance	28



# 1. INTRODUCTION

The Transformational Satellite Communications System (TSAT) is a major ongoing effort for improving the communication capabilities of our military. The overall system includes a terrestrial network, an optically connected network in space, and multiple links across various platforms. The goal is to enable communications from anywhere, at anytime, and under hostile conditions. In this work, we focus on the radio frequency (RF) link between a geostationary satellite in space and a Communications-on-the-Move (COTM) terminal on the ground, and the task is to model the statistics of the connectivity of this RF link. This study is an extension of the channel-modeling work presented by J. B. Schodorf of MIT Lincoln Laboratory in [1]. While this study focuses primarily on a scenario particular to the Army, lessons learned from this modeling problem can be extended to channel modeling for the Air Force or the Navy in the future.

In the rest of this section, we first describe the problem of interest and then explain our approach. At the end, we provide an outline for the report.

## 1.1 PROBLEM DESCRIPTION

The main task of this study is to provide models for the connectivity of the link between a geostationary satellite and a mobile COTM terminal. As the COTM terminal moves, objects in its environment, such as buildings and foliage, intermittently block its view to the satellite, and communications are interrupted. It is our goal to study and model the statistics of these blockages.

One of the main challenges is to model the blockages in different environments. Blockages caused by buildings in an urban environment are statistically different from blockages caused by foliage in a more open environment. Even if their models share the same structure, the model parameters are certainly going to be different.

The purpose of modeling the COTM channel is to allow accurate prediction of system behavior via either symbolic analysis or numerical simulation. Examples of systems of interest include a network with COTM links in it and a single-link system running link-layer ARQ<sup>1</sup> for blockage mitigation. A generic block diagram showing part of a system containing a COTM link is shown in Figure 1. Our goal is to provide an appropriate model for the COTM channel element so that system behavior can be analyzed in closed form and numerical simulations can be performed without having to collect extensive amounts of experimental data.

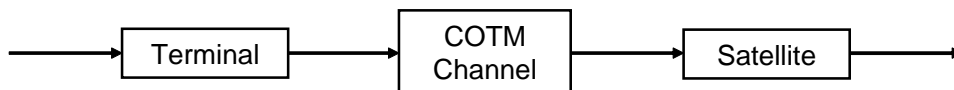


Figure 1. A generic block diagram of a portion of a system containing a COTM channel.

---

<sup>1</sup>With ARQ (automatic repeat request), packets that are dropped due to blockage are repeated.

## 1.2 THE APPROACH

We now describe our modeling approach, which is illustrated in Figure 2. The main idea is that models are not only built to capture the statistics of the experimental data, they are also validated by their ability to predict system behavior in particular scenarios.

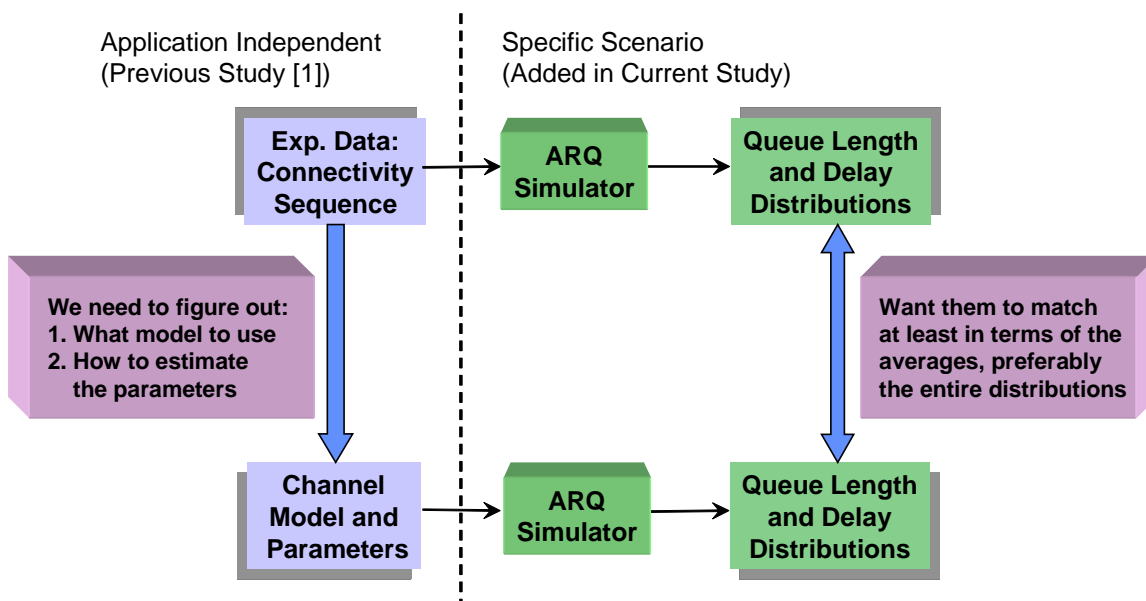


Figure 2. Our modeling approach which involves not only building a channel model based on experimental data, but also validating the model using a specific application, in particular, ARQ.

In the previous study by J. B. Schodorf [1], starting from the experimental data, the researchers built channel models and estimated parameters to match certain statistics of the data. However, when we first began extending their study, we found that since the models do not exactly reproduce the various statistics of the data, different ways of estimating the parameters led to different results. It was hard to decide which sets of parameters are really better.

To resolve this problem, in this study, we propose to validate the models built using specific scenarios from specific applications, since the purpose of building a model is to use it in a system to predict system behavior. We note that the model building itself is still application-independent. It is only the model validation that depends on the application. It is not our intention to build different models for different applications.

Throughout this study, we use the scenario of a single-link system between a satellite and a COTM terminal running link-layer ARQ, where the system behavior is measured by the queue length and the delay experienced by the data. These quantities are important because queue length reflects the amount of memory

needed, and delay affects the experience of the end users. More specifically, we take the experimental data and a channel sequence randomly generated using the model, feed them through an ARQ simulator, let the system run, and then compare the resulting queue length and delay as shown in Figure 2. The desired outcome is to have the experimental-data results and the model-predicted results match at least in terms of the averages, preferably in terms of the distributions. The expectation is that models validated using one scenario should be applicable to other scenarios as well, such as networks with COTM links.

In comparison, the previous study in [1] was application-independent, i.e., it only had the left side of Figure 2. The new feature in the current study is the validation of the models using the ARQ scenario, i.e., the addition of the right side.

This channel modeling problem is a particular case of a stochastic modeling problem. One important rule in stochastic modeling is that the models should always have some connections to the real world. The modeling process should not be a purely mathematical exercise.

The models should also be built in stages. The initial model should be simple and well known, such as those found in text books, e.g., Poisson, Bernoulli, Markov models, etc.; and the one that seems most appropriate for the problem should be chosen. In subsequent stages, the model accuracy can be improved by increasing its complexity, such as adding states and parameters. However, there are trade-offs between simple and more complex models. On one hand, simple models are amenable to symbolic analysis, which can lead to a deeper understanding of the system. We can see how the system behavior changes with certain parameters: what is within our control and what is not. But simple models are less accurate. On the other hand, the more complex a model is, hopefully, the more accurate it gets, although we must avoid overfitting and introducing too many parameters. But as the models become more complex, it gets harder to do symbolic analysis, and we must resort to numerical simulations.

### 1.3 OUTLINE

The rest of this report is organized as follows. We first review some background material in Section 2, including the description of the experimental data and the system model for link-layer ARQ. In Section 3, we present two ways of analyzing the statistics of the experimental data, including a new concept we call a memory-decay curve. We start the model-building process with the simplest two-state Markov model in Section 4. This model was previously studied in [1], but we employ a new way of estimating the parameters. We also analyze the ARQ behavior in closed form, which is a benefit of the model being sufficiently simple. In Section 5, we move on to higher-order models. We select a higher-order model from two potential ones after studying a deficiency of the two-state model. We then estimate the model parameters and evaluate the resulting ARQ behavior via numerical simulation, as symbolic analysis becomes too difficult. We summarize our results in Section 6.



## 2. BACKGROUND

This section briefly reviews some background material. We first describe how the experimental data were originally obtained, show some examples, and demonstrate how the data are initially processed. We then review the basics of link-layer ARQ and describe the model and assumptions, as well as the quantities of interest.

### 2.1 EXPERIMENTAL DATA

The experimental data used in this study were taken from July to November 2002 in an effort led by J. B. Schodorf of MIT Lincoln Laboratory. A HMMWV (High Mobility Multipurpose Wheeled Vehicle) equipped with a MILSTAR terminal was driven along three separate routes. Each route is two to three miles long, takes about ten minutes to complete, and is repeated eight times. One route is in an *urban environment* near the MIT campus in Cambridge, MA, where buildings are the primary contributors of blockage. The other two are in Ft. Devens, MA, where blockages are mostly due to foliage. The one with less foliage is referred to as the *open environment* route. The one with denser foliage is referred to as the *rural environment* route. The statistics measured include received signal power, which is used in this study. The sampling interval is 0.1 second. All aspects about the experiment can be found in [1].

A 30-second segment of data from the urban environment is shown in Figure 3 on the left. The x-axis is time in seconds and the y-axis is the received signal power in dB relative to the line-of-sight (LOS) power. We see that the signal suffers from severe degradation from time to time. When the degradation exceeds a certain fade margin built into the system, the communication is lost completely, and the channel is considered blocked. The raw data can be simplified by converting it to a binary channel connectivity sequence, as shown on the right side of Figure 3, where 1 means open and 0 means blocked. It is our goal to build models that can produce binary sequences with the same characteristics. For this study, the fade margin is chosen to be 10 dB. We note that different fade margins could lead to different amounts of blockage and channel models. Generally, a smaller margin makes the blockage more severe in the same environment. The effect of changing the fade margin was addressed to some extent in [1], but is not addressed in this study.

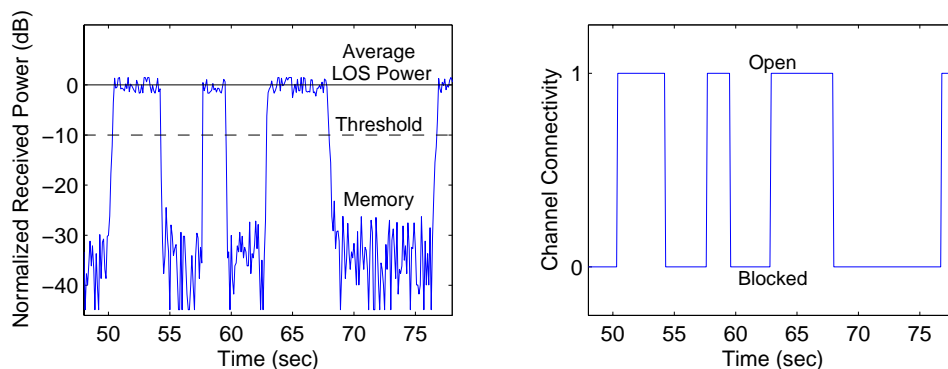


Figure 3. Example of experimental data and initial processing: A 30-second segment of experimental data from the urban environment (left) is converted to a binary sequence (right) using a fade threshold of 10 dB. The nature of the COTM channel is blockage with memory.

Figure 3 shows that each blockage can last up to several seconds at a time, so there is memory in the system. The intuition is that if some object is blocking communication currently, it is highly likely that it is still blocking a short time from now. It takes time to get out of blockage. Therefore, the nature of the COTM channel is *blockage with memory*.

Figure 4 displays the channel connectivity sequence for one complete run of each route in the three different environments. These are essentially all the data available for this study since the data sequences from the same route do not vary significantly. Each sequence is about ten minutes long. The most important and easily measurable channel statistic is the fraction of time that the channel is blocked. This quantity is 15%, 34%, and 26%, for the open, rural, and urban environments, respectively, for the 10-dB fade margin.

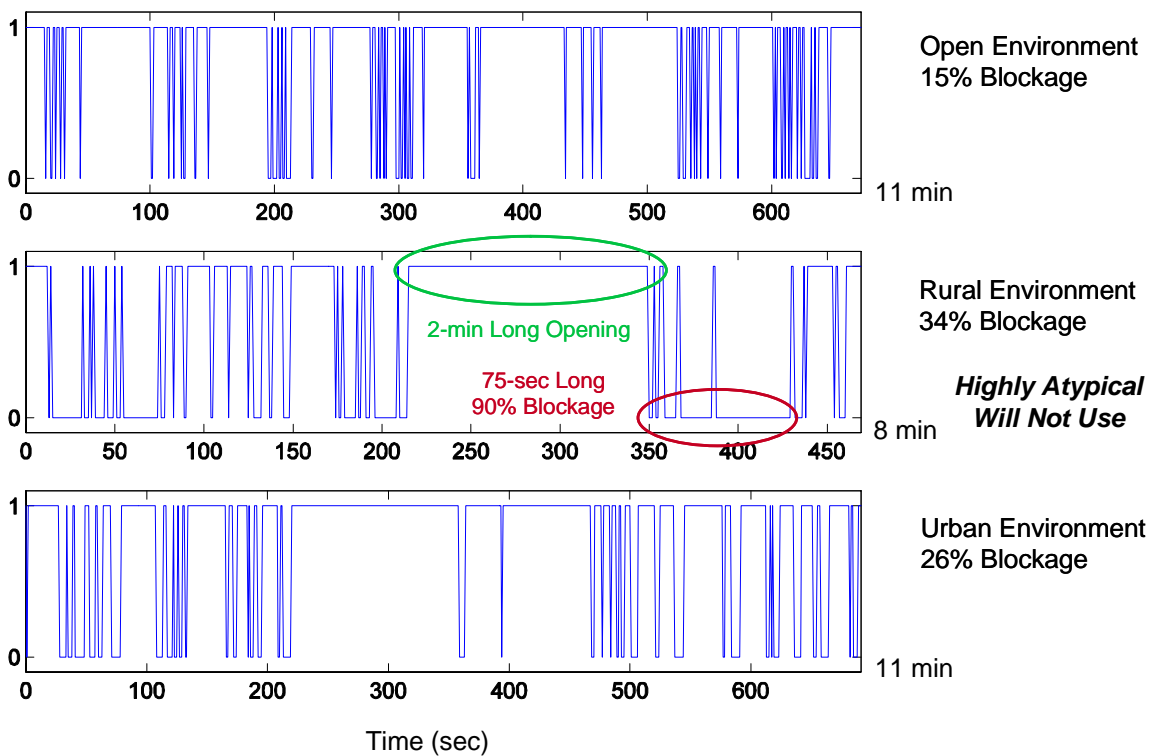


Figure 4. All available channel connectivity data (excluding repetition) from open, rural, and urban environments. Because the rural data are highly atypical, they are not used.

Figure 4 shows that the rural environment data are highly atypical. There is a 2-minute-long segment where the channel is open continuously, which is due to a large grassy field nearly a mile long, and a 75-second-long segment where the channel is blocked almost 90% of the time. These two possibly rare events occupy a significant portion of the 8-minute total length, making the data ill-suited for analysis. It is also highly unlikely that the data represent a “typical” outcome expected in a rural environment. For these

reasons, the rural data set is entirely omitted, and channel models will be developed and validated only for the open and urban environments.

We emphasize that the channel models built from these data sets are to some extent particular to the environment in which the data were collected and the fade margin used. Furthermore, the vehicle speed also affects the channel model in the sense that slower vehicles experience longer blockages in the same environment. Despite these limitations, the models built in this study do provide reasonable sample points for what could potentially happen in the real world. Further study on the robustness of the models is needed.

## 2.2 LINK-LAYER ARQ

This section serves as a review of the basics of link-layer ARQ (automatic repeat request) over a blockage channel. We discuss the motivation for using ARQ, describe the model and assumptions, and specify the quantities of interest.

The problem with transmission over a blockage channel is that packets are successfully delivered when the channel is open and dropped when the channel is blocked. The ARQ technique, which automatically repeats dropped packets, is a method of blockage mitigation. In the selective-repeat ARQ scheme used in this study, every time the receiver successfully receives a packet, it returns an acknowledgment back to the transmitter to confirm reception. If the transmitter does not get an acknowledgment by a certain time, it assumes the packet is lost and retransmits it.

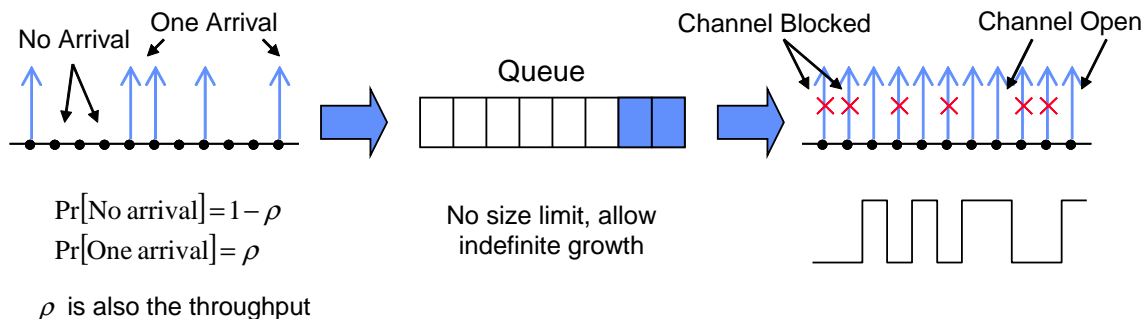


Figure 5. ARQ model at the transmitter in which packet arrivals are modeled by a stationary Bernoulli process and packet departures are governed by the blockage channel.

Figure 5 depicts what happens at the transmitter in a simple ARQ system. For simplicity, we consider a discrete-time system in which each packet arrives and departs instantaneously at integer multiples of a time unit. When the data rate is sufficiently high, this is a reasonable approximation. As shown in Figure 5, packets arrive at the transmitter from data sources, and they wait inside a queue for their turn to be transmitted. When the channel is blocked, the packets are dropped (indicated by “x”), and retransmissions

are required. So, the queue also serves the purpose of storing packets for potential retransmissions. A packet is only removed from the queue after it has been acknowledged. So the queue grows when the channel is blocked and shrinks when the channel is open at rates which are dependent on the system throughput. Since the amount of memory needed directly affects the cost of the system, queue length is a very important statistic. In this study, we do not put a size limit on the queue, so that we can observe the tail behavior and know what would be lost if the queue were finite.

In addition to queue length, another important statistic is delay. Since dropped packets are retransmitted, all packets are successfully delivered eventually. However, some packets might take many retransmissions and, as a consequence, have large delays. Since some applications are delay sensitive, it is important for us to know the delay profile associated with using ARQ. In this study, we do not put a limit on the maximum number of times each packet can be retransmitted, although a limit is usually enforced in practice.

To simplify analysis, we make two additional assumptions that are specific to this study. First, the packet arrivals are modeled as a stationary Bernoulli process with independent samples and an average arrival rate of  $\rho$  packets per discrete time unit, meaning that during each time unit, there is a probability of  $\rho$  that one packet arrives and a probability of  $1 - \rho$  that no packet arrives. Although more than one packet may arrive at a time in reality, this is a close approximation that captures the average arrival rate  $\rho$ , which is also the effective throughput, because whatever goes in must come out in order to maintain a stable queue. Secondly, we assume immediate and perfect acknowledgments. This means a successfully transmitted packet is immediately removed from the queue and a blocked one is immediately retransmitted. Therefore, one packet leaves the queue each time the channel is open unless the queue is empty, which is a very simple rule. These two assumptions will allow us to derive the averages and distributions of both queue length and delay for simple blockage channel models.<sup>2</sup>

In all the simulations performed in this study, the time unit for packet arrivals and departures is chosen to be the sampling interval of the channel sequences, which is 0.1 second. This avoids the need to upsample or downsample the channel sequences. We try to make most of our results relatively independent of this time unit by measuring delay in seconds and queue length in seconds' worth of transmission. The quantity  $\rho$ , which is both the packet arrival rate and the effective throughput, is dimensionless, as it measures the fraction of time in which packet arrivals and departures happen. Since only one packet can be transmitted per discrete-time unit and the channel must be open for that packet to go through, the maximum throughput is the fraction of time that the channel is open, which is 0.85 for the open environment and 0.74 for the urban environment for our data sets. The packet arrival rate of  $\rho = 1$  can only be supported if the channel is open continuously.

---

<sup>2</sup>In the ARQ study presented in [2], the case of non-immediate, but still perfect, acknowledgments is examined in depth, and experiments using non-Bernoulli arrival processes are briefly mentioned. Study to characterize the effect of imperfect acknowledgments may take place in the future.

### 3. STATISTICAL ANALYSIS METHODS

In order to build models that have the same statistical behavior as the experimental data, we must first extract key statistics from the data. One important question is which statistics to use. Since different statistics capture different aspects of the data and have different degrees of robustness, which statistics we use affects the models we build. Furthermore, since the experimental data are usually not modeled exactly, it is unlikely that multiple statistics can be matched simultaneously.

Opening and blockage duration distributions were used in the earlier study in [1] after the experimental data were originally taken. For a given class of model, model parameters were estimated to reproduce the empirical opening and blockage distributions. However, we will show that this is insufficient. In particular, even if the opening and blockage duration distributions were matched exactly, the loss of information embedded in the relative order of the opening and blockage segments could lead to large inaccuracies in terms of ARQ average queue-length estimates produced by the model. With the understanding of this deficiency of the opening and blockage duration distributions, we propose a new statistic called a memory-decay curve.

#### 3.1 OPENING AND BLOCKAGE DURATIONS

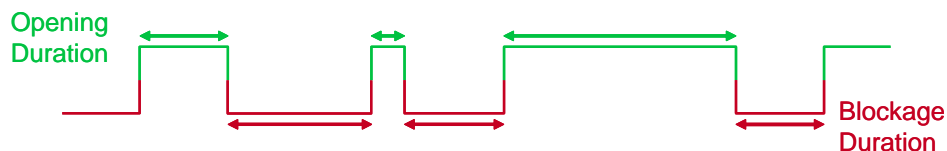


Figure 6. Demonstration of measurement of opening and blockage durations from channel connectivity sequence.

The opening and blockage durations, i.e., how long the channel stays open or blocked each time, can be measured from the channel connectivity sequence as shown in Figure 6. The distributions of the two durations can then be examined separately. Figure 7 shows the opening and blockage duration histograms for the open environment. The x-axis is the duration in seconds, and the y-axis is the count on a log scale. A distribution that appears linear on a log scale is exponential. So we see that the blockage duration is approximately exponential. The log scale also emphasizes the rare events of having really long openings. So, the opening duration is exponential with a heavy tail. Some consequences of this tail will be discussed later in Section 5.1, which addresses the issue of choosing higher-order models.

While the opening and blockage duration distributions are very important statistics, they do not capture any information embedded in the relative order of the opening and blockage segments. It turns out that if we were to preserve the duration distributions alone and ignore the relative order, the system behavior may not be accurately reproduced, at least in terms of ARQ average queue length.

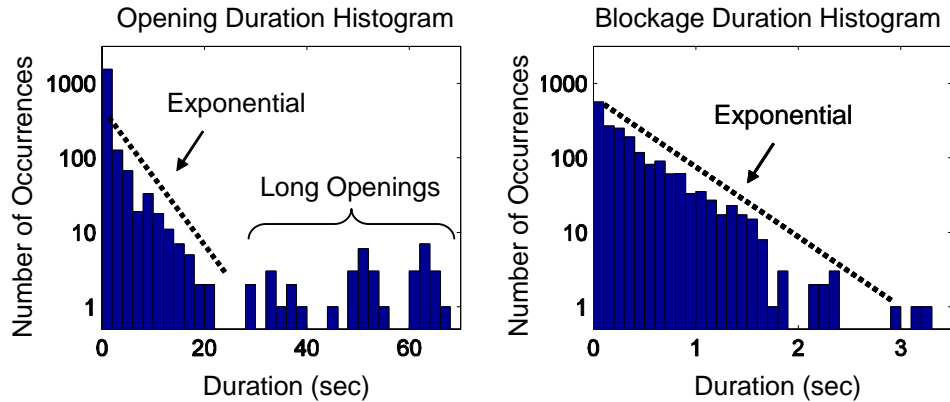


Figure 7. Opening duration and blockage duration histograms for the open environment. The log scale emphasizes the exponential distributions and the long openings.

Using a graphical example depicted in Figure 8, we first demonstrate a problem associated with ignoring the relative order. Suppose there are a few blockages of duration 10 and openings of duration 30, broken up by some opening and blockage segments of duration one. With the original ordering as shown on the left, there is effectively a 30-sample-long blockage, which will most likely lead to a large queue buildup in an ARQ scenario. However, if the segments were shuffled in a somewhat random fashion, as shown on the right, then the effectively long blockage would be broken up, and the queue would not build up as much. From this example, we see that the order clearly matters.

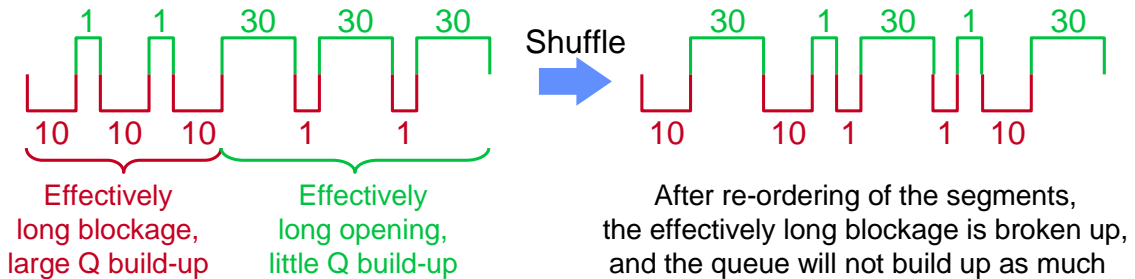


Figure 8. Example of shuffling opening and blockage durations to demonstrate that changing the order of the segments may affect ARQ queue behavior.

Although we have shown that the ordering of the opening and blockage segments may potentially affect the queue behavior in an ARQ scenario, we still need to evaluate how severe this phenomenon actually is in our data. To do this, we ran ARQ simulations for both open and urban environments, with throughputs  $\rho = 0.6$  and  $\rho = 0.4$ , respectively, with both the original data and three randomly shuffled versions, and evaluated the resulting average queue lengths, which are tabulated in Table 1. The throughputs were chosen to keep the average queue length reasonably small. The original data resulted in average queue lengths equaling 0.67 and 0.91 seconds' worth of transmission for open and urban environments, respectively. Once the opening and blockage segments were shuffled randomly three separate times, the results changed. We see that while there is little difference between the three random shuffles, compared to the original, the errors are about 40% and 20%, respectively. This demonstrates that the effect of shuffling is fairly severe in our data, especially for the open environment.

**Table 1**  
**The Effect of Shuffling Opening and Blockage Segments**

Environment	Throughput	Average Queue Length (seconds' worth of transmission)				
		Original	Shuffle Three Separate Times			Error
Open	0.6	0.67	0.42	0.40	0.42	-40%
Urban	0.4	0.91	1.14	1.07	1.08	+20%

In summary, we showed that even if we could perfectly capture the distributions of the opening and blockage durations, the loss of the ordering information could still lead to significant error in terms of average queue length in an ARQ scenario.

### 3.2 NEW IDEA: MEMORY-DECAY CURVE

We propose a new statistical measure that focuses on characterizing the channel memory, the memory-decay curve. We measure the decay in memory over time in terms of how much the current sample tells us about samples nearby and far away, both in the past and in the future. More specifically, for a discrete time system, we look at the  $t$  samples before and after the current sample, and measure the average fraction of time that the channel is the same as the current state. We believe this is more directly related to system

behavior, in particular, for ARQ. Mathematically, the memory-decay curve is defined as

$$M(t) = \frac{1}{P_B} \left( \frac{1}{2t} \sum_{1 \leq |i| \leq t} \Pr [s_i = 1 | s_0 = 1] - P_U \right) \quad (1)$$

$$= \frac{1}{P_U} \left( \frac{1}{2t} \sum_{1 \leq |i| \leq t} \Pr [s_i = 0 | s_0 = 0] - P_B \right). \quad (2)$$

In the first equation, given that the channel is currently open ( $s_0 = 1$ ), for all samples within  $t$  samples, we count the number of ones probabilistically and take the average.<sup>3</sup> We then normalize it by subtracting  $P_U$ , the fraction of time that the channel is unblocked or open, and dividing by  $P_B = 1 - P_U$ , the fraction of time that the channel is blocked. The normalization is chosen so that as  $t$  approaches infinity, when all the memory should have decayed,  $M(t)$  diminishes to zero.

One interesting property that the memory-decay curve has is that it can be computed both from the openings using Equation (1) and from the blockages using Equation (2), thereby showing that it truly captures the memory in the system. The proof for this equality is in Appendix A.

The memory-decay curves for the open and urban environments are shown in Figure 9, with the x-axis normalized to seconds. We see that they both decay monotonically, but have different time scales and slightly different shapes.

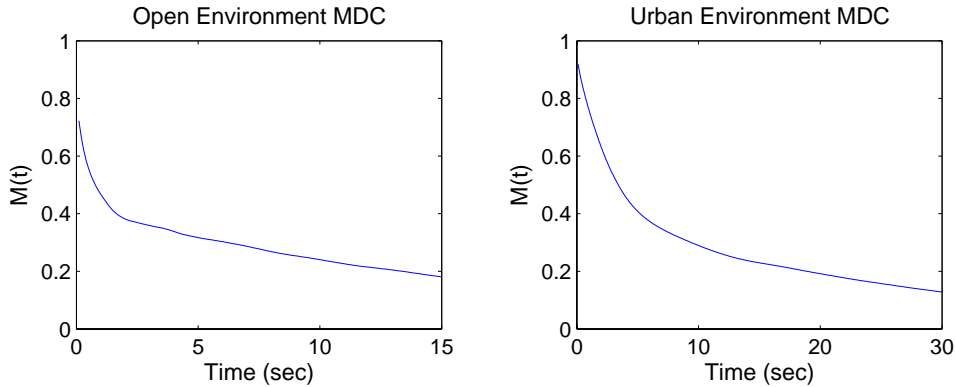


Figure 9. Experimentally measured memory-decay curves for the open and urban environments. They both decay monotonically, but have different time scales.

---

<sup>3</sup>For a continuous-time model, the summation and average should be replaced by an integral and average.

## 4. TWO-STATE MARKOV MODEL

In this section, we study the two-state Markov model, which is one of the simplest models that allows the channel to have memory. Higher-order models will be studied in the next section. In comparison, the main advantage of the two-state Markov model is that its simplicity enables us to perform analytical studies in closed form, from which useful insights can be obtained.

We first describe the model in Section 4.1. In Section 4.2, we derive the memory-decay curve expression for the model and estimate the model parameters by performing a least-squares fit for each experimentally measured memory-decay curve given in Figure 9. We then go on to evaluate the validity of the model in terms of how well it replicates the behavior of the experimental data in ARQ scenarios. We first derive the ARQ queue-length and delay distributions and averages in closed form for the two-state Markov model in Section 4.3 and then compare the results obtained from the experimental data and those predicted by the model in Section 4.4.

### 4.1 MODEL DESCRIPTION

In a two-state Markov model, the channel switches between an *unblocked* state and a *blocked* state with certain transition probabilities as shown in Figure 10. When the channel is in the unblocked state, or open, all packets successfully go through. When the channel is in the blocked state, all packets are dropped.

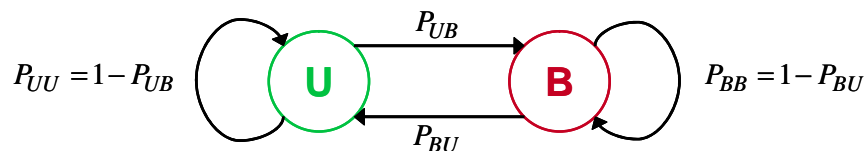


Figure 10. The simple two-state Markov model. The two states are labeled U for unblocked and B for blocked.

This simple model can be fully specified by just two parameters. The most intuitive pair is the average connection and blockage durations measured in seconds, denoted by  $T_U$  and  $T_B$ . Although the channel goes in and out of blockage continuously in reality, for the purpose of analysis, it is more convenient to consider a discrete-time model with a time unit  $\tau$ , i.e., the channel state only changes at integer multiples of  $\tau$ .<sup>4</sup> In this study, we assume all packets have equal size and choose  $\tau$  to be the packet arrival and departure interval in the ARQ model in Section 2.2. This way, for each time unit the channel is open, one packet may go through; for each time unit the channel is blocked, one packet may be blocked. While these discrete approximations are not accurate, they significantly simplify the problem setup without losing the essence of what we want to

---

<sup>4</sup>A discrete-time channel sequence can be obtained from a continuous-time one by performing sampling at integer multiples of  $\tau$ . The experimental data we have are essentially discrete.

study. With this discrete-time two-state Markov model, another convenient choice for parameterization is to use the state transition probabilities,  $P_{UB}$  and  $P_{BU}$ , as labeled in Figure 10. They are related to  $T_U$  and  $T_B$  by  $P_{UB} = P_B (1 - e^{-\tau/T_0})$  and  $P_{BU} = P_U (1 - e^{-\tau/T_0})$ , where  $T_0 = T_U T_B / (T_U + T_B) = T_U P_B = T_B P_U$ .<sup>5</sup> When  $\tau$  is small,  $P_{UB} \approx \tau/T_U$  and  $P_{BU} \approx \tau/T_B$ . From either pair of parameters, the fraction of time that the channel spends in the unblocked and blocked states can be computed using

$$P_B = \frac{T_B}{T_U + T_B} = \frac{P_{UB}}{P_{BU} + P_{UB}} = 1 - P_U \quad (3)$$

and

$$P_U = \frac{T_U}{T_U + T_B} = \frac{P_{BU}}{P_{BU} + P_{UB}} = 1 - P_B. \quad (4)$$

As in any stochastic modeling problem, it is important to understand the physical world scenario that could result in such a model. The two-state Markov model has exponentially distributed opening and blockage durations with their respective means. Therefore, if we were in an environment where the widths of the buildings that cause blockage and the gaps between them were both exponentially distributed with corresponding means, as illustrated in Figure 11, then as we drive by at a constant speed, the channel would be precisely modeled by the two-state Markov model. Obviously, this does not model the real world exactly; building spacing is usually not very irregular in an urban environment and vehicle velocity is usually not constant. However, we can always choose to consider more complex models as the need arises.

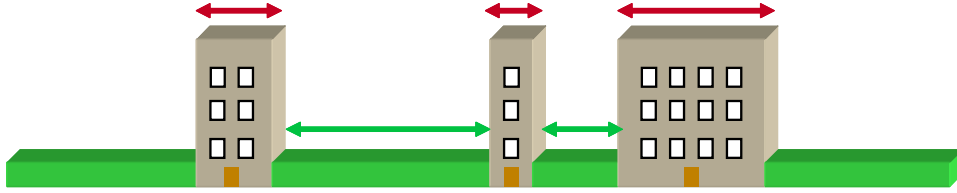


Figure 11. Physical world scenario associated with the two-state Markov model. The model is exact if both the widths of the buildings and the gaps between them are exponentially distributed and the vehicle velocity is constant.

## 4.2 PARAMETER ESTIMATION USING MEMORY-DECAY CURVE

We estimate the two-state Markov model parameters from the experimental data using the memory-decay curve concept introduced in Section 3.2. To do so, we first need to derive the memory-decay curve

---

<sup>5</sup>This can be found in many textbooks including Alberto Leon-Garcia, *Probability and Random Process for Electrical Engineering*. Addison-Wesley, 1989, Page 449.

expression for the two-state Markov model and then perform a least-squares fit of the experimentally measured memory-decay curves in Figure 9 to estimate the parameters.

We compute the memory-decay curve from both the openings using Equation (1) and from the blockages using Equation (2) and show that the two ways indeed lead to the same answer. The multistep transition probabilities,  $\Pr [s_i = 1|s_0 = 1]$  and  $\Pr [s_i = 0|s_0 = 0]$ , are the same for  $+i$  and  $-i$  due to the reversibility of the two-state Markov chain. They can be computed from

$$\begin{bmatrix} \Pr [s_i = 1|s_0 = 1] & \Pr [s_i = 1|s_0 = 0] \\ \Pr [s_i = 0|s_0 = 1] & \Pr [s_i = 0|s_0 = 0] \end{bmatrix} = \begin{bmatrix} 1 - P_{UB} & P_{BU} \\ P_{UB} & 1 - P_{BU} \end{bmatrix}^{|i|} \quad (5)$$

Using eigenvalue decomposition, it can be shown that

$$\begin{aligned} \begin{bmatrix} 1 - P_{UB} & P_{BU} \\ P_{UB} & 1 - P_{BU} \end{bmatrix}^{|i|} &= \begin{bmatrix} P_{BU} & 1 \\ P_{UB} & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \lambda^{|i|} \end{bmatrix} \begin{bmatrix} P_{BU} & 1 \\ P_{UB} & -1 \end{bmatrix}^{-1} \\ &= \begin{bmatrix} P_U + P_B \lambda^{|i|} & P_U (1 - \lambda^{|i|}) \\ P_B (1 - \lambda^{|i|}) & P_B + P_U \lambda^{|i|} \end{bmatrix} \quad \text{where } \lambda = 1 - P_{UB} - P_{BU}. \end{aligned}$$

Therefore,

$$\Pr [s_i = 1|s_0 = 1] = P_U + P_B \lambda^{|i|} \quad \text{and} \quad \Pr [s_i = 0|s_0 = 0] = P_B + P_U \lambda^{|i|}. \quad (6)$$

Substituting Equation 6 into Equations (1) and (2), it is clear that they lead to the same memory-decay curve for the two-state Markov model,

$$M(t) = \frac{1}{2t} \sum_{1 \leq |i| \leq t} \lambda^{|i|} = \frac{\lambda(1 - \lambda^t)}{t(1 - \lambda)}. \quad (7)$$

We note that the normalization in the definition of the memory-decay curve makes it independent of  $P_U$  and  $P_B$ . The decay of memory over time is only parameterized by  $\lambda = 1 - P_{UB} - P_{BU}$ , which represents how fast the channel changes.<sup>6</sup> Therefore, another good pair of parameters for characterizing the two-state model is  $P_B$  and  $\lambda$ .<sup>7</sup>

We are now ready to estimate  $P_B$  and  $\lambda$  from the experimental data and then subsequently obtain  $T_U$  and  $T_B$ .  $P_B$  can be directly estimated from the experimental data simply by counting the fraction of time that the channel is blocked. To estimate  $\lambda$ , we perform a least-squares fit for each experimentally measured memory-decay curve given in Figure 9 with the analytical expression in Equation (7). The results are shown in Figure 12 with the open environment plot on the left and the urban environment one on the

---

<sup>6</sup>The quantity  $\lambda$  is also referred to as channel memory in [3].

<sup>7</sup>The continuous-time model counterpart of  $\lambda$  is  $T_0 = T_U T_B / (T_U + T_B)$ . They are related by  $\lambda \approx 1 - \tau / T_0$ . Therefore,  $P_B$  and  $T_0$  are a good pair of parameters for characterizing the continuous-time two-state Markov model.

right. The blue solid curves are experimentally measured and the red dashed ones are the least-squares fits using Equation (7). We see that while the fit is reasonably close for the urban environment, there are large gaps for the open environment, suggesting that the open environment is probably not sufficiently well modeled by the two-state Markov model.

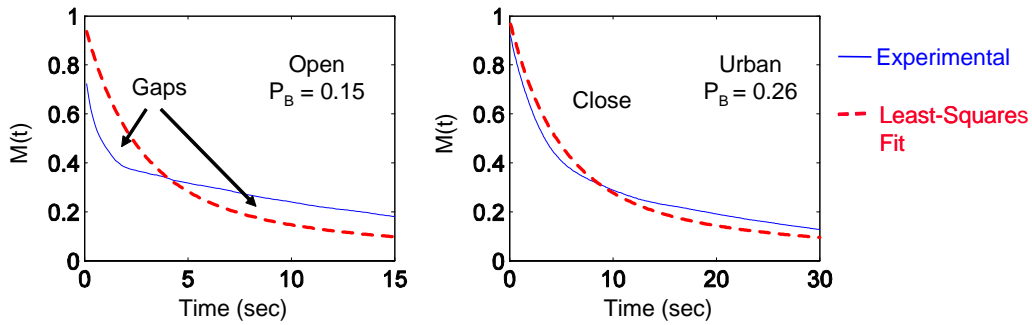


Figure 12. Estimation of two-state Markov model parameters by least-squares fit of the memory-decay curve for both the open and urban environments.

The parameters estimated for both the open and urban environments are tabulated in Table 2 in terms of the average opening and blockage durations,  $T_U$  and  $T_B$ . We see that the openings are about ten seconds long on average, and the blockages are around two and four seconds long on average for the open and urban environments, respectively. It is reasonable for the urban environment to have a longer blockage than the open environment, since the blockages are mainly caused by buildings instead of by foliage. The average squared errors from the least-squares fits are also given to demonstrate the difference in the closeness of the fits.

**Table 2**  
**Estimated Two-State Markov Model Parameters**

Environment	$T_U$	$T_B$	Squared Error
Open	10.2 sec	1.8 sec	0.013
Urban	11.2 sec	4.0 sec	0.002

### 4.3 ARQ BEHAVIOR ANALYSIS

One of the key features that distinguishes this study from the previous one in [1] is that the models are validated using specific scenarios, in particular, ARQ. In this section, we derive in closed form the queue-length and delay distributions and averages resulting from running ARQ over a blockage channel modeled by the two-state Markov model. This ability to perform closed-form analysis of the system is a key advantage of working with a simple model. We compare the results obtained from the experimental data and those predicted by the model in the next section.

The ARQ model and assumptions were introduced earlier in Section 2.2. In an ARQ scheme, dropped packets, which do not produce acknowledgments from the receiver, are retransmitted. If we assume that the success of each packet transmission is known immediately and perfectly at the transmitter, so retransmission can happen immediately, then one packet leaves the queue each time the channel is open unless the queue is empty. We also assume independent Bernoulli arrival statistics with an average rate of  $\rho$  packets per discrete-time unit  $\tau$ . For simplicity, we use the same discrete-time unit for both packet arrivals and channel transitions. These assumptions are not accurate, but the results can serve as good indicators of what happens in the real system.

The behavior of such an ARQ scheme over a blockage channel modeled by the two-state Markov model can be solved by building a composite Markov chain where each state is labeled with both the length of the queue and the state of the channel (effective for the next packet transmission)<sup>8</sup> as shown in Figure 13. This captures all of the memory in the system.

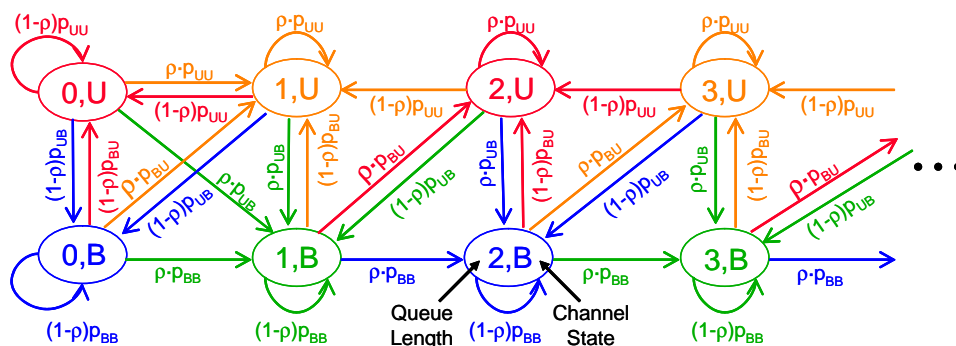


Figure 13. A two-layer semi-infinite Markov chain for analyzing the behavior of ARQ over a blockage channel modeled by the two-state Markov model, assuming perfect and immediate acknowledgments and Bernoulli arrivals.

<sup>8</sup>The order of events within one time unit is transmit packet, receive ACK or NACK, new packet arrives, channel state changes.

Let us examine this two-layer semi-infinite Markov chain closely. At each time, there can be zero or one arrival and the channel may switch state, so there are four branches leaving each state. Since arrivals and channel transitions are independent, the joint probability is the product of the two. To determine the new state, we also need the number of departures. There is a departure if and only if the state is  $(i, U)$ ,  $i \geq 1$ , and departures happen one at a time. All transitions and their probabilities are labeled in Figure 13. By analyzing this Markov chain, we can solve for the distributions and averages of both queue length and delay.

For brevity, we defer all derivations to Appendix B and just present the main results here. We also assume that the discrete-time unit  $\tau$  is small and present the queue-length results in seconds' worth of transmission and the delay results in seconds, as functions of  $T_B$ ,  $P_B$ ,  $P_U = 1 - P_B$ , and the arrival rate  $\rho$ . Most of these results are also presented in [2]. We emphasize that all the results and intuitions obtained are possible because the two-state Markov model is sufficiently simple.

1. The average queue length is

$$T_B P_B P_U \left( \frac{P_U}{\rho} - 1 \right)^{-1} \text{ seconds' worth of transmission.} \quad (8)$$

First of all, we use the unit of seconds' worth of transmission because the average queue length is proportional to the data rate. The greater the data rate, the more data that have to be queued. Secondly, the average queue length is proportional to the average blockage duration  $T_B$  and the probability of blockage  $P_B$ . The longer each blockage lasts, and the more frequently it happens, the more data that have to be queued. Thirdly, the average queue length turns out to be a product of two terms. The first one,  $T_B P_B P_U$ , is called the environment factor. It is entirely determined by the channel statistics, which are beyond our control. The second term,  $\left( \frac{P_U}{\rho} - 1 \right)^{-1}$ , is called the greedy factor, which is within our control. It approaches infinity as the arrival rate  $\rho$  approaches the maximum throughput possible,  $P_U$ , which is the fraction of time that the channel is unblocked. The more greedy we are, the closer  $\rho$  is to  $P_U$  and the longer the queue gets. When  $\rho \geq P_U$ , packets arrive faster than they can leave, the queue grows indefinitely, and the steady-state distribution does not exist. This is an important result. It can potentially help us determine what data rate to operate at, depending on how much we want to queue and the environment. However, this topic is beyond the scope of this report.

2. The queue-length probability density function is

$$p_Q(t) = \frac{P_U - \rho}{1 - \rho} \delta(t) + \frac{P_B}{1 - \rho} \alpha e^{-t\alpha}, \text{ where } \alpha = \frac{P_U - \rho}{T_B P_U (1 - \rho) \rho}, \quad (9)$$

and the unit is seconds' worth of transmission. The distribution is an impulse near zero followed by an exponential distribution. The impulse tells us how often the queue is actually (near) empty. The exponential tail tells us how much data would be lost if a finite-length queue were used.

3. The average delay, defined as from the moment a packet enters the queue to the moment it is removed after it is acknowledged, is

$$T_B P_B \left( 1 - \frac{\rho}{P_U} \right)^{-1} \text{ seconds.} \quad (10)$$

This can be obtained from the average queue-length result in Equation (8) using Little’s formula,<sup>9</sup> which says that the average queue length equals the average arrival rate times the average time each packet spends in the queue, which is the average delay. Just like the average queue length, the average delay is also proportional to the average blockage duration  $T_B$  and the probability of blockage  $P_B$ . It is also the product of an environment factor,  $T_B P_B$ , and a greedy factor,  $\left(1 - \frac{\rho}{P_U}\right)^{-1}$ .

4. The delay probability density function is

$$p_d(t) = \frac{P_U - \rho}{1 - \rho} \delta(t) + \frac{P_B}{1 - \rho} \beta e^{-t\beta}, \quad \text{where } \beta = \frac{P_U - \rho}{T_B P_U (1 - \rho)}, \quad (11)$$

where the unit is seconds. This distribution also decays exponentially after an initial impulse near zero. The exponential decay is a factor of  $\rho$  slower, and the impulse has the same magnitude compared to the queue-length distribution in Equation (9). Intuitively, if a packet arrives at an empty queue, at which time the channel is almost certainly open, the packet goes through right away.

#### 4.4 ARQ BEHAVIOR COMPARISON

In this section, we compare the ARQ behavior resulting from the experimental data and that predicted by the model. Because of the many similarities between the queue length and the delay, in terms of both the averages and the distributions, we only focus on the queue length and do not bother with the delay. As stated in Section 2.2, for all the numerical simulations performed, the discrete-time unit is chosen to be the channel sampling interval, 0.1 second.

The comparison is shown in Figure 14. The rugged, blue, solid curves are from the experimental data, and the more smooth, red, dashed ones are predictions using the two-state Markov model with the parameters we estimated. These curves are normalized queue-length distributions. The x-axis is queue length in seconds’ worth of transmission. The y-axis is normalized so that the area under each curve is the average queue length. For each open (left) and urban (right) environment, we run simulations for two sets of arrival rates because we want to test the validity of the model at different operating points. The simulations in the bottom figures have half the arrival rates relative to the top ones. We see that for the urban environment there is a reasonably close match between the distributions. The differences in the averages are 4% and 20%, which are not very large. However, for the open environment, there are some gross mismatches, especially for the lower arrival-rate one. The error in the average is almost 70%. Recall that in the memory-decay curve fits in Figure 12, we saw that by using the two-state Markov model, we were able to produce a reasonably close fit for the urban environment, but could not fit the open environment one very well. The accuracy of the match between the queue-length distributions seems to agree with the quality of the memory-decay curve fit.

---

<sup>9</sup>Little’s formula can be found in many textbooks including Ronald W. Wolff, *Stochastic Modeling and The Theory of Queues*. New Jersey:Prentice Hall, 1988, Page 236.

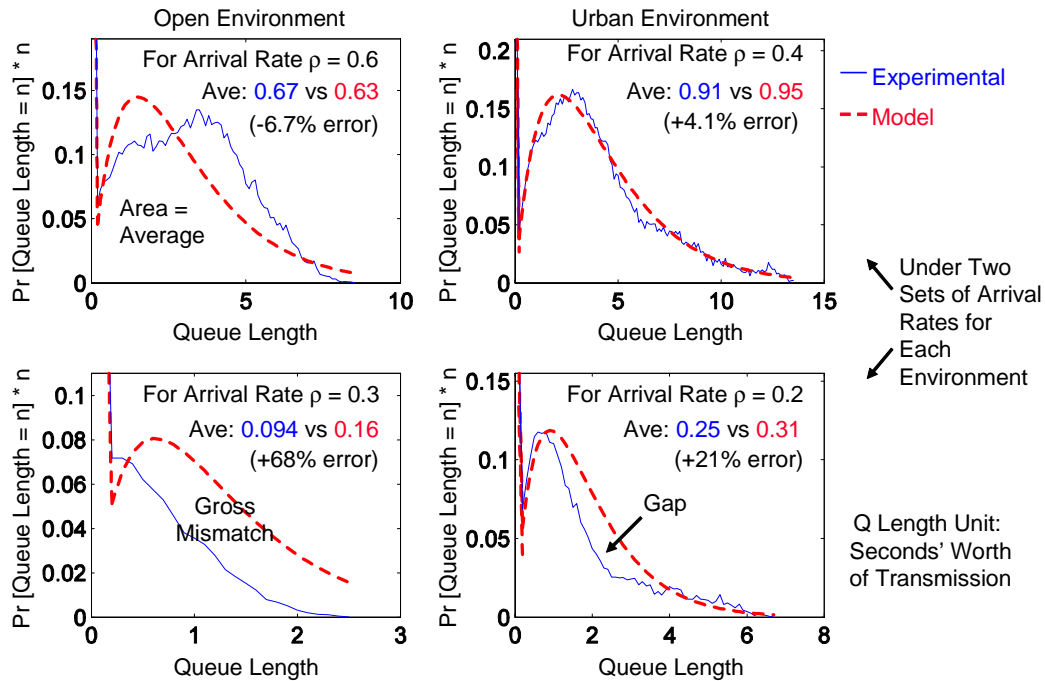


Figure 14. Comparison of the queue-length distributions and averages obtained from the experimental data and the two-state Markov model. There are large gaps for the open environment.

Table 3 summarizes the quality of the two-state Markov model in terms of its ability to predict ARQ behavior over a blockage channel. For the urban environment, the memory-decay curve fit is close, the average queue-length prediction is within 20%, and the queue-length distributions are close. All this leads to the conclusion that the two-state Markov model works reasonably well for the urban environment. However, for the open environment, the memory-decay curve fit has large gaps, the average queue-length prediction error is up to 70%, and there are large gaps between the queue-length distributions. These results indicate that the two-state Markov model is not as good for the open environment. Nevertheless, we re-emphasize that the main advantage of the two-state Markov model is that its simplicity allows closed-form analysis of the system.

**Table 3**  
**Summary of Two-State Markov Model Performance**

Environment	MDC Fit	Average Queue-Length Error	Queue-Length Distributions	Overall
Urban	Close	Within $\sim 20\%$	Some Gaps	Acceptable
Open	Large Gaps	Up to $\sim 70\%$	Large Gaps	Unsatisfactory

## 5. HIGHER-ORDER MODELS

In this section, we extend our study to higher-order models, building upon what we learned from the simple two-state Markov model in the last section. First, in Section 5.1, we study a deficiency of the two-state Markov model and choose an appropriate higher-order model. The rest of the section follows an outline similar to the two-state Markov model case. In Section 5.2, we again derive the memory-decay curve for the model and then estimate the model parameters by performing a least-squares fit for each experimentally measured memory-decay curve given in Figure 9. We show that excellent fits can be achieved for both open and urban environments. In Section 5.3, we compare the queue-length distributions and averages obtained from the experimental data and those predicted by the model. We show that improved accuracy is achieved. Unfortunately, with the increased complexity over the two-state Markov model, we can no longer obtain closed-form expressions for the queue-length and delay averages and distributions.

### 5.1 CHOOSING A HIGHER-ORDER MODEL

Unlike the simple two-state Markov model, there are many ways to construct higher-order models. In a study by J.S. Slack [4] modeling the satellite channel from an error-burst perspective, four well-known Markov models were examined. The simplest one is the *Gilbert model* [5], which has one additional degree of freedom relative to the two-state Markov model and allows packets to go through the channel with a certain probability, in a memoryless manner, even when the channel is in the blocked state.<sup>10</sup> In the *Elliot model* [6], one more degree of freedom is added by also allowing packets to be blocked with a certain probability, also in a memoryless manner, when the channel is in the unblocked state. In the *McCullough model* [7], also known as the binary regenerative-channel model, the Elliot model is modified with the constraint that a transition between states is allowed only after a blockage occurs. The most complex model studied in [4] is the *Fritchman model* [8]. It is a multistate Markov model in which states are partitioned into two groups, with one group designated as unblocked states and the other designated as blocked states. One of the models we examine later falls in this category.

In this study, to guide us in building higher-order models, we first examine a deficiency of the two-state Markov model. Based on the findings, we propose two potential higher-order models and select one by comparing their ease in analyzing model statistics. We then examine the chosen model in more depth.

#### 5.1.1 A Deficiency of the Two-State Markov Model

At the end of Section 4, it was concluded that the two-state Markov model does not work well for the open environment. There are large gaps in the memory-decay curve fit, and the queue-length prediction errors are large. To understand the cause of the problem, we re-examine the channel-connectivity sequence in Figure 15, which was shown earlier as part of Figure 4.

---

<sup>10</sup>In the previous study by J.B. Schodorf [1], during which the data sets we now use were originally taken, the Gilbert model was also examined in addition to the two-state Markov model.

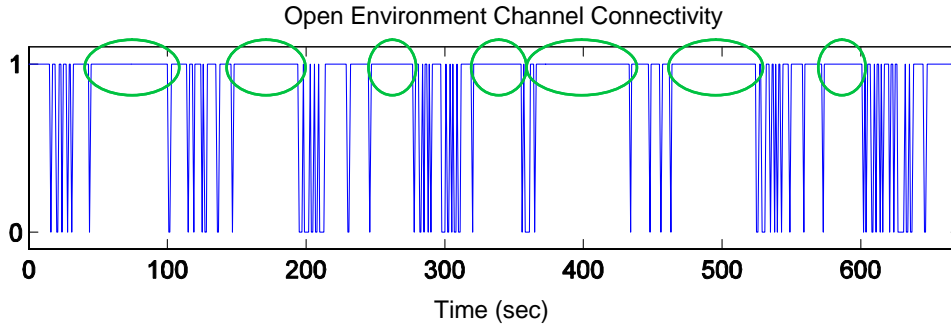


Figure 15. Demonstration of a deficiency of the two-state Markov model. In the open environment, there are particularly long openings that are more frequent than what the model predicts.

The cause of the problem seems to be the occurrences of many long openings, which are circled in green. Both the lengths and the frequency of these long openings turn out to be much more than what the two-state model predicts. This phenomenon actually manifested itself earlier in the opening duration histogram given in Figure 6. The histogram showed that there are long openings that are not modeled by the exponential distribution, which is associated with the two-state model. As a result of not capturing these long openings, the model underestimates the long-term memory; this phenomenon can be seen in the memory-decay curve fit in Figure 12. Therefore, the lesson is that we need to distinguish long openings from short openings; this then is how we next modify and improve our model.

### 5.1.2 Two Potential Higher-Order Models

We consider two potential higher-order models that both distinguish long and short openings. One is called a *multi-open-state* Markov model and is shown in Figure 16. The other one is called a *mixture* model and is shown in Figure 17. Both models are built upon the simple two-state model and are very similar in the channel characteristics they capture. We will choose between them by comparing their ease of analyzing model statistics.

In the multi-open-state model, starting from the two-state Markov model, we add a long open state relative to the original one, now called short-open, as shown in Figure 16. It is essentially a three-state model centered around the blocked state. This is a special case of the Fritchman model, except the transition between the long-open state and the short-open state is not allowed. The physical world scenario is very similar to the two-state model, except there are now two kinds of openings having different time scales and occurring with different probabilities.

In the mixture model, the original two-state Markov chain is treated as one joint state, and then a long-open state is added, as shown in Figure 17. The intuition is that the physical world is a mixture of two kinds of regions. In one region, called the unblocked region, the channel is always open, and in the other, called the blocked region, the channel is governed by the two-state model. The transition between regions

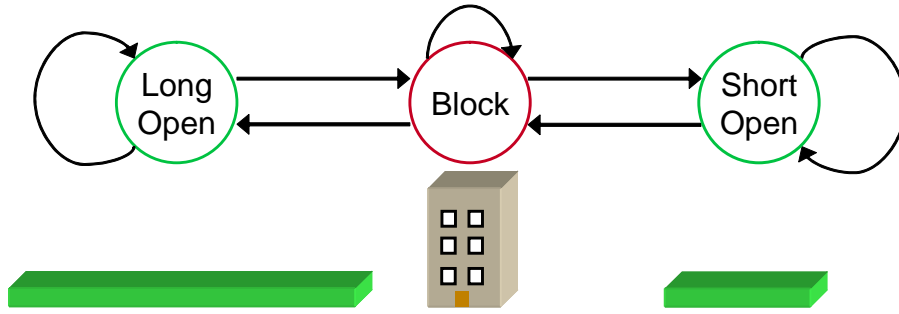


Figure 16. The multi-open-state Markov model and its corresponding physical world scenario. The model is obtained by adding a third long-open state to the original two-state Markov model.

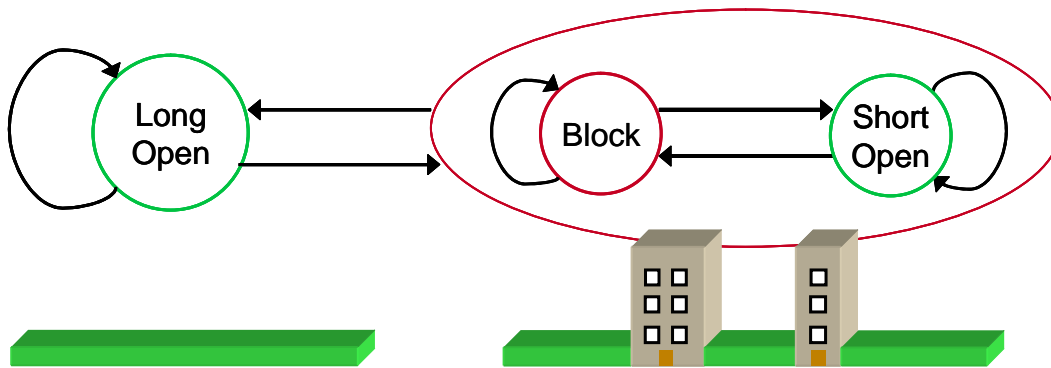


Figure 17. The mixture model and its corresponding physical world scenario. The model is obtained by treating the original two-state Markov model as one joint state and adding a long-open state.

happens at a much slower time scale compared to the change of state within the blocked region.<sup>11</sup> This model can be viewed as a generalization of the Gilbert model, the difference being while in the blocked region, channel openings are not memoryless, but governed by a relatively more complex two-state model.

These two models are in fact very similar conceptually. The only difference is in whether transition between the two open states is allowed. However, this difference turns out to be insignificant. To help us choose between these two models, we compare them in terms of the ease of analyzing certain model statistics, including steady-state probabilities and durations of various states, as tabulated in Table 4.

---

<sup>11</sup>We assume that while in the long-open state, the Markov process between the blocked state and the short-open state still goes on. This assumption is not related to any physical phenomenon, but is just to simplify analysis.

**Table 4**  
**Comparison of the Ease of Analyzing Model Statistics**

	Multi-Open-State Markov Model	Mixture Model
Steady-state probability of each state	Easy	Easy
Average duration of one stay in each state and its distribution	Easy	Easy
$P_U$ and $P_B$	Easy	Easy
$T_U$ and $T_B$	Easy	Easy
Blockage duration distribution	Easy	Easy
Opening duration distribution	Easy	<b>Hard</b>
Multistep transition probability, $\Pr [s_t   s_0]$	<b>Hard</b>	Easy
Memory-decay curve	<b>Hard</b>	Easy

Table 4 shows that the main differences between the two models are that it is relatively easier to compute the memory-decay curve with the mixture model, while it is relatively easier to compute the opening duration distribution with the multi-open-state Markov model. Since we mainly work with the memory-decay curve in this study, we choose the mixture model.

### 5.1.3 Mixture Model Parameterization

The mixture model is equivalent to two independent, two-state Markov chains with different statistics running in parallel as shown in Figure 18. The left Markov chain models the long-term behavior and determines which region we are in. In the  $U_1$  region, the channel is always open, independent of the state of the short-term Markov chain on the right. This corresponds to the long-open state. In the  $B_1$  region, the short-term Markov chain determines the state of the channel. So, the channel is blocked when both Markov chains are in state B. Being in state  $B_1$  and  $U_2$  simultaneously corresponds to being in the short-open state.

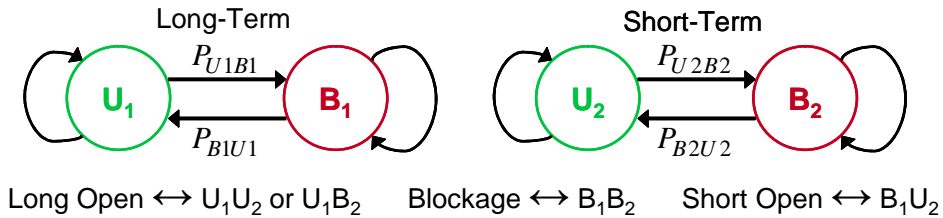


Figure 18. An equivalent representation of the mixture model as two parallel two-state Markov chains.

This mixture model can be fully characterized by four parameters since it consists of two two-state Markov models. The parameters can be the average durations of each state ( $T_{U1}, T_{B1}, T_{U2}, T_{B2}$ ) for a continuous-time model, or the transition probabilities ( $P_{U1B1}, P_{B1U1}, P_{U2B2}, P_{B2U2}$ ) for a discrete-time model. When the discrete time step  $\tau$  is small,  $P_{U1B1} \approx \tau/T_{U1}$ ,  $P_{B1U1} \approx \tau/T_{B1}$ ,  $P_{U2B2} \approx \tau/T_{U2}$ , and  $P_{B2U2} \approx \tau/T_{B2}$ . The steady-state probabilities of the states ( $P_{U1}, P_{B1}, P_{U2}, P_{B2}$ ) can be computed using Equations (3) and (4). However, given the mixture nature of the model, these quantities are not the most intuitive. The set of four parameters we prefer for describing the environment is the average long-open duration  $T_{\text{long open}}$ , average short-open duration  $T_{\text{short open}}$ , average blockage duration  $T_{\text{block}}$ , and the probability of blockage  $P_B$ . These are quantities that can be easily related to the physical world. They are related to ( $T_{U1}, T_{B1}, T_{U2}, T_{B2}$ ) by

$$T_{\text{long open}} = T_{U1}, \quad \frac{1}{T_{\text{short open}}} = \frac{1}{T_{B1}} + \frac{1}{T_{U2}}, \quad \frac{1}{T_{\text{block}}} = \frac{1}{T_{B1}} + \frac{1}{T_{B2}},$$

and

$$P_B = P_{B1}P_{B2} = \frac{T_{B1}}{T_{U1} + T_{B1}} \frac{T_{B2}}{T_{U2} + T_{B2}}.$$

## 5.2 PARAMETER ESTIMATION USING MEMORY-DECAY CURVE

In this section, we estimate the model parameters for the mixture model by performing least-squares fits of the experimentally measured memory-decay curves. We first derive the memory-decay curve expression for the mixture model. Since both Markov chains must be in state B for the channel to be blocked, conditioned on the channel being blocked, the states of both Markov chains are known. So it is easier to derive the memory-decay curve by conditioning on the blockage events, rather than on the openings. We have,

$$\Pr [s_i = 0 | s_0 = 0] = \Pr [s_i = B_1 | s_0 = B_1] \Pr [s_i = B_2 | s_0 = B_2] \quad (12)$$

$$= (P_{B1} + P_{U1} \lambda_1^t) (P_{B2} + P_{U2} \lambda_2^t), \quad (13)$$

where  $\lambda_i = 1 - P_{UiBi} - P_{BiUi}$  was defined earlier in Section 4.2. Substituting Equation (13) into the memory-decay curve expression in Equation (2), we have

$$M(t) = \frac{1}{1 - P_{B1}P_{B2}} \left( \frac{1}{2t} \sum_{1 \leq |i| \leq t} (P_{B1} + P_{U1} \lambda_1^t) (P_{B2} + P_{U2} \lambda_2^t) - P_{B1}P_{B2} \right) \quad (14)$$

$$= \frac{1}{1 - P_{B1}P_{B2}} \left( P_{U1}P_{B2} \frac{1}{2t} \sum_{1 \leq |i| \leq t} \lambda_1^t + P_{B1}P_{U2} \frac{1}{2t} \sum_{1 \leq |i| \leq t} \lambda_2^t + P_{U1}P_{U2} \frac{1}{2t} \sum_{1 \leq |i| \leq t} \lambda_1^t \lambda_2^t \right) \quad (15)$$

$$= \frac{1}{1 - P_{B1}P_{B2}} \left( P_{U1}P_{B2} \frac{\lambda_1(1 - \lambda_1^t)}{t(1 - \lambda_1)} + P_{B1}P_{U2} \frac{\lambda_2(1 - \lambda_2^t)}{t(1 - \lambda_2)} + P_{U1}P_{U2} \frac{\lambda_1 \lambda_2 (1 - \lambda_1^t \lambda_2^t)}{t(1 - \lambda_1 \lambda_2)} \right) \quad (16)$$

Note that  $P_{U1}P_{B2} + P_{B1}P_{U2} + P_{U1}P_{U2} = 1 - P_{B1}P_{B2}$ , so  $M(t)$  is a weighted sum of three terms, where each has the same form as the two-state model memory-decay curve expression in Equation (7). Generally speaking, since the long-term change is much slower than the short-term change,  $\lambda_1 \lambda_2 \approx \lambda_2$ , and

the last two terms act as one. So the memory-decay curve is essentially characterized by three parameters,  $\lambda_1$ ,  $\lambda_2$ , and the weight of the first term. We emphasize that it is these multiple terms that allow the multiple time scales of the channel behavior to be captured.

To estimate a set of four parameters needed to characterize the model, we first estimate  $P_B$  directly from the experimental data simply by counting the fraction of time that the channel is blocked. Then, we estimate the remaining three parameters by performing a least-squares fit of the experimentally measured memory-decay curve. The parameters estimated directly are  $\lambda_1$ ,  $\lambda_2$ , and the weight of the first term in Equation (16). From the four parameters including these three and  $P_B$ , all other parameters can be derived.

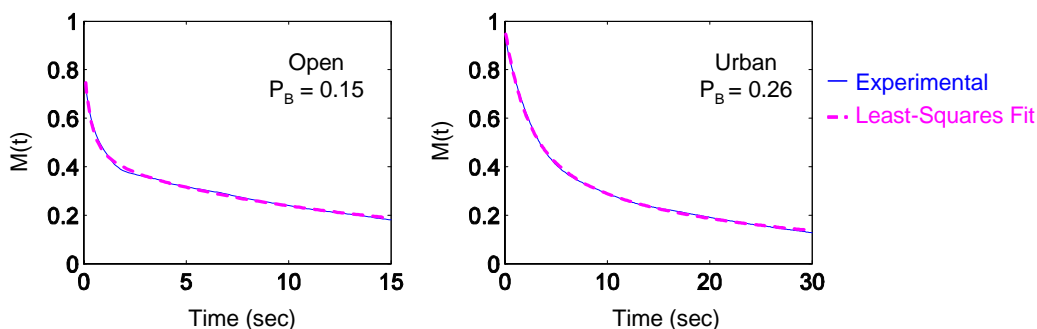


Figure 19. Estimation of the mixture model parameters by least-squares fits of memory-decay curves for both open and urban environments. Excellent fits are achieved.

The fitting of the memory-decay curves is shown in Figure 19. We see that with the mixture model, excellent fits are achieved for both open and urban environments. Compared to the two-state Markov model case shown in Figure 12, there is a significant improvement for the open environment. And although the two-state Markov model has acceptable performance for the urban environment, the mixture model does even better. It is worth noting that since the mixture model can achieve such a good fit, there is no longer a need to make the model any more complicated. Doing so could risk overfitting.

The parameters estimated for both the open and urban environments are tabulated in Table 5. We see that the average long openings are twenty to thirty seconds long, and the average blockage durations are relatively shorter compared to the two-state model case. The average squared errors resulting from the fit are about two to three orders of magnitude smaller. To facilitate implementation, the parameters  $T_{U1}$ ,  $T_{B1}$ ,  $T_{U2}$ , and  $T_{B2}$  are tabulated in Table 6.

**Table 5**  
**Estimated Mixture Model Parameters**

Environment	$T_{\text{long open}}$	$T_{\text{short open}}$	$T_{\text{block}}$	$P_B$	Squared Error
Open	27 sec	0.4 sec	0.4 sec	0.15	0.00003
Urban	22 sec	2.6 sec	2.4 sec	0.26	0.00003

**Table 6**  
**Estimated Mixture Model Parameters for Implementation**

Environment	$T_{U1}$	$T_{B1}$	$T_{U2}$	$T_{B2}$
Open	27 sec	12 sec	0.4 sec	0.4 sec
Urban	22 sec	26 sec	2.9 sec	2.7 sec

### 5.3 ARQ BEHAVIOR COMPARISON

In this section, we validate the mixture model by comparing the ARQ behavior predicted by the model to those resulting from the experimental data. Just as in the two-state model case, we focus on the queue-length distributions and averages. However, due to the increased complexity of the model, we can no longer analytically solve for the queue-length and delay distributions and averages. Instead, numerical simulations are performed. Fortunately, since the model is capable of generating channel-blockage sequences of any lengths, very accurate numerical simulation results can be obtained. In particular,  $10^7$ -sample long channel sequences are used in this study. Compared to the limited amount of experimental data available, this is a major advantage of using a model. As stated in Section 2.2, for all the numerical simulations performed, the discrete-time unit is chosen to be the channel sampling interval, 0.1 second.

The simulation results are shown in Figure 20, which has the same format as Figure 14 in the two-state Markov model case. The rugged, blue, solid curves are from the experimental data, and the more smooth, magenta, dashed ones are associated with the mixture model. These queue-length distribution curves are normalized so that the area under each curve indicates the average queue length. The two plots on the left are for the open environment, and the right side ones are for the urban environment. The top ones have double the arrival rate as the lower ones, so we can test the model at various operation points. We see that for the urban environment, the gaps are even smaller than for the two-state model case, especially for the lower arrival rate one. The curves almost overlap entirely. The mismatch in the tails is really insignificant given the limited amount of data available. For the open environment, the gaps are still significant, but are relatively smaller compared to the two-state model case. Furthermore, the average queue-length errors are reduced to below 20%, which is a considerable improvement over the 70% error in the two-state model case.

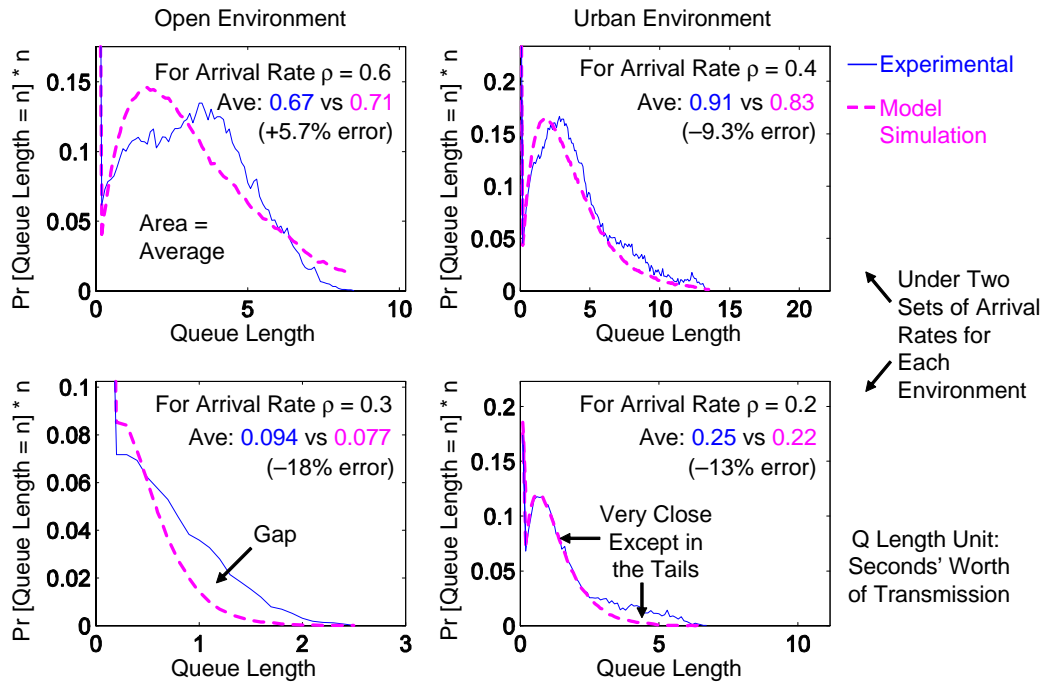


Figure 20. Comparison of the queue-length distributions and averages obtained from the experimental data and the mixture model. The gaps are smaller than for the two-state Markov model case.

Table 7 now summarizes the goodness of the mixture model in terms of how well it predicts the queue-length average and distribution. For both environments, the memory-decay curve fits are excellent. The errors on average queue length are 15% and 20%, respectively. In terms of the match of the queue-length distribution, for the urban environment, the gaps are very small except in the tails. For the open environment, there are still some gaps. Overall, the mixture model works well for both environments, slightly better for the urban environment. Compared to the two-state Markov model, there is a significant improvement. However, we must note that although excellent fits of the memory-decay curves and closer match of the queue-length behavior are achieved by the mixture model, the increased complexity makes it difficult to perform analytical study, so we cannot obtain as much intuition. It is harder to distinguish the effect of the environment and the influence of the design parameters.

Table 7

Summary of Mixture Model Performance

Environment	MDC Fit	Average Queue-Length Error	Queue-Length Distributions	Overall
Urban	Excellent	Within ~ 15%	Small Gaps	Good
Open	Excellent	Within ~ 20%	Some Gaps	Acceptable

## 6. SUMMARY AND FUTURE WORK

We now summarize and reiterate the major contributions of this work. The goal of this study is to build statistical models for COTM blockage channels in different environments, in particular, an open environment and an urban environment. The purpose is to enable accurate prediction of system behavior using symbolic analysis or numerical simulation. We started the model building with the simple and well-known two-state Markov model and proposed a new higher-order mixture model that is capable of capturing channel openings of different time scales. We found that the simple two-state Markov model is better for symbolic analysis and for obtaining intuitions, while the increased complexity of the higher-order mixture model allows it to produce more accurate results via numerical simulations. Comparing to the experimental data collected, the models also provide the capability of statistically generating channel sequences that are much longer, which can then be used in many ways.

In terms of parameter estimation, we identified a deficiency of the opening and blockage durations previously used in [1] and proposed a new concept called a memory-decay curve. We estimated model parameters by performing least-squares fits of the experimentally measured memory-decay curves. For the mixture model, we were able to achieve excellent fits for both the open and urban environments. The curves are essentially indistinguishable. The parameters estimated for the particular open and urban environments studied can be found in Table 5, in terms of a more intuitive description of the environments, and in Table 6, in terms of a more implementation-oriented description. In comparison, the memory-decay curve fits in the two-state Markov model case have larger gaps, especially for the open environment, due to the fewer degrees of freedom of the simpler model. The parameters estimated can be found in Table 2. We re-emphasize that the channel models developed and the parameters estimated from the limited amount of data available are to some extent particular to the environment and the vehicle speed at which the data were collected and the fade margin used. For both models, the parameters may be modified to simulate different environments in other studies.

Once the models were built, we evaluated their validity by using them to predict system behavior, via symbolic analysis or numerical simulation, and comparing the results to those obtained using actual experimental data. The closer the results are, the more valid the models are. In particular, we considered the system of link-layer ARQ and predicted the queue-length distributions and averages. The expectation is that models validated using one scenario should be applicable to other scenarios as well. For the simple two-state Markov model, we were able to derive closed-form expressions for the ARQ queue-length and delay distributions and averages. Each of the averages is expressed as a product of an environment factor and a greedy factor. This decomposition can potentially help system designers to set operating points for different environments depending on queue-length and delay tolerances. The comparison between the queue-length results generated by the experimental data and those predicted by the model shows that the two-state model does not work well for the open environment, where the prediction error of average queue length is up to 70%, but works reasonably well for the urban environment, where the prediction error is within 20%. For the higher-order mixture model, we were unable to derive closed-form solutions due to the increased complexity of the model. Simulation results show that the mixture model works reasonably well for both environments, with prediction errors below 20%. Depending on the desired degree of accuracy, either model may be used. Again, all the numerical results are particular to the scenarios used in this study.

In terms of future work, it would be beneficial to have additional data from different environments to help us improve the fidelity of the models. To perform a loose estimation of the amount of data needed, we consider the hypothetical environments that are modeled exactly by the mixture model with parameters in Table 5. It is found that for the open environment, two hours of data can yield an accuracy of less than 15% error with 88% probability. The time needed becomes eight hours for the urban environment. Since the model is not exact in reality, it is possible that more data are actually needed.

Another component of future work is to improve the understanding of the memory-decay curves, in particular, how it is related to system performance. We saw that for the open environment, although we were able to obtain an excellent fit of the memory-decay curve using the mixture model, the queue-length distribution prediction was still not very accurate. Also, the models we built still have independent opening and blockage durations. It would be interesting to experiment with models that do not have this property. It could further guide the model-building effort.

## APPENDIX A

### PROOF OF MEMORY-DECAY CURVE ON-OFF SYMMETRY

In this appendix, we prove that the two methods of memory-decay curve computation in Equations (1) and (2) in Section 3.2 yield the same answer, i.e., the memory-decay curve computed by conditioning on the current state being one equals the one computed by conditioning on the current state being zero.

**Theorem 1** *For any non-trivial stationary binary random process  $s_i = 0, 1$  with*

$$\Pr [s_i = 1] = P_1 \neq 0 \text{ and } \Pr [s_i = 0] = P_0 = 1 - P_1 \neq 0,$$

*the memory-decay curves*

$$M_1(t) = M_0(t), \tag{A.1}$$

*where*

$$M_1(t) = \frac{1}{P_0} \left( \frac{1}{2t} \sum_{1 \leq |i| \leq t} \Pr [s_i = 1 | s_0 = 1] - P_1 \right)$$

*and*

$$M_0(t) = \frac{1}{P_1} \left( \frac{1}{2t} \sum_{1 \leq |i| \leq t} \Pr [s_i = 0 | s_0 = 0] - P_0 \right).$$

*Proof:*

Let us consider the average fraction of 1's in the  $t$  samples before and after the current sample, but not including it, for any integer  $t, t \geq 1$ ,

$$\mathcal{F} \stackrel{\text{def}}{=} \frac{1}{2t} \sum_{1 \leq |i| \leq t} \Pr [s_i = 1] = \frac{1}{2t} \sum_{1 \leq |i| \leq t} P_1 = P_1. \tag{A.2}$$

This quantity can also be computed by conditioning on the value of the current sample  $s_0$ .

$$\begin{aligned} \mathcal{F} &= P_1 \cdot \left( \frac{1}{2t} \sum_{1 \leq |i| \leq t} \Pr [s_i = 1 | s_0 = 1] \right) + P_0 \cdot \left( \frac{1}{2t} \sum_{1 \leq |i| \leq t} \Pr [s_i = 1 | s_0 = 0] \right) \\ &= P_1 \cdot (M_1(t) \cdot P_0 + P_1) + P_0 \cdot (1 - (M_0(t) \cdot P_1 + P_0)) \\ &= P_0 P_1 (M_1(t) - M_0(t)) + P_1^2 - P_0^2 + P_0 \end{aligned} \tag{A.3}$$

Combining Equations (A.2) and (A.3), we have

$$\begin{aligned} P_0 P_1 (M_1(t) - M_0(t)) &= P_1 - P_1^2 + P_0^2 - P_0 \\ &= P_1(1 - P_1) + P_0(P_0 - 1) \\ &= P_1 P_0 - P_0 P_1 = 0. \end{aligned} \tag{A.4}$$

Since,  $P_0$  and  $P_1$  are both non-zero,

$$P_0 P_1 (M_1(t) - M_0(t)) = 0 \implies M_1(t) = M_0(t). \tag{A.5}$$

■



## APPENDIX B

### DERIVATION OF ARQ BEHAVIOR FOR THE TWO-STATE MARKOV MODEL

In this appendix, we analyze the behavior of the ARQ scheme described in Section 2.2 over a blockage channel modeled by the two-state Markov model and obtain the queue-length and delay averages and distributions in Equations (8)–(11) in Section 4.3. The results are obtained by solving the two-layer semi-infinite Markov chain in Figure 13, which is repeated below. The analysis presented in this appendix is a more detailed version of the immediate-acknowledgment case analysis in [2], with the addition of some continuous time analysis.

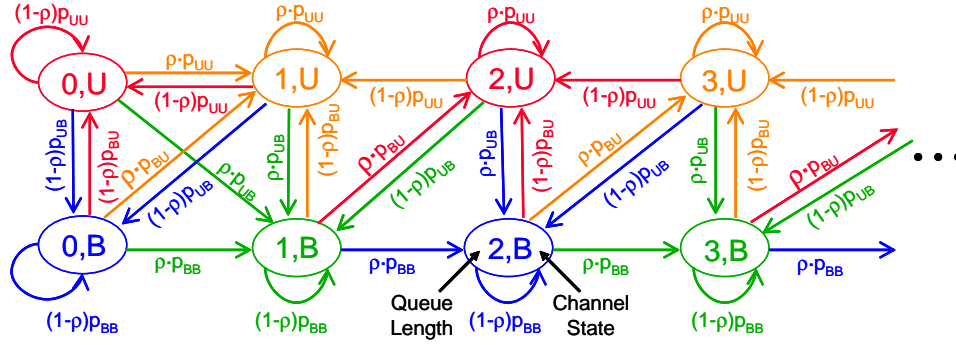


Figure B-1. Repeat of Figure 13: A two-layer semi-infinite Markov chain for analyzing the behavior of ARQ over a blockage channel modeled by the two-state Markov model, assuming perfect and immediate acknowledgments and Bernoulli arrival.

#### B.1 QUEUE-LENGTH RESULTS

Let  $P_{i,U}$  and  $P_{i,B}$  be the steady-state probabilities of the system being in states  $(i, U)$  and  $(i, B)$ , respectively. The quantity  $P_i = P_{i,U} + P_{i,B}$  is the probability that the queue is  $i$  units long independent of the channel state. We now solve for  $P_{i,U}$  and  $P_{i,B}$  explicitly as functions of the arrival rate  $\rho$  and the two-state Markov model transition statistics.

Let the channel state transition matrix be

$$\mathbb{P} \stackrel{\text{def}}{=} \begin{bmatrix} P_{UU} & P_{BU} \\ P_{UB} & P_{BB} \end{bmatrix}, \quad (\text{B.1})$$

where

$$P_{UU} + P_{UB} = P_{BU} + P_{BB} = 1.$$

From the Markov chain in Figure 13, considering the change in queue length first and then the channel-state transition, we have

$$\begin{bmatrix} P_{0,U} \\ P_{0,B} \end{bmatrix} = \mathbb{P} \begin{bmatrix} P_{0,U}(1-\rho) + P_{1,U}(1-\rho) \\ P_{0,B}(1-\rho) \end{bmatrix} \quad (\text{B.2})$$

$$\begin{bmatrix} P_{1,U} \\ P_{1,B} \end{bmatrix} = \mathbb{P} \begin{bmatrix} P_{0,U}\rho + P_{1,U}\rho + P_{2,U}(1-\rho) \\ P_{0,B}\rho + P_{1,B}(1-\rho) \end{bmatrix} \quad (\text{B.3})$$

$$\begin{bmatrix} P_{i,U} \\ P_{i,B} \end{bmatrix} = \mathbb{P} \begin{bmatrix} P_{i,U}\rho + P_{i+1,U}(1-\rho) \\ P_{i-1,B}\rho + P_{i,B}(1-\rho) \end{bmatrix} \quad i \geq 2. \quad (\text{B.4})$$

This family of matrix equations can be solved using the transform method. Generally, the transform of an integer random variable  $x$ , which takes the value  $n$  with probability  $P_x(n)$ , is  $F_x(z) = \sum_{n=-\infty}^{\infty} P_x(n) z^n$ . In this problem, let us define

$$F_U(z) = P_{1,U} + P_{2,U}z + P_{3,U}z^2 + \dots = \sum_{i=1}^{\infty} P_{i,U}z^{i-1}, \quad (\text{B.5})$$

$$F_B(z) = P_{0,B} + P_{1,B}z + P_{2,B}z^2 + \dots = \sum_{i=0}^{\infty} P_{i,B}z^i, \quad (\text{B.6})$$

and

$$R(z) = (1-\rho) + \rho z. \quad (\text{B.7})$$

Multiplying both sides of Equations (B.2), (B.3), and (B.4) with corresponding powers of  $z$  and summing them, we have

$$\begin{bmatrix} P_{0,U} + P_{1,U}z + P_{2,U}z^2 + \dots \\ P_{0,B} + P_{1,B}z + P_{2,B}z^2 + \dots \end{bmatrix} = \mathbb{P} \begin{bmatrix} P_{0,U}((1-\rho) + \rho z) + \sum_{i=1}^{\infty} P_{i,U}((1-\rho)z^{i-1} + \rho z^i) \\ \sum_{i=1}^{\infty} P_{i,B}((1-\rho)z^i + \rho z^{i+1}) \end{bmatrix}. \quad (\text{B.8})$$

Using Equations (B.5), (B.6), and (B.7), we can rewrite the above as

$$\begin{bmatrix} P_{0,U} + z \cdot F_U(z) \\ F_B(z) \end{bmatrix} = \mathbb{P} \begin{bmatrix} P_{0,U}R(z) + F_U(z)R(z) \\ F_B(z)R(z) \end{bmatrix}. \quad (\text{B.9})$$

Separating the known and unknown terms, we have

$$\frac{1}{P_{0,U}} \begin{bmatrix} F_U(z) \\ F_B(z) \end{bmatrix} = \begin{bmatrix} P_{UU}R(z) - z & P_{BU}R(z) \\ P_{UB}R(z) & P_{BB}R(z) - 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 - P_{UU}R(z) \\ -P_{UB}R(z) \end{bmatrix}. \quad (\text{B.10})$$

The right side of Equation (B.10) turns out to be a two-by-one array of first-order rational functions sharing the same denominator, i.e.,  $a+bz/(1-\alpha z) = a+b \sum_{i=1}^{\infty} \alpha^{i-1} z^i$  and  $c+dz/(1-\alpha z) = c+d \sum_{i=1}^{\infty} \alpha^{i-1} z^i$ . By matching terms with  $F_U(z)$  and  $F_B(z)$ , we can express  $P_{i,U}$  and  $P_{i,B}$ ,  $i \geq 1$ , as multiples of  $P_{0,U}$ , i.e.,

$$\begin{aligned} P_{1,U} &= a P_{0,U} & P_{0,B} &= c P_{0,U} \\ P_{i,U} &= \alpha^{i-2} b P_{0,U} \quad i \geq 2 & P_{i,B} &= \alpha^{i-1} d P_{0,U} \quad i \geq 1. \end{aligned} \quad (\text{B.11})$$

Finally,  $P_{0,U}$  can be solved using the normalization condition  $\sum P_{i,U} + \sum P_{i,B} = 1$ . The result is summarized below.

$$\begin{aligned}
P_{0,U} &= P_U - \rho & P_{0,B} &= \frac{P_{UB}}{\eta} P_{0,U} \\
P_{1,U} &= \frac{\rho}{1-\rho} \left(1 + \frac{P_{UB}}{\eta}\right) P_{0,U} & P_{1,B} &= \frac{\rho}{1-\rho} \frac{P_{UB}}{\eta^2} P_{0,U} \\
P_{2,U} &= \left(\frac{\rho}{1-\rho}\right)^2 \frac{P_{UB}}{\eta^2} P_{0,U} \\
P_{i+1,U} &= \alpha P_{i,U}, \quad i \geq 2 & P_{i+1,B} &= \alpha P_{i,B}, \quad i \geq 1
\end{aligned} \tag{B.12}$$

where

$$\eta = P_{BU}(1 - \rho) + P_{UU} \rho \quad \text{and} \quad \alpha = \frac{\rho}{1 - \rho} \frac{1 - \eta}{\eta}.$$

It can be verified that  $\sum P_{i,U} = P_U$  and  $\sum P_{i,B} = P_B$ .

The above result is in discrete time. When the time unit  $\tau$  is small, it is more meaningful to write the result in continuous time. Let the probability density function of the queue length be  $p_Q(t)$  where the unit of queue length is one second's worth of transmission. The relationship between  $p_Q(t)$  and  $P_i = P_{i,U} + P_{i,B}$  is

$$p_Q(t) = \lim_{\tau \rightarrow 0} \frac{P_{t/\tau}}{\tau}. \tag{B.13}$$

Since  $P_i$  decays geometrically after  $i \geq 2$ ,  $p_Q(t)$  decays exponentially after an initial impulse near 0. We will first evaluate the magnitude of the impulse and then derive the exponential tail. Since  $P_{UB} = \tau/T_U \rightarrow 0$  as  $\tau \rightarrow 0$ , the magnitude of the impulse is

$$P_{0,U} + P_{1,U} = (P_U - \rho) \left(1 + \frac{\rho}{1 - \rho}\right) = \frac{P_U - \rho}{1 - \rho}. \tag{B.14}$$

To derive the exponential tail, we note that when  $\tau$  is small,  $t/\tau$  is large, and the decay factor  $\alpha$  is very close to unity, so

$$\begin{aligned}
P_{t/\tau} &= P_{t/\tau,U} + P_{t/\tau,B} \\
&= \alpha^{t/\tau-2} P_{2,U} + \alpha^{t/\tau-1} P_{1,B} \\
&= \alpha^{t/\tau} (P_{2,U} + P_{1,B}).
\end{aligned} \tag{B.15}$$

To express  $\alpha$  in terms of continuous-time parameters, we need to first rewrite  $\eta$  using  $P_{BU} = \tau/T_B$  and  $P_{UB} = \tau/T_U$ , as  $\tau \rightarrow 0$ . We have

$$\eta = \frac{\tau}{T_B}(1 - \rho) + \left(1 - \frac{\tau}{T_U}\right) \rho = \rho + \frac{\tau}{T_B} - \frac{\tau\rho}{T_B} - \frac{\tau\rho}{T_U} = \rho + \frac{\tau}{T_B} \frac{P_U - \rho}{P_U}, \tag{B.16}$$

where we used  $1/T_B + 1/T_U = (T_U + T_B)/(T_U T_B) = 1/(T_B P_U)$ . Let  $\epsilon = \eta - \rho = \tau(P_U - \rho)/(T_B P_U)$ , which diminishes as  $\tau \rightarrow 0$ . We can rewrite  $\alpha$  as

$$\alpha = \frac{1 - \rho - \epsilon}{1 - \rho} \frac{\rho}{\rho + \epsilon} = \left(1 - \frac{\epsilon}{1 - \rho}\right) \left(1 - \frac{\epsilon}{\rho}\right) \approx 1 - \frac{\epsilon}{(1 - \rho)\rho} = 1 - \frac{\tau(P_U - \rho)}{T_B P_U (1 - \rho)\rho}. \tag{B.17}$$

Define  $\alpha'$  using  $\alpha = 1 - \alpha'\tau$ , so  $\alpha' = (P_U - \rho)/(T_B P_U(1 - \rho)\rho)$ , then the first factor in Equation (B.15) becomes

$$\lim_{\tau \rightarrow 0} \alpha^{t/\tau} = \lim_{\tau \rightarrow 0} (1 - \alpha'\tau)^{t/\tau} = e^{-\alpha't}. \quad (\text{B.18})$$

The second factor of Equation (B.15) turns out to be

$$P_{2,U} + P_{1,B} = \left(1 + \frac{\rho}{1 - \rho}\right) \frac{\rho}{1 - \rho} \frac{\tau}{T_U(\rho + \epsilon)^2} (P_U - \rho) = \frac{\tau(P_U - \rho)P_B}{(1 - \rho)^2 \rho T_B P_U} = \frac{\tau P_B}{1 - \rho} \alpha'. \quad (\text{B.19})$$

Combining Equations (B.18) and (B.19), we have

$$\lim_{\tau \rightarrow 0} \frac{P_{t/\tau}}{\tau} = \frac{P_B}{1 - \rho} \alpha' e^{-\alpha't}, \quad \text{for } t > 0, \quad (\text{B.20})$$

which is the exponential tail portion of  $p_Q(t)$ . Combining with the impulse in Equation (B.14), we have

$$p_Q(t) = \frac{P_U - \rho}{1 - \rho} \delta(t) + \frac{P_B}{1 - \rho} \alpha' e^{-t\alpha'}, \quad \text{where } \alpha' = \frac{P_U - \rho}{T_B P_U(1 - \rho)\rho}. \quad (\text{B.21})$$

This completes the derivation for Equation (9) in Section 4.3. We note that  $(P_U - \rho)$  is the difference between the maximum throughput  $P_U$  and the arrival rate  $\rho$ . The steady-state distribution exists only when  $\rho < P_U$ . Otherwise, data arrive faster than they can leave, and the queue would grow indefinitely.

Another important observation is the exponential decay in queue length. Empirical observations show that this is true even when the independent and stationary arrival statistics are not Bernoulli, i.e., more than one arrival can happen at a time. This is most likely because the resulting Markov chain, like the one in Figure 13, starts repeating after a certain point for any independent and stationary arrival statistics with finite support.

From the discrete-time steady-state distribution in Equation (B.12), we can compute the average queue length

$$\mathcal{L} = P_{1,U} + \sum_{i=2}^{\infty} P_{2,U} \alpha^{i-2} i + \sum_{i=1}^{\infty} P_{1,B} \alpha^{i-1} i \quad (\text{B.22})$$

$$= P_{1,U} + P_{2,U} \frac{2 - \alpha}{(1 - \alpha)^2} + P_{1,B} \frac{1}{(1 - \alpha)^2} \quad (\text{B.23})$$

$$= \frac{P_B P_U / P_{BU}}{P_U / \rho - 1} + \rho, \quad (\text{B.24})$$

where the unit is the amount of data that can be transmitted in one time unit  $\tau$ . When  $\tau$  is small, it is more meaningful to express the result in continuous time with the unit of seconds' worth of transmission. When  $\tau \rightarrow 0$ , the  $+\rho$  term is negligible, since  $\rho < 1$ , and one  $\tau$  second's worth of memory is insignificant. We can normalize  $\mathcal{L}$  with the transmission rate per second,  $1/\tau$ , and the queue length as multiples of one second's worth of transmission is

$$\mathcal{L}' \stackrel{\text{def}}{=} \tau \mathcal{L} = T_B P_B P_U \left( \frac{P_U}{\rho} - 1 \right)^{-1}, \quad (\text{B.25})$$

where we used  $\tau/P_{BU} = T_B$  as  $\tau \rightarrow 0$ . This is the derivation for Equation (8).

## B.2 DELAY RESULTS

We now derive the distribution and the average of the delay experienced by each packet or data segment from the time it enters the queue to the moment it leaves the queue after it is acknowledged.<sup>12</sup> Just like in the derivation for the queue-length result, we first derive the results in discrete time and then obtain the continuous time results by taking the limit of  $\tau \rightarrow 0$ .

The basic idea is that, with immediate acknowledgments, the same packet is repeatedly transmitted until the channel opens and then it goes through successfully. So all packets are transmitted in order. (This would not hold exactly if the acknowledgment were not immediate.) When a packet enters the queue during steady state, the queue length and channel state pair has the distribution given in Equation (B.12). If there are  $N$  packets already in the queue, then all  $N$  packets must be delivered before this packet is. Therefore, if the channel is unblocked when a packet arrives, then the delay it experiences is the amount of time it takes for the channel to be open for another  $N$  times, unless  $N = 0$ , in which case, since the packet transmission cannot be scheduled right away, it must wait for the next time the channel is open. On the other hand, if the channel is blocked when a packet arrives, then we must add the initial delay for the channel to open up.

This problem is a classic random sum of random variables problem since the number of packets already in the queue is random and the amount of time between two adjacent channel openings is also random. A standard way to solve this problem is using the transform method: Suppose  $y = x_1 + x_2 + \dots + x_N$ , where  $x_i$  are independent and identically distributed with transform  $F_x(z)$ , and the transform for  $N$  is  $F_N(z)$ , then  $F_y(z) = F_N(F_x(z))$ . We also use the technique that summation of independent random variables corresponds to multiplication of their transforms.

The transform for the number of additional open channels needed when the channel is initially unblocked and for when it is initially blocked are

$$\begin{aligned} F_{N_U}(z) &= (P_{0,U} + P_{1,U})z + P_{2,U}z^2 + P_{3,U}z^3 + \dots \\ &= (P_{0,U} + P_{1,U})z + P_{2,U}z^2/(1 - \alpha z) \end{aligned} \quad (\text{B.26})$$

$$\begin{aligned} F_{N_B}(z) &= P_{0,B} + P_{1,B}z + P_{2,B}z^2 + P_{3,B}z^3 + \dots \\ &= P_{0,B} + P_{1,B}z/(1 - \alpha z). \end{aligned} \quad (\text{B.27})$$

The transform for the time between two adjacent channel openings is

$$\begin{aligned} F_{t_{UU}}(z) &= P_{UU}z + P_{UB}P_{BU}z^2 + P_{UB}P_{BB}P_{BU}z^3 + \dots \\ &= P_{UU}z + P_{UB}P_{BU}z^2/(1 - P_{BB}z). \end{aligned} \quad (\text{B.28})$$

Similarly, the transform for the time from a blocked channel state to the next open one is

$$\begin{aligned} F_{t_{BU}}(z) &= P_{BU}z + P_{BB}P_{BU}z^2 + P_{BB}^2P_{BU}z^3 + \dots \\ &= P_{BU}z/(1 - P_{BB}z). \end{aligned} \quad (\text{B.29})$$

---

<sup>12</sup>Another way to measure delay is from the time a packet enters the queue to the time it is received. Since we are assuming perfect and immediate acknowledgments, this version of delay is longer by exactly half of the round trip time.

Substituting the above expressions for  $F_{N_U}(z)$ ,  $F_{N_B}(z)$ ,  $F_{t_{UU}}(z)$ , and  $F_{t_{BU}}(z)$  into the transform of the delay distribution,

$$F_d(z) = F_{N_U}(F_{t_{UU}}(z)) + F_{N_B}(F_{t_{UU}}(z)) \cdot F_{t_{BU}}(z), \quad (\text{B.30})$$

it can be shown that

$$F_d(z) = Az + Bz^2/(1 - \beta z) \quad (\text{B.31})$$

where

$$A = \frac{P_U - \rho}{1 - \rho}, \quad B = A \cdot \frac{P_{UB}}{1 - \rho}, \quad \text{and} \quad \beta = \frac{1 - \eta}{1 - \rho}.$$

Let  $d_i$  be the probability that the delay is  $i$ , we have

$$\begin{aligned} d_1 &= \frac{P_U - \rho}{1 - \rho}, \\ d_2 &= \frac{P_{UB}}{1 - \rho} d_1, \\ d_{i+1} &= \beta \cdot d_i \quad \text{for } i \geq 2. \end{aligned} \quad (\text{B.32})$$

We note that the minimum delay is one since according to our setup every new packet must wait for at least one time unit to be scheduled for transmission.

When  $\tau$  is small, the continuous-time delay distribution is an impulse followed by an exponential tail, just like the queue-length distribution. We will also perform the derivation in a similar fashion.

Since  $P_{UB} = \tau/T_U \rightarrow 0$  as  $\tau \rightarrow 0$ , the magnitude of the impulse is

$$d_1 = \frac{P_U - \rho}{1 - \rho}. \quad (\text{B.33})$$

When  $\tau$  is small,  $t/\tau$  is large, and  $\beta$  is close to unity, so the exponential tail can be derived from  $d_i$  using

$$p_d(t) = \lim_{\tau \rightarrow 0} \frac{d_{t/\tau}}{\tau} = \lim_{\tau \rightarrow 0} \frac{\beta^{t/\tau-2} d_2}{\tau} = \lim_{\tau \rightarrow 0} \beta^{t/\tau} \frac{d_2}{\tau}, \quad \text{for } t > 0. \quad (\text{B.34})$$

We can first rewrite  $\beta$  using the expression for  $\eta$  in Equation (B.16),

$$\beta = \frac{1 - \eta}{1 - \rho} = 1 + \frac{\tau(P_U - \rho)}{T_B P_U (1 - \rho)}. \quad (\text{B.35})$$

Define  $\beta'$  using  $\beta = 1 - \beta'\tau$ , so  $\beta' = (P_U - \rho)/(T_B P_U (1 - \rho))$ , then the first factor in Equation (B.34) becomes

$$\lim_{\tau \rightarrow 0} \beta^{t/\tau} = \lim_{\tau \rightarrow 0} (1 - \beta'\tau)^{t/\tau} = e^{-\beta' t}. \quad (\text{B.36})$$

Using  $P_{UB} = \tau/T_U$ , the second factor in Equation (B.34) can be rewritten as

$$\frac{d_2}{\tau} = \frac{d_1}{T_U(1 - \rho)} = \frac{(P_U - \rho)P_B}{(1 - \rho)^2 T_B P_U} = \frac{P_B}{1 - \rho} \beta'. \quad (\text{B.37})$$

Combining with the impulse in Equation (B.33), we have

$$p_d(t) = \frac{P_U - \rho}{1 - \rho} \delta(t) + \frac{P_B}{1 - \rho} \beta' e^{-t\beta'}, \quad \text{where } \beta' = \frac{P_U - \rho}{T_B P_U (1 - \rho)}. \quad (\text{B.38})$$

This completes the derivation for Equation (11) in Section 4.3.

From the discrete-time steady-state delay distribution in Equation (B.32), we can compute the average delay,

$$\mathcal{D} = \sum_{i=1}^{\infty} i \cdot d_i = \frac{P_B/P_{BU}}{1 - \rho/P_U} + 1, \quad (\text{B.39})$$

where the unit is multiples of  $\tau$ . When  $\tau$  is small, we can ignore the +1 term. We can then use  $P_{BU} = \tau/T_B$  and express the average delay in seconds as

$$\mathcal{D}' \stackrel{\text{def}}{=} \tau \mathcal{D} = T_B P_B \left(1 - \frac{\rho}{P_U}\right)^{-1}. \quad (\text{B.40})$$

This is the derivation for Equation (10).

The average delay results in Equations (B.39) and (B.40) can also be obtained from the average queue-length results in Equations (B.24) and (B.25) using Little's formula, i.e., average queue length equals average delay times the arrival rate,

$$\mathcal{L} = \mathcal{D} \cdot \rho \quad \text{and} \quad \mathcal{L}' = \mathcal{D}' \cdot \rho. \quad (\text{B.41})$$



## REFERENCES

1. J.B. Schodorf, "EHF satellite communications on the move: experimental results," MIT Lincoln Laboratory Technical Report 1087, August 2003.
2. H. Yao, "Link level ARQ for satellite-mobile communications over blockage channels with memory," to be published in *Proc. Military Communications Conference*, October 31–November 3, 2004.
3. M. Mushkin and I. Bar-David, "Capacity and coding for the Gilbert-Elliott Channels," *IEEE Transactions on Information Theory*, vol. 35, no. 6, pp. 1277–1290, November 1989.
4. J.S. Slack, "Finite state Markov models for error bursts on the ACTS land mobile satellite channel," MSEE Thesis, Brigham Young University, Provo, UT. (Also available as NASA Technical Report ACTS-96-046), 1996.
5. E.N. Gilbert, "Capacity of a burst-noise channel," *The Bell System Technical Journal*, pp. 1253–1265, September 1960.
6. E.O. Elliott, "Estimates of error rates for codes on burst-noise channel," *The Bell System Technical Journal*, vol. 42, pp. 1977–1997, September 1963.
7. R.H. McCullough, "The binary regenerative channel," *The Bell System Technical Journal*, vol. 47, pp. 1713–1735, October 1968.
8. B.D. Fritchman, "A binary channel characterization using partitioned Markov chains," *IEEE Transactions on Information Theory*, vol. 13, pp. 221–227, April 1967.



# REPORT DOCUMENTATION PAGE

*Form Approved*  
*OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

<b>1. REPORT DATE (DD-MM-YYYY)</b> 2 November 2004		<b>2. REPORT TYPE</b> Technical Report		<b>3. DATES COVERED (From - To)</b>	
<b>4. TITLE AND SUBTITLE</b> EHF Satellite Communications-on-the-Move Blockage Channel Modeling				<b>5a. CONTRACT NUMBER</b> F19628-00-C-0002	
				<b>5b. GRANT NUMBER</b>	
				<b>5c. PROGRAM ELEMENT NUMBER</b>	
<b>6. AUTHOR(S)</b> Huan Yao				<b>5d. PROJECT NUMBER</b>	
				<b>5e. TASK NUMBER</b>	
				<b>5f. WORK UNIT NUMBER</b>	
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b>  MIT Lincoln Laboratory 244 Wood Street Lexington, MA 02420-9108				<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>  TR-1098	
<b>9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b> MILSATCOM Joint Program Office SMC/MCX 2420 Vela Way, Suite 1467-A8 Los Angeles, AFB, CA 90245				<b>10. SPONSOR/MONITOR'S ACRONYM(S)</b>	
				<b>11. SPONSOR/MONITOR'S REPORT NUMBER(S)</b>  ESC-TR-2004-079	
<b>12. DISTRIBUTION / AVAILABILITY STATEMENT</b> Approved for public release; distribution is unlimited.					
<b>13. SUPPLEMENTARY NOTES</b>					
<b>14. ABSTRACT</b> In the next-generation Transformational Satellite Communications System, one essential element is to provide satellite communication to on-the-move terminals, which suffer from intermittent channels due to blockages by objects in their environments. Presented is our approach for modeling these blockage channels for different environments, in order to allow accurate prediction of system behavior via numerical simulation or symbolic analysis. In particular, we focus on the scenario of link-layer ARQ (automatic repeat request) and the prediction of the averages and distributions of the queue length and delay experienced by the data packets. We investigate two models. The simple two-state Markov model is amenable to symbolic analysis, but is less accurate, while a more complex mixture model leads to better predictions, but can be studied only via numerical simulation. We also introduce a new memory-decay curve concept for the characterization of channel memory and the estimation of model parameters.					
<b>15. SUBJECT TERMS</b>					
<b>16. SECURITY CLASSIFICATION OF:</b>			<b>17. LIMITATION OF ABSTRACT</b>  Same as report	<b>18. NUMBER OF PAGES</b>  56	<b>19a. NAME OF RESPONSIBLE PERSON</b>
<b>a. REPORT</b> Unclassified	<b>b. ABSTRACT</b> Unclassified	<b>c. THIS PAGE</b> Unclassified			<b>19b. TELEPHONE NUMBER (include area code)</b>