ABSTRACT

Title of dissertation: FLOW CONTROL IN WIRELESS AD-HOC NETWORKS

Georgios Papageorgiou,
Doctor of Philosophy, 2009

Dissertation directed by: Professor John S. Baras,
Department of Electrical and
Computer Engineering

We are interested in maximizing the Transmission Control Protocol (TCP) throughput between two nodes in a single cell wireless ad-hoc network. For this, we follow a cross-layer approach by first developing an analytical model that captures the effect of the wireless channel and the MAC layer to TCP. The analytical model gives the time evolution of the TCP window size which is described by a stochastic differential equation driven by a point process. The point process represents the arrival of acknowledgments sent by the TCP receiver to the sender as part of the self-regulating mechanism of the flow control protocol. Through this point process we achieve a cross-layer integration between the physical layer, the MAC layer and TCP. The intervals between successive points describe how the packet drops at the wireless channel and the delays because of retransmission at the MAC layer affect the window size at the TCP layer. We fully describe the statistical behavior of the point process by computing first the p.d.f. for the inter-arrival intervals and then the compensator and the intensity of the process parametrized by the quantities that describe the MAC layer and the wireless channel.

To achieve analytical tractability we concentrate on the pure (unslotted) Aloha for the MAC layer and the Gilbert-Elliott model for the channel. Although the Aloha protocol is simpler than the more popular IEEE 802.11 protocol, it still exhibits the same exponential backoff mechanism which is a key factor for the performance of TCP in a wireless network. Moreover, another reason to study the Aloha protocol is that the protocol and its variants gain popularity as they are used in many of today's wireless networks.

Using the analytical model for the TCP window size evolution, we try to increase the TCP throughput between two nodes in a single cell network. We want to achieve this by implicitly informing the TCP sender of the network conditions. We impose this additional constraint so we can achieve compatibility between the standard TCP and the optimized version. This allows the operation of both protocol stacks in the same network.

We pose the optimization problem as an optimal stopping problem. For each packet transmitted by the TCP sender to the network, an optimal time instance has to be computed in the absence of an acknowledgment for this packet. This time instance indicates when a timeout has to be declared for the packet. In the absence of an acknowledgment, if the sender waits long for declaring a timeout, the network is underutilized. If the sender declares a timeout soon, it minimizes the transmission rate. Because of the analytical intractability of the optimal stopping time problem, we follow a Markov chain approximation method to solve the problem numerically.

FLOW CONTROL IN WIRELESS AD-HOC NETWORKS

by

Georgios Papageorgiou

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2009

Advisory Committee:
Professor John S. Baras, Chair/Advisor
Professor Armand M. Makowski
Professor Sennur Ulukus
Professor Nuno C. Martins
Professor A. Udaya Shankar, Dean's Representative

DEDICATION

To my late father Aristides, my mother Antonia and
my sister Efrosyni.

ACKNOWLEDGMENTS

I am grateful to my advisor, Prof. John S. Baras, for his continuous support and encouragement during the entire period of my graduate studies at the University of Maryland. His patience and generosity, even during tough times, were limitless. His energy, enthusiasm, persistence, deep mathematical insight and expertise in different and varying topics have always been a constant source of motivation for me. The fact that he fosters a research environment free of constraints, where a student is encouraged to explore different research areas and problems, allowed me to work on various problems that may not be part of this thesis, but introduced me to various mathematical tools and methodologies. This involvement helped me to build a concrete mathematical background and increased research maturity.

I am also grateful to my thesis committee members, Prof. Armand M. Makowski, Prof. Sennur Ulukus, Prof. Nuno C. Martins and Prof. A. Udaya Shankar, for agreeing to serve on my committee. Prof. Makowski and Prof. Ulukus provided me with useful feedback during my Thesis Proposal Examination.

During my graduate studies at the University of Maryland I had the privilege to take classes with some exceptional teachers. I would like to take this opportunity and thank Profs. Gilmer L. Blankenship, P. S. Krishnaprasad, Steve Markus (control theory, optimization), Adrian Papamarcou, Armand M. Makowski, Mark I. Freidlin (probability theory, stochastic processes and information theory). Their dedication to teaching and the extra effort they always made to organize and present the course material in an interesting manner made the task of learning more enjoyable. Having had a background in Computer Science, I could not have wished for better teachers to take upon themselves the task of introducing me to a whole new and exciting world of science. Their deep knowledge, experience, intuition, relaxed style and graciousness towards their students had a profound impact.

The life of a graduate student involves dealing with bureaucratic issues from time to time. I would like to thank Kim Edwards, Althia Kirlew and Diane Hicks for their efficiency in keeping all these administrative details to an absolute minimum. I would also like to thank the staff of both ECE and ISR for always trying to do their best helping students with official matters.

I would also like to thank a number of colleagues that made the long hours in the office or the lab to seem not so bad after all: Pedram Hovareshti, Punyaslok Purkayastha, Ion Matei, Vahid Tabatabaee, Kiran Kumar Somasundaram, Kaustubh Jain, Vladimir Ivanov, Tao Jiang, and Ayan Roy-Chowdhury. I also feel very fortunate to have made some good friends here at Maryland whose support and help I could always count on, including Senni Perumal, Vasilis Botopoulos, Konstantinos Bitsakos, Nikolaos Frangiadakis, Maben Rabi, Yadong Shang, Nick Pavlounis and Christine Demkowych.

Most and above all I would like to thank my family for their never-ending support and unconditional love.

This work was supported by the National Aeronautics and Space Administration under Cooperative Agreements No. NCC8-235 and NAG3-2844, by Communications and Networks Consortium sponsored by the U.S. Army Research Laboratory under the

Collaborative Technology Alliance Program Cooperative Agreement DAAD19-01-2-001, and by the DARPA NMS Project under the Collaborative Agreement N6600100C8063.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

This chapter serves as an introduction to the rest of the thesis, by providing the motivation for the current work. Moreover, it introduces the problems that are addressed and the approach and tools used to solve these problems.

1.1 Motivation

Wireless ad-hoc networks had been used until recently in military applications and also in emergency situations where the nature of the situation does not allow the deployment and usage of a communication infrastructure, e.g. physical destruction, etc. In the recent years however, wireless ad-hoc networks have become quite popular and are widely used for commercial applications. New standards have been introduced (IEEE 802.11, Bluetooth, etc.) that helped to increase the popularity of Wireless Local Area Networks (WLAN's), which is one class of ad-hoc networks. Presently, there is also an increasing interest for sensor networks, not only for military purposes but also for commercial use.

Key features of wireless ad-hoc networks include the lack of infrastructure, the need to operate under energy constraints, the fact that each node in the network operates both as a host and as a router (i.e. a node is responsible to relay information not destined to it to other nodes in the network), mobility and the physical characteristics of the wireless channel. Such features yield the design and implementation of a wireless ad-hoc network, a non trivial task. The study of ad-hoc networks has also unveiled the need to diverge from the paradigm of layering. Since the early days of packet networking, the designers of such networks have been following a "divide-and-conquer" approach in defining and implementing the various functions necessary for the operation of the network. Related operations are grouped together forming a layer and each layer is treated as a black box with a very well and formally-defined interface. Although this approach makes the control and maintenance of networks easier, it fails to reveal the dependencies among quantities that exist in different layers, thus causing suboptimal performance.

In a wireless ad-hoc network it makes sense to consider a cross-layer (i.e. vertical across the protocol stack) integration, especially among the lower layers of the protocol stack: physical, Medium Access Control (MAC), network, transport, since these are the layers that participate the most during a packet exchange both from a hop-by-hop and an end-to-end point of view. Nevertheless, inter-dependencies can be found even between the upper and lower layers, e.g. depending on the specific characteristics of the wireless channel (physical layer) different source coding (application layer) and channel coding (link layer) schemes can be used to adapt to available bandwidth, fading, latency, etc. Another example where the dependency of one layer to another is evident is related to power control. Traditionally, power control was treated as a physical layer characteristic. Most of the schemes that were developed in the context of power control aimed to adjust the transmitting power of a node in order to maximize the signal to noise and interference ratio subject to the channel characteristics (fading, interference, noise). In a wireless ad-hoc network however, the power level at which a node is transmitting defines not only the interference this node is creating to other nodes but also what nodes in the network are immediately accessible (one-hop away). Thus, power control determines the connectivity of the wireless ad-hoc network and in consequence it affects how packets are routed in the network (network layer).

An example which is more relevant to the current work and makes the need for layer coupling even more obvious is the operation of the Transmission Control Protocol (TCP) over a wireless network. The main functionality of TCP is to control the rate of information sent from a host node to the network. At the same time TCP controls the number of outstanding packets, i.e. packets whose receipt has not been acknowledged by the destination. This is done by using a window based mechanism. In the event of a packet loss, TCP assumes congestion exists in the network and abruptly reduces its window size. Although such an assumption is reasonable in the case of the Internet where most of the packet losses are due to buffer overflow, it is not always true for a wireless environment. On wireless links a packet may be lost because of errors at the channel. A simple reaction in this case would had been the retransmission of the lost packet. But TCP cannot distinguish between the two cases and reacts to the packet loss because of errors on the channel as if it were caused by congestion. This results to the under-utilization of the channel since the end host is forced to send packets in the network with a lower rate.

1.2 Problems Addressed and Approach

The problem that is explored in this thesis is that of maximizing the TCP throughput for a connection between two nodes in a wireless ad-hoc network. It is known that TCP is impaired by performance degradation in a wireless network and the main reason for this is the inherent assumption that a packet loss is due to congestion in the network. In a wireless network however, it is possible that packets are dropped because of bad channel quality (e.g. fading). In such a case, the reaction of TCP, which is to reduce the sending rate of data, does not solve the problem it only makes things worse.

Instead of explicitly informing the TCP sender of the nature of a packet loss, our approach to the problem is to inform the sender implicitly. We impose this constraint because we want to keep any changes to the TCP protocol stack to a minimum. By keeping protocol changes small and only local to the TCP sender, we can achieve compatibility between the TCP protocol stack optimized for a wireless node and the standard TCP protocol stack operating on a host in the Internet. This can lead to an easier adoption of the new scheme.

In our approach, the implicit notification the TCP sender receives regarding the condition of the network and the possible cause of a packet drop is achieved through the feedback mechanism of acknowledgments (or the lack of them). In more detail, we assume the TCP sender and receiver are one hop away and thus there can be no congestion in intermediate queues. In addition, we assume the MAC mechanism is that of unslotted Aloha. The choice of unslotted Aloha allows for tractability in the analysis, since our focus is on the dynamics of the system, rather than the average behavior. Moreover, Aloha captures the behavior of the backoff machinery that is present in more popular wireless networks such as IEEE 802.11 systems. Finally, we model the effects of the physical layer using a two-state Markov chain where a packet is lost or not with probability 1 depending on the state of this chain.

Using this model and results from [17,34], we are able to describe the TCP window size evolution using a stochastic differential equation driven by a point process. The driving point process describes the arrival of acknowledgments to the TCP sender and its intensity is given as a function of the parameters that describe the unslotted Aloha and the Markov chain for the physical layer. Thus, a cross-layer (i.e. vertical across layers) integration is achieved through the protocol stack on the wireless node.

Using the stochastic differential equation that describes the evolution of the window size, we try to pose an optimization problem. The objective is to maximize the TCP throughput with constraints that come from the stochastic differential equation. The problem can be posed as an optimal stopping problem where the TCP sender needs to choose the optimal time to declare a timeout event in the absence of a received acknowledgment for a packet that has been sent to the network. This is because at any point in time, the TCP sender can either continue waiting to receive an acknowledgment or since an acknowledgment has not been received, declare a timeout and retransmit the packet. The first choice incurs delays and leaves the connection idle and thus decreases the throughput. In the event of a received acknowledgment, the TCP sender can resume operation from the stopping point without having to minimize its window size and begin from slow-start again. The second choice minimizes the time the connection is kept idle, at the expense of using the connection with a small sending rate, since a timeout event is declared and the TCP sender has to switch to the slow-start mode of operation minimizing at the same time its window size.

Because of analytical intractability for this optimal stopping problem, and motivated by the approximation method of Kushner [28–31] we choose to approximate the original problem with a discrete time version, by appropriately defining a Markov chain and solving a dynamic programming problem on that Markov chain.

1.3 Contributions of the Thesis

Because of its importance and popularity, TCP has been analyzed frequently under different assumptions and for different communication environments (i.e. Internet, wireless, satellite networks) and the related literature is extensive (see for example [3,4,9–12, 18–20,23,26,27,32,33,37,39,44–47,55].

The vast majority of the literature, concentrates on the average behavior of TCP, generally ignoring in most of the cases the dynamic evolution of quantities such as the window size. Even when the dynamics of TCP are studied, the round-trip time of packets are typically constant and are not associated with the layers below TCP such as the physical layer, the MAC or the routing. In Chapter 2 of this thesis we develop a simple stochastic differential equation that describes the evolution of the window size. The equation is driven by a point process that represents the sequence of acknowledgments arriving to the TCP sender. The statistical properties of this point process are described in terms of quantities that relate to the MAC layer and the channel.

Previous work on the analysis of TCP concentrates on the congestion avoidance part of the protocol ignoring the slow-start phase. Typically, this is justified by the fact that in the Internet, most of the packet losses happen because of buffer overflow and these losses are detected at the TCP sender by the duplicate acknowledgment mechanism. When this happens, the TCP sender starts operating in the congestion avoidance phase.

But, if TCP operates over a wireless network, packet losses may also occur due to poor channel quality. If a burst of packets is lost the duplicate acknowledgment mechanism does not work and thus a timeout is declared. In such a case the sender switches to the slow-start phase. Thus, it is important to analyze the timeout mechanism of TCP and try to associate its performance with the underlying wireless medium.

Building on the work of Chapter 2 we develop a framework that allows the designer of a wireless network to properly tune and optimize the timeout mechanism of TCP in order to increase the throughput of the network. This approach is presented in Chapter 3 of the thesis.

Chapter 2

Cross-layer Integration between TCP and Aloha

In this chapter the interaction between the Additive-Increase, Multiplicative-Decrease (AIMD) algorithm of the Transmission Control Protocol (TCP) and the random access channel is investigated. In particular, we examine the effect of the Medium Access Control (MAC) and the physical layer on the window size evolution of TCP. The problem of coupling the window size evolution of TCP with a random access channel is addressed using point processes.

2.1 Introduction

TCP is very popular in wired networks and is also used in the first generation of many wireless networks. Thus, it is important to investigate its performance over a wireless communication environment. In a wireless environment the characteristics of communication are quite different compared to those of a wired environment, yielding the existing layering approach of protocols inefficient. It is believed that a closer interconnection between various layers in the protocol stack of a mobile node would allow for a better utilization of the wireless network. One aspect of this cross-layer integration should include the flow-control, which is an end-to-end function, and access control of the shared wireless channel, an operation which is local to each mobile node. The integration of flow-control with Medium Access Control would prevent a data source in a mobile node from overloading the network and hence, decrease its overall performance.

In this chapter, we consider the window-based mechanism of flow-control of TCP on top of an Aloha-based MAC protocol. All nodes in the network are in hearing distance from each other, thus the hidden terminal problem does not exist. The characteristics of the physical channel are captured through the use of a simple two-state Markov process. Aloha is chosen as the MAC protocol since it provides a very simple channel access mechanism and also because it captures in a simple way the random waiting time before retransmission present in many other random access protocols. Moreover, various forms of the Aloha protocol are in use today in most of the current digital cellular networks, increasing the interest for this protocol [5–8, 35, 38]. In this chapter we attempt to develop a simple, yet complete TCP model for a wireless communications environment that captures the behavior of TCP in such an environment. To this end, we consider a single persistent TCP connection over one wireless hop, i.e. the TCP sender and the TCP receiver are one hop away. Because of this, no buffering is performed in any intermediate node, and thus, the round-trip time (RTT) consists mainly of the delay incurred by the MAC in its effort to successfully transmit the packet. Since we are interested in examining the effect of timeouts due to MAC and the physical layer, we assume the forward channel (i.e. the channel from the TCP sender to the receiver) to be ideal, in the sense that there are no packet losses. Thus, there are no duplicate acknowledgments (ACKs) received at the TCP sender. The same situation, i.e. detection of packet losses through timeouts rather than duplicate acknowledgments, arises in the case where the bandwidth-delay product is small [4]. Timeout events are produced because of ACK losses in the backward channel (from the TCP receiver to the TCP sender). In the backward channel the MAC layer introduces delays and thus increases the RTTs, while the physical layer is responsible for ACK losses.

Due to the great importance of TCP in the Internet, various models of it have been developed. These models try to capture the operation of the main mechanisms of TCP and to give insights on how these mechanisms can be improved in order to fine-tune TCP under various networking environments.

Low, Paganini and Doyle [36] study TCP from a control theory point of view in the case of the Internet. They interpret TCP as an optimal controller optimizing specific utility functions at equilibrium and also look at the dynamics of TCP employing linear models to exhibit stability limitations. Their analytical model incorporates general dynamic models for rate control at the source and for pricing at each link. The main assumption is that each source has access to the aggregate price (i.e. congestion indication) of all links in its route. Under the assumption of constant round-trip time and not considering at all the slow-start phase of TCP or timeouts, they are able to provide the utility functions that TCP-Reno and TCP-Vegas are implicitly using. Although this was one of the first works to show that TCP is maximizing certain utility functions, its applicability to the wireless ad-hoc networks is questionable. The main reason for this is the absence of the notion of links in such networks, and thus the interpretation of them as price producers.

In [40] Mascolo et al. propose sender-side modifications to TCP to improve the throughput especially over wireless links. They call the new flavor TCP Westwood (TCPW). The key point in their work is the introduction of bandwidth estimation at the sender and appropriate adjustments for the window size and the slow-start threshold after a packet loss is detected, either by a duplicate acknowledgment or a timeout. They aim for end-to-end extensions of TCP rather than “localized” solutions where packet inspection and/or interception is required at specific points in the network (e.g. base stations). An issue not addressed in [40] but investigated in the current work, relates to cases where the bottleneck link is in the reverse direction from the receiver to the sender. In such cases, the bottleneck link needs to be shared in a fair manner between data packets and acknowledgment packets.

Bansal et al. [11] study the performance of TCP in a multi-hop, wireless network such that of IEEE 802.11. They focus simultaneously on two performance metrics: energy efficiency and TCP session throughput. The motivation to explore these two metrics comes from the fact that they are both important to the operation of a wireless network but also that behave differently with respect to the transmission range, and hence the number of hops traversed in a communication path. A short transmission range increases the number of hops in a communication path, but at the same time decreases the overall energy consumption for the end-to-end communication. Thus, minimal energy consumption favors a short transmission range. On the other hand, it is well known [41,47] that the TCP throughput of a session is inversely proportional to the round-trip time (i.e. the number of hops traversed by the TCP session) and the square root of the packet loss rate. This means that a trade off needs to take place in order to achieve good performance for both these metrics. In [11] it is shown that a smaller transmission range is beneficial from an energy perspective up to a point, but it comes with a cost associated with the TCP session goodput.

Abouzeid, Roy and Azizoglou [1] also investigate the performance of TCP over a wireless link. In their work phenomena relevant to fading are considered as well as queuing. More specifically, they consider a TCP connection where one end is a server in the wired network and the other end resides on a wireless node. The communication path includes one wireless hop and a series of wired links. In their analysis they consider a two-state continuous time Markov chain that models the wireless link behavior and provides for the packet losses because of fading on the channel. Along with packet losses because of the wireless link, they include in their analysis packet losses associated with queuing. Another novelty of their work is the consideration of variable round-trip times and of timeouts, an issue which is almost always neglected in similar studies of TCP. On the other hand, they do not include the effect of the MAC layer to the performance of TCP. Their observations include the fact that timeouts (that are frequent in a wireless communication environment) may cause underutilization of TCP for periods of time much longer than the time the channel is in a "bad" state because of fading. They suggest that a modification to the exponential backoff algorithm and sampling the channel can improve performance, but they do not provide any guidelines to this direction. Based on their model, they point out that it is always recommended to use a higher packet size for a given fading rate, unlike what some earlier work has suggested.

An extensive amount of work has also been performed regarding the analysis and modeling of MAC and especially IEEE 802.11. In his seminal work Bianchi [16] considers saturated users with ideal (no channel losses) and homogeneous (equal physical data rate) channel conditions with no hidden terminals. He develops a model for the analysis of the DCF scheme in both basic packet transmission and RTS/CTS transmission mechanisms. Using this model he provides a throughput performance evaluation of the DCF scheme. In [52], Sharma et al. consider the same problem but they use a more general approach. They compute again the throughput of the network and provide a justification for the assumptions used in [16].

Tobagi et al. [25, 43] develop a model where blocking and interference are taken into account. In [43] they compute throughput of one hop connections and every node can transmit to a single node. In [25] the model is extended for the case of one single path in the network and is explained that the same methodology can by used to model multiple paths as long as there is no common node between the paths.

In [13, 14], Baras et al., develop and evaluate a new method for estimating and optimizing various performance metrics for multi-hop wireless networks, including MANETs. They introduce an approximate (throughput) loss model that couples the physical, MAC and routing layers effects. The model provides quantitative statistical relations between the loss parameters that are used to characterize multiuser interference and physical path conditions on the one hand and the traffic rates between origin-destination pairs on the other. The model takes into account effects of the hidden nodes, scheduling algorithms, IEEE 802.11 MAC and PHY layer transmission failures and finite packet transmission retries at the MAC layer in arbitrary network topologies where multiple paths share nodes. They apply Automatic Differentiation (AD) to these implicit performance models, and develop a methodology for sensitivity analysis, parameter optimization and trade-off analysis for key wireless protocols.

In [18, 19], Bruno et al. consider the interplay between the TCP dynamics and the IEEE 802.11 access method. They present a simple closed-form expression of the per-connection TCP throughput as a function of the average duration of collisions, the average backoff period and the TCP packet size.

The research efforts mentioned above compute averages for the various quantities of interest. Our approach differs from them in that we try to directly analyze and model the dynamics of TCP and more specifically the window size evolution parametrized by quantities relevant to the MAC layer and the wireless channel. Because of the complexities of IEEE 802.11 considering for the MAC layer yield the analysis intractable. Instead, we choose to consider the Aloha protocol which provides for a more tractable MAC model but which at the same time exhibits the exponential backoff periods that is also part of IEEE 802.11.

2.2 Model Description

The flow-control mechanism of TCP is modeled according to the AIMD paradigm. Two quantities are defined, the TCP window size W and the slow-start threshold S. The window size W in the sender changes dynamically based on the reception of acknowledgments (ACKs). During normal operation of the protocol, where no packet losses occur, the arrival of an acknowledgment causes the increase of the window size. An acknowledgment is sent from the receiver to the sender every time a TCP packet is correctly received by the receiver. An acknowledgment is cumulative in the sense that by sending an acknowledgment packet, the receiver acknowledges that all the packets sent by the sender with sequence number less than the number indicated in the ACK packet were correctly received. In the case where a packet from a series of packets is lost, the receiver produces acknowledgments for every packet that is correctly received after the lost packet. Each of these ACKs acknowledges that packets before the lost one are correctly received but

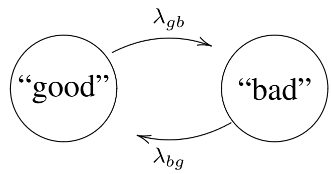

Figure 2.1: The Gilbert-Elliott model.

there is still one packet missing.

If the current window size is less than the slow-start threshold, then TCP is in the slow-start phase and $W_{new} = W_{cur} + 1$ each time an acknowledgment is received, otherwise TCP is in the congestion avoidance phase and $W_{new} = W_{cur} + 1$ per round-trip time. The latter suggests that the increase in the window size is according to $W_{new} = W_{cur} + 1/W_{cur}$ for every received acknowledgment. TCP assumes a packet has been lost either by receiving three duplicate ACKs for a specific packet, or by a timeout. In both cases it is assumed by TCP that the packet was lost because of congestion in the network. In the first case though, since the network is able to deliver packets to the final destination the congestion cannot be severe, so TCP reacts less aggressively. Particularly, the slow-start threshold is updated by taking half the value of the current window size, $S_{new} = W_{cur}/2$, and $W_{new} = S_{new}$. Thus, after the arrival of three duplicate acknowledgments, TCP enters the congestion avoidance phase. The second case implies a severe congestion in the network, so TCP reacts aggressively by setting $S_{new} = W_{cur}/2$, and $W_{new} = 1$. This reaction causes TCP to enter the slow-start phase.

The packet loss model associated with the wireless channel can be described by a continuous time Markov chain with two states as shown in Fig. 2.1 and known as the Gilbert-Elliott model [21,22] which is frequently used in the literature [1,24,42,56]. One state corresponds to the channel being "good", i.e. packets are not lost w.p. 1, and the other state corresponds to the channel being "bad" having as an effect packet losses w.p. 1. The transition rate from the "bad" to the "good" state is λ_{bg} and the transition rate from the "good" to the "bad" state is λ_{gb}.

The MAC layer model assumed in this work is based on the pure Aloha mechanism [2, 15, 51]. In pure Aloha, if a packet transmission overlaps at all with that of another packet, then the transmission is unsuccessful for all the packets that participated in the collision. A packet that failed to be transmitted successfully is retransmitted after a random delay. It is assumed that this delay is a random variable following the exponential distribution with mean $1/\lambda_{ret}$ and it is independent of any previous delays. It is also assumed that a feedback is immediately available to the nodes informing them about the successful or unsuccessful transmission of a packet.

If the new arrivals to the system are Poisson with rate λ, and the number of nodes that have packets that participated in a collision (backlog) is n, then the initiation times of attempted transmissions follow a time-varying Poisson distribution with rate $G(n) = \lambda + n\lambda_{ret}$. In this setting the probability of a successful transmission of a packet is,

$$p_{mac} = e^{-2G(n)T_p} \tag{2.1}$$

where $T_p = L/C$ is the transmission time of the packet of constant length L bits over the wireless channel of capacity C bps. We assume that each node is capable of estimating the number n of nodes with which it competes for bandwidth and therefore can estimate the probability of successful transmission p_{mac}.

The problem attempted to be solved then is to increase the TCP performance based on the models described above. Usually the approach taken is to change the way TCP

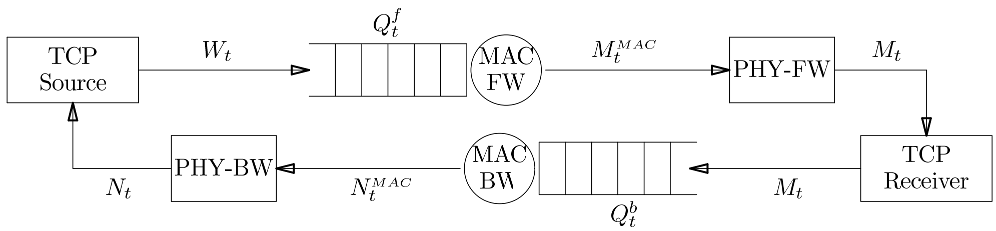

Figure 2.2: System model.

operates by either changing the finite state machine of the protocol or introducing new fields in the TCP header. In contrast to that, our approach keeps to a minimum any required changes to TCP, so that compatibility with the standard TCP implementations can be achieved.

The poor performance of TCP over wireless stems from the fact that TCP cannot distinguish between packet losses due to congestion and packet losses due to poor channel quality. As a result treats all packet losses as if they happen because of the congestion in the network. The TCP sender sets a timeout period for each packet sent to the network. If no acknowledgment for this particular packet is received during this period, the packet is considered lost. Each packet is delayed in the MAC layer because of collisions and probably is lost in the physical layer because of the poor channel quality.

In this chapter we attempt to develop a simple model for the window size evolution of TCP which incorporates the dependencies from the MAC and the physical layers.

2.3 Analysis

In this section we give the mathematical models that describe the operation of each of the layers involved in our analysis. In the following we assume there exists a complete probability space $(\Omega, \mathcal{F}, P)$. A representation of the system is given in Fig. 2.2.

2.3.1 Physical Layer

We elaborate on the model for the Physical layer that was briefly introduced in Section 2.2. We define the continuous time Markov chain $P = (P_t)_{t \geq 0}$ with a state space $\mathcal{P} = \{0, 1\}$. When $P_t = 0$, it means the channel is in the "bad" state and the transmitted packet is dropped w.p. 1, when $P_t = 1$, it means the channel is in the "good" state and the transmission of the packet is successful w.p. 1. As was mentioned in Section 2.2, the transition rates from "bad" to "good" and from "good" to "bad" are λ_{bg} and λ_{gb}, respectively. Then, the transition probabilities for the chain in a small time interval $h > 0$ can be given by:

$$p_{00}(h) = 1 - \lambda_{bg}h + o(h), \qquad p_{01}(h) = \lambda_{bg}h + o(h)$$

$$p_{10}(h) = \lambda_{gb}h + o(h), \qquad p_{11}(h) = 1 - \lambda_{gb}h + o(h)$$

where $o(h)$ is such that $\lim_{h \to 0} \frac{o(h)}{h} = 0$.

In Appendix A.1 we compute the probability $p_0(t)$ the chain is in the "bad" state at time t to be:

$$p_0(t) = \frac{\lambda_{gb}}{\lambda_{bg} + \lambda_{gb}} + \left(p_0(0) - \frac{\lambda_{gb}}{\lambda_{bg} + \lambda_{gb}}\right) e^{-(\lambda_{bg} + \lambda_{gb})t} \tag{2.2}$$

for $t \geq 0$ and some initial probabilities $p_0(0)$ and $p_1(0)$ for the chain to be in the "bad" and the "good" state respectively, at time $t = 0$.

Since $p_0(t) + p_1(t) = 1$ for all $t \geq 0$, we also have:

$$p_1(t) = \frac{\lambda_{bg}}{\lambda_{bg} + \lambda_{gb}} + \left(p_1(0) - \frac{\lambda_{bg}}{\lambda_{bg} + \lambda_{gb}}\right) e^{-(\lambda_{bg} + \lambda_{gb})t} \tag{2.3}$$

for $t \geq 0$. The stationary distribution for the Markov chain corresponds to the case where $t \uparrow \infty$ in (2.2) and (2.3). The stationary probabilities π_b, π_g of being in the "bad" and the "good" states respectively, are:

$$\pi_b = \lim_{t \to \infty} p_0(t) = \frac{\lambda_{gb}}{\lambda_{bg} + \lambda_{gb}} \tag{2.4a}$$

$$\pi_g = \lim_{t \to \infty} p_1(t) = \frac{\lambda_{bg}}{\lambda_{bg} + \lambda_{gb}} \tag{2.4b}$$

In Appendix A.1 we also prove that the waiting time $T_i, i \in \mathcal{P}$ in each state is exponentially distributed and more specifically,

$$P\{T_0 \geq t\} = e^{-\lambda_{bg} t}, \quad t \geq 0 \tag{2.5a}$$

$$P\{T_1 \geq t\} = e^{-\lambda_{gb} t}, \quad t \geq 0 \tag{2.5b}$$

As was mentioned in Section 2.1 we focus on the effect of the MAC and the physical layer on the timeout mechanism of TCP. Thus, we assume the physical channel (PHY-FW) in the forward direction to be ideal. This means that M_t and M_t^{MAC} are indistinguishable and there are no duplicate ACKs produced at the TCP receiver. Assuming that the Markov chain that describes the physical layer operates in the stationary regime, then the effect of the physical layer (PHY-BW) in the backward direction is described by (2.4) and (2.5). In particular, the process $N = (N_t)_{t \geq 0}$ in Fig. 2.2 is a thinned version of the point process $N^{MAC} = (N_t^{MAC})_{t \geq 0}$ and this thinning is done with the stationary probability π_g the channel is in the good state, given by (2.4b).

2.3.2 MAC Layer

In this section we give a more detailed description of the MAC layer model that we use in our analysis. In this chapter we consider the pure Aloha protocol. Each packet i is successfully transmitted (i.e. without any collisions at the MAC layer) with probability

p_{mac} that is given by (2.1) and the transmission time in this case will be constant and equal to $T_p = L/C$ as was mentioned in Section 2.2. A collision happens with probability $1 - p_{mac}$ and the packet has to wait a random time that is exponentially distributed with mean $1/\lambda_{ret}$. At the end of this time period another transmission is attempted. If there is another collision the packet has to wait again for some time which is exponentially distributed with mean $1/\lambda_{ret}$ and is independent of any previous waiting periods. Since $p_{mac} > 0$, the packet will eventually be transmitted successfully.

Since each packet transmission happens independently of any transmissions of previous packets, if we define D_i^{MAC} to be the service time (the time from the moment the packet goes to the head of the queue until it is successfully transmitted) of packet i in the MAC layer, then the random variables $\{D_i^{MAC},\ i = 1, 2, \dots\}$ form an i.i.d. sequence represented by the generic random variable D^{MAC}. We know that it is always true $D^{MAC} \geq T_p$. In particular,

$$D^{MAC} = T_p + \sum_{j=1}^{K} X_j$$

where $\{X_j,\ j = 1, 2, \dots, K\}$ are i.i.d. exponentially distributed random variables with mean $1/\lambda_{ret}$, and K is a geometrically distributed random variable with parameter p_{mac}, such that

$$P\{K = k\} = p_{mac}(1 - p_{mac})^k,\ k = 0, 1, 2, \dots$$

In Appendix A.2 we show that the characteristic function of the random variable D^{MAC} is given by

$$\mathrm{E}\Big[e^{isD^{MAC}}\Big] = p_{mac}\, e^{isT_p} + (1 - p_{mac})\frac{\lambda_{ret}p_{mac}}{\lambda_{ret}p_{mac} - is}\, e^{isT_p} \tag{2.6}$$

From (2.6) we immediately get the p.d.f. of the random variable D^{MAC} to be

$$f_{D^{MAC}}(t) = p_{mac}\delta(t - T_p) + p_{mac}(1 - p_{mac})\lambda_{ret}e^{-\lambda_{ret}p_{mac}\cdot\left(t - T_p\right)}u_0(t - T_p) \qquad (2.7)$$

for $t \geq 0$, where $u_0(\cdot)$ is the Heaviside function:

$$u_0(t) = \begin{cases} 0, & t \leq 0 \\ 1, & t > 0 \end{cases}$$

The times between successful packet transmissions at the MAC layer (assuming there are always packets to be transmitted) are independent and distributed according to (2.7), forming a renewal process. If we define the corresponding point process to be $\{T_n^{MAC},\ n = 0, 1, \ldots\}$, with $T_0^{MAC} = 0$ P-a.s., then

$$T_n^{MAC} = D_1^{MAC} + \cdots + D_n^{MAC} \qquad (2.8)$$

where $D_1^{MAC}, \ldots, D_n^{MAC}$ are i.i.d. random variables with p.d.f. given by (2.7) and

$$D_i^{MAC} = T_p + \sum_{j=1}^{K_i} X_j, \quad i = 1, 2, \ldots, n$$

where K_i is geometrically distributed with parameter p_{mac} and X_j is exponentially distributed with parameter λ_{ret}. Then,

$$\begin{aligned} T_n^{MAC} &= nT_p + \sum_{j=1}^{K_1} X_j + \cdots + \sum_{j=1}^{K_n} X_j \\ &= nT_p + \sum_{j=1}^{K} X_j \end{aligned} \qquad (2.9)$$

where K is the sum of n i.i.d. geometrically distributed random variables with parameter p_{mac}. It is shown in Appendix A.3 that the random variable K has a negative binomial distribution with parameters n and p_{mac},

$$P\{K = k\} = \binom{n + k - 1}{k} p_{mac}^n (1 - p_{mac})^k$$

for $k = 0, 1, \ldots$. Regarding $\sum_{j=1}^{K} X_j$:

$$\mathrm{E}\Big[e^{is\sum_{j=1}^{K} X_j}\Big] = \sum_{k=0}^{+\infty} \big(\mathrm{E}\big[e^{isX_1}\big]\big)^k \binom{n+k-1}{k} p_{mac}^n (1-p_{mac})^k$$
$$= \left(\frac{\lambda_{ret}p_{mac} - ip_{mac}s}{\lambda_{ret}p_{mac} - is}\right)^n$$
$$= p_{mac}^n \sum_{k=0}^{n} \binom{n}{k} \left(\frac{1-p_{mac}}{p_{mac}}\right)^k \left(\frac{\lambda_{ret}p_{mac}}{\lambda_{ret}p_{mac} - is}\right)^k \tag{2.10}$$

which means that

$$f_{\sum_{j=1}^{K} X_j}(t) = p_{mac}^n \delta(t) + p_{mac}^n \sum_{k=1}^{n} \binom{n}{k} \left(\frac{1-p_{mac}}{p_{mac}}\right)^k \frac{(\lambda_{ret}p_{mac})^k}{(k-1)!} t^{k-1} e^{-\lambda_{ret}p_{mac}t} \tag{2.11}$$

for $t \geq 0$. From (2.9) and (2.11), the p.d.f. for T_n^{MAC} is computed

$$f_{T_n^{MAC}}(t) = f_{\sum_{j=1}^{K} X_j}(t - nT_p)$$
$$= p_{mac}^n \delta(t - nT_p) + p_{mac}^n \sum_{k=1}^{n} \binom{n}{k} \left(\frac{1-p_{mac}}{p_{mac}}\right)^k \frac{(\lambda_{ret}p_{mac})^k}{(k-1)!} (t - nT_p)^{k-1} e^{-\lambda_{ret}p_{mac}(t-nT_p)} \tag{2.12}$$

for $t \geq nT_p$.

We define the counting process $N^{MAC} = (N_t^{MAC})_{t\geq 0}$ that corresponds to the point process $\{T_n^{MAC},\ n = 0, 1, \ldots\}$,

$$N_t^{MAC} = \sum_{i=1}^{\infty} \mathbf{1}[T_n^{MAC} \leq t]\,,\ t \geq 0 \tag{2.13}$$

where $\mathbf{1}[\cdot]$ is the indicator function. Although the sequence $\{T_{n+1}^{MAC} - T_n^{MAC},\ n = 0, 1, \ldots\}$ is an i.i.d. sequence defining a renewal process, the counting process N^{MAC} does not have stationary and independent increments because of the fact that $T_p > 0$. We define the history of the N^{MAC} process as the right continuous filtration $\mathcal{F}^{MAC} = (\mathcal{F}_t^{N^{MAC}})_{t\geq 0}$, such that,

$$\mathcal{F}_t^{N^{MAC}} = \sigma\{N_s^{MAC},\ s \leq t\} = \sigma\{T_{N_s^{MAC}}^{MAC},\ s \leq t\}$$

To compute the $\mathcal{F}_t^{N^{MAC}}$−compensator $\mathcal{N}_t$ of the N^{MAC} process, we define the conditional distribution functions:

$$F_1(t) = P\{T_1^{MAC} \leq t\}$$

$$F_i(t) = P\{T_i^{MAC} \leq t \mid T_{i-1}^{MAC}, \ldots, T_1^{MAC}\}, \quad i \geq 2$$

From (2.8) we have that

$$T_1^{MAC} = D_1^{MAC}, \quad P\text{-a.s.}$$

and using (2.7) we compute the conditional distribution $F_1(\cdot)$ and the corresponding p.d.f. $f_1(\cdot)$:

$$f_1(t) = p_{mac}\delta(t - T_p) + p_{mac}(1 - p_{mac})\lambda_{ret} e^{-\lambda_{ret} p_{mac} \cdot \left(t - T_p\right)} u_0(t - T_p)$$

and

$$F_1(t) = \begin{cases} 0, & t < T_p \\ p_{mac} + (1 - p_{mac})\Big(1 - e^{-\lambda_{ret} p_{mac}\left(t - T_p\right)}\Big), & t \geq T_p \end{cases}$$

From (2.8) we notice that

$$T_i^{MAC} = T_{i-1}^{MAC} + D_i^{MAC}, \quad i \geq 2$$

thus,

$$F_i(t) = P\{D_i^{MAC} \leq t - T_{i-1}^{MAC} \mid T_{i-1}^{MAC}\}, \quad i \geq 2$$

Using (2.7) we get the conditional distribution $F_i(\cdot)$ and the corresponding p.d.f. $f_i(\cdot)$, $i \geq 2$:

$$f_i(t) = p_{mac}\delta(t - T_{i-1}^{MAC} - T_p) + p_{mac}(1 - p_{mac})\lambda_{ret}e^{-\lambda_{ret}p_{mac}\cdot\left(t - T_{i-1}^{MAC} - T_p\right)}u_0(t - T_{i-1}^{MAC} - T_p)$$

and

$$F_i(t) = \begin{cases} 0, & t < T_{i-1}^{MAC} + T_p \\ p_{mac} + (1 - p_{mac})e^{-\lambda_{ret}p_{mac}\left(t - T_{i-1}^{MAC} - T_p\right)}\Big), & t \geq T_{i-1}^{MAC} + T_p \end{cases}$$

We proceed by defining

$$\Lambda_t^{(i)} = \int_0^{t \wedge T_i^{MAC}} \frac{dF_i(u)}{1 - F_i(u^-)} = \int_0^{t \wedge T_i^{MAC}} \frac{f_i(u)}{1 - F_i(u^-)}du, \quad i \geq 1$$

Then,

$$\Lambda_t^{(1)} = \begin{cases} 0, & 0 \leq t < T_p \\ \lambda_{ret}p_{mac}\big(t \wedge T_1 - T_p\big), & T_p \leq t \end{cases}$$

and

$$\Lambda_t^{(i)} = \begin{cases} 0, & 0 \leq t < T_{i-1}^{MAC} + T_p \\ \lambda_{ret}p_{mac}\big(t \wedge T_i - T_{i-1}^{MAC} - T_p\big), & T_{i-1}^{MAC} + T_p \leq t \end{cases}$$

for $i \geq 2$. From [17, T7 Theorem, p.61] and [34, Theorem 18.2, p.270] the $\mathcal{F}_t^{N^{MAC}}$−compensator Λ_t of the N^{MAC} process is given by

$$\Lambda_t^{MAC} = \sum_{i \geq 1} \Lambda_t^{(i)}$$

and using $\Lambda_t^{(i)},\ i \geq 1$ computed above, we have

$$\Lambda_t^{MAC} = \begin{cases} \lambda_{ret} p_{mac} \big(T_i^{MAC} - iT_p \big), & T_i^{MAC} \leq t < T_i^{MAC} + T_p \\ \\ \lambda_{ret} p_{mac} t - (i+1) \lambda_{ret} p_{mac} T_p, & T_i^{MAC} + T_p \leq t < T_{i+1}^{MAC} \end{cases} \tag{2.14}$$

The $\mathcal{F}_t^{N^{MAC}}-$intensity λ_t^{MAC} can be computed directly from (2.14) to be:

$$\lambda_t^{MAC} = \begin{cases} 0, & T_i^{MAC} \leq t < T_i^{MAC} + T_p \\ \\ \lambda_{ret} p_{mac}, & T_i^{MAC} + T_p \leq t < T_{i+1}^{MAC} \end{cases} \tag{2.15}$$

2.3.3 Transport Layer

To describe the evolution of the window size, two stochastic processes $W = (W_t)_{t \geq 0}$ and $H = (H_t)_{t \geq 0}$ are defined, where W_t is the window size of the TCP flow, and H_t is the corresponding slow-start threshold at time t.

2.3.3.1 Underlying Point Processes

Given the description in Section 2.2, there exist two underlying strictly increasing sequences of random variables representing two point processes:

- for the arrival of acknowledgments $\{T_n,\ n = 0, 1, \dots\}$ with $T_0 = 0$, P-a.s. and intensity $\lambda_t > 0$, and

- for the timeout events $\{S_n,\ n = 0, 1, \dots\}$ with $S_0 = 0$, P-a.s. and intensity $\mu_t > 0$.

The point process $\{T_n,\ n = 0, 1, \dots\}$ represents the arrival of acknowledgments at the TCP sender, and is closely related to the MAC and the physical layer. With the assumption that there are always acknowledgments waiting transmission at the MAC layer at the TCP receiver side, it was shown in Section 2.3.2 that the successful (i.e. without collisions) transmissions of acknowledgments at the MAC layer form a renewal process.

The assumption that there are always acknowledgments at the receiver side waiting for transmission at the MAC layer is a strong one. The ramification of this assumption is two fold: (i) we underestimate the inter-ack time intervals, and (ii) we decouple the forward and backward channel in the sense that any delays introduced to the data packet transmission times by the MAC layer in the forward channel are not preserved in the acknowledgment point process in the backward channel.

Those acknowledgments that survived collisions at the MAC layer are subject to the quality of the physical layer. Thus, each of these acknowledgments is successfully received at the TCP sender with probability π_g that is given by (2.4b) and this happens independently of the operation of the MAC layer (thinning of the point process), assuming the Markov chain that represents the physical layer operates at the stationary regime.

If $\mathcal{F}^N = (\mathcal{F}_t^N)_{t\geq 0}$ is the right continuous filtration that represents the history of the point process $\{T_n,\, n = 0, 1, \ldots\}$, then the $\mathcal{F}^N$-intensity λ_t of the process is

$$\lambda_t = \pi_g \lambda_t^{MAC} \tag{2.16}$$

where π_g is given by (2.4b) and λ_t^{MAC} is given by (2.15).

For each packet sent to the network, TCP expects an acknowledgment back from the receiver acknowledging the receipt of the packet. In the case of poor channel quality such an acknowledgment may be lost. If the TCP sender does not receive the acknowledgment in certain amount of time it will assume the packet was not properly received by the receiver and will retransmit it, minimizing at the same time its window size and in effect the throughput of the connection. In a wireless network though an acknowledgment may experience delays because of the MAC and the collisions that take place when accessing the channel. Thus, the TCP sender should not be anxious declaring a timeout and in effect minimizing the sending rate to the network. On the other hand, the more the TCP sender is waiting for the arrival of an acknowledgment, the more the connection remains idle resulting in performance degradation.

2.3.3.2 The Slow-Start Threshold Process $H = (H_t)_{t \geq 0}$

Based on the point processes defined above, the stochastic process H that represents the slow-start threshold in TCP is given by:

$$H_0 = h, \quad P\text{-a.s.}$$
$$H_t = h + \sum_{n=1}^{\infty} \mathbf{1}[S_n \leq t] \, \Delta H_{S_n}, \quad t > 0 \tag{2.17}$$

where h is given. The sample paths of H defined by (2.17) are piecewise constant and right continuous with left limits (càdlàg process). The magnitude of each jump at the points of the process $\{S_n,\ n = 1, 2, \dots\}$ is given by

$$\begin{aligned}
\Delta H_{S_n} &= H_{S_n} - H_{S_n^-} \\
&= H_{S_n} - H_{S_{n-1}} \\
&= \max\left\{2, \frac{W_{S_n}^-}{2}\right\} - \frac{W_{S_{n-1}}^-}{2} \\
&= \begin{cases} \frac{1}{2}\left(W_{S_n}^- - W_{S_{n-1}}^-\right), & \text{if } W_{S_n}^- \geq 4 \\ 2 - \frac{W_{S_{n-1}}^-}{2}, & \text{otherwise} \end{cases}
\end{aligned}$$

for $n = 2, 3, \ldots$, and

$$\begin{aligned}
\Delta H_{S_1} &= H_{S_1} - H_{S_1^-} \\
&= H_{S_1} - H_{S_0} \\
&= \max\left\{2, \frac{W_{S_1}^-}{2}\right\} - H_{S_0} \\
&= \begin{cases} \frac{W_{S_1}^-}{2} - h, & \text{if } W_{S_1}^- \geq 4 \\ \\ 2 - h, & \text{otherwise} \end{cases}
\end{aligned}$$

and it is zero for all the other time instances.

2.3.3.3 The Window Size Process $W = (W_t)_{t \geq 0}$

As was described in Section 2.2, the window size evolution is driven by the two point processes $\{T_n,\, n = 0, 1, \ldots\}$, and $\{S_n,\, n = 0, 1, \ldots\}$.

During the slow-start phase the window size is increased by 1 for every received acknowledgment. Define the counting process $N = (N_t)_{t \geq 0}$ that is associated with the point process $\{T_n,\, n = 0, 1, \ldots\}$ and counts the received acknowledgments:

$$N_t = \sum_{i=1}^{\infty} \mathbf{1}[T_n \leq t]\,,\; t \geq 0$$

Then, in the slow-start phase the window size evolves according to

$$W_t = W_0 + N_t,\; t \geq 0 \tag{2.18}$$

where W_0 is given. As was described in Section 2.2 the evolution of the window size in congestion avoidance phase is more conservative compared to the case of the slow-start phase. Based on that description the window size evolution in the congestion avoidance phase is described by

$$dW_t = \frac{1}{W_t} dN_t,\ t \geq 0 \tag{2.19}$$

The continuous time evolution of the window size is characterized by (2.18) and (2.19) for slow-start and congestion avoidance respectively. It starts in the slow-start phase and if there are no timeouts it switches to the congestion avoidance phase whenever the window size W_t becomes larger than the slow-start threshold H_t. Whenever the window size W_t reaches its maximum allowable value W_{max}, it remains to this value. In any case, whenever a timeout occurs, TCP switches to slow-start and sets the window size to its minimum value W_0:

$$W_{S_n} = W_0,\ n = 1, 2, \ldots$$

and the window size evolves according to (2.18). To summarize, the window size evolution is described by:

$$\begin{aligned} dW_t &= \begin{cases} dN_t, & W_t < H_t \\ \frac{1}{W_t} dN_t, & W_t \geq H_t \end{cases} \\ W_{S_n} &= W_0, \quad n = 1, 2, \ldots \\ W_0 &\leq W_t \leq W_{max}, \quad t \geq 0 \end{aligned} \tag{2.20}$$

Note that both processes $W = (W_t)_{t \geq 0}$ and $H = (H_t)_{t \geq 0}$ are fully observable by the TCP sender (controller).

2.4 Validation of the Model

We compare the analytical results given by (2.20) against ns-2 [54] simulations. For the ns-2 simulations a detailed model of the pure (unslotted) Aloha protocol was developed for the wireless network protocol stack. The wireless network is a single cell where all nodes can hear each other. In all the experiments we establish a TCP Tahoe connection between two wireless nodes. The TCP connection is used to serve a persistent FTP transfer for the duration of each experiment. In the neighborhood of these nodes there are pairs of nodes that exchange data packets over UDP connections that serve CBR traffic. The CBR traffic provides the background traffic that contributes to packet collisions at the MAC layer. Both TCP and UDP data packets are 1024 bytes long. Each CBR connection has a rate of 64kbps and implements the ns-2 mechanism of random perturbation of the packet transmission times. Moreover, we introduce random packet losses that simulate packet errors due to fading and channel quality degradation. The channel capacity in all experiments is 2Mbps.

In Fig. 2.3 the window size of the TCP Tahoe connection is shown for both the analytical results given by (2.20) and the ns-2 simulation. The network consists of the two nodes that participate in the TCP connection and 20 nodes that provide the background traffic and thus the collisions at the MAC layer. In this scenario there are no packet losses due to low channel quality and thus there are no timeout events at the TCP sender. Because of that, the TCP sender starts in the slow-start mode of operation and when the slow-start threshold is reached it switches to the slower regime of congestion avoidance. The mean exponential backoff duration for the MAC is 0.1sec.

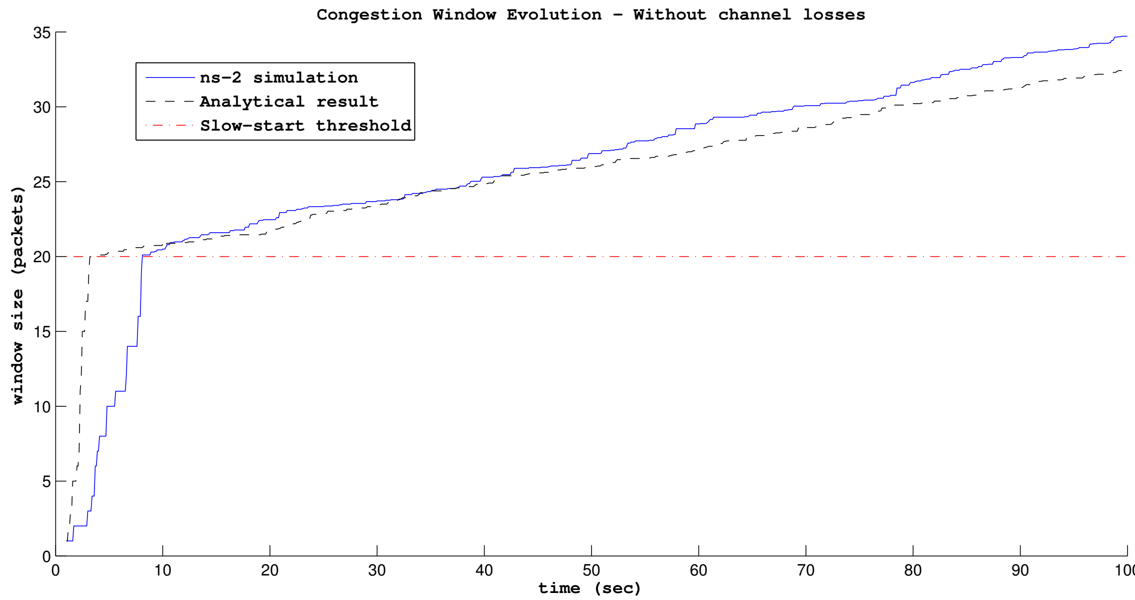

Figure 2.3: Window size evolution in the absence of channel losses (22 nodes).

Similarly, Fig. 2.4 shows the window size for the analytical as well as the simulation results. In this case, the background traffic is created by 40 nodes. As before, there are no channel errors but there are timeouts because of packet losses due to multiple collisions at the MAC layer.

In Fig. 2.5 the window size is shown for the same network of 22 nodes but in the presence of packet losses due to channel quality degradation. The mean exponential backoff duration after a collision is 0.1sec. In this experiment, the transition rates between the two states of the Markov chain that models the channel losses are $\lambda_{gb} = 1$ and $\lambda_{bg} = 2$. From (2.5a) and (2.5b) we can then compute the expected duration of each state (and thus the effect of channel losses to the data exchange) to be

$$\mathrm{E}[T_0] = \frac{1}{\lambda_{bg}} = 0.5 \text{ sec}$$

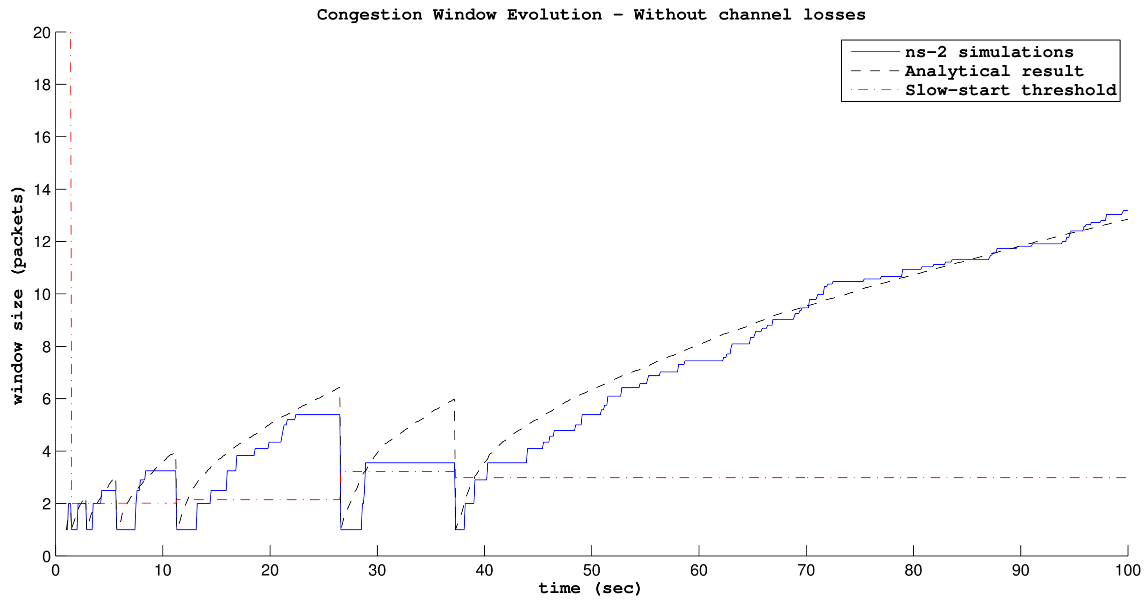

Figure 2.4: Window size evolution in the absence of channel losses (42 nodes).

$$\mathrm{E}[T_1] = \frac{1}{\lambda_{gb}} = 1 \text{ sec}$$

The scenario in Fig. 2.6 is for a network of 42 nodes. The mean exponential backoff duration is 0.1sec and the transition rates for the Markov chain are $\lambda_{gb} = 0.1$ and $\lambda_{bg} = 1$ giving mean durations for the ”bad” and the ”good” state $\mathrm{E}[T_0] = 1$ sec and $\mathrm{E}[T_1] = 10$ sec respectively.

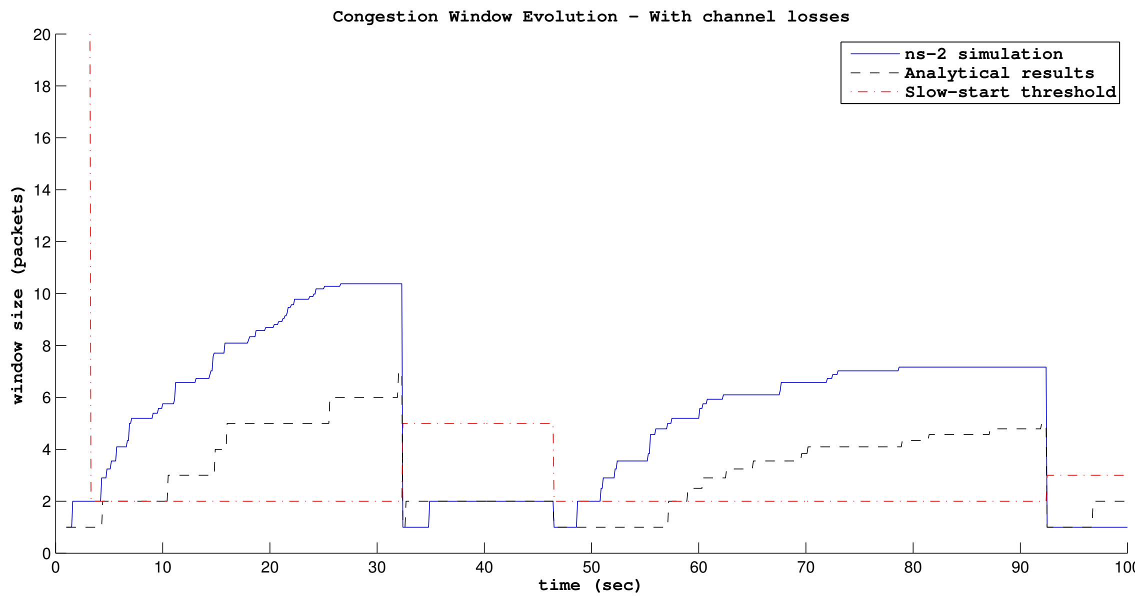

Figure 2.5: Window size evolution with channel losses (22 nodes).

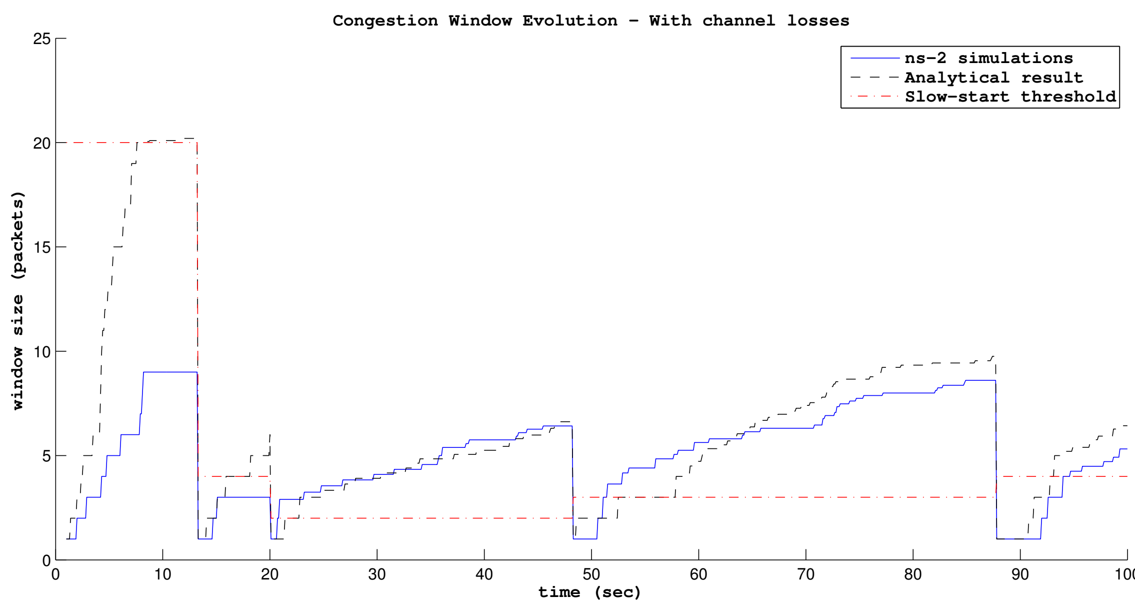

Figure 2.6: Window size evolution in the absence of channel losses (44 nodes).

Chapter 3

Optimal Timeout Mechanism of TCP over Aloha

This chapter presents an optimization problem that aims to maximize the throughput of a Transmission Control Protocol (TCP) connection between two nodes in a wireless ad-hoc network. More specifically, the setting is the one introduced in Chapter 2, where a persistent TCP connection is established between two nodes that are one hop away in a wireless unslotted Aloha network. The optimization is over the TCP timeout period, i.e. the problem is to find the optimal waiting period before the TCP sender declares a timeout event in the absence of a received acknowledgment for a transmitted packet. The problem is formulated as an optimal stopping problem. In the absence of a tractable analytical solution to the problem, a numerical method is proposed to achieve performance improvement of the system.

3.1 Introduction

As was mentioned in Chapters 1 and 2, the performance of TCP over a wireless network is poor. This is because the TCP sender assumes that the reason for a packet loss is congestion in the network. Although this is a reasonable assumption for the Internet, it might not be the case for wireless networks. The successful transmission of a packet over a wireless channel depends on the channel quality. In wireless networks the channel quality exhibits high variability and it often causes unrecoverable errors for the packet at the receiver. In these cases, the TCP receiver cannot acknowledge the received packet and drops it. When packets are dropped due to bursts of errors introduced at the channel level, the TCP sender declares a timeout in the absence of received acknowledgments. When this happens, TCP enters the slow-start phase minimizing its window size, and thus its throughput. Clearly, this is not the best remedy to the problem, since the lost packets are dropped because of a temporary quality degradation at the channel and not because of congestion.

The same situation arises when the channel affects the transmission of the acknowledgments themselves. Typically, in the case of a bidirectional TCP connection, the acknowledgments for the one direction are piggybacked onto the data packets to the opposite direction. Hence, acknowledgments are subject to the effect of the channel as data packets do. Even in unidirectional TCP connections where the acknowledgments are sent to the sender on their own, the bursty nature of the channel errors increases the probability that more than three consecutive acknowledgments are lost and thus a data packet loss is falsely detected at the TCP sender and a timeout is declared.

It makes sense then to try to maximize the TCP throughput by appropriately tuning the timeout interval for each packet the TCP source sends to the network. This tuning should take into consideration the parameters that characterize the operation of the Medium Access Control (MAC) layer as well as the channel. In the setting considered here, these parameters are the mean backoff time λ_{ret} for unslotted Aloha and the probabilities the wireless channel is in the "good" or the "bad" state, π_g, π_b respectively.

After a packet is sent to the network the TCP sender starts a timer which is set to a value according to an estimate of the round-trip time (RTT). In general, in current TCP implementations this estimate does not take into account the fact that the communication takes place over a wireless network and thus it considers any delays to be associated with congestion. In a wireless environment however, especially in random access networks such as the IEEE 802.11 or Aloha (as is the case here), packets are delayed because of collisions and retransmissions at the MAC layer. If a packet is delayed because there is heavy background traffic (i.e. traffic from or to nodes in the neighborhood of the TCP sender or receiver), its acknowledgment might not reach the TCP sender before the expiration of the timeout timer. Then, the TCP sender will declare a timeout, it will minimize the window size and enter slow-start. This will happen even if there is no congestion between the TCP sender and receiver. In this case the TCP throughput is minimized without any congestion being present in the network.

Consider the situation from the point of view of the TCP sender. In the current TCP implementations the TCP sender at some point sends a packet and starts the timeout timer. If no acknowledgment is received for that packet when the timer expires, the TCP sender declares a timeout and enters slow-start. The TCP sender has no way to know the exact cause for the lack of a received acknowledgment and it always assumes there is congestion in the network. If we want to maximize the TCP performance we need to incorporate in the timeout mechanism some information regarding the wireless medium. Consider again the situation where the TCP sender sends a packet to the network and waits for an acknowledgment. There are two reasons for an acknowledgment to be delayed. Either there is congestion in the network or the packet is delayed because of collisions and retransmissions at the MAC layer. In the first case the TCP sender would have liked to declare immediately a timeout and enter slow-start and thus minimize the traffic that is sent to the network in order to compensate for the congestion. In the second case however, a timeout event is not the best thing to do because the cause of the delay is the MAC layer and not congestion at the transport layer. In this case the TCP sender would have liked to wait and let the MAC layer resolve the collision. Then, the received acknowledgment would trigger the transmission of packets from the current state of the TCP sender and thus no performance decrease would be observed regarding the TCP throughput.

It is clear then, that the TCP sender has to choose between two different actions in the absence of a received acknowledgment. Either stop the waiting period and enter slow-start (by essentially declaring a timeout), or continue waiting for the acknowledgment, hoping that it is delayed at the MAC layer and not because of congestion. Thus, an optimal stopping problem can be defined. The solution to this problem provides the TCP sender with the optimal timeout period in order to increase its throughput.

3.2 Problem Formulation

The setting is the same as in Chapter 2. We assume there exists a complete probability space $(\Omega, \mathcal{F}, P)$. We consider a wireless network where the MAC layer is unslotted (pure) Aloha. Each node in the network can hear the transmissions from any other node (single cell). We model the wireless channel as a two-state continuous time Markov chain. One state corresponds to the case the channel is "good", i.e. the packets are not lost w.p. 1, and the other state corresponds to the case the channel is "bad" which means the transmitted packets are lost w.p. 1. The transition rate from the "bad" to the "good" state is λ_{bg} and the transition rate from the "good" to the "bad" state is λ_{gb}.

In the unslotted Aloha protocol a packet is immediately transmitted to the network. If this transmission overlaps with another packet transmission from another node, then there is a collision and neither of the packets are received by the corresponding receivers, and have to be retransmitted. To avoid another collision, the nodes that participated in the collision have to delay their retransmission for an exponentially distributed random interval. The mean of this exponential distribution is the same for all nodes and is denoted by $1/\lambda_{ret}$. Moreover, this randomly selected interval is independent of any possible previous delays at each node and across nodes.

We focus on a TCP connection between two nodes in the network that are one hop away. We assume this is a persistent TCP connection which implies that the sender has always packets to send. If W_t is the TCP window size at the sender at time t, we can define a stochastic process $W = (W_t)_{t\geq 0}$ that evolves according to the dynamics of the TCP protocol and affected by the MAC layer and the wireless channel.

In Section 2.3 of Chapter 2 we provided Eq. (2.20) which is a stochastic differential equation that describes the evolution of the window size as this evolution is driven by a point process that represents the arrival of acknowledgments from the TCP receiver to the TCP sender. More specifically, we have:

$$dW_t = \begin{cases} dN_t, & W_t < H_t \\ \frac{1}{W_t} dN_t, & W_t \geq H_t \end{cases}$$

$$W_{S_n} = W_0, \quad n = 1, 2, \ldots \tag{3.1}$$

$$W_0 \leq W_t \leq W_{max}, \quad t \geq 0$$

where $H = (H_t)_{t\geq 0}$ is the process that describes the slow-start threshold and is given by (2.17). W_0 and W_{max} are the initial (minimum) and maximum values of the window size, and $\{S_n, n = 0, 1, \dots\}$ is the sequence of the time instances a timeout event is declared by the TCP sender. The stochastic process $N = (N_t)_{t \geq 0}$ is the counting process that corresponds to the point process $\{T_n^{MAC}, n = 0, 1, 2, \dots\}$ that represents the arrival of acknowledgments at the TCP sender. If $\mathcal{F}^N = (\mathcal{F}_t^N)_{t \geq 0}$ is the right continuous filtration that represents the history of this point process, it was shown in Section 2.3.3 of Chapter 2 that the $\mathcal{F}^N$-intensity λ_t of the process is

$$\lambda_t = \frac{\lambda_{bg}}{\lambda_{bg} + \lambda_{gb}} \lambda_t^{MAC} \tag{3.2}$$

where

$$\lambda_t^{MAC} = \begin{cases} 0, & T_i^{MAC} \leq t < T_i^{MAC} + T_p \\ \lambda_{ret} p_{mac}, & T_i^{MAC} + T_p \leq t < T_{i+1}^{MAC} \end{cases} \tag{3.3}$$

and $T_p = L/C$ is the transmission time of the packet of constant length L bits over the wireless channel of capacity C bps, and p_{mac} is the probability of a successful transmission of a packet for unslotted Aloha given by (2.1)

As was mentioned in Section 3.1 we want to increase the TCP throughput by computing the optimal timeout period τ for the TCP sender to declare a timeout. This can be formulated as an optimal stopping time problem in the probability space $(\Omega, \mathcal{F}, P)$. If $\mathcal{F}^{TCP} = (\mathcal{F}_t^{TCP})_{t \geq 0}$ is the right continuous filtration that represents the history of events observed by the TCP sender we want to find the optimal $\mathcal{F}^{TCP}$-stopping time τ such that

$$J(w, \tau; h) = \mathrm{E}_w \left[\int_0^\tau e^{-\beta t} k(W_t; h) dt + e^{-\beta \tau} g(W_\tau; h) \right] \tag{3.4}$$

is maximized over τ, for $\beta > 0$, where $\mathrm{E}_w[\cdot]$ represents the expected value conditioned on the event that the initial value of the process W is w and the slow-start threshold is h.

The value h of the slow-start threshold remains constant for the duration of the timeout period and it only affects the size of the window if an acknowledgment is received. The value function $V(\cdot;h)$ for the problem can then be defined as:

$$V(w;h) = \sup_{\tau} J(w,\tau;h) \tag{3.5}$$

Notice that the two filtrations, $\mathcal{F}^N$ and $\mathcal{F}^{TCP}$ are equal since the TCP sender observes the arrival of acknowledgments. Moreover, the arrival of acknowledgments dictates the evolution of the process $W = (W_t)_{t\geq 0}$.

3.3 Hamilton-Jacobi-Bellman (HJB) Equation

Let B denote the optimal stopping set, i.e. the process stops when the set B is reached for the first time. Then, the equation satisfied by the value function $V(\cdot;h)$ is known as the Hamilton-Jacobi-Bellman (HJB) equation and is [31, 49]:

$$\begin{cases} \mathcal{L}V(w;h) - \beta V(w;h) + k(w;h) = 0, & w \notin B \\ V(w;h) = g(w;h), & w \in B \end{cases} \tag{3.6}$$

where $\mathcal{L}$ is the infinitesimal generator of the process W defined on a function $f : \Re \rightarrow \Re$ as:

$$\mathcal{L}f(x) = \lim_{t\downarrow 0} \frac{\mathrm{E}_x\left[f(W_t)\right] - f(x)}{t} \tag{3.7}$$

To see the validity of (3.6) take $\Delta > 0$ and small, and for convenience, drop the parameter h from all the expressions in the following. Suppose that at some point the system is at state w. At that moment we can either stop or continue the process. The gain from stopping the process is $g(w)$, which means that

$$V(w) = g(w)$$

for $w \in B$, the stopping set. If the decision is to continue, the value function takes the value

$$V(w) = \mathbf{E}_w \left[\int_0^\tau e^{-\beta t} k(W_t)\, dt \right], \quad \text{for } w \notin B$$

In this case we have:

$$\begin{aligned}
V(w) &= \mathbf{E}_w \left[\int_0^{\tau \wedge \Delta} e^{-\beta t} k(W_t)\, dt + \int_\Delta^\tau e^{-\beta t} k(W_t)\, dt \cdot \mathbf{1}[\tau > \Delta] \right] \\
&= \mathbf{E}_w \left[\int_0^{\tau \wedge \Delta} e^{-\beta t} k(W_t)\, dt + e^{-\beta \Delta} \int_\Delta^\tau e^{-\beta (t-\Delta)} k(W_t)\, dt \cdot \mathbf{1}[\tau > \Delta] \right] \\
&= \mathbf{E}_w \left[\int_0^{\Delta} e^{-\beta t} k(W_t)\, dt + \int_0^\tau e^{-\beta t} k(W_t)\, dt \cdot \mathbf{1}[\tau \leq \Delta] \right] \\
&\quad + \mathbf{E}_w \left[e^{-\beta \Delta} \int_\Delta^\tau e^{-\beta (t-\Delta)} k(W_t)\, dt \cdot \mathbf{1}[\tau > \Delta] \right] \\
&= \mathbf{E}_w \left[\int_0^{\Delta} e^{-\beta t} k(W_t)\, dt + \int_0^\tau e^{-\beta t} k(W_t)\, dt \cdot \mathbf{1}[\tau \leq \Delta] \right] \\
&\quad + \mathbf{E}_w \left[e^{-\beta \Delta} \cdot \mathbf{E}_{W_\Delta} \left[\int_\Delta^\tau e^{-\beta (t-\Delta)} k(W_t)\, dt \cdot \mathbf{1}[\tau > \Delta] \right] \right]
\end{aligned}$$

where $\mathbf{1}[\cdot]$ is the indicator function. But

$$V(W_\Delta) = \mathbf{E}_{W_\Delta} \left[\int_\Delta^\tau e^{-\beta (t-\Delta)} k(W_t)\, dt \cdot \mathbf{1}[\tau > \Delta] \right]$$

Then, the equation for the value function becomes

$$\begin{aligned}
V(w) = \mathbf{E}_w \left[\int_0^{\Delta} e^{-\beta t} k(W_t)\, dt \right] + \mathbf{E}_w \left[\int_0^\tau e^{-\beta t} k(W_t)\, dt \cdot \mathbf{1}[\tau \leq \Delta] \right] \\
+ e^{-\beta \Delta} \mathbf{E}_w \left[V(W_\Delta) \right]
\end{aligned}$$

Add and subtract $\mathrm{E}_w\left[V(W_\Delta)\right]$ to get

$$0 = \mathrm{E}_w\left[V(W_\Delta)\right] - V(w) + \left(e^{-\beta\Delta} - 1\right)\mathrm{E}_w\left[V(W_\Delta)\right] + \mathrm{E}_w\left[\int_0^\Delta e^{-\beta t} k(W_t)\, dt\right] + \mathrm{E}_w\left[\int_0^\tau e^{-\beta t} k(W_t)\, dt \cdot \mathbf{1}[\tau \le \Delta]\right]$$

Divide both sides by Δ:

$$0 = \frac{1}{\Delta}\Big(\mathrm{E}_w\left[V(W_\Delta)\right] - V(w)\Big) + \frac{e^{-\beta\Delta} - 1}{\Delta}\mathrm{E}_w\left[V(W_\Delta)\right] + \frac{1}{\Delta}\mathrm{E}_w\left[\int_0^\Delta e^{-\beta t} k(W_t)\, dt\right] + \frac{1}{\Delta}\mathrm{E}_w\left[\int_0^\tau e^{-\beta t} k(W_t)\, dt \cdot \mathbf{1}[\tau \le \Delta]\right]$$

Letting $\Delta \downarrow 0$:

$$0 = \mathcal{L}V(w) - \beta V(w) + k(w)$$

Moreover, whenever

$$\mathcal{L}V(w) - \beta V(w) + k(w) = g(w)$$

it does not matter whether we stop or continue since this has no effect on the total gain.

In general, as it can be seen from the computation presented above, it is always true that

$$V(w) \ge g(w)$$

and it is $V(w) = g(w)$ only when the process is stopped. Thus, the optimal strategy is simple enough, we should stop whenever the condition $V(w) = g(w)$ holds and this happens for $w \in B$, the stopping set. The problem of solving the HJB equation (3.6) using analytical methods is hard because the stopping set B is not known, but it is part of the solution [49]. In probabilistic terms, Snell [53] has shown that the stochastic process $V(W_t)$ is the smallest supermartingale that dominates the process $g(W_t)$. However, it is not easy to find this supermartingale using analytical methods and someone has to rely on numerical methods for its computation.

It can also be shown [49] that the optimal stopping problem for a Markov process X is equivalent to the problem of finding the smallest superharmonic function which dominates a properly defined gain function on the state space of the Markov process X. This connection can be seen through (3.6) which appears in both types of problems.

3.4 Preliminary Analysis for the Numerical Approximation

This section presents some results that will be needed for the numerical approximation to the optimal stopping problem.

3.4.1 Jump Process

In Section 2.3 of Chapter 2 we have defined the point process $\{T_n^{MAC}, n = 0, 1, 2, \dots\}$ and the corresponding counting process $N = (N_t)_{t \geq 0}$. The analysis in Chapter 2 provides the $\mathcal{F}^N$-compensator Λ_t for N which is:

$$\Lambda_t = \begin{cases} \pi_g \lambda_{ret} p_{mac} \big(T_i^{MAC} - iT_p \big), & T_i^{MAC} \leq t < T_i^{MAC} + T_p \\ \pi_g \lambda_{ret} p_{mac} t - (i+1) \pi_g \lambda_{ret} p_{mac} T_p, & T_i^{MAC} + T_p \leq t < T_{i+1}^{MAC} \end{cases} \tag{3.8}$$

Because of the way this compensator was computed it is implied [17] that the process $M = (M_t)_{t \geq 0}$ defined as $M_t = N_t - \Lambda_t$ for $t \geq 0$, is an $\mathcal{F}^N$-martingale. That means that for $0 \leq s \leq t$ it holds that:

$$\mathrm{E}[M_t \mid \mathcal{F}_s] = M_s \Rightarrow$$

$$\mathrm{E}[\mathrm{E}[M_t \mid \mathcal{F}_s]] = \mathrm{E}[M_s] \Rightarrow$$

$$\mathrm{E}[M_t] = \mathrm{E}[M_s] \Rightarrow$$

$$\mathrm{E}[M_t] = \mathrm{E}[M_0] = 0, \quad P\text{-a.s. for all } t \geq 0$$

Then, we have that

$$0 = \mathrm{E}[M_t] = \mathrm{E}[N_t] - \mathrm{E}[\Lambda_t] \Rightarrow \mathrm{E}[N_t] = \mathrm{E}[\Lambda_t], \quad t \geq 0$$

To compute the value of $\mathrm{E}[\Lambda_t]$ we consider two cases:

$T_i \leq t < T_i + T_p$: In this case $\Lambda_t = \pi_g \lambda_{ret} p_{mac} \big(T_i^{MAC} - iT_p\big)$ so we get:

$$\begin{aligned}\mathrm{E}[N_t] &= \mathrm{E}[\Lambda_t] \\ &= \pi_g \lambda_{ret} p_{mac} \big(\mathrm{E}[T_i^{MAC}] - iT_p\big) \\ &= \pi_g \lambda_{ret} p_{mac} \mathrm{E}\Big[\textstyle\sum_{j=1}^{K} X_j\Big], \quad \text{from (2.9)} \\ &= \pi_g p_{mac}^i S(i, p_{mac}), \quad \text{see Appendix B.1}\end{aligned}$$

where $S(i,p) = \sum_{k=0}^{i} \binom{i}{k} k \left(\frac{p}{1-p}\right)^k$.

$T_i + T_p \leq t < T_{i+1}$: In this case $\Lambda_t = \pi_g \lambda_{ret} p_{mac} t - (i+1)\pi_g \lambda_{ret} p_{mac} T_p$, so we have

$$\mathrm{E}[N_t] = \mathrm{E}[\Lambda_t] = \pi_g \lambda_{ret} p_{mac} t - (i+1)\pi_g \lambda_{ret} p_{mac} T_p$$

Notice that if we set $T_p = 0$, then

$$\mathrm{E}\big[N_t\big] = \pi_g \lambda_{ret} p_{mac} t, \quad t \geq 0$$

and the process becomes a Poisson point process.

If $\mathcal{B}(\Re)$ is the Borel σ-algebra on $\Re$ and $U \in \mathcal{B}(\Re)$ such that $0 \notin \overline{U}$ (i.e. 0 does not belong to the closure of U), we can define the Poisson measure $\mathbb{N} : [0, \infty) \times \mathcal{B}(\Re) \rightarrow \{0, 1, 2, \dots\}$ associated with the counting process $N = (N_t)_{t \geq 0}$ as

$$\mathbb{N}(t, U) = \sum_{0 < s \leq t} \mathbf{1}[\Delta N_s \in U]$$

where $\Delta N_s = N_s - N_{s^-}$. The Poisson measure counts the number of jumps up to time t that have size that belongs to U.

The Lévy measure $\nu : \mathcal{B}(\Re) \rightarrow \Re$ is defined as

$$\nu(U) = \mathrm{E}\big[\mathbb{N}(1, U)\big]$$

for $U \in \mathcal{B}(\Re)$ and $0 \notin \overline{U}$.

In our case, the Lévy measure for the counting process $N = (N_t)_{t \geq 0}$ can be computed as

$$\begin{aligned}
\nu(U) &= \mathrm{E}\big[\mathbb{N}(1, U)\big] \\
&= \mathrm{E}\big[\textstyle\sum_{0 < s \leq t} \mathbf{1}[\Delta N_s \in U]\big] \\
&= \mathrm{E}\big[N_1 \cdot \mathbf{1}[\text{jump size } \in U]\big] \\
&= \mathrm{E}\big[N_1\big] \cdot \mathrm{E}\big[\mathbf{1}[\text{jump size } \in U]\big]
\end{aligned}$$

where we used the fact that the time instances the jumps occur are independent of the jump size. To compute $\mathrm{E}\big[N_1\big]$ we assume that $T_p < 1$ which is always the case in real wireless networks. Then,

$$\mathrm{E}\big[N_1\big] = \pi_g \lambda_{ret} p_{mac} \big(1 - T_p\big)$$

To compute $\mathrm{E}\big[\mathbf{1}[\text{jump size } \in U]\big]$, recall that for the point process that corresponds to $N = (N_t)_{t\geq 0}$, the jumps are always of size 1. Thus,

$$\mathrm{E}\big[\mathbf{1}[\text{jump size } \in U]\big] = P\{1 \in U\} = \delta_1(U)$$

where δ_x is the Dirac measure at x, such that for $A \subset \Re$

$$\delta_x(A) = \begin{cases} 1, & \text{if } x \in A \\ \\ 0, & \text{if } x \notin A \end{cases}$$

Then,

$$\nu(U) = \pi_g \lambda_{ret} p_{mac} \cdot \big(1 - T_p\big) \cdot \delta_1(U)$$

for $U \in \mathcal{B}(\Re)$ and $0 \notin \overline{U}$.

Now, we can rewrite the window size evolution of (3.1) as

$$W_t = W_0 + \int_0^t \int_{[0,\infty)} q(W_{s^-}, a; h)\, \mathbb{N}(ds, da), \quad t \geq 0 \tag{3.9}$$

where

$$q(w, a; h) = \begin{cases} a, & \text{if } w < h \text{ (slow-start)} \\ \\ \frac{a}{w}, & \text{if } w \geq h \text{ (congestion avoidance)} \end{cases}$$

3.5 Numerical Approximation

Because of the analytical intractability of the optimal stopping problem introduced in Section 3.2 we need to solve the problem numerically. Following Kushner's Markov chain approximation method [31], we proceed by discretizing the time and thus moving from the original optimal stopping problem in continuous time to an equivalent problem in discrete time. In discrete time, the evolution of the system is described by a Markov chain, with transition probabilities that can be computed from the original continuous time dynamical system. We then formulate an optimal stopping time problem associated with the Markov chain and solve the corresponding dynamic programming equation using the value iteration method [50].

3.5.1 Markov Chain Approximation

The discretization of the time is based on the transmission time T_p of a packet. In particular, we define the time increment δ to be

$$\delta = \frac{T_p}{K} \tag{3.10}$$

where K is a positive integer. For larger values of K, a smaller increment δ is defined and as K increases to infinity, the discretization becomes finer.

The stochastic process $W = (W_t)_{t\geq 0}$ that describes the evolution of the window size takes values in the interval $[W_0, W_{max}]$. When TCP operates in the slow-start regime, the window size takes values in the set of positive integers, and when TCP is in the congestion avoidance phase the window size is a positive real number. For the optimal stopping problem of the approximating Markov chain the window size does not need to be discretized. We only need to differentiate between the current value of the window size and the value that the window size will take after a new acknowledgment is received or a maximum waiting time is reached. Therefore, we do not discretize the interval $[W_0, W_{max}]$ where the process $W = (W_t)_{t\geq 0}$ takes its values.

Suppose at time $t = t_0$ there is a jump to $W_{t_0} = w$ for the original, continuous time system (3.1) and the slow-start threshold H_t is h. We define the Markov chain that describes the evolution of the system given that at time $t = t_0$ the size of the window is w. We want to solve the optimal stopping problem for the Markov chain for any such initial condition (w, t_0).

The approximating Markov chain $X^{\delta,\ h,\ t_0,\ w} = \{X_n^{\delta,\ t_0,\ w},\ n = 0, 1, \dots\}$ that represents the evolution of the original, continuous time dynamical system (3.1) has a two dimensional state space:

$$\mathcal{X}^{\delta,\ h,\ t_0,\ w} = [W_0, W_{max}] \times \{t_0,\ t_0 + \delta,\ t_0 + 2\delta,\ \dots,\ t_0 + (K + M + 1) \cdot \delta\} \quad (3.11)$$

where M is the number of time increments that we allow after the time $t_0 + K\delta$ before we declare a timeout. Therefore, the parameter M defines an upper bound on the optimal stopping time of our problem and assures that the algorithm that computes this stopping time terminates.

In order to describe the transitions of the Markov chain $X^{\delta,\ h,\ t_0,\ w}$, suppose that the chain is in the state $(w, t_0 + i \cdot \delta)$. For $i = 0, 1, 2, \dots, (K - 1)$, the chain can only move in time leaving the first component of the state unchanged. This is true because in the original continuous time dynamical system (3.1) there is no new jump for a time duration equal to T_p (the transmission time of a packet) after a jump (which we assumed it happened at time t_0). Thus, for $i = 0, 1, 2, \dots, (K-1)$ the state that follows $(w, t_0 + i \cdot \delta)$ can only be $(w, t_0 + (i + 1) \cdot \delta)$ and this transition happens with probability 1.

After time $T_p = K \cdot \delta$ has elapsed from t_0 a new jump may occur. If the Markov chain $X^{\delta,\ h,\ t_0,\ w}$ is in state $(w, t_0 + (K + j) \cdot \delta)$ for $j = 0, 1, 2, \dots, (K + M - 1)$, there are two different events that may happen, and thus two possible transitions out of this state that represent these events.

The first event represents a new jump of the original system (3.1). Thus the second component of the state will increase by one to $t_0 + (K + j + 1) \cdot \delta$ and the first component of the state will be a new window size w'. The value of w' depends on whether TCP is in slow-start $(w < h)$ or congestion avoidance phase $(w \geq h)$. If TCP is in slow-start phase, then $w' = w + 1$, and if TCP is in congestion avoidance phase then $w' = w + \frac{1}{w}$. Such a transition indicates that a new acknowledgment has arrived at the TCP sender which implies that there is no need to find a timeout interval, and thus solve the optimal stopping problem. The transition from $(w, t_0 + (K + j) \cdot \delta)$ to $(w', t_0 + (K + j + 1) \cdot \delta)$ happens with probability p_j that is computed from the p.d.f. (2.7) for D^{MAC}. In this new state the only transition that is allowed is to itself with probability 1.

The second event represents the fact that no new arrival (and thus jump) has occurred. In this case, the first component of the next state remains the same and equal to w, and the second component indicates the increase in time by the increment δ. The transition from $(w, t_0 + (K + j) \cdot \delta)$ to $(w, t_0 + (K + j + 1) \cdot \delta)$ happens with probability $1 - p_j$.

Finally, if the Markov chain is in the state $(w, t_0 + (K + M) \cdot \delta)$ it means that the maximum allowed waiting period has been reached, therefore a timeout has to be declared. This is indicated by a transition to $(1, t_0 + (K + M + 1) \cdot \delta)$ with probability 1.

The two dimensional Markov chain $X^{\delta,\, h,\, t_0,\, w}$ can be represented schematically as in Fig. 3.1.

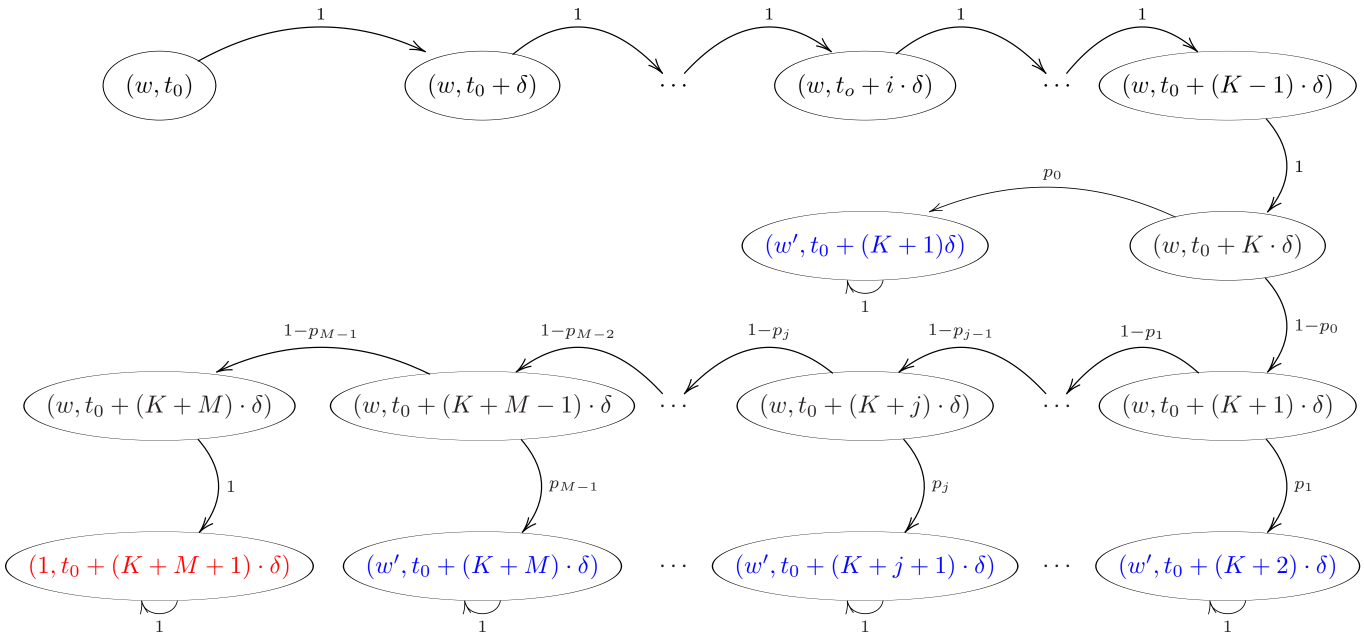

Figure 3.1: The Markov chain approximation for the Optimal Stopping problem.

3.5.2 Transition Probabilities

To better represent the Markov chain $X^{\delta,\, h,\, t_0,\, w}$, we name each of the two dimensional state of the chain as follows:

$$\begin{aligned} S_i &= \Big(w, t_0 + i \cdot \delta\Big), \quad i = 0, 1, 2, \ldots, (K+M) \\ F_j &= \Big(w', t_0 + (K+j) \cdot \delta\Big), \quad j = 1, 2, \ldots, M \\ R &= \Big(1, t_0 + (K+M+1) \cdot \delta\Big) \end{aligned} \tag{3.12}$$

Using the representation of (3.12) the Markov chain $X^{\delta,\, h,\, t_0,\, w}$ is shown in Fig. 3.2. The transition probabilities can then be given as

- for $i = 0, 1, 2, \ldots, (K-1)$,

$$Pr\{X_{n+1}^{\delta,\, h,\, t_0,\, w} = X \mid X_n^{\delta,\, h,\, t_0,\, w} = S_i\} = \begin{cases} 1, & \text{if } X = S_{i+1} \\ 0, & \text{otherwise} \end{cases} \tag{3.13}$$

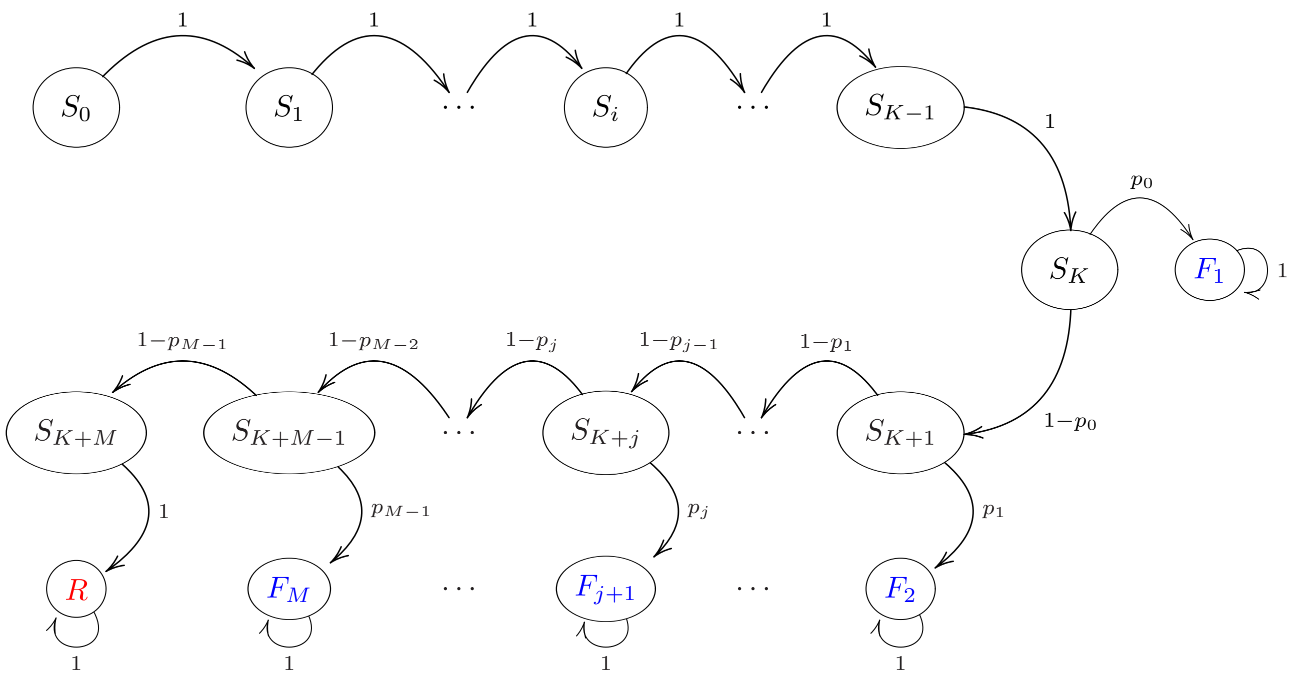

Figure 3.2: The Markov chain approximation for the Optimal Stopping problem with different state representation.

- for $i = K, (K+1), \ldots, (K+M-1)$,

$$Pr\{X_{n+1}^{\delta,\, h,\, t_0,\, w} = X \mid X_n^{\delta,\, h,\, t_0,\, w} = S_i\} = \begin{cases} 1 - p_{i-K}, & \text{if } X = S_{i+1} \\ p_{i-K}, & \text{if } X = F_{i-K+1} \\ 0, & \text{otherwise} \end{cases} \tag{3.14}$$

- for $i = K + M$,

$$Pr\{X_{n+1}^{\delta,\, h,\, t_0,\, w} = X \mid X_n^{\delta,\, h,\, t_0,\, w} = S_i\} = \begin{cases} 1, & \text{if } X = R \\ 0, & \text{otherwise} \end{cases} \tag{3.15}$$

- for $i = 1, 2, \ldots, M$,

$$Pr\{X_{n+1}^{\delta,\, h,\, t_0,\, w} = X \mid X_n^{\delta,\, h,\, t_0,\, w} = F_i\} = \begin{cases} 1, & \text{if } X = F_i \\ 0, & \text{otherwise} \end{cases} \tag{3.16}$$

- and,

$$Pr\{X_{n+1}^{\delta,\, h,\, t_0,\, w} = X \mid X_n^{\delta,\, h,\, t_0,\, w} = R\} = \begin{cases} 1, & \text{if } X = R \\ \\ 0, & \text{otherwise} \end{cases} \tag{3.17}$$

The cardinality of the state space for the Markov chain $X^{\delta,\, h,\, t_0,\, w}$ is $K + 2M + 2$. If we order the states in the following manner:

$$\Big(S_0,\ S_1,\ \ldots,\ S_{K+M},\ F_1,\ F_2,\ \ldots,\ F_M,\ R\Big) \tag{3.18}$$

then using the transition probabilities defined above we can write the $(K + 2M + 2) \times (K + 2M + 2)$ transition matrix $Q^{(\delta)}$ for the Markov chain $X^{\delta,\, h,\, t_0,\, w}$ in block form:

$$Q^{(\delta)} = \begin{pmatrix} \mathbb{O}_{K\times 1} & \mathbb{I}_K & \mathbb{O}_{K\times M} & \mathbb{O}_{K\times M} & \mathbb{O}_{K\times 1} \\ \mathbb{O}_{M\times 1} & \mathbb{O}_{M\times K} & \mathbb{I}_M - \mathbb{P}_M & \mathbb{P}_M & \mathbb{O}_{M\times 1} \\ 0 & \mathbb{O}_{1\times K} & \mathbb{O}_{1\times M} & \mathbb{O}_{1\times M} & 1 \\ \mathbb{O}_{M\times 1} & \mathbb{O}_{M\times K} & \mathbb{O}_{M\times M} & \mathbb{I}_M & \mathbb{O}_{M\times 1} \\ 0 & \mathbb{O}_{1\times K} & \mathbb{O}_{1\times M} & \mathbb{O}_{1\times M} & 1 \end{pmatrix} \tag{3.19}$$

where $\mathbb{O}_{\cdot\times\cdot}$ is the zero matrix, $\mathbb{I}_{\cdot}$ is the identity matrix and $\mathbb{P}_M$ is a diagonal matrix with the element at the i^{th} diagonal position equal to p_i:

$$\mathbb{P}_M = \begin{pmatrix} p_1 & 0 & \cdots & 0 & \cdots & 0 & 0 \\ 0 & p_2 & \cdots & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & p_i & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & p_{M-1} & 0 \\ 0 & 0 & \cdots & 0 & \cdots & 0 & p_M \end{pmatrix} \tag{3.20}$$

As an example, for $K = 2$ and $M = 3$, the transition matrix $Q^{(\delta)}$ becomes:

$$Q^{(\delta)} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & (1-p_0) & 0 & 0 & p_0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & (1-p_1) & 0 & 0 & p_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & (1-p_2) & 0 & 0 & p_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.21)$$

3.5.2.1 Computation of the Transition Probability p_i

To compute the transition probabilities p_i, $i = 0, 1, \ldots, (M-1)$ we use the results from Chapter 2. In Chapter 2 the random variable D^{MAC} represents the time between two successive packet transmissions at the MAC layer. The p.d.f. of D^{MAC} is given by (2.7):

$$f_{D^{MAC}}(t) = p_{mac}\delta(t-T_p) + p_{mac}(1-p_{mac})\lambda_{ret}e^{-\lambda_{ret}p_{mac}\cdot\left(t-T_p\right)}u_0(t-T_p) \quad (3.22)$$

for $t \geq 0$.

We first compute the probability:

$$\begin{aligned} Pr\{D^{MAC} > s\} &= \int_s^\infty f_{D^{MAC}}(t)\,dt \\ &= \int_s^\infty \Big(p_{mac}\delta(t-T_p) + p_{mac}(1-p_{mac})\lambda_{ret}e^{-\lambda_{ret}p_{mac}\cdot\left(t-T_p\right)}u_0(t-T_p)\Big)\,dt \end{aligned}$$

$$= \begin{cases} 1, & 0 \le s \le T \\ (1-p_{mac})e^{-\lambda_{ret} p_{mac} \cdot (s-T_p)}, & s > T \end{cases} \tag{3.23}$$

For $i = 0, 1, 2, \ldots$, we compute the probability $q_i^{T_p,\,\delta} = Pr\{T_p + (i+1)\delta > D^{MAC} > T_p + i\delta\}$

$$q_i^{T_p,\,\delta} = \int_{T_p+i\delta}^{T_p+(i+1)\delta} f_{D^{MAC}}(t)\,dt$$

$$= \begin{cases} 1-(1-p_{mac})e^{-\lambda_{ret} p_{mac} \delta}, & i = 0 \\ (1-p_{mac})e^{-\lambda_{ret} p_{mac} i\delta}\big(1-e^{-\lambda_{ret} p_{mac}\delta}\big), & i = 1, 2, \ldots \end{cases} \tag{3.24}$$

Using (3.23) and (3.24) we compute the probability:

$$Pr\Big\{T_p + (i+1)\delta > D^{MAC} > T_p + i\delta \mid D^{MAC} > T_p + i\delta\Big\}$$

$$= \frac{Pr\Big\{T_p + (i+1)\delta > D^{MAC} > T_p + i\delta,\ D^{MAC} > T_p + i\delta\Big\}}{Pr\Big\{D^{MAC} > T_p + i\delta\Big\}}$$

$$= \frac{Pr\Big\{T_p + (i+1)\delta > D^{MAC} > T_p + i\delta\Big\}}{Pr\Big\{D^{MAC} > T_p + i\delta\Big\}}$$

$$= \begin{cases} 1-(1-p_{mac})e^{-\lambda_{ret} p_{mac} \delta}, & i = 0 \\ 1-e^{-\lambda_{ret} p_{mac}\delta}, & i = 1, 2, \ldots \end{cases} \tag{3.25}$$

Using (3.25) and taking into account the operation of the wireless channel, we can compute the transition probability p_i:

$$p_i = \pi_g \cdot Pr\Big\{T_p + (i+1)\delta > D^{MAC} > T_p + i\delta \mid D^{MAC} > T_p + i\delta\Big\}$$
$$= \begin{cases} \pi_g \cdot \Big(1 - (1-p_{mac})e^{-\lambda_{ret}p_{mac}\delta}\Big), & i = 0 \\ \pi_g \cdot \Big(1 - e^{-\lambda_{ret}p_{mac}\delta}\Big), & i = 1, 2, \ldots, (M-1) \end{cases} \tag{3.26}$$

for $i = 0, 1, 2, \ldots, (M-1)$, where π_g is given by (2.4b):

$$\pi_g = \frac{\lambda_{bg}}{\lambda_{bg} + \lambda_{gb}} \tag{3.27}$$

3.5.3 Running and Final Rewards

The running reward $k(\cdot; h)$ is defined in such a way as to represent our unwillingness to declare a timeout and thus minimize the window size. At the same time though this unwillingness should be decreasing with time, since as time increases and no event has occurred (no arrival of an acknowledgment) is an indication of bad channel quality. This means that the chances of finally receiving an acknowledgment become smaller. On the other hand, if we have already built a large window size we might be reluctant to declare a timeout since declaring a timeout brings the window size to its minimum value. Thus, the running reward $k(\cdot; h)$ is an increasing function of the current window size and a decreasing function of the waiting time.

In case an acknowledgment has arrived and the Markov chain $X^{\delta,\, h,\, t_0,\, w}$ has moved to an F_i, $i = 1, 2, \ldots, M$ state, there is no need to declare a timeout and thus solve the optimal stopping problem. Because of this, the running cost is 0 for these states.

If no acknowledgment has arrived and the maximum waiting time has been reached, the Markov chain $X^{\delta,\, h,\, t_0,\, w}$ will move to state R. When this transition happens a timeout is declared anyway, and no further action is needed. Hence, the running reward for the R state is 0.

More precisely, the running reward is given by (3.28)

$$
\begin{aligned}
k(S_i; h) &= \frac{w}{W_{max}}\Big(K + M + 1 - i\Big)\alpha\delta, \quad i = 0, 1, 2, \ldots, (K + M) \\
k(F_i; h) &= 0, \quad i = 1, 2, \ldots, M \\
k(R; h) &= 0
\end{aligned}
\tag{3.28}
$$

where α is a parameter that depends on whether the TCP sender is in the slow-start $(w < h)$ or the congestion avoidance phase $(w \geq h)$ and can be tuned based on the performance we want to achieve through the optimization problem.

The final reward $g(\cdot; h)$ also depends on both components of the state of the Markov chain $X^{\delta,\, h,\, t_0,\, w}$. For the states S_i, $i = 0, 1, 2, \ldots, (K - 1)$, the final reward should be 0, since these states represent waiting time during transmission of a packet and thus no event will occur with probability 1. For the rest of the S_i, $i = K, (K + 1), \ldots, (K + M)$ states, the final reward is defined to be an increasing function on both the window size and the waiting time.

If an acknowledgment is received, and thus the Markov chain $X^{\delta,\, h,\, t_0,\, w}$ moves to an F_i, $i = 1, 2, \ldots, M$, state, the final reward depends on the new window size which is different and depends on whether the TCP sender is in the slow-start $(w < h)$ or the congestion avoidance phase $(w \geq h)$.

Finally, the final reward for the state R depends on the maximum waiting time that we allow before we declare a timeout.

More specifically, the final reward $g(\cdot; h)$ is given by (3.29)

$$
\begin{aligned}
& g(S_i; h) = 0, \quad i = 0, 1, \ldots, (K-1) \\
& g(S_i; h) = \frac{w}{W_{max}}\, i\alpha, \quad i = K, (K+1), \ldots, (K+M) \\
& g(F_i; h) = \frac{w'}{W_{max}}(i+K)\alpha, \quad i = 1, 2, \ldots, M \\
& g(R; h) = (K+M+1)\alpha
\end{aligned}
\tag{3.29}
$$

where α is as in the case of the running reward $k(\cdot)$.

3.5.4 Optimal Stopping and Dynamic Programming

The optimal stopping problem presented in Section 3.2 can now be posed on the Markov chain $X^{\delta,\, h,\, t_0,\, w}$. If N_δ is a stopping time for the approximating chain $X^{\delta,\, h,\, t_0,\, w}$, we define the discounted reward according to (3.4) to be:

$$
J^{\delta,\, h}(x, N_\delta) = \mathrm{E}_x \left[\sum_{n=0}^{N_\delta - 1} e^{-\beta\, t_n^\delta} \cdot k\big(X_n^{\delta,\, h,\, t_0,\, w}; h\big) \cdot \delta + e^{-\beta\, N_\delta} \cdot g\big(X_{N_\delta}^{\delta,\, h,\, t_0,\, w}; h\big) \right]
\tag{3.30}
$$

where $t_n^\delta = n\delta$ and $\beta > 0$ the discount factor. If

$$
V^{\delta,\, h}(x) = \sup_{N_\delta} J^{\delta,\, h}(x, N_\delta) \tag{3.31}
$$

is the corresponding value function for the problem, it satisfies the dynamic programming equation:

$$
V^{\delta,\, h}(x) = \max\Big\{ \sum_y e^{-\beta\, \delta} Q^{(\delta)}(x, y) \cdot V^{\delta,\, h}(y) + k(x) \cdot \delta, \;\; g(x) \Big\} \tag{3.32}
$$

For numerical purposes, we can approximate $e^{-\beta\delta}$ in (3.32) with $\frac{1}{1+\beta\cdot\delta}$ [31].

Because of the discounting, the metric $J^{\delta,\,h}(x, N_\delta)$ in (3.30) is finite and well defined. The dynamic programming equation (3.32) is also well defined since the term

$$\sum_y e^{-\beta\,\delta} Q^{(\delta)}(x,y) \cdot V^{\delta,\,h}(y) + k(x) \cdot \delta \tag{3.33}$$

is a contraction mapping.

3.5.5 Simulation Results

To solve the optimal stopping problem for the approximating Markov chain $X^{\delta,\,h,\,t_0,\,w}$ we use the dynamic programming equation (3.32). Because of the contraction mapping property of (3.33) we use a combination of iteration methods in both the value and the policy space to get the solution to the stopping problem. The value iteration method solves for the value function $V^{\delta,\,h}(\cdot)$ and the policy iteration provides the optimal stopping set B of (3.6) for the problem. The stopping set B defines practically our optimal policy, in the sense that, whenever the Markov chain $X^{\delta,\,h,\,t_0,\,w}$ moves into a state $x \in B$ then we stop.

The parameters that define the experiments are related to the wireless channel, the MAC and the TCP. They are summarized in Table 3.1.

Wireless Channel	MAC	TCP	Approximation
λ_{bg}	λ_{ret}	T_p	δ, K
λ_{gb}	p_{mac}	β	M
			α

Table 3.1: Simulation parameters.

We run different experiments for different values of the parameters in Table 3.1 and compare against the standard timeout mechanism of TCP [48]. The channel capacity in all experiments is 2Mbps.

In Fig. 3.3 and Fig. 3.4 we compare the instantaneous rate when using the timeouts that are solutions to the optimal stopping time against the standard timeout mechanism of TCP. In the case of Fig. 3.3, the mean waiting time before a retransmission at the MAC layer is 0.1sec ($\lambda_{ret} = 10$), and in Fig. 3.4 the corresponding mean waiting time is 0.01sec ($\lambda_{ret} = 100$). In both cases, the are no losses at the wireless channel ($\pi_g = 1$). The probability of a successful transmission at the MAC layer is $p_{mac} = 0.3$ and the discount parameter β is 0.9. Also, in both cases the parameter α in the running and final rewards is 1 when TCP is in slow-start and 10 when in congestion avoidance. As it can be seen in both Fig. 3.3 and Fig. 3.4 the timeout mechanism that is produced from the numerical approximation to the stopping problem has better performance than the standard implementation of the timeout mechanism.

The case of channel losses is shown in Fig. 3.5 and Fig. 3.6. In both cases the channel loss probability is $\pi_b = 0.5$. Fig. 3.5 shows the case where the mean waiting time after a collision is 0.1sec ($\lambda_{ret} = 10$), and Fig. 3.6 corresponds to the case where the mean waiting time after a collision at the MAC layer is 0.01sec ($\lambda_{ret} = 100$). As before, the probability p_{mac} of not having a collision at the MAC layer is 0.3 for both cases. The discount parameter β is 0.9, and the parameter α in the running and final rewards is 1 for slow-start and 10 for congestion avoidance.

Finally, In Fig. 3.7 the results of the comparison are shown where the discount factor β is 0.001. The loss probability ($1 - \pi_g$) at the wireless channel is 0.3, and the

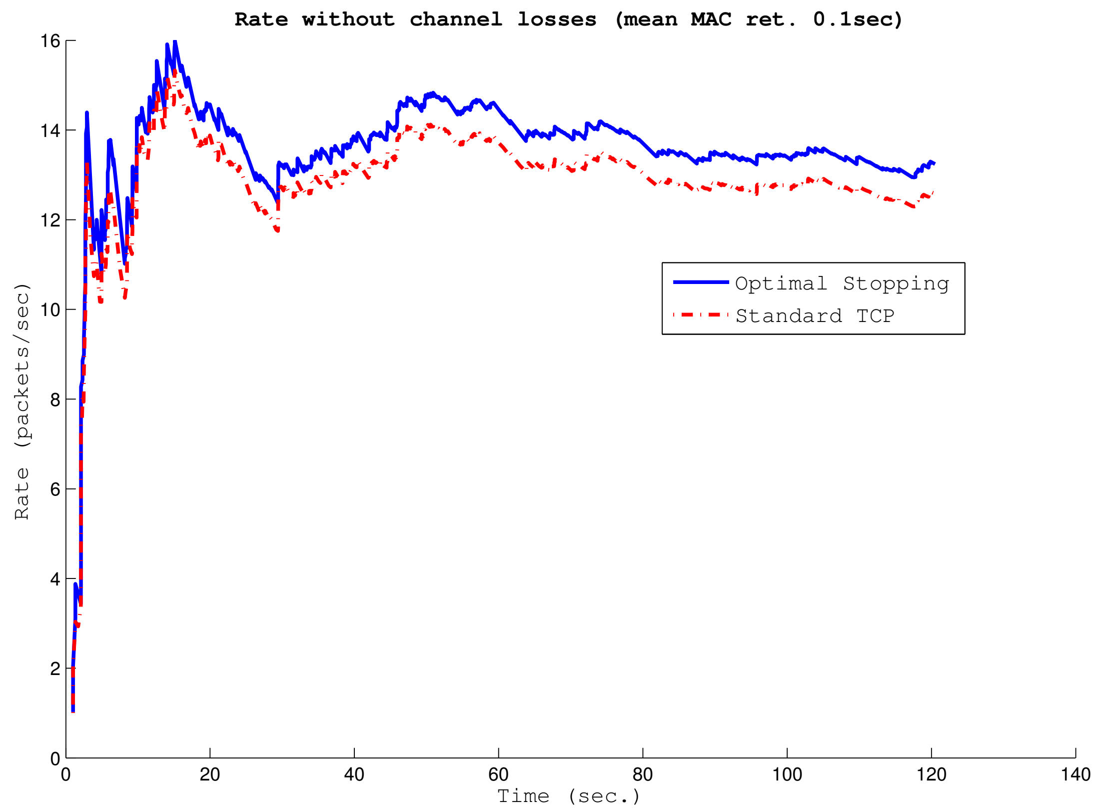

Figure 3.3: Comparison of the instantaneous rate between the stopping problem and the TCP mechanism in the absence of losses at the channel and mean retransmission waiting time 0.1sec at the MAC layer.

mean waiting time before retransmission at the MAC layer is 0.01sec ($\lambda_{ret} = 100$).

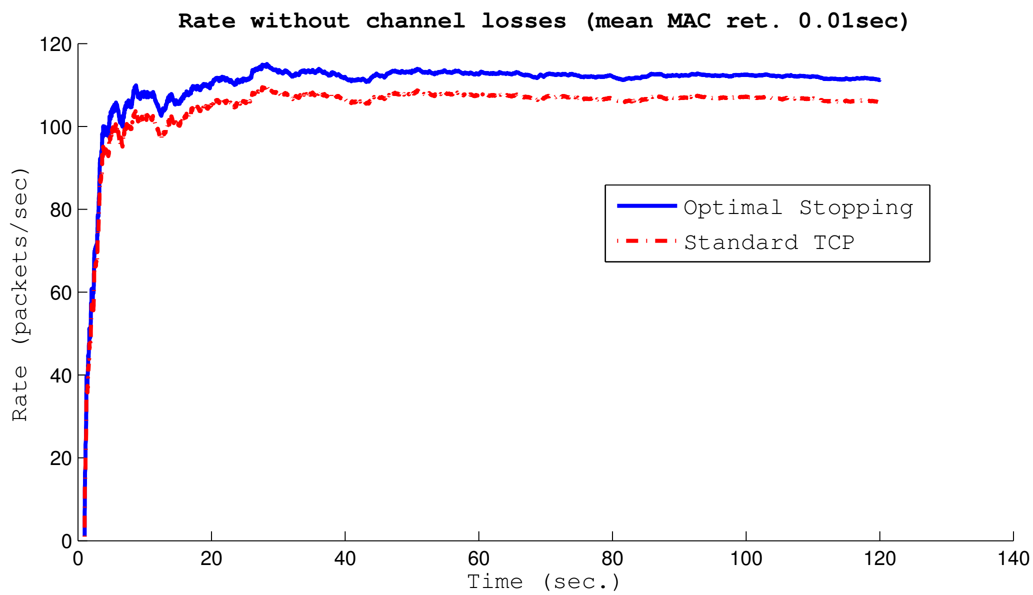

Figure 3.4: Comparison of the instantaneous rate between the stopping problem and the TCP mechanism in the absence of losses at the channel and mean retransmission waiting time 0.01sec at the MAC layer.

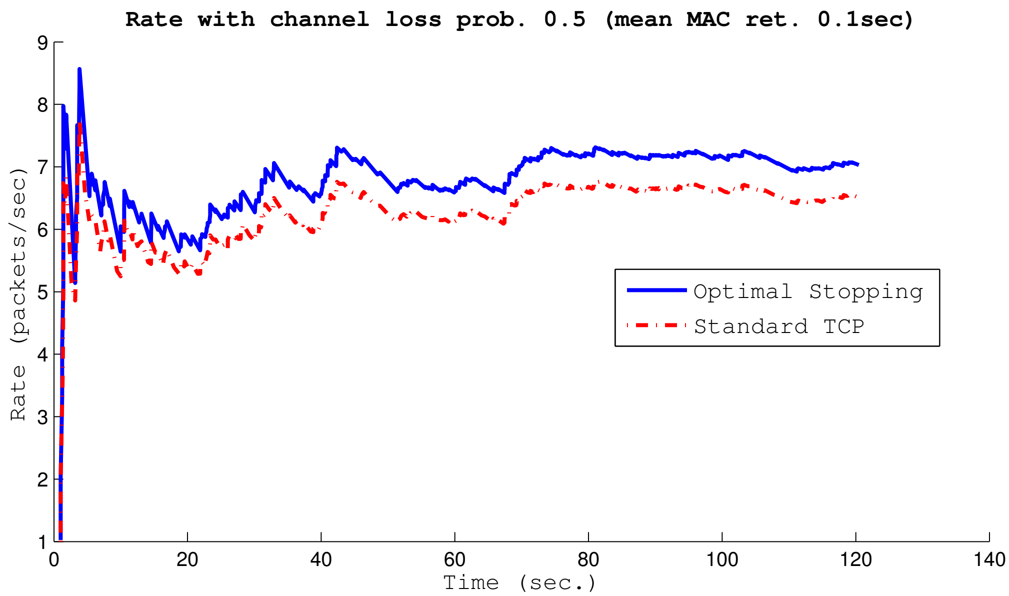

Figure 3.5: Comparison of the instantaneous rate between the stopping problem and the TCP mechanism with a loss prob. 0.5 at the channel and mean retransmission waiting time 0.1sec at the MAC layer.

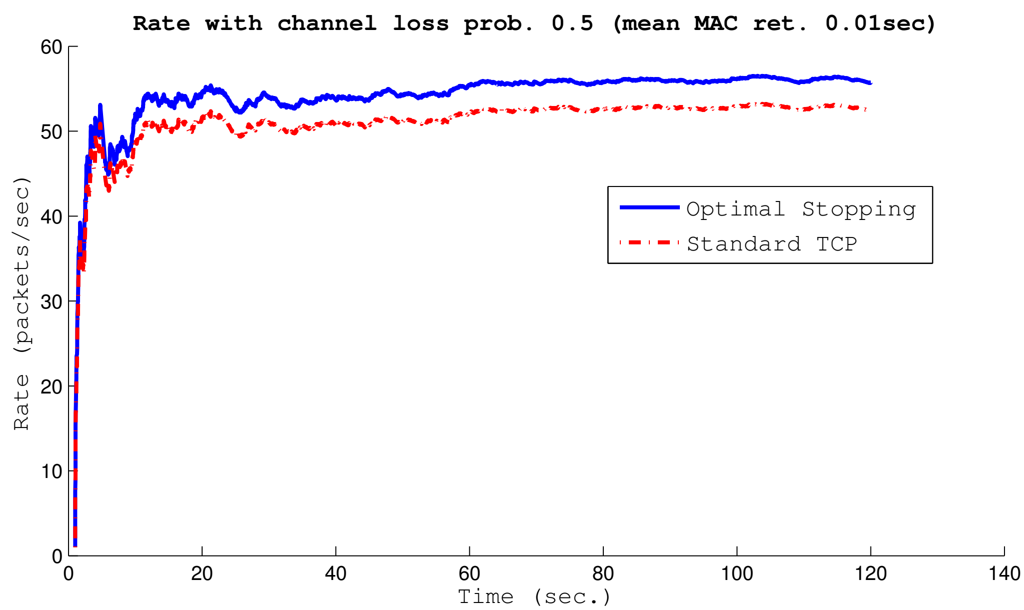

Figure 3.6: Comparison of the instantaneous rate between the stopping problem and the TCP mechanism with a loss prob. 0.5 at the channel and mean retransmission waiting time 0.01sec at the MAC layer.

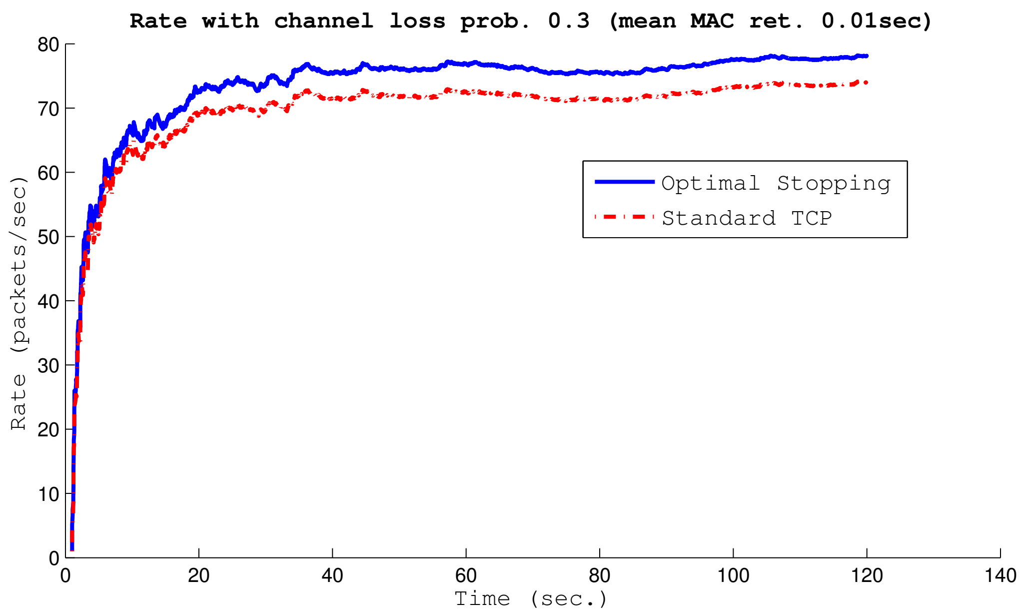

Figure 3.7: Comparison of the instantaneous rate between the stopping problem and the TCP mechanism with a loss prob. 0.3 at the channel, mean retransmission waiting time 0.01sec at the MAC layer and discount factor $\beta = 0.001$.

Chapter 4

Conclusions

The focus of this thesis is the flow control in wireless networks and specifically the behavior of the Transmission Control Protocol (TCP) in conjunction with a random access MAC protocol. The current work investigates issues that are not usually dealt with in the research literature, namely the dynamical behavior of TCP and more specifically the evolution of its window size and also the effect of the slow-start phase and the related mechanism of timeouts to the TCP throughput.

The general approach that is followed is based on the idea that taking advantage of the dependency among quantities that characterize different layers of the protocol stack, can improve the performance of the network. At the same time, we constrain ourselves to a minimum number of changes to the standard TCP operation. We do that in order to achieve interoperability between our solution and the standard TCP protocol.

4.1 TCP and Aloha

In Chapter 2 we have developed a stochastic differential equation that describes the evolution of the TCP window size with time. The random access MAC is the pure Aloha and the behavior of the channel is modeled through a two state Markov chain according to the Gilbert-Elliott model.

The equation is driven by a point process that represents the arrival of acknowledgments to the sender. We take into consideration the packet length and the finite capacity of the channel. This introduces for each point of the process, a time interval which is constant and during which no event occurs with probability 1. Because of this, the point process is not a Poisson process.

We fully characterize the point process by first computing the statistics of the inter-arrival intervals and then computing its compensator and the corresponding intensity.

Finally we compare our analytical model against simulations for different values of the parameters that characterize the channel and the Aloha protocol. We see that although our analysis is based on certain assumptions, our model is valid and closely matches the simulation results.

4.2 Timeout Mechanism

Chapter 3 builds on the results of Chapter 2 and uses the dynamics of the window size to properly tune the timeout mechanism of TCP in order to increase the throughput.

We pose the problem as an optimal stopping problem with the stochastic differential equation of Chapter 2 as a constraint. The fact that the driving point process is not a Poisson process makes the analysis intractable. Motivated by the approximation method of Kushner for stochastic control problems in continuous time, we develop a numerical approximation to the original problem. Using dynamic programming we solve the discrete time version of the original problem and retrieve stopping policies that define the new timeout mechanism. We verify the performance increase by comparing our solution to the standard TCP timeout mechanism using simulation and for different values of the involved parameters.

Appendix A

Appendix to Chapter 2

This appendix provides proofs to intermediate results needed in Chapter 2.

A.1 Physical Layer

To compute the probability $p_0(t)$ the chain is in the "bad" state at time t we formulate a differential equation in $p_0(t)$ which we solve. For any time $t \geq 0$ and $h > 0$ small enough:

$$p_0(t+h) = p_0(t)p_{00}(h) + p_1(t)p_{10}(h)$$

$$= p_0(t)\left[1 - \lambda_{bg} + o(h)\right] + p_1(t)\left[1 - \lambda_{gb} + o(h)\right] \Rightarrow$$

$$\frac{p_0(t+h) - p_0(t)}{h} = -\lambda_{bg}p_0(t) + \lambda_{gb}p_1(t) + \frac{o(h)}{h}\left(p_0(t) + p_1(t)\right)$$

Letting $h \downarrow 0$ and using $p_0(t) + p_1(t) = 1$ for all $t \geq 0$, we have:

$$\frac{d}{dt}p_0(t) = -(\lambda_{bg} + \lambda_{gb})p_0(t) + \lambda_{gb}, \quad \text{for } t \geq 0$$

The solution to this differential equation is:

$$p_0(t) = \frac{\lambda_{gb}}{\lambda_{bg} + \lambda_{gb}} + \left(p_0(0) - \frac{\lambda_{gb}}{\lambda_{bg} + \lambda_{gb}}\right)e^{-(\lambda_{bg}+\lambda_{gb})t}$$

for $t \geq 0$ and some initial probabilities $p_0(0)$ and $p_1(0)$ for the chain to be in the "bad" and the "good" state respectively, at time $t = 0$.

To compute the time the chain spends in each state $i \in \mathcal{P}$, we define the random variable T_i as the waiting time of P_t in state $i \in \mathcal{P}$ given the process is in state i. If $P\{T_i \geq t\} = G_i(t)$ and $h \geq 0$ small enough, then for $t \geq 0$,

$$\begin{aligned} G_i(t+h) &= P\{T_i \geq t+h\} \\ &= P\{T_i \geq t\}P\{T_i \geq h\} \\ &= G_i(t)\left(p_{ii}(h) + o(h)\right) \\ &= G_i(t)\left(1 - \nu_i h\right) + o(h) \Rightarrow \end{aligned}$$

$$\frac{G_i(t+h) - G_i(t)}{h} = -\nu_i G_i(t)$$

where

$$\nu_i = \begin{cases} \lambda_{bg}, & \text{if } i = 0 \\ \lambda_{gb}, & \text{if } i = 1 \end{cases}$$

By letting $h \downarrow 0$, we get

$$\frac{d}{dt}G_i(t) = -\nu_i G_i(t), \quad \text{for } t \geq 0$$

Setting $G_i(0) = 1$, we get

$$G_i(t) = e^{-\nu_i t}, \quad t \geq 0$$

That means, the waiting time in each state is exponentially distributed and more specifically,

$$P\{T_0 \geq t\} = e^{-\lambda_{bg} t}, \quad t \geq 0$$

$$P\{T_1 \geq t\} = e^{-\lambda_{gb} t}, \quad t \geq 0$$

A.2 MAC Layer

We first compute the characteristic function of the random variable $\sum_{j=1}^{K} X_j$ in Section 2.3.2.

$$\begin{aligned}
\mathrm{E}\left[e^{is\sum_{j=1}^{K} X_j}\right] &= \sum_{k=0}^{+\infty} \mathrm{E}\left[e^{is\sum_{j=1}^{K} X_j} \mid K = k\right] P\{K = k\} \\
&= \sum_{k=0}^{+\infty} \mathrm{E}\left[e^{is\sum_{j=1}^{k} X_j}\right] P\{K = k\} \\
&= \sum_{k=0}^{+\infty} \left(\frac{\lambda_{ret}}{\lambda_{ret} - is}\right)^k (1 - p_{mac})^k \, p_{mac} \\
&= \frac{\lambda_{ret} - is}{\lambda_{ret} p_{mac} - is} p_{mac}
\end{aligned}$$

where we used the fact that the random variable K is geometrically distributed with p.m.f.

$$P\{K = k\} = (1 - p_{mac})^k \, p_{mac}, \; k = 0, 1, 2, \ldots$$

The characteristic function of the random variable $D^{MAC} = T_p + \sum_{j=1}^{K} X_j$ can then be computed:

$$\begin{aligned}
\mathrm{E}\left[e^{isD^{MAC}}\right] &= \mathrm{E}\left[e^{is\left(T_p + \sum_{j=1}^{K} X_j\right)}\right] \\
&= \mathrm{E}\left[e^{isT_p} \cdot e^{is\sum_{j=1}^{K} X_j}\right] \\
&= e^{isT_p} \, \mathrm{E}\left[e^{is\sum_{j=1}^{K} X_j}\right] \\
&= \frac{\lambda_{ret} - is}{\lambda_{ret} p_{mac} - is} p_{mac} e^{isT_p} \\
&= p_{mac} e^{isT_p} + (1 - p_{mac}) \frac{\lambda_{ret} p_{mac}}{\lambda_{ret} p_{mac} - is} e^{isT_p}
\end{aligned}$$

A.3 Negative Binomial Distribution

Let $K_1, K_2, \ldots, K_n$ be geometrically distributed i.i.d. random variables with parameter p_{mac}, such that

$$P\{K_i = k\} = p_{mac}(1 - p_{mac})^k,\ k = 0, 1, 2, \ldots$$

for $i = 1, 2, \ldots, n$. The characteristic function for K_i is $\mathrm{E}\big[e^{isK_i}\big] = \frac{p_{mac}}{1-(1-p_{mac})e^{is}}$ for $i = 1, 2, \ldots, n$.

If $K = \sum_{j=1}^{n} K_j$, then the characteristic function of the random variable K can easily be shown to be

$$\mathrm{E}\big[e^{isK}\big] = \left(\frac{p_{mac}}{1-(1-p_{mac})e^{is}}\right)^n$$

which corresponds to the negative binomial distribution with parameters n, p_{mac} and p.m.f.

$$P\{K = k\} = \binom{n+k-1}{k} p_{mac}^n (1 - p_{mac})^k,\ k = 0, 1, 2, \ldots$$

A.4 Thinning

If D is the generic random variable representing the interarrival time between points in the thinned point process at the physical layer, then

$$D = \sum_{j=1}^{K} D_j^{MAC}$$

where K is geometrically distributed with pmf,

$$P\{K = k\} = \pi_g(1 - \pi_g)^{k-1}, \quad k = 1, 2, \ldots$$

Using the same methodology as in Appendix A.2, the characteristic function of D can be computed,

$$\begin{aligned}
\mathrm{E}\big[e^{isD}\big] &= \mathrm{E}\big[e^{is\sum_{j=1}^{K} D_j^{MAC}}\big] \\
&= \sum_{k=1}^{+\infty} \mathrm{E}\big[e^{is\sum_{j=1}^{K} D_j^{MAC}} \mid K = k\big] P\{K = k\} \\
&= \sum_{k=1}^{+\infty} \Big(\mathrm{E}\big[e^{isD^{MAC}}\big]\Big)^k \pi_g (1 - \pi_g)^{k-1} \\
&= \frac{\pi_g \mathrm{E}\big[e^{isD^{MAC}}\big]}{1 - (1 - \pi_g)\mathrm{E}\big[e^{isD^{MAC}}\big]}
\end{aligned}$$

Appendix B

Appendix to Chapter 3

This appendix provides proofs to intermediate results needed in Chapter 3.

B.1 Computation of $\mathrm{E}\left[\sum_{j=1}^{K} X_j\right]$

From (2.10) of Chapter 2 we have that

$$\begin{aligned}
\phi(s) &= \mathrm{E}\left[e^{is\sum_{j=1}^{K} X_j}\right] \\
&= p_{mac}^{n} \sum_{k=0}^{n} \binom{n}{k} \left(\frac{1-p_{mac}}{p_{mac}}\right)^{k} \left(\frac{\lambda_{ret}p_{mac}}{\lambda_{ret}p_{mac}-is}\right)^{k}
\end{aligned}$$

We know that $\mathrm{E}\left[\sum_{j=1}^{K} X_j\right] = \frac{1}{i}\frac{d}{ds}\phi(s)|_{s=0}$. But

$$\begin{aligned}
\frac{d}{ds}\phi(s) &= p_{mac}^{n} \sum_{k=0}^{n} \binom{n}{k} \left(\frac{1-p_{mac}}{p_{mac}}\right)^{k} k \left(\frac{\lambda_{ret}p_{mac}}{\lambda_{ret}p_{mac}-is}\right)^{k} \frac{\lambda_{ret}p_{mac}i}{(\lambda_{ret}p_{mac}-is)^{2}} \\
&= ip_{mac}^{n} \sum_{k=0}^{n} \binom{n}{k} k \left(\frac{1-p_{mac}}{p_{mac}}\right)^{k} \frac{(\lambda_{ret}p_{mac})^{k+1}}{(\lambda_{ret}p_{mac}-is)^{k+2}}
\end{aligned}$$

Then,

$$\mathrm{E}\left[\sum_{j=1}^{K} X_j\right] = \frac{1}{i}\frac{d}{ds}\phi(s)|_{s=0} \tag{B-1}$$

$$= p_{mac}^{n} \sum_{k=0}^{n} \binom{n}{k} k \left(\frac{1-p_{mac}}{p_{mac}}\right)^{k} \frac{1}{\lambda_{ret}p_{mac}} \tag{B-2}$$

$$= p_{mac}^{n} \frac{1}{\lambda_{ret}p_{mac}} S(n, p_{mac}) \tag{B-3}$$

where $S(n,p) = \sum_{k=0}^{n} \binom{n}{k} k \left(\frac{1-p_{mac}}{p_{mac}}\right)^{k}$.

Bibliography

[1] ABOUZEID, A. A., ROY, S., AND AZIZOGLU, M. Comprehensive performance analysis of a TCP session over a wireless fading link with queueing. *IEEE Transactions on Wireless Communications 2*, 2 (Mar. 2003), 344–356.

[2] ABRAMSON, N. The aloha system-another alternative for computer communications. In *AFIPS Conference Proceedings* (Nov. 1970), vol. 37, pp. 281–285.

[3] ANJUM, F., AND TASSIULAS, L. Comparative study of various TCP versions over a wireless link with correlated losses. *IEEE/ACM Transactions on Networking 11*, 3 (June 2003), 370–383.

[4] AVRACHENKOV, K. E., KHERANI, A. A., VILCHEVSKY, N. O., AND ZABOROVSKI, V. S. Optimal tuning of the TCP retransmission timeout for small-BDP lossy wireless networks. In *Proceedings of the* 6^{th} *ITC Specialist Seminar on Performance Evaluation of Wireless and Mobile Systems* (Antwerp, Belgium, Sept. 2004).

[5] BACCELLI, F., BŁASZCZYSZYN, B., AND MÜHLETHALER, P. A spatial reuse Aloha MAC protocol for multihop wireless mobile networks. Tech. Rep. INRIA/RR-4955, INRIA, Oct. 2003.

[6] BACCELLI, F., BŁASZCZYSZYN, B., AND MÜHLETHALER, P. An Aloha protocol for multihop mobile wireless networks. *IEEE Transactions on Information Theory 52*, 2 (Feb. 2006), 421–436.

[7] BACCELLI, F., BŁASZCZYSZYN, B., AND MÜHLETHALER, P. Stochastic analysis of spatial and opportunistic Aloha. http://hal.inria.fr/inria-00360800/en/, May 2009.

[8] BACCELLI, F., BŁASZCZYSZYN, B., AND MÜHLETHALER, P. Stochastic analysis of spatial and opportunistic Aloha. *IEEE Journal on Selected Areas in Communications 27*, 7 (Sept. 2009), 1105–1119.

[9] BACCELLI, F., AND MCDONALD, D. R. Mellin transforms for TCP throughput with applications to cross layer optimization. In *Proceedings of the Conference on Information Sciences and Systems (CISS) 2006* (Princeton, NJ, Mar. 2006).

[10] BAIOCCHI, A., AND VACIRCA, F. TCP fluid modeling with a variable capacity bottleneck link. In *Proceedings of the IEEE Conference on Computer Communications, (INFOCOM 2007)* (Anchorage, AK, May 2007).

[11] BANSAL, S., GUPTA, R., SHOREY, R., ALI, I., RAZDAN, A., AND MISRA, A. Energy efficiency and throughput for TCP traffic in multi-hop wireless networks. In *Proceedings of the Conference on Computer Communications, (INFOCOM 2002)* (New York, NY, June 2002), pp. 210–219.

[12] BANSAL, S., SHOREY, R., AND KHERANI, A. A. Performance of TCP and UDP protocols in multi-hop multi-rate wireless networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference, (WCNC 2004)* (Atlanta, GA, Mar. 2004).

[13] BARAS, J. S., TABATABAEE, V., PAPAGEORGIOU, G., AND RENTZ, N. Modelling and optimization for multi-hop wireless networks using fixed point and automatic differentiation. In *Proceedings of the* 6^{th} *Intl. Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt 2008)* (Berlin, Germany, Mar. 2008).

[14] BARAS, J. S., TABATABAEE, V., PAPAGEORGIOU, G., AND RENTZ, N. Performance metric sensitivity computation for optimization and trade-off analysis in wireless networks. In *Proceedings of the IEEE Global Communications Conference (IEEE GLOBECOM 2008)* (New Orleans, LA, Nov. 2008).

[15] BERTSEKAS, D., AND GALLAGER, R. *Data Networks*, second ed. Prentice-Hall, Englewood Cliffs, NJ, 1992.

[16] BIANCHI, G. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications 18*, 3 (Mar. 2000), 535–547.

[17] BRÉMAUD, P. *Point Processes and Queues, Martingale Dynamics*. Springer-Verlag, New York, 1981.

[18] BRUNO, R., CONTI, M., AND GREGORI, E. An accurate closed-form formula for the throughput of a long-lived TCP connection in IEEE 802.11 WLANs. *Computer Networks 52*, 1 (Jan. 2008), 199–212.

[19] BRUNO, R., CONTI, M., AND GREGORI, E. Throughput analysis and measurements in IEEE 802.11 WLANs with TCP and UDP traffic flows. *IEEE Transactions on Mobile Computing 7*, 2 (Feb. 2008), 171–186.

[20] BUDHIRAJA, A., HERNANDEZ-CAMPOS, F., KULKARNI, V. G., AND SMITH, F. D. Stochastic differential equation for TCP window size. *Probability in the Engineering and Informational Sciences 18*, 1 (Jan. 2004), 111–140.

[21] ELLIOTT, E. O. Estimates of error rates for codes on burst-noise channels. *Bell System Technical Journal 42*, 3 (Sept. 1963), 1977–1997.

[22] GILBERT, E. N. Capacity of a burst-noise channel. *Bell System Technical Journal 39*, 3 (Sept. 1960), 1253–1265.

[23] HANBALI, A. A., ALTMAN, E., AND NAIN, P. A survey of TCP over ad hoc networks. *IEEE Commun. Surveys and Tutorials 7*, 3 (July 2005), 22–36.

[24] HARTWELL, J. A., AND FAPOJUWO, A. O. Modeling and characterization of frame loss process in IEEE 802.11 wireless local area networks. In *Proceedings of the Vehicular Technology Conference, (VTC 2004)* (Los Angeles, CA, Sept. 2004), pp. 4481–4485.

[25] HIRA, M. M., TOBAGI, F. A., AND MEDEPALLI, K. Throughput analysis of a path in an IEEE 802.11 multihop wireless network. In *Proceedings of the IEEE Wireless Communications and Networking Conference, (WCNC 2007)* (Hong Kong, China, Mar. 2007).

[26] KHERANI, A. A., AND SHOREY, R. Performance improvement of TCP with delayed acks in IEEE 802.11 wireless LANs. In *Proceedings of the IEEE Wireless Communications and Networking Conference, (WCNC 2004)* (Atlanta, GA, Mar. 2004).

[27] KHERANI, A. A., AND SHOREY, R. Throughput analysis of TCP in multi-hop wireless networks with IEEE 802.11 MAC. In *Proceedings of the IEEE Wireless Communications and Networking Conference, (WCNC 2004)* (Atlanta, GA, Mar. 2004).

[28] KUSHNER, H. J. Numerical methods for stochastic control problems in continuous time. *SIAM Journal on Control and Optimization 28*, 5 (Sept. 1990), 999–1048.

[29] KUSHNER, H. J. Numerical methods for stochastic control problems in finance. In *Mathematics of Derivative Securities*, M. A. H. Dempster and S. R. Pliska, Eds., vol. 16. Cambridge University Press, 1997, pp. 504–527.

[30] KUSHNER, H. J., AND DIMASI, G. Approximations for functionals and optimal control problems on jump diffusion processes. *Journal of Mathematical Analysis and Applications 63*, 3 (May 1978), 772–800.

[31] KUSHNER, H. J., AND DUPUIS, P. *Numerical Methods for Stochastic Control Problems in Continuous Time*, second ed. Springer, 2001.

[32] LAKSHMAN, T. V., MADHOW, U., AND SUTER, B. TCP/IP performance with random loss and bidirectional congestion. *IEEE/ACM Transactions on Networking 8*, 5 (Oct. 2000), 541–555.

[33] LEUNG, K.-C., AND LI, V. O. Transmission control protocol (TCP) in wireless networks: Issues, approaches, and challenges. *IEEE Commun. Surveys and Tutorials 8*, 4 (Sept. 2006), 64–79.

[34] LIPTSER, R. S., AND SHIRYAEV, A. N. *Statistics of Random Processes, II. Applications*, second ed. Springer-Verlag, 2001.

[35] LIU, J., STOLYAR, A. L., CHIANG, M., AND POOR, H. V. Queue back-pressure random access in multihop wireless networks: Optimality and stability. *IEEE Transactions on Information Theory 55*, 9 (Sept. 2009), 4087–4098.

[36] Low, S. H., Paganini, F., and Doyle, J. C. Internet congestion control. *IEEE Control Systems Magazine 22*, 1 (Feb. 2002), 28–43.

[37] Ma, L., Barner, K. E., and Arce, G. R. Statistical analysis of TCP's retransmission timeout algorithm. *IEEE/ACM Transactions on Networking 14*, 2 (Apr. 2006), 383–396.

[38] Ma, R. T. B., Misra, V., and Rubenstein, D. An analysis of generalized slotted-aloha protocols. *IEEE/ACM Transactions on Networking 17*, 3 (June 2009), 936–949.

[39] Marsan, M. A., Garetto, M., Giaccone, P., Leonardi, E., Schiattarella, E., and Tarello, A. Using partial differential equations to model TCP mice and elephants in large IP networks. *IEEE/ACM Transactions on Networking 13*, 6 (Dec. 2005), 1289–1301.

[40] Mascolo, S., Casetti, C., Gerla, M., Sanadidi, M. Y., and Wang, R. TCP Westwood: Bandwidth estimation for enhanced transport over wireless links. In *Proceedings of the ACM Mobicom 2001* (Rome, Italy, July 2001), pp. 287–297.

[41] Matthis, M., Semke, J., Mahdavi, J., and Ott, T. The macroscopic behavior of the TCP congestion avoidance algorithm. *ACM SIGCOMM Computer Communication Review 27*, 3 (July 1997), 67–82.

[42] McDougall, J., and Miller, S. Sensitivity of wireless network simulations to a two-state markov model channel approximation. In *Proceedings of the IEEE Global Telecommunications Conference, (GLOBECOM 2003)* (San Francisco, CA, Dec. 2003), pp. 697–701.

[43] Medepalli, K., and Tobagi, F. A. Towards performance modeling of IEEE 802.11 based wireless networks: A unified framework and its applications. In *Proceedings of the IEEE Conference on Computer Communications, (INFOCOM 2006)* (Barcelona, Spain, Apr. 2006).

[44] Misra, A., Baras, J., and Ott, T. Generalized TCP congestion avoidance and its effect on bandwidth sharing and variability. In *Proceedings of Global Telecommunications Conference, 2000 (GLOBECOM 2000)* (Nov. 2000), pp. 329–337.

[45] Misra, A., Ott, T., and Baras, J. The window distribution of multiple TCPs with random loss queues. In *Proceedings of Global Telecommunications Conference, 1999 (GLOBECOM 1999)* (Dec. 1999), pp. 1714–1726.

[46] Misra, A., and Ott, T. J. The window distribution of idealized TCP congestion avoidance with variable packet loss. In *Proceedings of the Conference on Computer Communications, (INFOCOM 1999)* (Mar. 1999), pp. 1564–1572.

[47] Padhye, J., Firoiu, V., Towsley, D. F., and Kurose, J. F. Modeling TCP Reno performance: A simple model and its empirical validation. *IEEE/ACM Transactions on Networking 8*, 2 (Apr. 2000), 133–145.

[48] PAXSON, V., AND ALLMAN, M. *Computing TCP's Retransmission Timer*. RFC2988, Nov. 2000.

[49] PESKIR, G., AND SHIRYAEV, A. *Optimal Stopping and Free-Boundary Problems*. Birkhauser, 2006.

[50] PUTERMAN, M. L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley-Interscience, 2005.

[51] ROBERTS, L. G. Aloha packet system with and without slots and capture. *ACM SIGCOMM Computer Communications Review 5*, 2 (Apr. 1975), 28–42.

[52] SHARMA, G., GANESH, A., AND KEY, P. Performance analysis of contention based medium access control protocols. *IEEE Transactions on Information Theory 55*, 4 (Apr. 2009), 1665–1682.

[53] SNELL, J. L. Applications of martingale system theorems. *Transactions of the American Mathematical Society 73*, 2 (Sept. 1952), 293–312.

[54] THE NETWORK SIMULATOR. ns-2. http://nsnam.isi.edu/nsnam/index.php/Main_Page.

[55] WANG, J., LI, L., LOW, S. H., AND DOYLE, J. C. Cross-layer optimization in TCP/IP networks. *IEEE/ACM Transactions on Networking 13*, 3 (June 2005), 582–595.

[56] YAJNIK, M., MOON, S., KUROSE, J., AND TOWSLEY, D. Measurement and modelling of the temporal dependence in packet loss. In *Proceedings of the Conference on Computer Communications, (INFOCOM 1999)* (New York, NY, Mar. 1999), pp. 345–352.