Performance Analysis and Enhancement of MAC Protocols

Chuan Heng Foh

B.Sc. , M.Sc.

A Thesis submitted in total fulfillment of the requirements of the degree of

Doctor of Philosophy

Department of Electrical and Electronic Engineering

November 2002

(Produced on acid-free paper)

Abstract

The challenge of designing an efficient Medium Access Control (MAC) protocol and analyzing it has been an important research topic for over 30 years. This thesis focuses on the performance analysis and enhancement of MAC protocols, particularly the random access protocols. Performance of the two widely known random access MAC protocols – the IEEE 802.3 MAC protocol that employs the Carrier Sense Multiple Access with Collision Detection (CSMA/CD) protocol, and the IEEE 802.11 MAC protocol that uses the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol, are first studied. They are analyzed using a new analytical approach proposed in this thesis. The new approach is based on the idea that the service process of a MAC protocol can be modeled by a Phase-Type (PH) distribution. This way, the arrival process as well as the service process of the actual protocol can be described by a certain multidimensional continuous time Markov chain. The advantages of this novel technique over the traditional approach for performance analyses of MAC protocols are that: (i) it provides a unified model for the analysis of MAC protocols; (ii) it significantly simplifies the analytical model of a MAC protocol which makes it possible to include a more complex and realistic traffic model without compromising the protocol details; (iii) vast knowledge is available on the analysis of a continuous time Markov chain, which allows for more insight to the performance of a MAC protocol. Extensive demonstrations of the use of the approach on performance analyses of MAC protocols are provided, some of which conducted under realistic bursty traffic conditions.

As a result of the performance analyses, it is found that the IEEE 802.3 MAC protocol exhibits relatively low and unattractive performance, especially when it is operated at 1Gb/s data rate. In response to this finding, two new protocols are introduced in this thesis. Firstly, we introduce a novel

technique, *Reservations by Interruptions*, to provide an efficient reservation scheme for CSMA/CD. The resulting protocol is named CSMA with Reservations by Interruptions (CSMA/RI). Performance of CSMA/RI is evaluated and compared with CSMA/CD. The stability of CSMA/RI is also studied. In addition, two realistic scenarios, namely the saturation and disaster scenarios, are used to demonstrate the performance advantage of CSMA/RI. Furthermore, performance analyses based on the new approach under realistic traffic conditions are performed to show the performance benefit of CSMA/RI. Our analytical results show that CSMA/RI always offers better performance than CSMA/CD, and in some cases, the delay performance of CSMA/RI approaches that of a perfect scheduling G/D/1 system. A performance comparison between CSMA/CD, CSMA/RI as well as the token ring protocol is provided. Finally, some implementation issues and limitations of CSMA/RI are addressed.

The second MAC protocol introduced in this thesis is called the *Request Contention Multiple Access* (RCMA) protocol. RCMA is a distributed gigabit MAC protocol. It is designed to operate in the passive star optical network such as the 10BASE-FP version of Ethernet at a gigabit data rate. Unlike the existing IEEE 802.3z Gigabit Ethernet MAC protocol that employs CSMA/CD, RCMA is efficient and stable for a wide range of user numbers based on our study under the saturation scenario. Moreover, RCMA can easily accommodate service differentiation within the MAC layer with no additional overhead. In terms of implementation, it does not appear to be difficult. The implementation of RCMA may lead to a cost competitive yet efficient solution for the future gigabit LANs.

Declaration

I hereby certify that this thesis is my own work, except where due reference is made in the text and that, to my best knowledge and belief, it has not been submitted to this university or to any other university or institution for a degree.

Signed

Chuan Heng Foh

11th November, 2002

Acknowledgments

I would like to thank my thesis supervisor, Professor Moshe Zukerman, for dedicating his knowledge, unreserved encouragement and support throughout this PhD. It was his inspiring course in Network Design that gave me a starting point for this study.

I would also like to thank the members of my PhD examination committee for their valuable time and advice.

Finally, and always, I thank my lovely wife, Yuk Yee Leung. Without her love, dedication and devotion, none of these is ever possible.

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

With the rapid development of the Internet in the last decade, the role of Local Area Networks (LANs) for users to access the Internet has become more important. Two architectures have been used in developing a LAN: the switched architecture mainly implemented in the backbone connecting several individual LANs, and the shared architecture based on a particular Medium Access Control (MAC) protocol used extensively in providing network access for a collection of end users.

As the network access part plays a vital role in a network, it is important to design an efficient yet reliable MAC protocol for a LAN. Many MAC protocols used in LANs are based on random access schemes, for example Ethernet that implements the *Carrier Sense Multiple Access with Collision Detection* (CSMA/CD) protocol, and the wireless LAN that uses the *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA). One unique property of a random access protocol is its randomness of the access scheme that leads to a possible collision of a transmission making the transmission unsuccessful. Such an unsuccessful transmission requires retransmissions that may suffer further collisions. Because of the collisions, the performance of a random access protocol can be seriously degraded.

As a result, the performance evaluation of a random access protocol is necessary in the process of protocol design, development and deployment. Due to the random nature of the protocol operation and the bursty nature of the traffic, it has been considered difficult to analyze a random access protocol with all protocol details under a realistic traffic condition. Performance analyses of random access protocols usually rely on simplifications of protocol details as well as arrival traffic models. Hence,

the scope of the results of such performance analyses is somewhat limited. One of the goals of this thesis is to propose a new analytical approach for the performance analysis of a MAC protocol. The approach should not only retain all protocol details of the protocol operation, but also allow the use of a more general and realistic arrival process. The use of the approach will lead to more accurate performance results of the protocol.

Besides, the recent growth in research, development and deployment of multimedia multi-service networks has prompted a need for a MAC protocol that can cope with the growing bandwidth demands as well as feature service differentiation to support transmission of real-time multimedia traffic with a certain quality of service (QoS). However, it is found that Ethernet that employs CSMA/CD is unable to fulfill this requirement. As another goal of this dissertation, two new protocols are introduced and analyzed, they are the *CSMA with Reservations by Interruptions* (CSMA/RI) protocol and the *Request Contention Multiple Access* (RCMA) protocol.

In brief, CSMA/RI introduces a novel reservation technique, Reservations by Interruptions, on top of the CSMA/CD operation to reduce the chances of collisions. By reducing the collision probability, higher channel utilization and better performance can be achieved. Since CSMA/RI itself uses a reservation scheme, thus the support of a priority operation, which enables service differentiation, is straightforward.

Unfortunately, at a very high data rate such as 1Gb/s, both the CSMA/CD and the CSMA/RI protocols suffer significant performance degradation. As a result, we develop the RCMA protocol that is tailored for 1Gb/s data rate LANs. Similar to CSMA/CD that has been chosen for shared Gigabit Ethernet LANs, RCMA is based on a distributed control principle. In addition, it achieves efficient scheduling and fairness with minimum overhead and complexity. Unlike CSMA/CD, RCMA is efficient and stable even with a reasonably large population of users. Moreover, it can easily accommodate a priority operation that enables service differentiation.

1.2 Overview

This thesis deals with the performance analyses and enhancements of MAC protocols, particularly the random access protocols. Examinations of the existing random access MAC protocols as well as their performance analyses are first provided. This dissertation then presents a new approach for performance analyses to achieve more accurate and realistic performance results. In an attempt to enhance the existing standard MAC protocols, this thesis introduces two new random access protocols. Details of the protocol operations and their performance are also given. The organization of this thesis is presented in the following.

Chapter 2 reviews several popular random access protocols. They include Aloha, slotted Aloha, CSMA and CSMA/CD protocols. Along with the reviews, the current efforts in performance analyses of the protocols are provided. According to the analytical results, these protocols exhibit an unstable phenomenon. To stabilize the channel of these protocols, it is necessary to implement an appropriate retransmission algorithm. Several important retransmission algorithms are described in this chapter, and their performance analyses are also provided. Two widely known MAC protocols, the IEEE 802.3 and the IEEE 802.11 MAC protocols that are based on a particular random access protocol and a certain retransmission algorithm are then reviewed. Since all the presented performance analyses introduce certain assumptions to simplify the protocol operation and/or the arrival process, the obtained results may not be realistic. In the final section of this chapter, we discuss implications of the assumptions and the scope of applications of the obtained results.

Chapter 3 begins with describing two realistic scenarios, namely the saturation and the disaster scenarios, that may occur in a LAN. The analyses of two widely known MAC protocols, the IEEE 802.3 and the IEEE 802.11 MAC protocols based on CSMA/CD and CSMA/CA respectively are presented under these two realistic scenarios. The applications of the results are also described. Moreover, to further provide insight to the protocol

performance under a certain statistical arrival process, in this chapter, a new approach for performance analysis of a random access protocol is introduced. In contrast to the analyses reviewed in Chapter 2, the approach allows a more complex arrival process without the need of simplifying any protocol detail. Extensive examples on performance analyses of the IEEE 802.3 and the IEEE 802.11 MAC protocols are provided under various arrival processes to demonstrate the accuracy and the versatility of the approach. Many performance results of the two protocols under some realistic traffic conditions including the bursty Markov Modulated Poisson Process (MMPP) traffic are given.

Chapter 4 presents a new protocol named CSMA/RI that uses a novel reservation scheme to enhance the performance of CSMA/CD. The performance of CSMA/RI is analyzed and compared with its counterpart, CSMA/CD. The stability of CSMA/RI is then demonstrated. A performance comparison of various MAC protocols including the token ring protocol is also given. This chapter is finally concluded by addressing issues related to the implementations, and the boundary of the application of CSMA/RI.

Chapter 5 firstly addresses the impact of a high data rate on the performance of a MAC protocol, illustrating by using the MAC protocol of Gigabit Ethernet. By understanding the characteristics of gigabit LANs, we design a new protocol for high data rate LANs such as gigabit LANs, called RCMA. It is efficient, robust, capable of carrying multimedia traffic with a certain QoS, and most importantly easy to implement. The detail operation of RCMA is given and its performance is studied. A performance comparison between RCMA and CSMA/CD is also illustrated to show the performance benefit of RCMA. Finally, limitations and extensions of RCMA are discussed.

Chapter 6 concludes this dissertation by summarizing the contributions and highlighting important results obtained in this thesis.

1.3 Contributions

The contribution of this thesis is mainly threefold. Firstly, a new approach for performance analyses of a random access protocol is introduced. Performance analyses of a random access protocol with all protocol details are made possible by this approach. Furthermore, the approach facilitates a more complex arrival process so that a random access protocol can be analyzed under some realistic traffic conditions. As a result of the approach, many new analytical results of some important MAC protocols are produced.

Secondly, a new reservation scheme is proposed. It is demonstrated by enhancing CSMA/CD. The resulting protocol is named CSMA/RI. Because of this reservation scheme, the performance of the protocol is significantly improved. Besides, the proposed new scheme supports a priority operation with no additional overhead, enabling service differentiation that is difficult to achieve in CSMA/CD.

Thirdly, this thesis presents a newly invented protocol, named RCMA, for gigabit LANs. RCMA inherits many properties of random access protocols such as simplicity, easy of implementation, distributed such that no intelligent central controller is needed, and cost competitive when implemented. Unlike many random access protocols, including CSMA/CD which has been chosen for the Gigabit Ethernet MAC protocol, RCMA is very efficient, and stable even when operated in a LAN that consists of a reasonably large number of users. Moreover, it features a priority operation without additional overhead. RCMA offers a cost competitive solution for future gigabit LANs.

The detail of the contributions of this thesis is listed in the following in approximate order of appearance.

1. Saturation and disaster scenario analysis for the IEEE 802.3 MAC protocol (see Section 3.2) [FoZu01b].

2. Disaster scenario analysis for the IEEE 802.11 MAC protocol (see Section 3.3.2) [FoZu01a].
3. Proposal of a novel approach for performance analysis for random access protocols (see Section 3.4) [FoZu02a, FoZu02b].
4. Performance analyses for the IEEE 802.3 MAC protocol under various realistic arrival models (see Section 3.5) [FoZu02b].
5. Performance analysis for the IEEE 802.11 MAC protocol under various realistic arrival models (see Section 3.6) [FoZu02a, FoZu02b].
6. Proposal of a new reservation scheme, Reservations by Interruptions, for CSMA/CD (see Section 4.1) [FoZu00a, FoZu00b].
7. Saturation and disaster throughput analysis for CSMA/RI with Binary Exponential Backoff retransmission algorithm (see Section 4.2) [FoZu01b].
8. Performance analyses for CSMA/RI under various realistic arrival models (see Section 4.3).
9. Saturation throughput analysis for the Gigabit Ethernet MAC protocol (see Section 5.1.2).
10. Invention of a new protocol, named RCMA, for gigabit LANs (see Section 5.2) [FoZu02c].
11. Saturation throughput analysis for RCMA (see Section 5.2.3) [FoZu02c].

1.4 Publications

Publications due to the work presented in this thesis are listed as follows.

[FoZu00a] C. H. Foh and M. Zukerman, “Improving the Efficiency of CSMA using Reservations by Interruptions,” in *Proc. IEEE ICC ‘00*, June 2000.

[FoZu00b] C. H. Foh and M. Zukerman, "CSMA with Reservations by Interruptions (CSMA/RI): A Novel Approach to Reduce Collisions in CSMA/CD," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 9, pp. 1572-1580, September 2000.

[FoZu01a] C. H. Foh and M. Zukerman, "Performance Evaluation of IEEE 802.11" in *Proc. IEEE VTC '01*, May 2001.

[FoZu01b] C. H. Foh and M. Zukerman, "Performance Comparison of CSMA/RI and CSMA/CD with BEB," in *Proc. IEEE ICC '01*, June 2001.

[FoZu02a] C. H. Foh and M. Zukerman, "Performance Analysis of the IEEE 802.11 MAC Protocol," in *Proc. European Wireless '02*, February 2002.

[FoZu02b] C. H. Foh and M. Zukerman, "A New Technique for Performance Evaluation of Random Access Protocols," in *Proc. IEEE ICC '02*, April 2002.

[FoZu02c] C. H. Foh and M. Zukerman, "A Novel and Simple MAC Protocol for High Speed Passive Optical LANs," in *Proc. Networking '02*, May 2002.

2 Random Access Protocols

Random access protocols are some of the most important and widely implemented protocols in local area networks. Due to their simple operation, their implementations are generally straightforward. Random access protocols are different from other types of MAC protocols in one property: a transmission is not guaranteed to be a successful one. This is mainly due to their random access operation, where a transmission is initiated without a prior negotiation with other possible transmissions. When two transmissions are commenced at the same time, a *collision* of transmissions occurs, and no useful information can be received and read correctly by receivers.

In the case of a collision, retransmissions are required for all unsuccessful transmissions until they are successful. To avoid further collisions of the same set of collided transmissions, the retransmission of an unsuccessful transmission is scheduled in a randomly chosen future time hoping that other retransmissions will not be scheduled to the same future time. This leads to the development of retransmission algorithms for random access protocols. A retransmission algorithm is responsible for arranging, in a distributive way, all transmissions involved in a collision to be retransmitted in different future time period so that the chances of further collisions can be minimized.

In the following sections, some important random access protocols and retransmission algorithms are revisited, and their performance analyses are reviewed. Sections 2.1 to 2.4 deal with Aloha, slotted Aloha, CSMA and CSMA/CD respectively. The instability of Aloha is then addressed in Section 2.5. It is also indicated that with a retransmission algorithm, the channel of a random access protocol can be stabilized. Two important retransmission algorithms are described, which are the Binary Tree

algorithm described in Section 2.6 and the Binary Exponential Backoff (BEB) algorithm covered in Section 2.7. Sections 2.8 and 2.9 provide reviews of the two popular random access type MAC protocols implemented in LANs – the IEEE 802.3 and the IEEE 802.11 MAC protocols. Finally in Section 2.10, the models and assumptions used in the performance analysis of the described random access protocols are discussed.

Throughout the protocol description, we use the term *station* to refer to a transmission source in a network. We further use the concept of a *ready station* to describe a station that has one or more data frames waiting for transmission.

2.1 Aloha

The concept of random access protocols was first introduced and demonstrated by Abramson and his colleagues at the University of Hawaii in 1970 [Abra70]. The protocol was named *Aloha*. It was intended to provide communications between several stations in a wireless environment.

The operation of Aloha is simple. When a station is ready for a data frame transmission, it transmits the data frame immediately. Since all stations share the same radio channel, when two or more stations commence their transmissions at the same time, a collision occurs such that all the collided data frame transmissions cannot be detected and read correctly.

In the case of a collision, all collided transmissions must be scheduled for retransmission at a randomly chosen future time. However, no retransmission algorithm is specified in Aloha. In other words, an Aloha station does nothing if its transmission suffers a collision. It is assumed that a protocol providing a reliable communication service residing in the upper layer of a protocol stack will discover a failure of the transmission some time after the collision and will then trigger a retransmission. The operation of an Aloha transceiver can be described by a finite state machine shown in

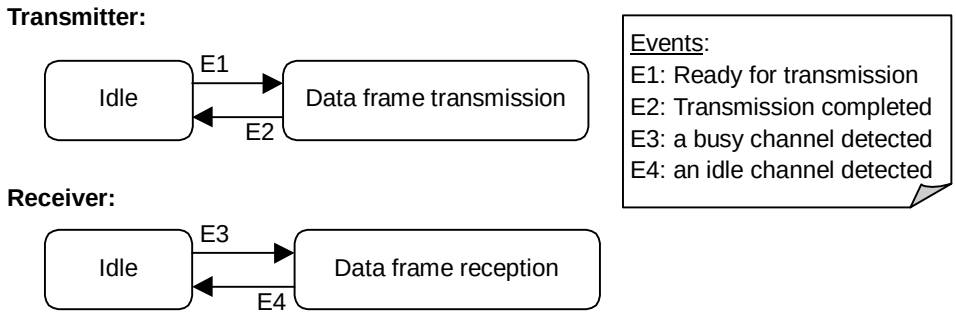


Figure 2.1: The finite state machine of an Aloha transceiver

Figure 2.1. It is worth noting that in the implementation, the transmitter and the receiver cannot be operated at the same time.

2.1.1 Throughput Analysis

Figure 2.2(a) presents an accurate model for the performance analysis of Aloha. As in Figure 2.2(a), the load traffic with arrival rate λ represents the aggregated traffic from individual stations. The load traffic is further merged with the retransmission traffic with arrival rate r before entering the Aloha system. The resulting traffic is called the attempt traffic with arrival rate g such that $g=\lambda+r$. To obtain realistic performance results of Aloha, it is required to (i) choose a realistic load traffic model, and (ii) accurately model both the Aloha protocol and the implemented retransmission algorithm.

Due to the randomness of the protocol operation, the performance analysis of Aloha with the above requirements is considered difficult. Two key assumptions are introduced to enable the performance analysis [KlTo75], they are listed as follows.

1. **Poisson arrivals**: Assume that the generation of data frames from all stations follows a Poisson process, and each station only holds at most one transmission in its buffer. Let the arrival rate of the Poisson process be λ data frames per second.

2. **Large retransmission time**: Assume that each collided data frame must be retransmitted at some later time in the future. Further

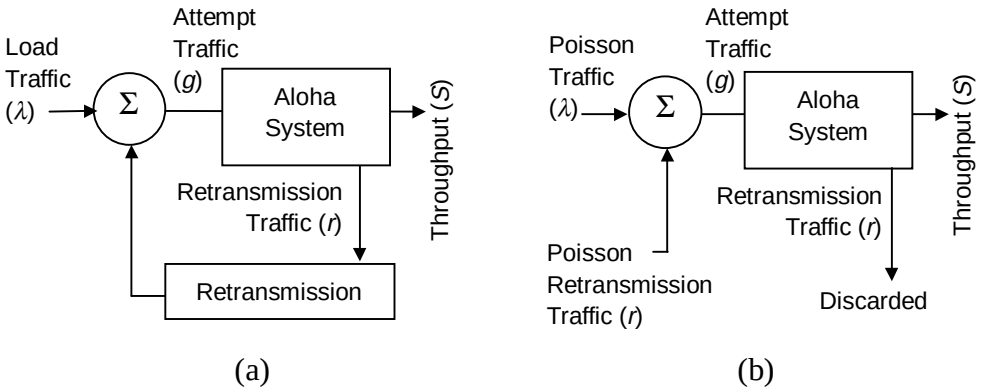


Figure 2.2: The model of Aloha for performance analysis: (a) the accurate model, (b) the simplified model

collisions will trigger additional retransmissions until that data frame is successfully transmitted. It is assumed that the chosen future time for a retransmission is very long compared to the data frame transmission time.

Figure 2.2(b) depicts the roles of the above assumptions in modeling and analyzing Aloha. Based on the first assumption, a new arrival represents a transmission from a new ready station. Since the arrival process is a Poisson process, and the number of arrivals given a certain period of time can be infinity with a positive probability, hence the first assumption implies that the number of stations in a network is infinite. It is shown that this assumption is a good approximation for a reasonable large, but not necessary infinite number of stations [KlLa75]. The review of the justification of this assumption is given in subsection 2.2.2.

Table 2.1: Summary of the key assumptions for performance analysis for Aloha

Assumption	Implication
Poisson Load Traffic	The number of stations in a network is infinite. Each station carries at most one data frame in its local buffer for transmissions.
Long Retransmission Time	Due to the long retransmission time, the correlation between the retransmission and the load traffic vanishes so that the retransmission traffic is independent to the load traffic. Moreover, it can be approximately described by Poisson traffic.

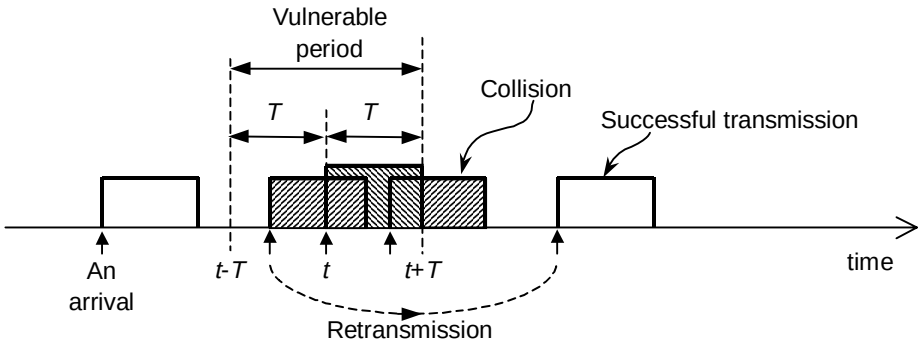


Figure 2.3: A snapshot of the Aloha channel

From the second assumption, it is found that the arrival process of the retransmission traffic can be approximately described by a Poisson process [Lam74]. The approximation vanishes when the retransmission time approaches infinity. Moreover, due to the long retransmission time, the correlation between the load and the retransmission traffic becomes very insignificant so that the retransmission traffic can be seen as independent Poisson traffic. The justification of this assumption is reviewed in subsection 2.2.3.

The summary of the two key assumptions is given in Table 2.1. Using these assumptions, the model for the performance analysis of Aloha can be simplified to the one presented in Figure 2.2(b), where the arrival process of the load traffic is a Poisson arrival process, and the retransmission traffic is also Poisson traffic that is independent to the load traffic.

Based on [KlTo75], the size of data frames are considered to be fixed and their transmission time is assumed to be T units of time. A transmission that appears on the radio channel can be a new data frame transmission from the load traffic, or a transmission from the Poisson retransmission traffic. If a data frame arrives at some time t as in Figure 2.3, this data frame transmission will not suffer a collision if no other data frames are transmitted within the *vulnerable period*. This vulnerable period is the interval that two data frame transmissions may overlap and interfere with each other. Since the transmission time of the data frame initiated at time t is

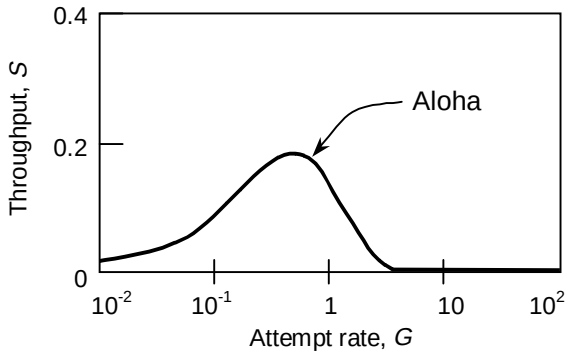


Figure 2.4: Throughput versus attempt rate for Aloha

T, the data frame will suffer a collision if at least one data frame transmission commences between *t*-*T* and *t*+*T*, hence the vulnerable period is 2*T*.

As in Figure 2.2(b), the arrival process of the attempt traffic is assumed to be a Poisson process with arrival rate *g*, thus the probability that *n* arrivals during the vulnerable period 2*T* is given by the Poisson distribution

$$\Pr\{X = n\} = \frac{(2T \cdot g)^n e^{-2T \cdot g}}{n!} \tag{2-1}$$

where *X* is a Poisson random variable. Given a transmission at some time *t*, to guarantee that it is a successful one, no other stations must commence their transmissions during that vulnerable period, thus by putting *n*=0 into Equation (2-1), we obtain the probability of a success transmission, P_{succ}, to be

$$P_{succ} = \Pr\{X = 0\} = \frac{(2T \cdot g)^0 e^{-2T \cdot g}}{0!} = e^{-2T \cdot g}. \tag{2-2}$$

Given that the arrival rate of the attempt traffic is *g*, hence the rate of successful data frame transmissions is $g \cdot P_{succ}$. Define the throughput, *S*, to be the fraction of time that the broadcast channel carries successful data frame transmissions, then the throughput *S* is

$$S = g \cdot P_{succ} \cdot T = gTe^{-2gT}. \tag{2-3}$$

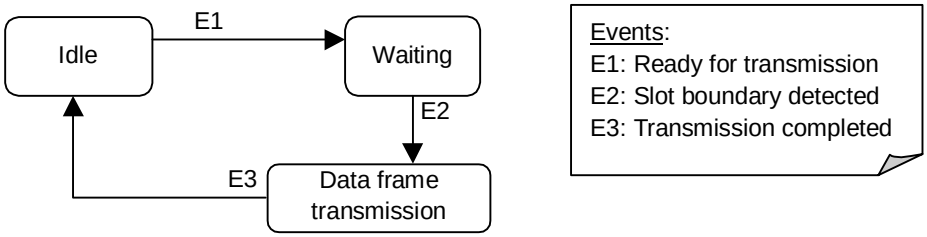


Figure 2.5: The finite state machine of a slotted Aloha transmitter

We further normalize g to the data frame transmission time T by defining G to be the normalized attempt rate, i.e. $G=gT$. By putting $G=gT$ into Equation (2-3), we get

$$S = Ge^{-2G}. \tag{2-4}$$

Figure 2.4 plots the throughput versus attempt rate for Aloha. Note that the throughput for the Aloha channel is at its peak value of $S=1/2e$ or around 0.18 at $G=0.5$. When G is smaller than 0.5, the system is not fully utilized so that the channel is dominated by idle periods. In contrast, when G is above 0.5, the system is overloaded, too many collisions occur and thus the throughput drops below its peak.

2.2 Slotted Aloha

An effort to improve the performance of Aloha was proposed by Roberts in [Robe72]. His proposal was to divide time into discrete intervals called *slots*. All stations are synchronized and can only commence their data frame transmissions in the beginning of a slot. The size of a slot is equal to the transmission time of a data frame. This modified version of Aloha is known as the slotted Aloha. The operation of a slotted Aloha transmitter can be described by a finite state machine shown in Figure 2.5. The receiver operation is the same as that of an Aloha receiver.

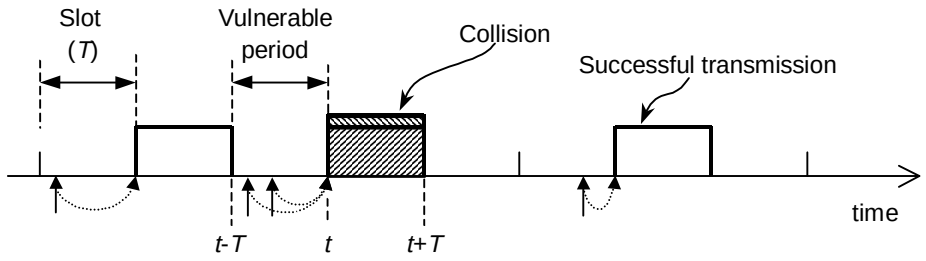


Figure 2.6: A snapshot of the slotted Aloha channel

2.2.1 Throughput Analysis under Poisson Traffic Assumption

Since a data frame transmission can only be commenced at the beginning of a slot, if a data frame arrives some time between t-T and t, it will be transmitted at t as in Figure 2.6. Hence, given a transmission at t, a collision will occur only if there is at least one additional data frame arrives within the time interval between t-T and t, of which the transmission is also scheduled at time t. As a result, the vulnerable period is now T, half of that in Aloha. Note that if any data frame arrives within the time interval between t and t+T, it will be scheduled to transmit at the beginning of the next slot.

With the new vulnerable period, by repeating the performance analysis of Aloha performed in the previous section, we obtain the throughput, S, of slotted Aloha to be

$$S = Ge^{-G}. \qquad (2\text{-}5)$$

The throughput as a function of the arrival rate of the attempt traffic for slotted Aloha is shown in Figure 2.7. It is also compared with the performance of Aloha. As can be seen, slotted Aloha achieves its maximum throughput value of $1/e$, or 0.36 at G=1, which is doubled the maximum throughput of Aloha.

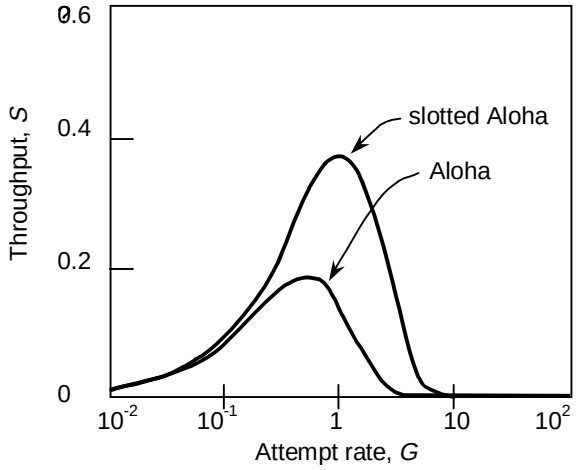


Figure 2.7: Throughput versus attempt rate for Aloha and slotted Aloha

2.2.2 Throughput Analysis under a Finite Station Assumption

As the previous subsection indicated, the performance analysis under a Poisson arrival process suggests an infinite number of stations. In a LAN, the number of stations is often small. To make the previous model more realistic, performance analysis under a finite station assumption is conducted by Kleinrock et al. [KlLa75]. The modified model is known as the "*linear feedback model*".

According to [KlLa75], a network consists of a finite number of stations, *M*. Each station is assumed to be in one of two states – *thinking* and *backlogged*. A station that is in the thinking state does not have a data frame to transmit. However, with probability σ, a new data frame will be generated within a slot. If a data frame is generated, the station will transmit the newly generated data frame in the next slot, otherwise, it will remain in the thinking state. The station will switch to the backlogged state only if the data frame transmission suffers a collision. A station that is in the backlogged state cannot generate further data frames as each station can only carry at most one data frame. It will remain in the backlogged state until it has completed the data frame transmission successfully. The data frame transmission is scheduled for retransmission based on a certain

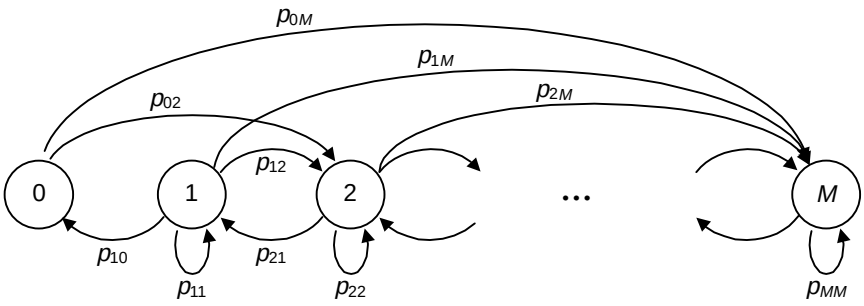


Figure 2.8: State transition diagram of the finite station model for slotted Aloha

retransmission algorithm. According to the retransmission algorithm, a station that suffers a collision in a data frame transmission should attempt a retransmission of the same data frame with probability v in every slot in the future. The station continues this retransmission attempt until the data frame is successfully transmitted, and it then returns back to the thinking state.

Let $N(t)$ be a random variable representing the number of backlogged stations at the beginning of slot t. Since $N(t+1)$ depends only on $N(t)$, a discrete time Markov chain can be used to describe this system, where the state of the Markov chain represents the number of backlogged stations in the slotted Aloha system. Figure 2.8 shows the state transition diagram of this model.

Define π_i to be the steady-state probability of the system being in state i, and p_{ij} to be the transition probability from state i to state j. To obtain the steady-state probability, it is necessary to solve the following finite set of linear equations

$$\begin{aligned} \boldsymbol{\pi} &= \boldsymbol{\pi}\mathbf{P} \\ 1 &= \sum_i \pi_i \end{aligned} \tag{2-6}$$

with $\boldsymbol{\pi}$ the row vector whose elements are π_i, and $\mathbf{P}$ the transition probability matrix.

The transition probability matrix, **P**, can be formulated as [KlLa75]

$$p_{ij} = \begin{cases} 0, & j < i-1 \\ P_b(1,i)\cdot P_t(0,i), & j = i-1 \\ [1-P_b(1,i)]P_t(0,i) + P_b(0,i)\cdot P_t(1,i), & j = i \\ [1-P_b(0,i)]P_t(1,i), & j = i+1 \\ P_t(j-i,i), & j > i+1 \end{cases} \quad (2\text{-}7)$$

with $P_b(n,i) = \binom{i}{n} v^n (1-v)^{i-n}$ and $P_t(n,i) = \binom{M-i}{n} \sigma^n (1-\sigma)^{M-i-n}$. $P_b(n,i)$ is the probability that n out of i backlogged stations transmit their data frames in a particular slot, and $P_t(n,i)$ is the probability that n thinking stations transmit in a particular slot given that there are already i backlogged stations. Having formulated **P**, the steady-state probability vector can be obtained numerically.

By definition, the throughput, *S*, is proportion of time that the channel carries successful transmissions. Consider that data frame sizes are fixed and its transmission time is equal to one unit of time, *S* is thus equal to the probability that a particular slot carries a successful transmission. To obtain a successful transmission in a slot, it is required to have exactly one transmission, either from the thinking or the backlogged stations. The probability, $P_{succ}(i)$, that there is exactly one transmission occurred in a slot given i backlogged stations is

$$\begin{aligned} P_{succ}(i) &= P_b(1,i)P_t(0,i) + P_b(0,i)P_t(1,i) \\ &= iv(1-v)^{i-1}(1-\sigma)^{M-i} + (1-v)^i (M-i)\sigma(1-\sigma)^{M-i-1} \end{aligned} \quad (2\text{-}8)$$

and hence the throughput, *S*, is

$$S = \sum_{i=0}^{M} P_{succ}(i)\cdot \pi_i \,. \quad (2\text{-}9)$$

A special case of this model is to consider that a backlogged station transmits its data frame with the same probability as a thinking station, that is $v=\sigma$. Substituting it into Equation (2-8) gives

$$P_{succ}(i) = M\sigma(1-\sigma)^{M-1}. \tag{2-10}$$

This result suggests that $P_{succ}(i)$ is no longer a function of i, this is because there is no difference between thinking and backlogged stations in this case. Thus the throughput, S, can be expressed as

$$\begin{aligned} S &= \sum_{i=0}^{M} P_{succ}(i)\pi_i \\ &= P_{succ}(i)\cdot \sum_{i=0}^{M} \pi_i \\ &= M\sigma(1-\sigma)^{M-1}. \end{aligned} \tag{2-11}$$

It is further assumed that the overall arrivals of both the thinking and backlogged stations generate a rate denoted by G expressed in number of transmissions per slot. It is clear that $G=M\sigma$ since at any time, there are M stations in the network and each of them generate a transmission with probability σ in every slot. By substituting $G=M\sigma$ into Equation (2-11), the throughput, S, is found to be

$$S = G\left[1-\frac{G}{M}\right]^{M-1}. \tag{2-12}$$

Recall that M is the number of stations in the network, by bringing M to infinity, Equation (2-12) yields $S=Ge^{-G}$, which is exactly the same as Equation (2-5), the result given in the previous subsection obtained under the assumption of an infinite number of stations.

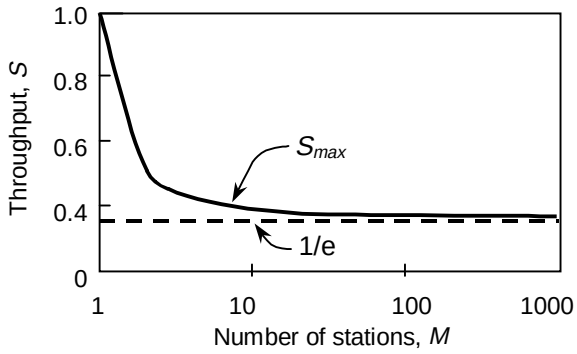


Figure 2.9: Throughput versus number of stations for slotted Aloha

By differentiating Equation (2-12) in terms of M and setting the result to zero, we obtain the maximum throughput of the slotted Aloha channel, S_{max}, in terms of the number of stations to be

$$S_{\max} = \left[1 - \frac{1}{M}\right]^{M-1}. \qquad (2\text{-}13)$$

Finally, we plot S_{max} as a function of the number of stations in Figure 2.9. As can be seen, if the number of stations grows above 15, the maximum throughput converges to that of the analytical result derived under the assumption of an infinite number of stations in subsection 2.2.1.

2.2.3 Throughput Analysis under a Linear Retransmission Algorithm

In the model developed in all the previous subsections, no retransmission algorithm is considered. It is assumed that the retransmission traffic is Poisson traffic that is independent to the load traffic. To justify this assumption, Lam has performed an analysis on slotted Aloha under a linear retransmission algorithm [Lam74, HaOr88]. The following assumptions are made in his model.

- The transmission time of a data frame is the time duration of one slot.
- The number of stations in a network is infinite and the overall arrival process is a Poisson process with mean arrival rate G data frames per slot.

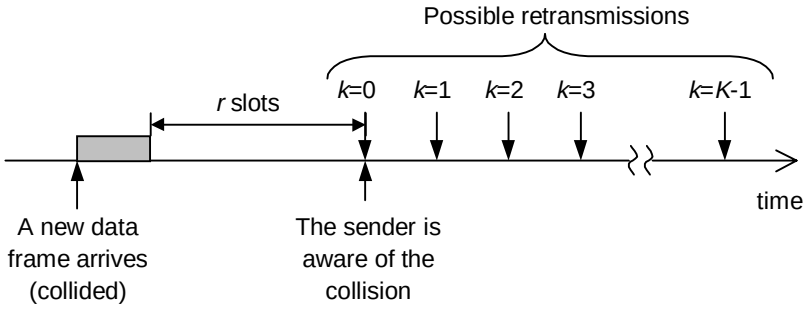


Figure 2.10: Illustration of a linear retransmission algorithm for slotted Aloha

- A station that suffers a collision during its data frame transmission will discover the collision *r* slots after the end of the transmission.
- Each station, after discovering the failure of its earlier transmission, schedules the retransmission based on a random integer number, *k*, uniformly chosen between 0 and *K*-1. The chosen number *k* specifies the delay in slots for the retransmission attempt. If zero is chosen, the retransmission is carried out immediately after the discovery of the failure of an earlier transmission.

An illustration of the retransmission algorithm is given in Figure 2.10. The throughput analysis involves in obtaining the following quantities:

- q_n – the probability that a newly arrived data frame will be transmitted successfully.
- q_t – the probability that a collided data frame will be transmitted successfully.
- p_i – the probability that a collided data frame is successfully transmitted after *i* retransmission attempts, where $i \geq 1$.
- *H* – The average number of retransmissions required to successfully transmit a data frame, that is E[p_i].
- *S* – The system throughput.

According to [Lam74], the quantity of q_n and q_t can be derived to be

$$q_t = \left[\frac{e^{-G/K} - e^{-G}}{1 - e^{-G}}\right] \cdot \left[e^{-G/K} + \frac{G}{K} e^{-G}\right]^{K-1} \cdot e^{-S} \qquad (2\text{-}14)$$

and

$$q_n = \left[e^{-G/K} + \frac{G}{K} e^{-G} \right]^K \cdot e^{-S} . \tag{2-15}$$

The quantity p_i can be expressed in terms of q_n and q_t as

$$p_i = (1-q_n)(1-q_t)^{i-1} q_t . \tag{2-16}$$

Given p_i, the expectation of p_i, H, can be obtained. It is found to be

$$H = \mathrm{E}[p_i] = \frac{1-q_n}{q_t} . \tag{2-17}$$

Finally, the system throughput, S, is determined by the aggregated attempt rate G, and the average number of retransmissions, H. The average number of transmission attempts is the mean retransmission attempt plus the first transmission attempt, that is H+1. Given that a data frame requires H+1 average transmission attempt to obtain a successful transmission, then the probability, P_{succ}, that a data frame transmission will be successful is $\frac{1}{H+1}$. According to the definition of throughput, we yield

$$\begin{aligned} S &= G \cdot P_{succ} \\ &= G \cdot \frac{1}{1+H} \\ &= G \cdot \left[\frac{q_t}{1-q_n+q_t} \right] . \end{aligned} \tag{2-18}$$

By taking the limit $K \to \infty$, both q_n and q_t become e^{-G}, and the throughput S can be reduced to

$$\begin{aligned} \lim_{K\to\infty} S &= G \cdot \left[\frac{e^{-G}}{1-e^{-G}+e^{-G}} \right] \\ &= G e^{-G} \end{aligned} \tag{2-19}$$

which becomes identical to the throughput obtained under the assumption of the Poisson retransmission traffic given in Section 2.2.1. This indicates that if a linear retransmission algorithm is implemented and the delay of the retransmission delay is large, then we can consider the retransmission traffic as Poisson traffic that is independent to the load traffic.

2.3 Carrier Sense Multiple Access

In Aloha and slotted Aloha, a station initiates a transmission without making sure that the broadcast channel is clear for transmission. As a result, during a data frame transmission, any other station may initiate a new transmission that will collide with an ongoing transmission. It is found that in some networks such as LANs, a station can detect whether the broadcast channel is clear, hence by requiring all stations to initiate transmissions only when the broadcast channel is sensed idle, the chances of collisions can be reduced. This additional operation is sometimes called “listen before transmit”. The protocol that employs this operation is called the Carrier Sense Multiple Access (CSMA) protocol.

As mentioned, this additional enhancement guarantees that any ongoing transmission that has been sensed by all stations will not be destroyed by collisions. However, collisions are still possible. If two or more stations start their transmissions at the same time, due to the signal propagation delay, they may not be aware of other transmissions at that time instance and go ahead with their transmissions. In this case, the transmissions collide.

In the case that a ready station senses a busy channel, it may defer its transmission based on various schemes described in the following:

- **Non-persistent CSMA Protocol**: In the non-persistent CSMA protocol, if the broadcast channel is sensed busy, a ready station will immediately reschedule its transmission to some later time in the future.
- **p-persistent CSMA Protocol**: In the p-persistent CSMA protocol, if the broadcast channel is sensed busy by a ready station, it will persist in

sensing the channel until the channel becomes idle. As soon as the channel is sensed idle, with probability *p*, the station transmits the data frame, or with probability (1-*p*), it waits for a predefined time period before sensing the channel again. The same process is repeated then.

- **1-persistent CSMA Protocol**: The 1-persistent CSMA protocol is a special case of the p-persistent CSMA protocol with *p*=1. If the channel is sensed busy, a ready station will keep sensing the busy channel until the channel turns idle. As soon as the channel is sensed idle, the station starts its transmission immediately, that is, with probability one.

Figure 2.11 describes the various versions of CSMA protocols by showing the operation of the CSMA transmitters using a finite state machine. The operation of a CSMA receiver is the same as that of an Aloha receiver.

There are several attempts in analyzing CSMA. The performance of various versions of CSMA protocols was first evaluated by Kleinrock et al. [KlTo75] under an infinite station assumption. In [TaKl85, TaKl87], Takagi et al. analyzed the performance of CSMA under a finite station assumption. A simple analysis of the 1-persistent CSMA protocol under an infinite station assumption was presented by Sohraby et al. [SoMV87]. In this review, we will concentrate on the non-persistent and the 1-persistent CSMA protocols, which are related to the focus of this thesis.

2.3.1 Throughput Analysis for Non-persistent CSMA

In this subsection, we review the performance analysis presented by Kleinrock et al. in [KlTo75]. We follow the definitions made in the study and let τ be the maximum signal propagation delay of the broadcast channel among stations. The normalized signal propagation time, *a*, is defined as

$$a = \frac{\tau}{T} \tag{2-20}$$

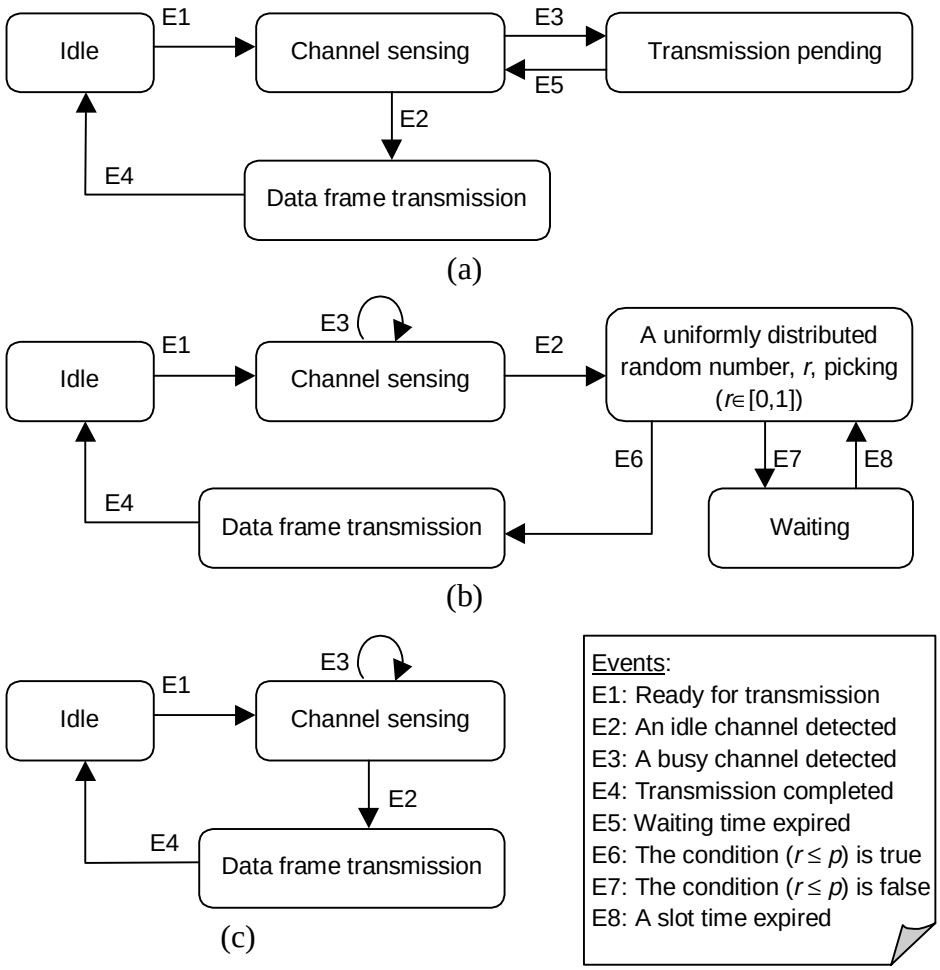


Figure 2.11: The finite state machine of a CSMA transmitter: (a) non-persistent CSMA; (b) p-persistent CSMA; (c) 1-persistent CSMA

where T is the data frame transmission time. This performance analysis is based on the two key assumptions summarized in Table 2.1, that are used in modeling Aloha, namely, the number of stations in a network is infinite, the aggregated traffic arrival process is a Poisson process, and the retransmission time for the collided transmission is very long so that the retransmission traffic is also Poisson traffic that is independent to the load traffic. Let the arrival rate of the combined load and retransmission traffic be G data frames per data frame transmission time.

To analyze the non-persistent CSMA protocol, we first observe that the broadcast channel is repeating two periods: an idle period and a busy period

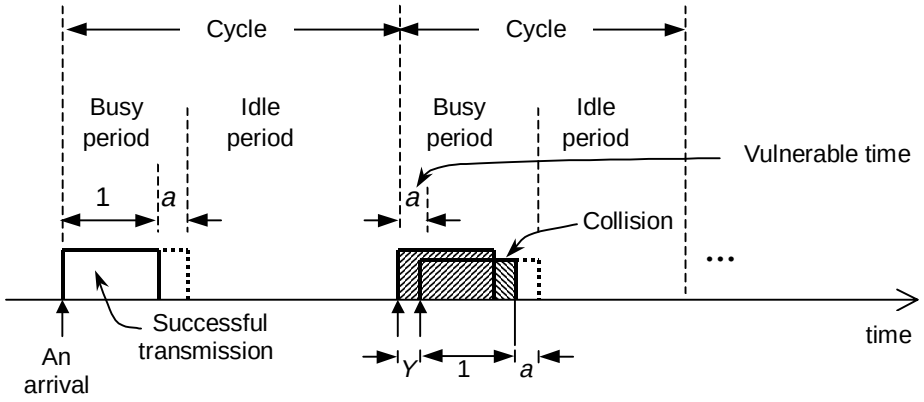


Figure 2.12: A snapshot of the non-persistent CSMA channel

(see Figure 2.12). Define B to be the duration of the busy period, I to be the duration of the idle period, and (B+I) to be the duration of a cycle. Note that B and I variables are two independent random variables. Let U denote the useful period, which is the duration that the channel carries useful information within a cycle. Using the results from the renewal theory, the throughput can be expressed as

$$S = \frac{\mathrm{E}[U]}{\mathrm{E}[B+I]} = \frac{\mathrm{E}[U]}{\mathrm{E}[B]+\mathrm{E}[I]}. \tag{2-21}$$

To compute the mean useful period, E[U], we observe when a transmission occurs, it takes a units of time to reach all other stations. To get a successful transmission, it is required to have no other stations initiating transmissions during vulnerable period, a, when a transmission is started. Given that the arrival process is a Poisson process, the mean useful period, E[U], is equal to the production of the probability that the transmission is successful and the normalized data frame transmission time, which is

$$\mathrm{E}[U] = 1 \cdot e^{-aG}. \tag{2-22}$$

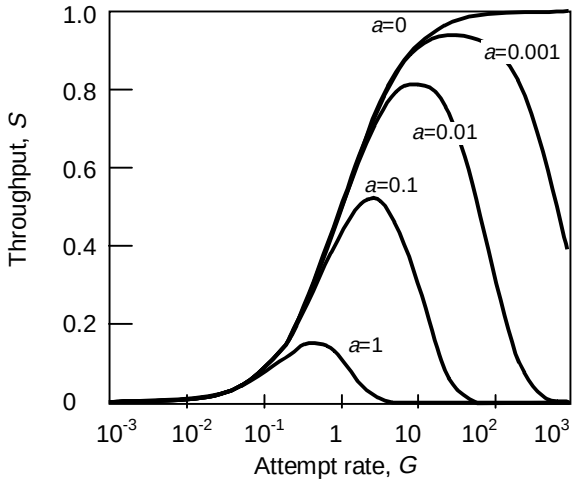


Figure 2.13: Throughput versus attempt rate for non-persistent CSMA

The idle period, *I*, is the duration from the end of a transmission to the arrival of the next transmission. In other words, *I* is the inter-arrival time of the arrival process. Since the arrival process is a Poisson process, thus the idle period is exponentially distributed, and its mean value is given by

$$\mathrm{E}[I] = \frac{1}{G}. \tag{2-23}$$

Finally, as depicted in Figure 2.12, the busy period, *B*, is the time difference between the first and the last arrivals within a vulnerable period, denoted *Y*, plus the data frame transmission time and the signal propagation time. The value *Y* is zero if there is only one transmission occurs in the vulnerable period. The cumulative distribution function of *Y* can be expressed as

$$\Pr\{Y \le y\} = e^{-G(a-y)}, \quad 0 \le y \le a \tag{2-24}$$

with expectation

$$\mathrm{E}[Y] = a - \frac{1 - e^{-aG}}{G}. \tag{2-25}$$

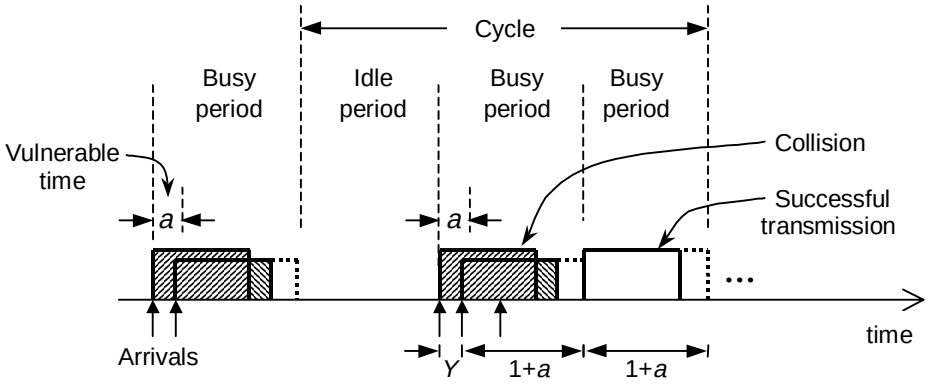


Figure 2.14: A snapshot of the 1-persistent CSMA channel

Hence, E[*B*] can be obtained by

$$\mathrm{E}[B] = \mathrm{E}[Y + 1 + a] = \left(a - \frac{1 - e^{-aG}}{G} \right) + 1 + a \,. \tag{2-26}$$

By substituting the expressions for E[*U*], E[*I*], E[*B*] into Equation (2-21), the throughput for the non-persistent CSMA protocol is thus

$$S = \frac{Ge^{-aG}}{G(1+2a) + e^{-aG}} \,. \tag{2-27}$$

The above result is shown in Figure 2.13 for several values of normalized propagation delay, *a*. It is clear that non-persistent CSMA performs better for smaller *a*.

2.3.2 Throughput Analysis for 1-persistent CSMA

The following throughput analysis of 1-persistent CSMA is taken from [SoMV87] performed by Sohraby et al. Similar to the assumptions used for non-persistent CSMA and Aloha (see Table 2.1), it considers an infinite number of stations. The aggregated arrival process is a Poisson process. The retransmission delay is assumed to be large so that the retransmission traffic is also Poisson traffic that is independent to the load traffic. Let the arrival rate of the combined load and retransmission traffic be *G* data frames per data frame transmission time.

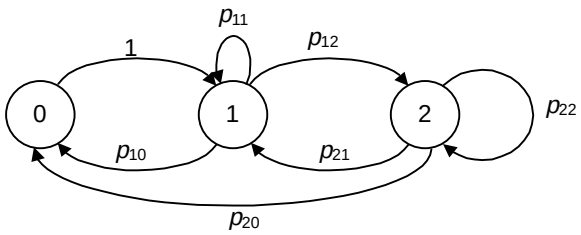


Figure 2.15: State transition diagram for the 1-persistent CSMA channel

Like non-persistent CSMA, the 1-persistent CSMA broadcast channel also exhibits a repeating cycle. A cycle contains an idle period and one or more busy periods as depicted in Figure 2.14. A busy period can be either a successful transmission period or an unsuccessful transmission period. Consider a three state Markov chain where states 0, 1 and 2 represent an idle, a successful and an unsuccessful transmission periods respectively. These three states correspond to a three state Markov chain embedded at the beginning of the periods. The state transition diagram of the three state Markov chain is shown in Figure 2.15.

According to the three state Markov chain, after an idle period (i. e. state 0), with probability one, a busy period (i. e. state 1) will occur. The occurrence of a busy period is due to an arrival. The busy period can be either a successful or an unsuccessful transmission period depends on whether there are other arrivals appear during the vulnerable period within the busy period. The vulnerable period has the duration of the signal propagation time, *a*. After a busy period, the following period will be an idle, busy, or unsuccessful period if there is no, one, or more than one arrival appears during that busy period respectively. If exactly one arrival appears during the busy period, then the next period on the channel will be another busy period where it can be a successful transmission period provided that no further arrivals appear during its vulnerable period.

Let T_i be the time the system spends in state i, i=0, 1, 2, and π_i the stationary probability for the system being in state i. The throughput of 1-persistent CSMA is thus

$$S = \frac{\pi_1 \cdot e^{-aG}}{\sum_{i=0}^{2} \mathrm{E}[T_i]\pi_i} \tag{2-28}$$

where the signal propagation time is a and the data frame transmission time is one unit of time. The reason that the useful time in each cycle is $\pi_1 e^{-aG}$ is because the condition for a busy period to be a successful transmission period is the probability that no transmission occurs during the vulnerable time during the beginning of the busy period.

Due to the similarity of the non-persistent CSMA and the 1-persistent CSMA protocols, the idle and busy periods of both protocols are the same. According to Equation (2-23), the mean idle period, E[T_0], for 1-persistent CSMA is

$$\mathrm{E}[T_0] = \frac{1}{G} \tag{2-29}$$

and according to Equation (2-26), the mean busy period, that can either be a successful or an unsuccessful transmission period, is

$$\mathrm{E}[T_1] = \mathrm{E}[T_2] = \mathrm{E}[Y + 1 + a] = \left(a - \frac{1 - e^{-aG}}{G} \right) + 1 + a \tag{2-30}$$

where Y is the random variable describing the time difference between the first and the last arrivals during a vulnerable time in a busy period. It is given in Equation (2-24).

From the state transition diagram shown in Figure 2.15, we obtain the following linear equations

$$\begin{aligned}
\pi_0 &= \pi_1 p_{10} + \pi_2 p_{20} \\
\pi_1 &= \pi_0 + \pi_1 p_{11} + \pi_2 p_{21} \\
\pi_2 &= \pi_1 p_{12} + \pi_2 p_{22} \\
1 &= \pi_1 + \pi_2 + \pi_3 .
\end{aligned} \tag{2-31}$$

It is found that when the channel is in a busy period, i.e. state 1 or state 2, according to the 1-persistent CSMA protocol, the probability that the channel will be busy again or will turn idle is regardless of whether the busy period is a successful or an unsuccessful transmission. Hence

$$p_{1j} = p_{2j}, \quad j = 0,1,2. \tag{2-32}$$

By substituting Equation (2-32) into Equation (2-31), we yield

$$\begin{aligned}
\pi_0 &= \frac{p_{10}}{1 + p_{10}} \\
\pi_1 &= \frac{p_{10} + p_{11}}{1 + p_{10}} .
\end{aligned} \tag{2-33}$$

Furthermore, according to [SoMV87], we have the following

- p_{10}, the probability that the channel is an idle period after the end of a successful transmission period, is

$$\begin{aligned}
p_{10} &= \Pr\{\text{no arrival appears in 1 unit of time} \mid \text{success}\} \\
&\quad + \Pr\{\text{no arrival appears in } Y+1 \text{ units of time} \mid \text{collision}\} \\
&= e^{-G} \cdot e^{-aG} + \int_0^a e^{-G(1+y)} \cdot G e^{-G(a-y)} dy \\
&= (1 + aG) e^{-G(1+a)} .
\end{aligned}$$

- p_{11}, the probability that the channel is a successful transmission period after the end of a successful transmission period, is

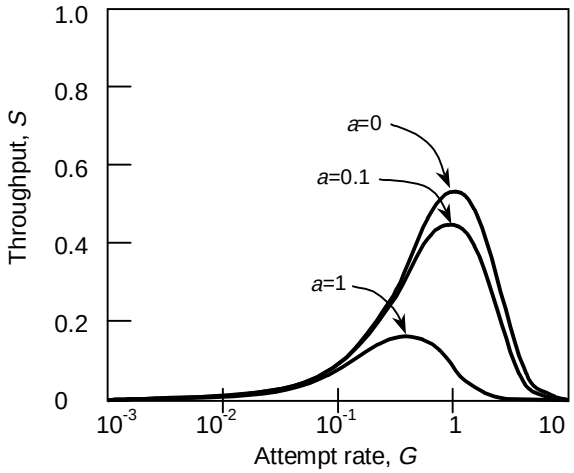


Figure 2.16: Throughput versus attempt rate for 1-persistent CSMA

$$
\begin{aligned}
p_{11} &= \Pr\{\text{one arrival appears in 1 unit of time} \mid \text{success}\} \\
&\quad + \Pr\{\text{one arrival appears in } Y+1 \text{ units of time} \mid \text{collision}\} \\
&= Ge^{-G}\cdot e^{-aG} + \int_0^a G(1+y)e^{-G(1+y)}\cdot Ge^{-G(a-y)}dy \\
&= \left(\frac{G^2}{2}\left[(1+a)^2-1\right]+G\right)e^{-G(1+a)} \quad .
\end{aligned}
$$

Putting all the obtained results into Equation (2-28), the throughput for the 1-persistent CSMA protocol is

$$
S = \frac{Ge^{-G(1+a)}\left[1+G+aG\left(1+G+aG/2\right)\right]}{G(1+2a)-\left(1-e^{-aG}\right)+(1+aG)e^{-G(1+a)}} \quad . \tag{2-34}
$$

The throughput versus attempt rate for 1-persistent CSMA is given in Figure 2.16. A comparison between the non-persistent and the 1-persistent versions of CSMA is presented in Figure 2.17. The 1-persistent CSMA version of CSMA was introduced to reduce the wastage due to idle periods. As shown in Figure 2.17, when the attempt rate is low, particularly below 1, the channel throughput for 1-persistent CSMA is higher than that for non-persistent CSMA, this indicates that the idle period in 1-persistent CSMA is indeed reduced. However, when the attempt rate is high, due to the high probability of collisions in 1-persistent CSMA and significant wastage of a

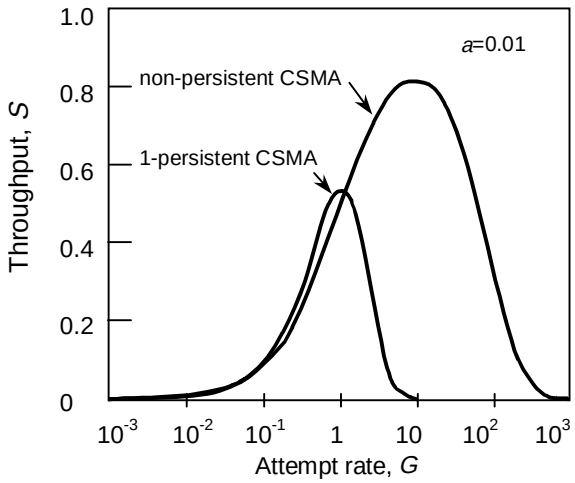


Figure 2.17: Throughput versus attempt rate for non-persistent CSMA and 1-persistent CSMA (a=0.01)

collision, its throughput drops below the channel throughput of non-persistent CSMA.

2.4 CSMA with Collision Detection

It is further realized that in wired LANs, in addition to sensing the broadcast channel before a transmission, a station can also detect if the data frame transmission suffers a collision during the transmission, this is sometimes called “listen while transmit”. With this feature, instead of blindly transmitting the entire data frame even though it has collided with others, a station can detect a collision and quickly abort the collided transmission so that the duration of a collision can be reduced. The protocol that implemented this feature is called the Carrier Sense Multiple Access with Collision Detection (CSMA/CD).

The operation of the CSMA/CD protocols is the same as that of the CSMA protocols, except that in the case of a collision, all data frame transmissions involved in a collision will be aborted immediately after the collision is detected. The retransmissions of the unsuccessful transmissions are scheduled at some time later in the future.

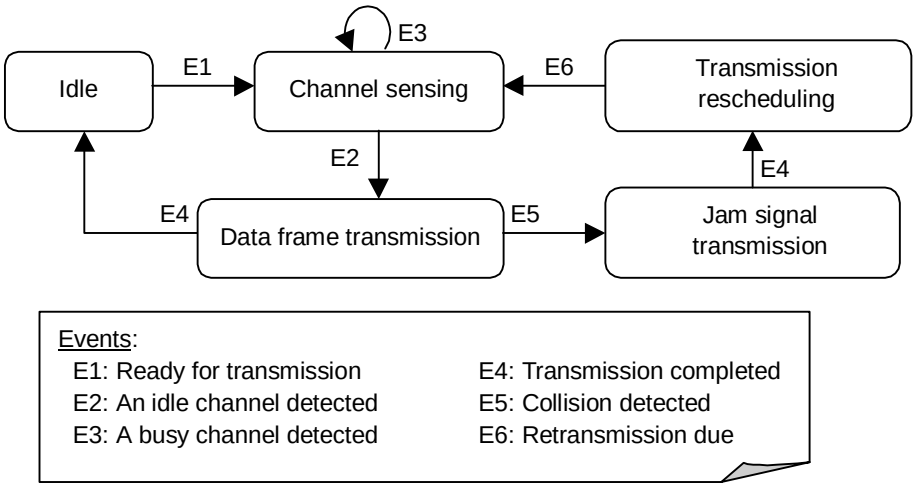


Figure 2.18: The finite state machine of a 1-persistent CSMA/CD transmitter

To perform the collision detection operation, each station is equipped with an interference detector. During a data frame transmission, the station also detects signals on the broadcast channel. If the detected signal is different from the transmitted signal, a collision of transmissions occurs and the station will abort the transmission immediately.

To further improve the reliability of collision detection operation in CSMA/CD, a *collision consensus enforcement* mechanism is introduced to ensure that all stations involved in a collision will detect the collision [MeBo76]. To achieve this, each station, when detected a collision during a data frame transmission, must jam the broadcast channel by transmitting the predefined jam signal. The transmission of the jam signal makes the collision more obvious.

The description of the CSMA/CD transmitter operation implementing the 1-persistent version is depicted in Figure 2.18.

2.4.1 Throughput Analysis for 1-persistent CSMA/CD

Several throughput analyses for CSMA/CD are available in the literature. One of the first appearances was due to Tobagi et al. [ToHu80]. Their analysis adopted the classical assumptions used for throughput performance

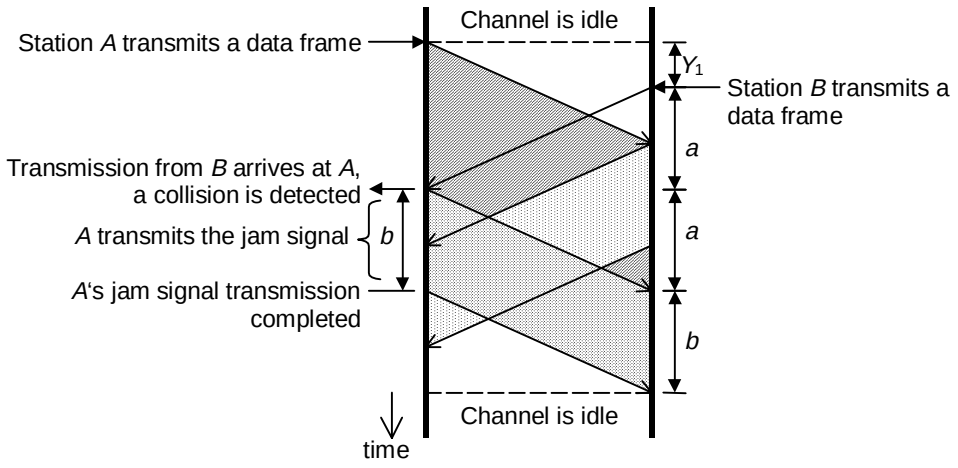


Figure 2.19: Timing diagram for 1-persistent CSMA/CD during a collision

analysis for Aloha and CSMA summarized in Table 2.1. In [TaKl85, TaKl87], Takagi et al. provided the channel throughput of CSMA/CD under a finite station assumption. Sohraby et al. [SoMV87] later presented a simple approach for analyzing 1-persistent CSMA/CD under the classical assumptions. The following treatment follows the work done by Sohraby et al. [SoMV87].

Similar to the performance analysis for 1-persistent CSMA, a three state Markov chain is first constructed to describe the channel activities of the 1-persistent CSMA/CD protocol (see Figure 2.15). States 0, 1, 2 of the Markov chain represent an idle period, a busy period and an unsuccessful transmission period respectively. The system throughput is

$$S = \frac{\pi_1 \cdot e^{-aG}}{\sum_{i=0}^{2} \mathrm{E}[T_i]\pi_i} \tag{2-35}$$

where T_i is the time the system spends in state i, i=0, 1, 2, and π_i is the stationary probability for the system being in state i. Again, a is the signal propagation time normalized to the data frame transmission time.

Figure 2.19 depicts the timing diagram for the 1-persistent CSMA/CD channel when a collision occurs. In the figure, b is the time duration of the jam signal, and the collision detection time by a transceiver is ignored. It is

noticed that with collision detection, the unsuccessful transmission period depends on the arrival time of the first and the second transmissions involved in the collision. Denote Y_1 be the arrival time difference between the first and the second arrivals in a vulnerable period. Since the traffic arrival process is a Poisson process, it is easy to show that

$$\Pr\{Y_1 \le y\} = 1 - e^{-Gy}, \quad 0 \le y \le a \tag{2-36}$$

To complete the analysis, it is required to find E[T_0], E[T_1], E[T_2], p_{10}, p_{11}, p_{20} and p_{21}. To start with, we notice that the mean duration of an idle period, E[T_0], for 1-persistent CSMA with or without collision detection is the same. Hence Equation (2-29) applies here for E[T_0].

The mean duration of a busy period, E[T_1], can be obtained to be

$$\begin{aligned} \mathrm{E}[T_1] &= \mathrm{E}[T_1 \mid \text{success}] + \mathrm{E}[T_1 \mid \text{collision}] \\ &= (1+a)e^{-aG} + \mathrm{E}[2a + b + Y_1 \mid \text{collision}] \\ &= (1 - e^{-aG})(2a + b + 1/G) + e^{-aG} \quad . \end{aligned} \tag{2-37}$$

The transition probabilities from state 1 to states 0 and 1 can be found to be

$$\begin{aligned} p_{10} &= \Pr\{\text{no arrival appears in 1 unit of time} \mid \text{success}\} \\ &\quad + \Pr\{\text{no arrival appears in } a + b + Y_1 \text{ units of time} \mid \text{collision}\} \\ &= e^{-G} \cdot e^{-aG} + \int_0^a e^{-G(a+b+y)} \cdot G e^{-Gy} dy \\ &= e^{-G(a+1)} + \frac{1}{2} e^{-G(a+b)} \left[1 - e^{-2aG}\right]. \end{aligned} \tag{2-38}$$

and

$$\begin{aligned} p_{11} &= \Pr\{\text{one arrival appears in 1 unit of time} \mid \text{success}\} \\ &\quad + \Pr\{\text{one arrival appears in } a + b + Y_1 \text{ units of time} \mid \text{collision}\} \\ &= G e^{-G(1+a)} + \int_0^a G(a + b + y) e^{-G(a+b+y)} \cdot G e^{-Gy} dy \\ &= G e^{-G(1+a)} + \frac{1}{4} e^{-G(a+b)} \left[1 - e^{-2aG}\right]\left[1 + 2G(a+b)\right]. \end{aligned} \tag{2-39}$$

For E[T_2], p_{20} and p_{21}, as illustrated in Figure 2.19, if two or more stations become ready during a busy period, then the following unsuccessful transmission will last for $2a+b$ units of time. That is

$$\mathrm{E}[T_2] = 2a + b, \tag{2-40}$$

$$p_{20} = e^{-G(a+b)} \tag{2-41}$$

and

$$p_{21} = G(a+b)e^{-G(a+b)}. \tag{2-42}$$

Using Equations (2-35) to (2-42) and the balance equation set given in Equation (2-31), the stationary probabilities are

$$\begin{aligned}
\pi_1 &= \frac{p_{20} + p_{21}}{(1 - p_{10} - p_{11})(1 + p_{20}) + (1 + p_{10})(p_{20} + p_{21})} \\
\pi_2 &= \frac{1 - p_{10} - p_{11}}{(1 - p_{10} - p_{11})(1 + p_{20}) + (1 + p_{10})(p_{20} + p_{21})} \\
\pi_3 &= 1 - \pi_1 - \pi_2
\end{aligned} \tag{2-43}$$

and the throughput for 1-persistent CSMA/CD is therefore

$$\begin{aligned}
S = (p_{20} + p_{21})e^{-aG} \cdot \Bigg[& \frac{(1 - p_{11})p_{20} + p_{10}p_{21}}{G} \\
& + (2a + b)(1 - p_{10} - p_{11}) \\
& + \left((1 - e^{-aG}) \left(2a + b + \frac{1}{G} \right) + e^{-aG} \right) (p_{20} + p_{21}) \Bigg]^{-1}.
\end{aligned} \tag{2-44}$$

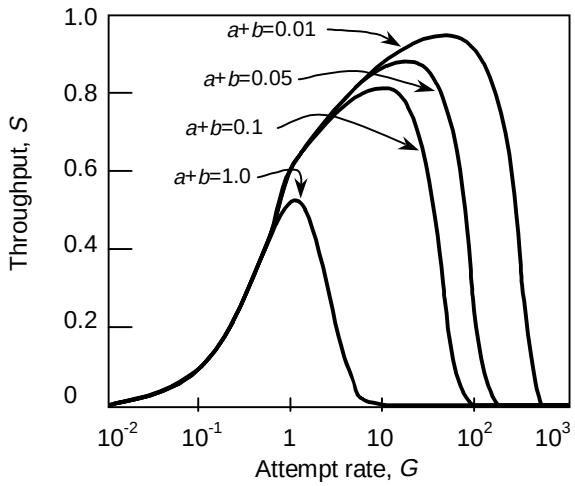


Figure 2.20: Throughput versus attempt rate for 1-persistent CSMA/CD (a=0.01)

The throughput versus attempt rate for 1-persistent CSMA/CD is plotted in Figure 2.20. Unlike 1-persistent CSMA, the protocol with collision detection offers much higher achievable throughput since an unsuccessful transmission can be quickly aborted when a collision is detected and the channel bandwidth wastage is significantly reduced.

A comparison between the non-persistent CSMA, the 1-persistent CSMA, as well as the 1-persistent CSMA/CD protocols is shown in Figure 2.21. When collision detection is implemented, the performance is significantly

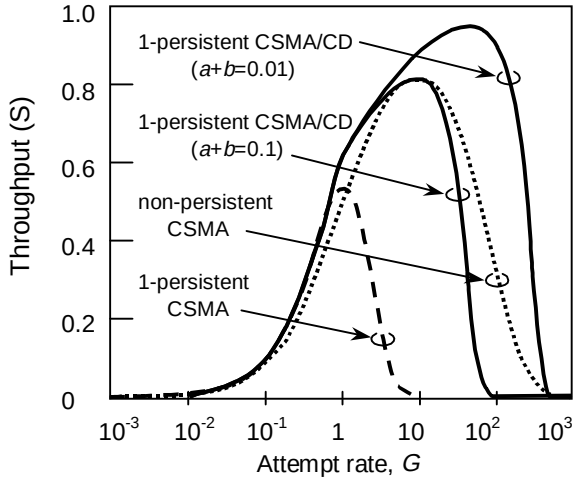


Figure 2.21: Throughput versus attempt rate for various carrier sensing random access protocols (a=0.01)

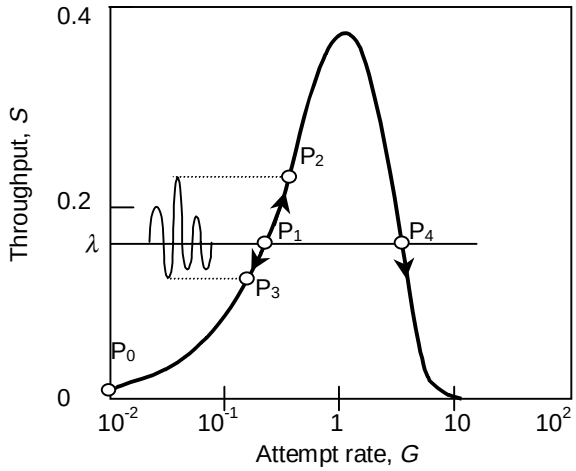


Figure 2.22: Illustration of instability of slotted Aloha

improved.

2.5 Instability of Aloha

The instability of an Aloha channel was identified in [KlLa75], [CaHe75] and [FLBG74]. It was then found that the instability property also occurs in other random access protocol such as CSMA [ToKl77]. To understand the instability of Aloha, we revisit the throughput curve of slotted Aloha obtained in Section 2.2 (see Figure 2.22). The protocol is said to be operated in a steady state if the arrival rate of the load traffic is equal to the service rate of the protocol. The service rate is related to the channel throughput of the protocol. In the case where the arrival rate is changed so that it is higher or lower than the service rate or a particular channel throughput level, the channel throughput will drift away from its original level to a new level where the service rate matches the arrival rate. At that operating point, the protocol enters a new steady state.

To demonstrate the instability of slotted Aloha, Figure 2.22 plots the throughput versus the attempt rate of slotted Aloha. Assume that initially the protocol is operated at a *channel operating point* P_0 on the throughput curve. The protocol is then loaded with Poisson traffic at a mean arrival rate of λ. This increase in input rate causes the protocol to move away from its steady

state. Since the load is higher than the throughput level, the attempt rate increases. According to the throughput curve, the increase in the attempt rate at point P_0 causes the operating point of the protocol to move rightwards, resulting in a higher channel throughput level. When the throughput level reaches point P_1, which is the operating point where the service rate is equal to λ, the protocol enters a new steady state.

Since the load traffic is Poisson traffic, its arrival rate fluctuates around the mean. If the arrival rate at one instance moves a little higher than its mean, a higher attempt rate is resulted. This causes the operating point of slotted Aloha to move rightwards from point P_1 to a new point P_2 where the service rate matches the higher arrival rate. In contrast, if the arrival rate suddenly drops a little below its mean, the attempt rate drops, causing the operating point of slotted Aloha to drift leftwards to a new channel operating point P_3. When the arrival rate of the load traffic returns to λ, the channel throughput returns to point P_1. Therefore, point P_1 is considered a stable point for operation provided that the fluctuation in the arrival rate of the load traffic is small.

Notice that in Figure 2.22, two different attempt rates are found to have the same throughput level, which are points P_1 and P_4. They are the points where the service rate is equal to the arrival rate. Hence we also expect the protocol to be operated steadily at point P_4. However, we can demonstrate that even a small fluctuation of arrival rate on the load traffic will easily drive the channel throughput towards zero. Consider that the protocol is operated at point P_4 initially. As we mentioned before, the arrival rate of the Poisson load traffic fluctuates around its mean. If the arrival rate is slightly higher than the mean value, the attempt rate increases. As a result of the increased attempt rate, the operating point drifts away from point P_4 and moves rightwards. This movement pushes the throughput level to an even lower value, which opens the gap between the arrival rate and the service rate. The channel operating point will keep drifting rightwards since it fails to find a steady state. As the drifting of the channel operating point

continues, the attempt rate increases and the throughput level decreases. Even though some time later, the arrival rate of the load traffic drops back to λ, the channel throughput has already dropped below λ. As a result of a higher arrival rate and a lower throughput level, the channel operating point will continue to drift rightwards until the channel throughput eventually drops to zero. Hence, it is said that P_4 is not a steady point of operation.

2.5.1 Stability Theory

In [LaKl75], a theory was developed by Lam et al. to analyze the stability of the slotted Aloha channel. To perform the stability analysis, the relationship between the channel throughput and the number of backlogged stations is first derived [KlLa75, HaOr88] to be

$$n = SH\left[r + \frac{1}{2} + \frac{K}{2}\right] \tag{2-45}$$

where S is the channel throughput, n is the number of backlogged stations, H is the average number of retransmission attempts required to successfully transmit a data frame (given in Equation (2-17)), r is the number of slots required for a sender to realize that its previous transmission is unsuccessful (a typical value for r in slotted Aloha channel is one), and K is the parameter of a linear retransmission algorithm describing the maximum number of slots to wait for a retransmission attempt after the discovery of an earlier unsuccessful transmission. In Figure 2.23, we plot the number of backlogged stations versus throughput for slotted Aloha. Several curves are shown for different values of K.

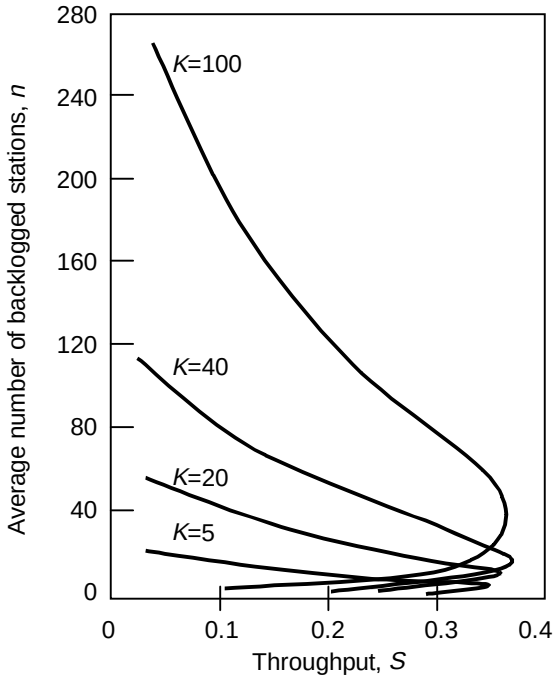


Figure 2.23: Average number of backlogged stations versus throughput for slotted Aloha (K=5, 20, 40, 100)

It is further considered that there are M stations in a network, and the average number of backlogged stations waiting for retransmission is n. Since each station only carries one data frame, clearly only (M-n) stations can generate a new data frame. If the access probability, that is the probability for any of the (M-n) stations to generate a new data frame is σ in each slot of the slotted Aloha channel, then the expected load, S_{in}, is

$$S_{in} = (M - n)\sigma \quad . \tag{2-46}$$

The previous equation is also known as the *channel load line* [LaKl75]. In Figure 2.24, we combine the curve of the number of backlogged stations versus throughput, and the channel load line in one figure.

According to [LaKl75], the intersection of the throughput curve and the channel load line is called an *equilibrium point*. At an equilibrium point, the expected load input rate is equal to the channel throughput. An equilibrium point can be a stable or an unstable operating point. It is defined that at an equilibrium point, if an increase in the number of backlogged stations will

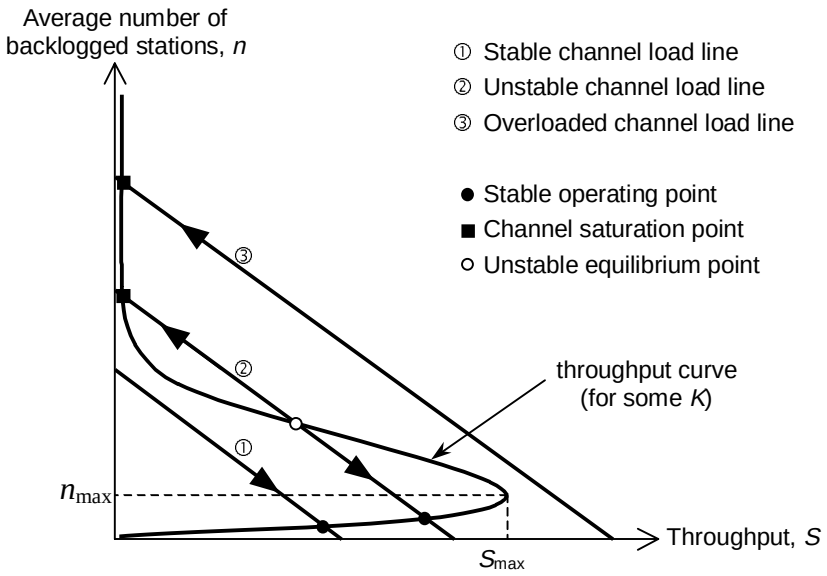


Figure 2.24: Stability of the slotted Aloha channel

cause the channel throughput to drop, then the point is an *unstable equilibrium point*. Otherwise, it is a *stable operation point*. Clearly in Figure 2.24, all intersection points above the n_{max} horizontal line are unstable since an increase in n will result in a decrease in S according to the throughput curve. Whereas those below the horizontal line are stable. Furthermore, at the intersection points where the throughput level is near zero, they are said to be *channel saturation points*.

As for a channel load line, it is said to be *stable* only when there is one equilibrium point on the line, and the point is not a channel saturation point. Three channel load lines are drawn in Figure 2.24, labeled ①–③. The only difference between the three channel load lines is the number of stations, M. The three load lines use the same attempt probability, σ.

Load line ① is stable since it has one stable equilibrium point. Whereas for Load line ②, it is unstable because it has three equilibrium points where one is stable, one is unstable and one is saturated. This unstable channel exhibits "bistable" behavior. As we have discussed earlier in this section, a small fluctuation in the input rate may cause the operating point to drift from a

stable equilibrium point, to an unstable equilibrium point, and finally to a channel saturation point.

Load line ③ is considered overloaded since it has only one equilibrium point, and that point is saturated. This means that given access probability for each station, the protocol cannot cope with the large number of stations.

2.5.2 Dynamic Control Procedure

In the previous subsection, the slotted Aloha channel is shown to be unstable. The channel instability is mainly caused by extensive collisions that occurred on the channel partly due to the retransmission algorithm. Lam and Kleinrock found that it is possible to stabilize the channel by changing the parameters of the used retransmission algorithm [LaKl75]. A demonstration of stabilizing the slotted Aloha channel with a retransmission algorithm is depicted in Figure 2.25. A linear retransmission algorithm is considered and the parameter K is used in stabilizing the channel. In the figure, given a particular channel load line, with parameter $K=K_1$, the channel load line intersects the throughput curve at three points, indicating an unstable channel. However, if the parameter K is changed to K_2, the number of intersections between the channel load line and the throughput curve is reduced to one, and the intersection point is a stable equilibrium point. As a result of the new K value, a stable channel is obtained.

While the slotted Aloha channel can be stabilized by using a particular set of parameters of the retransmission algorithm as demonstrated in Figure 2.25, the channel may become unstable again if the channel load line changes. A smart solution to handle this problem is to implement a retransmission algorithm of which its parameter changes according to the traffic condition. These schemes are called the *dynamic control procedures*. One essential element to enable dynamic control procedures is that each station must "learn" the channel status to decide a suitable parameter set for the retransmission algorithm. One possible way to achieve the "learning" is to somehow acknowledge all stations the instantaneous channel information.

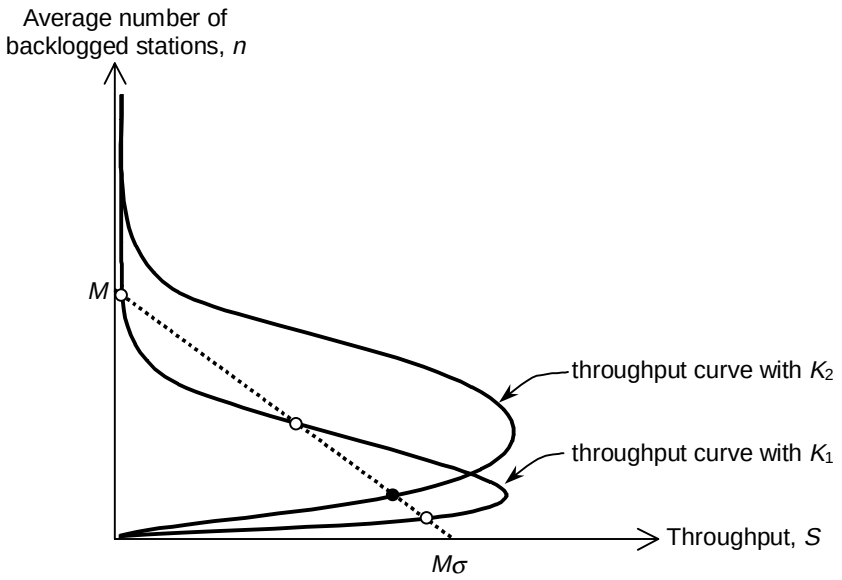


Figure 2.25: Illustration of the dynamic control procedure for stabilizing the slotted Aloha channel

For an arbitrary retransmission algorithm, given that there are Q backlogged stations, let each backlogged station retransmit its data frame in a particular slot with a probability p_a. This probability is called the *attempt probability*. With this information, the *acquisition probability*, denoted P_{succ}, that is the probability that a particular slot contains exactly one retransmission making it a successful transmission is

$$P_{succ} = Q \cdot p_a (1 - p_a)^{Q-1}. \qquad (2\text{-}47)$$

The "optimum" statistical decision on retransmissions that gives the highest acquisition probability can be obtained by differentiating Equation (2-47) with respect to p_a, and setting the result to zero. It is found that such a decision requires the attempt probability to be $1/Q$ [MeBo76]. To achieve this optimum decision, each station must learn the number of backlogged stations. Moreover, when the value Q is large, the acquisition probability approaches $1/e$, a non-zero probability. Thus the stability of the slotted Aloha can be achieved if an "optimum" retransmission algorithm can be developed.

2.6 Binary Tree Algorithm

The *Binary Tree Algorithm* is one of the most basic algorithms proposed to stabilize a broadcast channel. It was originally introduced by Caperanakis [Cape79], Tsybakov et al. [TsMi78], and Hayes [Haye78] independently.

The concept of this algorithm is that stations do not always retransmit their collided data frame with a constant probability in a slot, instead, they learn the system status from activities in previous slots and dynamically change the retransmission decision. Since stations are operated individually and have no common channel to exchange information, the retransmission decision among them is made asymmetrically. Moreover, the Binary Tree Algorithm requires all stations to monitor the broadcast channel at all times detecting whether each slot carries no, one or more than one transmission, even though the stations have nothing to transmit. This type of retransmission algorithm is known as the *full-sensing* type retransmission algorithm.

In the following, we describe the parallel search version of the static Binary Tree Algorithm proposed by Caperanakis [Cape79]. Consider that there are n data frame transmissions collided in a particular slot. This event is sometimes termed the "Big Bang" of n stations [Moll94]. Let the Big Bang slot be the first slot of the *collision resolution process*, the time period required for all n backlogged stations to successfully transmitted their data frame is the *collision resolution time* for n backlogged stations. During the collision resolution process, only stations that are involved in the Big Bang can transmit a new data frame.

The collision resolution process starts when a Big Bang occurs. Based on the Binary Tree Algorithm [Cape79], each backlogged station retransmits its data frame according to the following rules:

(R1) If a station suffers a collision in a slot, it may choose one of the two following options: retransmit immediately or wait. If the first option is chosen, (R2) will be used. Otherwise, (R3) will be executed.

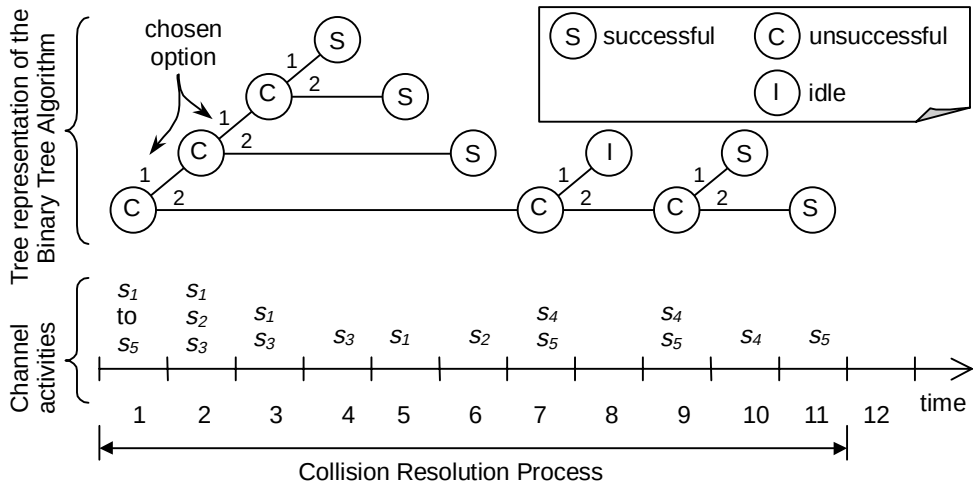


Figure 2.26: Example of the operation of the Binary Tree Algorithm

(R2) A station that has suffered a collision and chosen the first option in (R1) may retransmit its data frame immediately after the collision is detected. If a collision occurs again, (R1) will be repeated.

(R3) A station that has suffered a collision and chosen the second option in (R1) must wait until all other stations, which involved in the same collision and decided to retransmit immediately, successfully transmitted their data frames.

To illustrate the operation of the Binary Tree Algorithm, we construct the following example: We assume that there are five stations labeled s_1 to s_5 collide in a particular slot that starts the collision resolution process. The collision resolution process is depicted in Figure 2.26. The figure demonstrates the splitting of stations using a tree diagram. It also shows the activity in each slot on the broadcast channel. The channel activities during the collision resolution process of this example are described as follows.

All five backlogged stations transmit their data frames in the first slot, the Big Bang slot. According to (R1), all the five stations must choose either to retransmit immediately or wait. We assume that s_1 to s_3 choose to retransmit immediately, hence slot 2 carries three simultaneous retransmissions that will result in a collision. Since s_4 and s_5 choose to wait, then they can only retransmit when s_1 to s_3 have successfully transmitted their data frames. To

detect when they should become active again from the waiting, s_4 and s_5 must monitor the broadcast channel during their waiting.

After detected a collision in slot 2, s_1 to s_3 must choose one of the two options mentioned in (R1). We assume that s_1 and s_3 pick the first option, while s_2 chooses the second option. These decisions result in a collision in slot 3 caused by the immediate retransmissions of s_1 and s_3.

Again, s_1 and s_3 repeat (R1) and choose an option. Here we assume that s_3 chooses to retransmit immediately and s_1 chooses to wait, hence a successful transmission from s_3 finally occurs in slot 4. Upon the detection of the successful transmission, s_1 stops its waiting and performs its retransmission in slot 5, producing another successful transmission.

After a series of successful transmission slots indicating the successful transmission of all stations collided in slot 3, s_2 ends its waiting and transmits its data frame. With only one retransmission on the broadcast channel, the retransmission from s_2 is successful in slot 6.

It is then s_4 and s_5 become active from the waiting and transmit their data frames in slot 7 causing a collision. The resolution of the collision in slot 7 is illustrated in Figure 2.26. In this example, 11 slots were used for five data frame transmissions. In other words, the collision resolution time for the five stations is 11 slots in this example.

2.6.1 Collision Resolution Time Analysis

In this subsection, we review the performance of the Binary Tree Algorithm. The presented analysis is based on that of Massey [Mass81, RoSi90]. Define $B(n)$ to be the mean collision resolution time for n backlogged stations, that is the average number of slots needed to obtain n successful transmissions given that n transmissions collide in the Big Bang slot, and each station retransmits its data frame using the Binary Tree Algorithm.

In the Big Bang slot, each station can choose to retransmit immediately or wait. If i out of n backlogged stations choose to retransmit immediately, then $B(n)$, including the Big Bang slot, can be expressed recursively as

$$B(n) = 1 + \sum_{i=0}^{n} p(n,i) \cdot [B(i) + B(n-i)] \tag{2-48}$$

where $p(n,i)$ is the probability that i out of n stations choose option one in (R1). Since the choice of each station is binary and independent to each other, $p(n,i)$ is hence

$$\begin{aligned} p(n,i) &= \binom{n}{i}\left(\frac{1}{2}\right)^{i}\left(1-\frac{1}{2}\right)^{n-i} \\ &= \binom{n}{i}\left(\frac{1}{2}\right)^{n}. \end{aligned} \tag{2-49}$$

Note that when there is no or exactly one transmission in the Big Bang slot, the collision resolution process lasts for exactly one slot. That is

$$B(0) = B(1) = 1. \tag{2-50}$$

Using all the above results, B_n can be solved to be

$$B(n) = \frac{1 + \sum_{i=0}^{n-1} [p(n,i) + p(n,n-i)] \cdot B(i)}{1 - p(n,0) - p(n,n)}, \quad n \geq 2. \tag{2-51}$$

We plot $B(n)$ versus n in Figure 2.27. Our ultimate aim is to find the mean collision resolution time for one station, $L(n)$, that is the average number of slots needed for a station to successfully transmit its data frame. Since $B(n)$ is the number of slots needed for n successful transmissions, then $L(n)$ is simply

$$L(n) = \frac{B(n)}{n}, \quad n \geq 1. \tag{2-52}$$

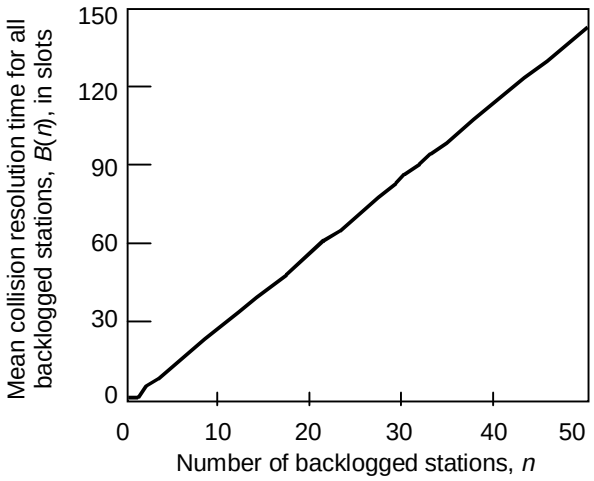


Figure 2.27: Mean collision resolution time of the Binary Tree Algorithm for n backlogged stations

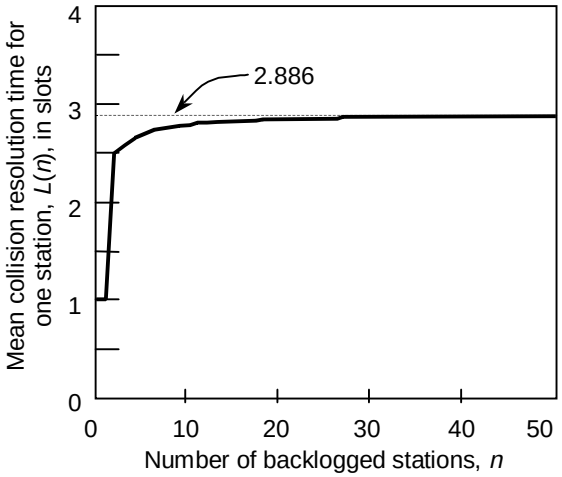


Figure 2.28: Mean collision resolution time of the Binary Tree Algorithm for one station

In Figure 2.28, we plot $L(n)$ as a function of n. One interesting observation from the figure is that $L(n)$ is rather insensitive to the number of backlogged stations, n. It is further shown that for $n \geq 6$, the following inequality holds [Mass81, RoSi90]

$$B(n) \leq 2.886n - 1, \quad n \geq 6. \tag{2-53}$$

Hence for a large value of n, the value of $L(n)$ approaches 2.886. In other words, regardless of the number of backlogged stations, 2.886 slots on average are needed for a station to transmit a data frame successfully.

Therefore the Binary Tree Algorithm is stable if the traffic arrival rate is less than 1/2.886 data frames per slot.

2.7 Binary Exponential Backoff Algorithm

In the previous section, we have studied the Binary Tree Algorithm. The Binary Tree Algorithm is a full-sensing type retransmission algorithm. The main property of the full-sensing type algorithm is that all stations must continuously monitor the broadcast channel to gain the knowledge of the system status. This operation introduces some drawbacks, mainly stations require more power to operate, and a noisy channel may result in incorrect channel feedback information received, leading to a wrong decision made by a station. Because of the wrong retransmission decision, the performance of the Binary Tree Algorithm may be seriously degraded.

To enhance the robustness of retransmission algorithms, instead of continuously monitoring the broadcast channel, a station only monitors the broadcast channel when it is transmitting a data frame. A station will receive feedback from the channel indicating whether its transmission is successful or unsuccessful. A retransmission algorithm based on this scheme is known as the *limited-sensing* retransmission algorithm.

A typical and widely known limited-sensing type retransmission algorithm is the Truncated *Binary Exponential Backoff* retransmission algorithm [MeBo76], or BEB in short. It is used in Ethernet, as well as the IEEE 802.11 MAC protocol in wireless LANs.

The operation of BEB is much simpler than that of the Binary Tree Algorithm since stations implementing BEB are not required to monitor the broadcast channel at all time. The rules for BEB implemented in Ethernet can be described as follows:

(R1) Each station resets its collision counter, c, to zero initially. Each station transmits its data frame, if any, based on a particular Aloha

type protocol such as slotted Aloha or CSMA/CD. If the transmission is unsuccessful, the retransmission is scheduled according to (R2).

(R2) Upon the detection of a collision, each station increases its collision counter by one. If the counter reaches 16, the data frame retransmission is aborted. Otherwise, it is retransmitted in a randomly and uniformly chosen future slot between slot r+1 and slot r+min(1024,2^c) provided that slot r is the slot that the station suffers a collision during its transmission.

(R3) If a retransmission is successful, (R1) is used then. Otherwise, (R2) is repeated.

The idea behind the design of the BEB retransmission algorithm is that each station starts to retransmit aggressively. As each station experiences a collision, its *backoff window*, that is the choices of future slots for retransmissions, is widened exponentially to reduce the chances of further collisions.

2.7.1 Collision Resolution Time Analysis

The performance as well as the stability of BEB has been studied in the literature. Aldous shows that the truncated BEB retransmission algorithm is unstable [Aldo87]. Later, Goodman et al. proved that BEB is stable if limited buffered stations are assumed [GGMM88]. The conflicting results are mainly due to the different assumptions in the two studies.

Zilic et al. attempted to model BEB using a Markov chain [ZiMo92]. Unfortunately, the approach is shown to be unrealistic because the resulting state space of the Markov chain is far too large for analysis. A simpler approach for the BEB analysis is later given by Molle [Moll94]. In the following, we present the analysis of BEB done by Molle [Moll94].

It is assumed that there are n stations involved in a Big Bang. After the detection of the collision, all n backlogged stations retransmit their data frames according to the BEB retransmission algorithm and no further

arrivals will occur during the collision resolution process. To compute the mean collision resolution time for a station, denoted $L(n)$, it is required to determine the attempt probability, P_k, that a station will retransmit its data frame in the k-th slot given that the Big Bang slot is the first slot.

Based on the BEB retransmission algorithm, a station may attempt a retransmission in the r-th slot only if the station has experienced an unsuccessful attempt previously between slot r-min(1024,2^c) and slot r-1, where c is the collision counter described in (R1). Thus P_r can be obtained by conditioning on the collision counter, that is

$$P_r = \sum_{c=0}^{15} P(r,c) , \qquad r = 1,2,\ldots \tag{2-54}$$

where $P(r,c)$ is the probability that slot r is chosen for a retransmission given c. $P(r,c)$ is determined by the following equation set

$$\begin{aligned}
&P(1,0) = 1 \\
&P(1,c) = 0, && c > 0 \\
&P(r,0) = P(r-1,15), && r > 1 \\
&P(r,c) = \sum_{k=\max(1,\, r-2^c)}^{r-1} \frac{P(k,c-1)}{2^c}, && r > 1,\ c < 10 \\
&P(r,c) = \sum_{k=\max(1,\, r-2^{10})}^{r-1} \frac{P(k,c-1)}{2^{10}}, && r > 1,\ c \geq 10
\end{aligned} \tag{2-55}$$

The attempt probability, P_r, of a particular station as a function of slot number, r, is shown in Figure 2.29. The attempt probability drops as r increases. This dropping of attempt probability is mainly because of the widening of the backoff window of each station as the collision resolution process progresses.

After 16 unsuccessful attempts, a station must abort its transmission and report this error to the protocol in the higher layer of a protocol stack. However, Molle assumes that the protocol in the higher layer will trigger an immediate retry, causing the station to transmit the same data frame again. From the MAC protocol point of view, it is considered as a new data frame

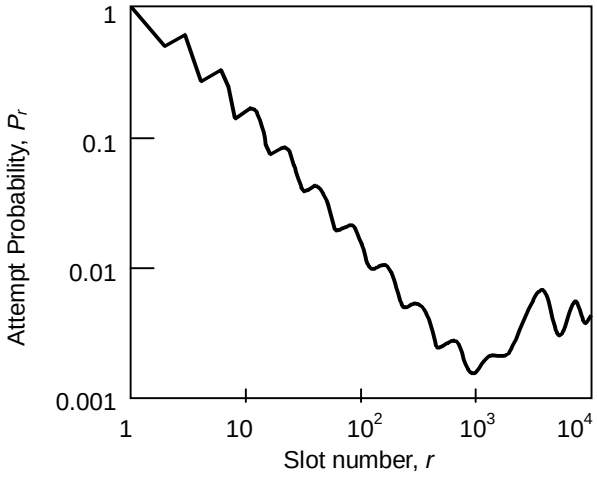


Figure 2.29: Attempt probability of a station for BEB in each slot after a Big Bang

transmission, hence the BEB operation for that station is restarted. This assumption is evident in the third formula of Equation (2-55), and it is also the reason why the attempt probability increases after around 1000 slots of collision resolution process time.

To find the mean collision resolution time for a station, $L(n)$, that is the average number of slots required for a station to successfully transmit its data frame, it is assumed that stations that participate in the collision resolution process are independent. The acquisition probability in the r-th slot, that is the probability that the r-th slot carries a successful transmission given n stations involving in the Big Bang, is

$$P_{succ}(n,r)=\left[\binom{n}{1}P_r(1-P_r)^{n-1}\right]\left[\prod_{j=1}^{r-1}\left(1-\binom{n}{1}P_j(1-P_j)^{n-1}\right)\right]. \qquad (2\text{-}56)$$

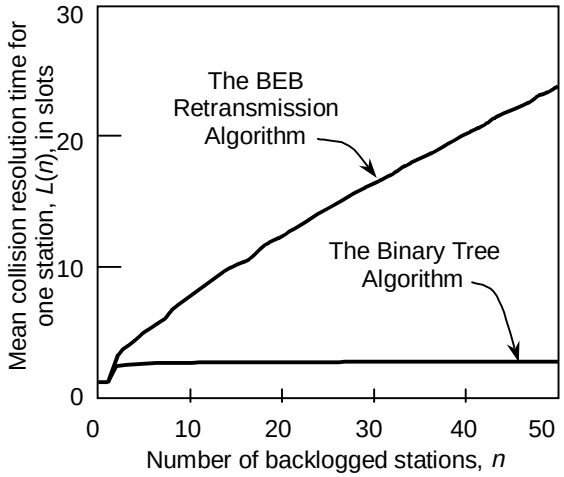


Figure 2.30: Mean collision resolution time for the BEB retransmission algorithm for one station

$P_{succ}(n,r)$ is essentially the probability that there is exactly one transmission in r-th slot and no successful transmission has occurred in all previous slots starting from the Big Bang slot. Given the acquisition probability, then $L(n)$ is

$$\begin{aligned} L(n) &= \sum_{r=1}^{\infty} r \cdot P_{succ}(n,r) \\ &= \sum_{k=1}^{\infty}\left[\prod_{r=1}^{k-1}\left(1-nP_r(1-P_r)^{n-1}\right)\right]. \end{aligned} \tag{2-57}$$

In Figure 2.30, $L(n)$ for BEB is plotted and compared with the results for the Binary Tree Algorithm. In the figure, we found that $L(n)$ for BEB is somewhat linear to the number of stations, n, participated in the collision resolution process. This indicates that the service rate of BEB, that is $1/L(n)$, is lower for a larger number of backlogged stations, n, which is unlike the Binary Tree Algorithm where the service rate is fixed regardless of the n value.

This result confirms the claim by Aldous that BEB is unstable [Aldo87], since new arrivals appear during a collision resolution process will increase the number of backlogged stations and lengthen the process. A longer collision resolution process will increase the likelihood of having further

new arrivals during the process, which will result in an even longer process. This positive feedback effect will continue, causing the number of backlogged stations to grow indefinitely and the service rate of BEB to drop to a near zero value. Hence BEB is unstable.

On the other hand, in [GGMM88], Goodman et al. consider the situation of a limited number of buffered stations, that is, a network consists of a certain number of stations and each station can only transmit at most one data frame. If all stations are attempting their data frame transmissions, then no further new arrivals will appear during the collision resolution process. Hence the number of backlogged stations will stop growing when it reaches the total number of stations in the network. Since the number of backlogged stations will cease to grow at some value, the service rate of BEB will remain at a level above zero, thus BEB is considered stable in this case.

2.8 The IEEE 802.3 MAC Protocol

In the early 1980s, the IEEE 802.3 working group was formed to study and standardize the Ethernet protocol originally proposed by Metcalfe et al. [MeBo76]. The Ethernet protocol is based on the 1-persistent CSMA/CD protocol. In case of collision, the BEB retransmission algorithm is used to resolve collisions. The IEEE 802.3 MAC protocol largely follows the original design of the Ethernet protocol.

The operation of a transmitter implementing the IEEE 802.3 MAC protocol is presented in Figure 2.31. The operation of an IEEE 802.3 MAC protocol receiver is the same as that of an Aloha receiver shown in Figure 2.1.

In the state diagram, a station when entering “Retransmission scheduling using BEB” state executes the BEB retransmission algorithm that is described in the previous section. Basing on the Collision Counter, c, the station randomly and uniformly chooses an integer, w, in the range $0 \le w < 2^{\min(c,10)}$. The station then waits for w slots before proceeding to “Channel sensing” state for the retransmission attempt.

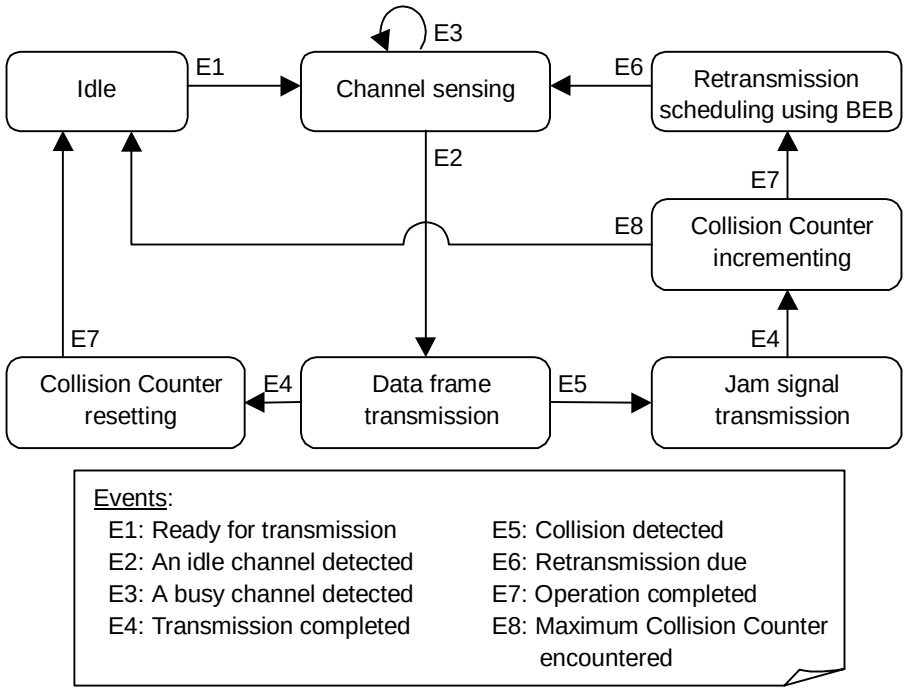
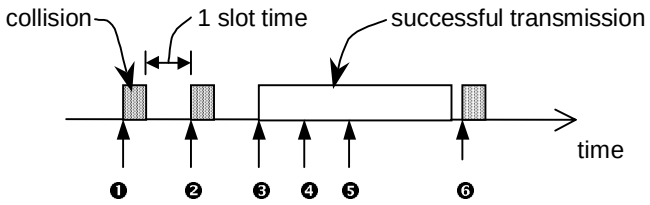


Figure 2.31: The finite state machine of the transmitter of the IEEE 802.3 MAC protocol

It is later discovered by Boggs et al. [BoMK88] and confirmed by Molle [Moll94] that the IEEE 802.3 MAC protocol possesses a serious fairness problem under an overloaded condition known as the *capture effect*. This effect is mainly due to the BEB retransmission algorithm. Figure 2.32 illustrates the capture effect of the IEEE 802.3 MAC protocol. From the presented example, it is shown that after a successful transmission, station *A* will enjoy a higher probability than others in competing the channel transmission right since station *A* uses a relatively small backoff window when executing the BEB retransmission algorithm. To eliminate this problem, Molle has proposed a new retransmission algorithm named *Binary Logarithmic Arbitration Method* (BLAM) to replace BEB [Moll94]. One important operation of BLAM in eliminating the capture effect is that when all ready stations "see" a successful transmission, they reset their Collision Counters so that after the successful transmission, all ready stations have an equal probability to acquire the broadcast channel for transmission.



1. Three stations (stations *A* to *C*) become ready and start their transmissions at the same time at ❶ causing a collision.
2. During the backoff, all the ready stations have decided to wait for a slot time before attempting their retransmissions at ❷.
3. A collision occurs again at ❷ due to the retransmissions of all the three stations.
4. Assume that after their jam signal transmissions, stations *A*, *B* and *C* have scheduled their retransmission at ❸, ❹ and ❺ respectively.
5. Station *A* retransmits its data frame at ❸. The transmission is successful.
6. When the retransmissions of stations *B* and *C* are due at ❹ and ❺ respectively, they sense a busy channel and defer their transmissions to the end of the ongoing transmission.
7. When station *A*'s transmission ends and an idle channel is detected by all other stations at ❻, we assume that station *A* has another data frame for transmission.
8. Upon the detection of an idle channel at ❻, all the three stations start their transmissions resulting in a collision. At this point, since station *A* is transmitting a new data frame, its Collision Counter is zero before executing the backoff, while the Collision Counters of both stations *B* and *C* are two. Hence station *A* has a higher probability to acquire the broadcast channel after the collision at ❻ which is unfair to others.

Figure 2.32: Illustration of the capture effect in Ethernet

2.8.1 Delay Performance Analysis

Many works had been conducted in the past to analyze the performance of the 1-persistent CSMA/CD protocol that is implemented in Ethernet. The performance measures of the analyses are usually the throughput of the protocol under Poisson or Poisson-like traffic [ToHu80, TaKl85, TaKl85, SoMV87], and/or the transmission delay experienced by each station [Lam80, ToHu80, Bux81, Toba82a, BeGa92]. Unfortunately, none of the performance results can be used to describe the performance of the IEEE 802.3 MAC protocol under actual operation since all of them assume a Poisson or Poisson-like arrival process, and they do not include the BEB retransmission algorithm in their models.

One of the analyses using the closest model to the IEEE 802.3 MAC protocol is the analysis presented by Lam [Lam80]. In the following, the

analysis for the 1-persistent CSMA/CD protocol performed by Lam is reviewed.

We begin by describing the assumptions made in the analysis, listed in the following.

1. Stations are arranged in a star topology connected to a multiport repeater, sometimes known as an Ethernet hub. The distance between any pair of stations is fixed. The signal propagation time from a station to all other stations is τ units of time.
2. The channel is slotted so that each station can only start its transmission in the beginning of a slot.
3. The collision detection time is ignored. The jam signal transmission is not included.
4. The retransmission algorithm used in this analysis is an optimum one that will guarantee the stability of the broadcast channel such as the Binary Tree Algorithm. The acquisition probability of each slot, *H*, is assumed to be a constant, and it is equal to e^{-1}.
5. The arrival process is a Poisson process generated from a large population of stations. The aggregated arrival rate is λ. This assumption is the same as the one used for Aloha and other protocols in previous sections (see the first assumption presented in Table 2.1).

Based on assumptions 1 and 3, to guarantee a normal operation of the 1-persistent CSMA/CD protocol, the duration of a slot, denoted *T*, must be at least 2τ. Figure 2.33 illustrates a snapshot of the broadcast channel. The random variables used in this analysis are described as follow.

q_n = number of ready stations in the queue after the departure of the n-th transmission;

y_{n+1} = time between the detection of the end of n-th transmission and the beginning of (n+1)th transmission;

u_{n+1} = number of new arrivals during y_{n+1} period;

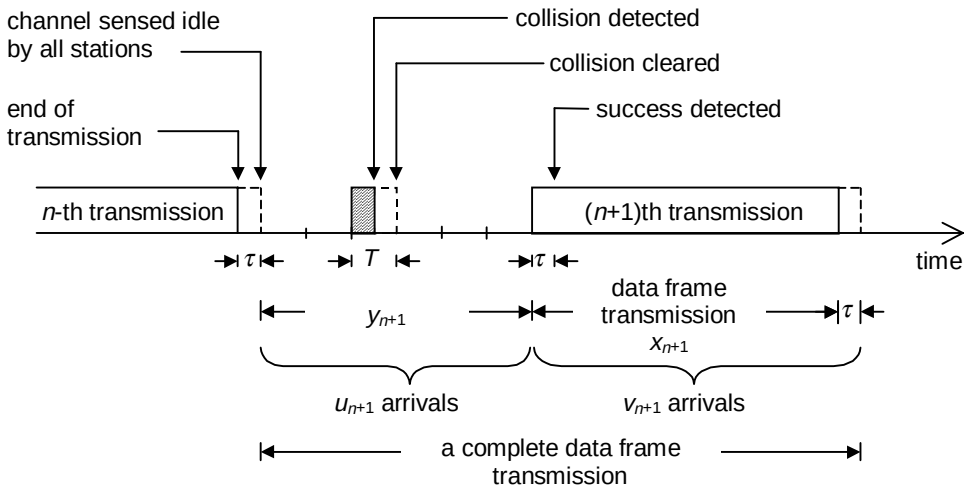


Figure 2.33: A snapshot of the 1-persistent CSMA/CD channel based on Lam's model

x_{n+1} = transmission time of a data frame;

v_{n+1} = number of new arrivals during $x_{n+1}+\tau$ period;

To obtain the average transmission delay, it is required to find the queue size distribution

$$Q_k = \lim_{n\to\infty} \Pr\{q_n = k\}, \qquad k = 0,1,2,\ldots. \tag{2-58}$$

The queue size, q_n, can be expressed recursively as

$$q_{n+1} = q_n + u_{n+1} + v_{n+1} - 1. \tag{2-59}$$

Since Poisson arrival process is assumed, the number of arrivals in each period solely depends on the duration of the period. According to Figure 2.33, to compute u_{n+1} and v_{n+1}, we need to find y_{n+1} and x_{n+1}.

Assume that the size of data frames follows a certain distribution, $\beta(x)$, with mean b_1, second moment b_2, and corresponding Laplace transform $\beta^*(s)$. Let $B(x)$ be the probability distribution function of $x_{n+1} + \tau$. Hence the Laplace transform of $B(x)$ is

$$B^*(s) = \beta^*(s)e^{-s\tau}. \tag{2-60}$$

Using the previous result, the z-transform of v_{n+1} is $B^*(\lambda\text{-}\lambda z)$.

To find u_{n+1}, we first realized that the random variable y_{n+1} is the sum of an idle period, I_{n+1}, and a contention period, r_{n+1}. That is $y_{n+1} = (I_{n+1} + r_{n+1})T$. An idle period starts when an idle channel is detected, it lasts as long as no arrival appears on the channel. If the probability that j arrivals appear in a slot is p_j, then given that there is no ready station after the completion of the n-th transmission, the probability density function of I_{n+1} is

$$\Pr\{I_{n+1} | q_n = 0\} = (1 - p_0) \cdot p_0^{k-1}, \qquad k = 1,2,\ldots. \tag{2-61}$$

When a collision occurs, given the acquisition probability, H, we get

$$\Pr\{r_{n+1} | \text{collision occurred}\} = H \cdot (1 - H)^{k-1}, \qquad k = 1,2,\ldots. \tag{2-62}$$

Finally, u_{n+1} can be obtained as

if $q_n = 0$,

$$u_{n+1} = \begin{cases} 1, & \text{with probability } \dfrac{p_1}{1 - p_0} \\ j + \text{number of arrivals during } r_{n+1} \text{ period} & \text{with probability } \dfrac{p_j}{1 - p_0} \\ & j = 2,3,\ldots \end{cases} \tag{2-63}$$

if $q_n = 1$,

$u_{n+1} = 0$

if $q_n \geq 2$,

$u_{n+1} = \text{number of arrivals during } r_{n+1} \text{ period}$.

Let $C^*(s)$ be the Laplace transform of a contention period, the z-transform of the contention period is then $C^*(\lambda\text{-}\lambda z)$. With all the developed results, and the following z-transform for Q_k,

$$Q(z) = \sum_{k=0}^{\infty} Q_k z^k \tag{2-64}$$

$Q(z)$ can be obtained as

$$Q(z) = B^*(\lambda - \lambda z)\left\{ Q_1 z\left[1 - C^*(\lambda - \lambda z)\right] + \frac{Q_0}{1 - p_0}\left[p_1 z(1 - C^*(\lambda - \lambda z)) - C^*(\lambda - \lambda z)(1 - e^{-\lambda T(1-z)})\right]\right\} \times (z - B^*(\lambda - \lambda z)C^*(\lambda - \lambda z))^{-1} \quad (2\text{-}65)$$

where

$$Q_0 = \frac{1 - \lambda(b_1 + \tau + T/H)}{\lambda T\left(\frac{1}{1 - p0} - \frac{1}{B^*(\lambda)H}\right)} \quad (2\text{-}66)$$

and

$$Q_1 = \left(\frac{1}{B^*(\lambda)} - \frac{p_1}{1 - p_0}\right)Q_0 . \quad (2\text{-}67)$$

Using Little's formula [Litt61], the mean transmission time, D, is found to be

$$D = \bar{x} + \frac{T}{H} + \frac{T}{H} - \frac{1 - p_0}{2[B^*(\lambda)H - (1 - p_0)]} \times \left(\frac{2}{\lambda} + HT - 3T\right) + \frac{\lambda[\overline{x^2} + 2\bar{x}(T/H) + T^2(1 + 2(1 - H)/H^2)]}{2[1 - \lambda(\bar{x} + T/H)]} \quad (2\text{-}68)$$

with $\bar{x} = b_1 + \tau$ and $\overline{x^2} = b_2 + 2b_1\tau + \tau^2$.

Define the normalized propagation delay, a, to be

$$a = \frac{\tau}{b_1} \quad (2\text{-}69)$$

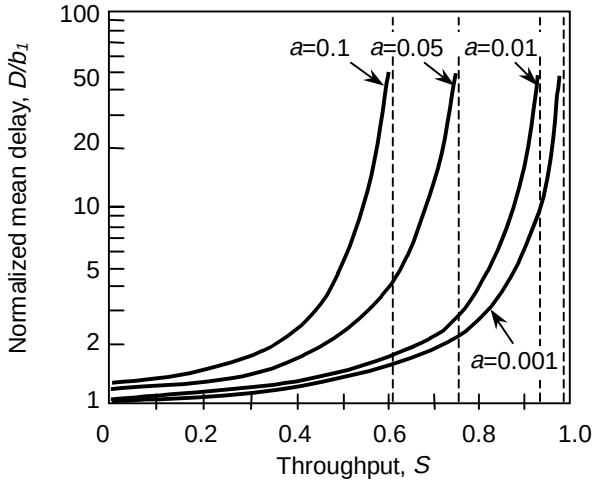


Figure 2.34: Mean transmission delay versus throughput for 1-persistent CSMA/CD based on Lam's model

and the system throughput, *S*, to be

$$S = \lambda b_1 \tag{2-70}$$

it is found that the upper bound of the system throughput is

$$S < \frac{H}{2a + (1+a)H} \ . \tag{2-71}$$

In Figure 2.34, the delay performance of 1-persistent CSMA/CD with an adaptive retransmission algorithm is plotted. Using computer simulation, it is found that the analytical results are applicable to the retransmission algorithm with rules defined as follows [Lam80]:

(R1) Each station reset its Collision Counter, *c*, to zero initially. The Collision Counter is also reset after a successful data frame transmission.

(R2) Upon a detection of a collision, the station increases its Collision Counter by one, and it retransmits its data frame in the next slot with probability $(0.5)^c$. If a collision is detected again during a retransmission attempt, (R2) is repeated. Otherwise if its retransmission is successful, (R1) is used.

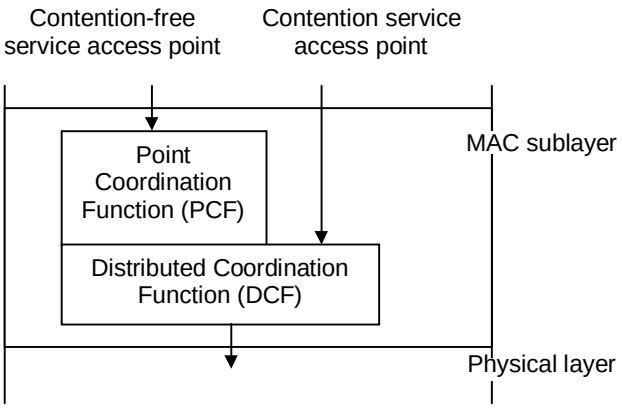


Figure 2.35: IEEE 802.11 MAC architecture

Notice that the above retransmission algorithm is similar but not identical to BEB. It is far less aggressive than BEB.

The results shown in Figure 2.34 verify that the shorter propagation delay will lead to a higher achievable throughput on the broadcast channel. The results also indicate that the protocol generally performs well under Poisson traffic in the range from light to moderate.

2.9 The IEEE 802.11 MAC Protocol

The IEEE 802.11 working group has chosen the CSMA with Collision Avoidance (CSMA/CA) protocol for wireless LANs [IEEE98]. CSMA/CA is also a random access protocol. It was derived from MACAW presented by Bharghavan et al. [BDSZ94], a protocol based on MACA originally proposed by Karn [Karn90].

Apart from the CSMA/CA protocol, the IEEE 802.11 MAC protocol also specifies an additional optional medium access scheme mainly for transmitting time sensitive information. Consequently, two functions are created in the MAC sublayer. They are depicted in Figure 2.35. For a contention-free service, a Point Coordination Function (PCF) is used, otherwise, a Distributed Coordination Function (DCF) is used.

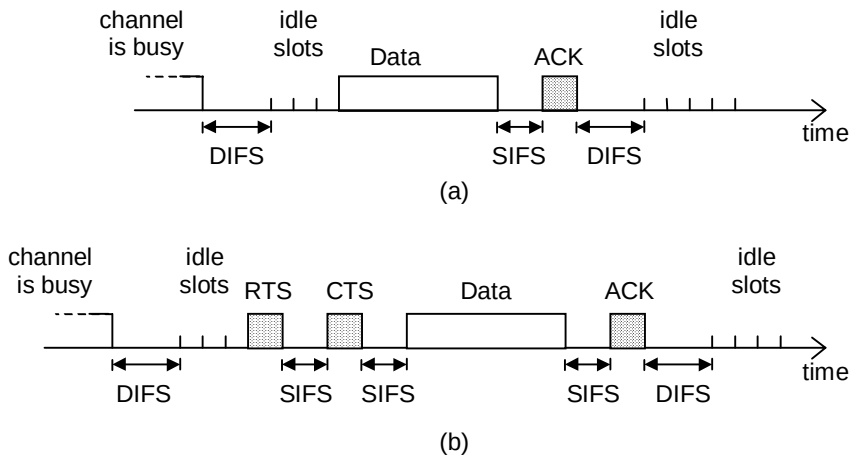


Figure 2.36: IEEE 802.11 MAC protocol access methods: (a) the basic access method; (b) the fourway handshaking access method

The PCF is based on a polling scheme to provide a contention free service. As it is not the primary concern of our study, its operation will not be covered here. The detailed operation of the PCF can be found in [IEEE 98].

The DCF is based on the CSMA/CA protocol. The IEEE 802.11 working group further specifies two access methods, which are the basic access method and the optional fourway handshaking access method.

Figure 2.36 illustrates the CSMA/CA protocol operation. Under the basic access method, a station, when ready for a new data frame transmission, first senses the channel status. If the channel is found to be busy, the station defers its transmission and continues to sense the channel until it is idle. After the channel is idle for a specified period of time called the distributed interframe space (DIFS) period, the station chooses a random number as a backoff timer. Note that the time immediately after the DIFS period is slotted. The slot duration is at least the time required for a station to detect an idle channel plus the time required to switch the transceiver from a listening mode to a transmitting mode. The backoff timer is decreased by one for each idle slot, stopped if the channel is sensed busy, and then reactivated if the channel turns idle and remains idle for more than a DIFS time period. When the backoff timer reaches zero, the data frame is transmitted.

The choice of the random number for the backoff timer is based on a slightly modified version of BEB. A random integer number is chosen randomly and uniformly from a backoff window [0,*CW*], where *CW* is the Contention Window parameter. Initially, the Contention Window parameter is set to CW_{min}. For every collision a station experienced, its *CW* value is doubled. Once its *CW* value reaches a predefined CW_{max}, its *CW* value remains at CW_{max}. Like BEB, when a station successfully transmitted a data frame, its backoff window is reset to [0,CW_{min}].

To determine whether a data frame transmission is successful, after the completion of a data frame transmission from a sender, the receiver replies to the sender with a positive acknowledgement (ACK) indicating the successful delivery of the data frame. The ACK is transmitted after a short interframe space (SIFS) period when the entire data frame has arrived with no errors. If ACK is not detected within a SIFS period plus the signal propagation time after the completion of the data frame transmission, the transmission is considered unsuccessful, and a retransmission is required.

In the case of the four-way handshaking access method, an additional operation is introduced on top of the basic access method before a data frame transmission is taken place. Before a data frame is transmitted, under the four-way handshaking access method, a station first transmits a Request To Send (RTS) control frame to request for a transmission right. Upon receiving the RTS frame, the receiver replies with a Clear To Send (CTS) control frame after a SIFS period. Once the RTS/CTS is exchanged successfully, the sender then transmits its data frame.

During the RTS/CTS exchange, the duration of a transmission is also specified. All other stations within the radio coverage will be aware of the transmission and its transmission time before it actually occurs. In the IEEE 802.11 MAC protocol, the transmission duration is used by those stations to predict the duration of the busy channel without physically sensing the channel. This is known as the virtual carrier sense mechanism, or the

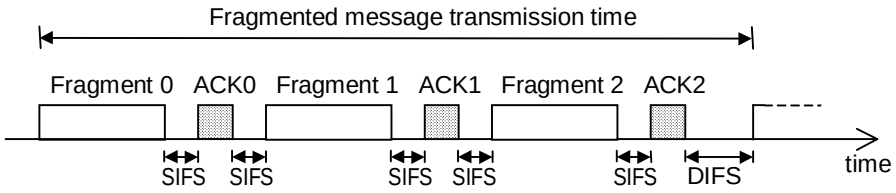


Figure 2.37: IEEE 802.11 MAC protocol fragmentation operation

network allocation vector (NAV). However, the introduction of NAV has no effect on the performance.

In the case where a sender wishes to transmit a message that is longer than the specified maximum IEEE 802.11 data frame size, the message must be fragmented before transmissions. The SIFS is used to provide an efficient transmission of the fragments of the message. As depicted in Figure 2.37, when the sender seized the channel and transmitted the first fragment, it may continue to transmit the next fragment if an acknowledgement of the previous fragment is returned. The sender can continue to transmit its fragments until either it has completed all the transmissions of the fragments, or a further fragment transmission will not cause the station to hold the broadcast channel for a duration longer than the predefined time duration called the *dwell time boundary* [IEEE 98].

In Figure 2.38, we present the operation of a transmitter implementing the IEEE 802.11 MAC protocol. For the purpose of understanding the IEEE 802.11 MAC protocol operation, we do not include the PCF, the NAV and the fragmentation operations.

The performance of the IEEE 802.11 MAC protocol or CSMA/CA has been studied extensively in the literature. Ho et al. studied the performance of CSMA/CA, with a retransmission algorithm different from the actual algorithm specified in the standard [HoCh96]. The considered retransmission algorithm is much simpler. In addition, a Poisson-like arrival process is assumed. Yue et al. analyzed CSMA/CA using a model similar to the linear feedback model [YuMa00]. Similar to the work done by Ho et al., a simpler retransmission algorithm is considered. Cali et al. provided the

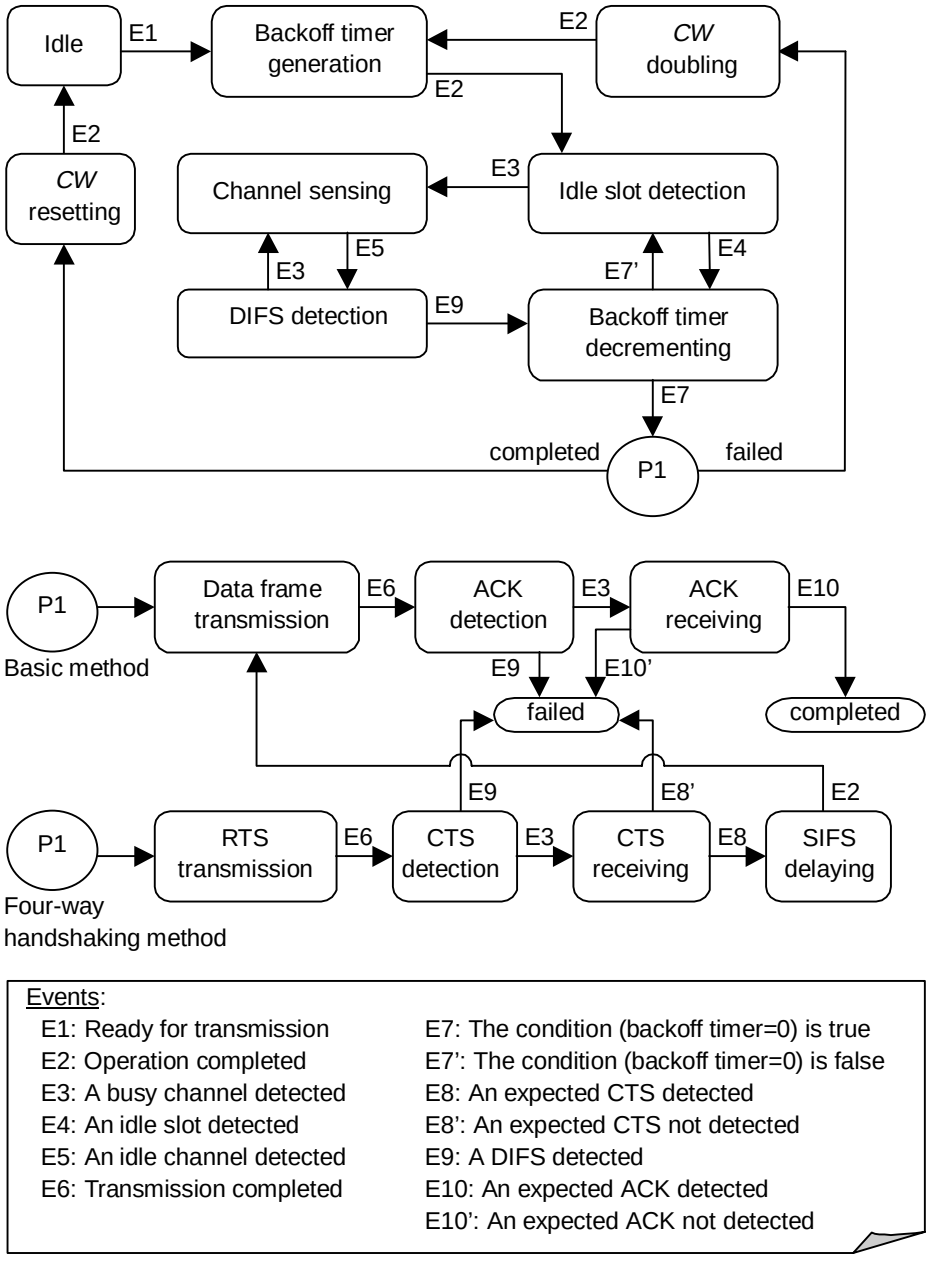


Figure 2.38: The finite state machine of the transmitter of the IEEE 802.11 MAC protocol

analysis of the IEEE 802.11 MAC protocol focusing on the channel capacity [CaCG98, CaCG00]. A simplified retransmission algorithm is used in their model.

One of the most accurate performance analyses for the IEEE 802.11 MAC protocol capturing all protocol details including the retransmission algorithm specified in the standard is performed by Bianchi

[Bian98,Bian00]. Both analyses presented in [CaCG98, CaCG00] and [Bian98,Bian00] measure the channel utilization under a continuous channel overloaded load traffic condition.

The review of the performance analysis for the IEEE 802.11 MAC protocol is given in Chapter 3, based on the analysis carried out by Bianchi.

2.10 Discussion

Understanding the performance of the IEEE 802.3 and the IEEE 802.11 MAC protocols is an important step to optimize the efficient use of the protocols and the efficient allocation of network resources. Performance analysis is the process of the development and study of mathematical models that predict the performance of networks in some well-defined sense [HaOr88]. In order to understand the behavior of the IEEE 802.3 and the IEEE 802.11 MAC protocol, performance analysis of the two MAC protocols should be conducted under the assumptions that is applicable to the "real-world".

Due to the random nature of the random access protocol operation, performance analysis for a random access protocol under a realistic traffic condition has been considered difficult. Virtually all analyses reviewed or cited in this chapter introduce certain model simplifications. Because of the employed simplifications, the application of the analytical results is limited.

Among the assumptions introduced for model simplifications, one of the most widely applied assumptions is the Poisson Load Traffic assumption as given in Table 2.1. The assumption states that the number of stations in a network is infinite, each station carries at most one data frame in its local buffer, and the aggregated arrival process is a Poisson process. This assumption has been used in many of the analyses for various random access protocols, especially in many early works. The weaknesses of this assumption are (i) in a LAN, the number of stations is usually small, and (ii) as indicated in [LTWW94], the LAN traffic is self-similar rather than

Poisson. Hence, the analyses based on this assumption are incapable to provide a good estimation of the performance of the actual network.

One attempt to make the analytical model more realistic is the use of the linear feedback model. In this model, a fixed number of stations is considered, each station is equipped with a buffer that can only hold one data frame. A station, if its buffer is empty, generates a new data frame with a certain probability during a given time period. However, the analyses based on this model often use a simple retransmission algorithm because a complex retransmission algorithm such as BEB makes the analysis intractable. Hence the application of the obtained results to a protocol implementing the BEB retransmission algorithm may introduce errors.

In [BoMK88], Boggs et al. reported that earlier efforts in analyzing CSMA/CD generally exhibit the following drawbacks: (i) many analyses assume a protocol that is not the IEEE 802.3 MAC protocol; (ii) to achieve analytic tractability, many analyses simplify their models by some assumptions such as assuming an infinite number of stations, constant data frame size, etc; (iii) the performance measures of the analyses are not particular useful.

Despite the study and criticisms reported by Boggs et al. in [BoMK88], many recent works in performance analyses of the IEEE 802.3, and the IEEE 802.11 MAC protocols reviewed or cited in this chapter still suffer from the shortcomings addressed in [BoMK88]. It is mainly because the analytic tractability of the analyses is achieved through simplifications of the protocol and the traffic arrival models.

To achieve a good prediction of the performance of a protocol, it is important to include all the detailed operations of the protocol in the protocol modeling for performance analysis. Moreover, any simplification introduced in the analysis should be justified. As for the traffic arrival model, a realistic traffic model should be considered. In the next chapter, we

present the performance analysis for MAC protocols that fulfills the requirements described above.

3 Performance Analysis

In the process of protocol designs and deployments, it is important to understand the performance of the protocol so that the protocol parameters can be tuned to achieve an optimum operation in an actual network environment. In the previous chapter, we reviewed some important random access MAC protocols and their performances. To make the performance analyses tractable, most presented performance analyses introduce some assumptions to simplify the protocol operation and/or the traffic arrival process. As a result, the obtained analytical results may not be realistic and their applications are somewhat limited.

To provide an accurate prediction of the protocol performance, it is essential to develop an analytical model that resembles the detail protocol operation as well as the traffic characteristics of an intended network operation environment. In this chapter, we first discuss two realistic scenarios that may occur in a LAN – the saturation and the disaster scenarios. The saturation scenario represents a continuous overload condition. The results of this scenario indicate a fundamental limit of a protocol – its worst performance for a given number of stations. As for the disaster scenario, it models the response of a protocol to the recovery (power up) from a major failure. This situation is likely to occur in a LAN when the shared channel in the network is temporarily inaccessible due to, for example, a broken cable or a long period of noise.

We then study the performance of the IEEE 802.3 and IEEE 802.11 MAC protocols under the two realistic scenarios. We analyze the throughput of the protocols under the saturation scenario, the duration of the recovery process under the disaster scenario, and the mean transmission delay that a station experienced under the scenarios.

Moreover, by using system approximation technique, we reuse the results obtained from the saturation throughput analysis for the study of the protocol performance under a statistical traffic arrival process. This approach allows the study of a protocol under a realistic traffic arrival process without compromising the detail of the protocol operation. Several performance studies are conducted and the analytical results are verified by simulation.

This chapter is organized as follows. In Section 3.1, we explain and discuss the saturation and the disaster scenarios in a LAN. In Sections 3.2 and 3.3, we analyze the IEEE 802.3 and the IEEE 802.11 MAC protocols respectively under the two realistic scenarios. In section 3.4, we introduce a new approach for performance evaluation of a MAC protocol using Markovian framework. Sections 3.5 and 3.6 provide several performance evaluations of the IEEE 802.3 and the IEEE 802.11 MAC protocols under realistic traffic assumptions using the Markovian framework. The verification of the obtained numerical results by simulation is also given.

3.1 The Saturation and the Disaster Scenarios

3.1.1 The Saturation Scenario

We consider that there are *m* stations in a shared LAN using a certain MAC protocol. All *m* stations share a common channel for data frame transmission. Under the saturation scenario, the stations are saturated in the sense that after a completion of each data frame transmission, the station immediately generates a new data frame for transmission. This scenario models the busiest situation in a network where the local buffer in each station is never empty. The results of this scenario indicate a fundamental limit of a protocol, that is the worst performance of a protocol for a given number of stations. The size of the generated data frames may be fixed or variable which can be described by a certain distribution function.

For this scenario, we measure the saturation throughput of a protocol and the average transmission delay experienced by a station.

3.1.2 The Disaster Scenario

The disaster scenario is termed by IEEE 802.14 workgroup to describe a power up situation in the hybrid fiber coax (HFC) networks [Bisd96]. Consider that there are *m* stations in a shared network, under this scenario, each station is assumed to carry a data frame for transmission, and all *m* data frame transmissions are commenced at the same time. Once a data frame is successfully transmitted, the station will remain idle. The process for all *m* stations to successfully transmit their data frames is defined as a *recovery process* of the disaster scenario. By the time the last station completes its data frame transmission, the recovery process ends.

Such a scenario is also likely to occur in a LAN when the common channel of the LAN is temporarily inaccessible because of a broken cable, a faulty hub, a long period of noise or other factors.

For this scenario, we mainly focus on the time it takes to clear the backlog of *m* stations, that is the duration of the recovery process. We also measure the mean transmission delay experienced by a station during the process.

3.2 IEEE 802.3 under the Saturation and the Disaster Scenarios

The description of the IEEE 802.3 MAC protocol is given in Section 2.8, in this analysis, we follow the assumptions made by Lam in [Lam80]. We assume that

1. Stations are arranged in a star topology connected to a multiport repeater, sometimes known as an Ethernet hub. The distance between any pair of stations is fixed. The signal propagation time from a station to all other stations is τ units of time.

2. The channel is slotted so that each station can only start its transmission in the beginning of a slot.

3. The collision detection time is ignored. The jam signal transmission is not included.

4. To use a common mechanism for detecting a collision and aborting collided transmissions in a slotted channel, the minimum duration of a slot is 2τ [Lam80]. Here we assume that the slot time is 2τ.

5. As discussed in Section 2.8, the current IEEE 802.3 MAC protocol exhibits the capture effect causing temporary performance boost and unfairness. To avoid this transient effect influencing the steady state performance results, in the analysis, we make a slight modification to the BEB retransmission algorithm. We adopt the solution proposed by Molle [Moll94] and consider that when a backlog station detects a successful transmission, it resets its Collision Counter so that after a successful transmission, all stations access the channel with the same probability. The BEB retransmission algorithm remains the same. This version of retransmission algorithm provides a conservative prediction to the performance.

In the following subsections, the IEEE 802.3 MAC protocol will be analyzed and simulated under the saturation and the disaster scenarios based on the assumptions given in the above.

3.2.1 The Saturation Analysis

In Figure 3.1, we observe that the broadcast channel is repeating a cycle that consists of (i) an idle period; (ii) a contention period; and (iii) a data frame transmission period. All cycles are statistically identical. Let $I(m)$, $C(m)$, $T(m)$ denote the random variables of the idle, contention, data frame transmission periods respectively for the case of m saturated stations, $m \geq 1$. They are expressed in slots. Furthermore, let $T_u(m)$ be the period within the data frame transmission that the channel carries useful information, and

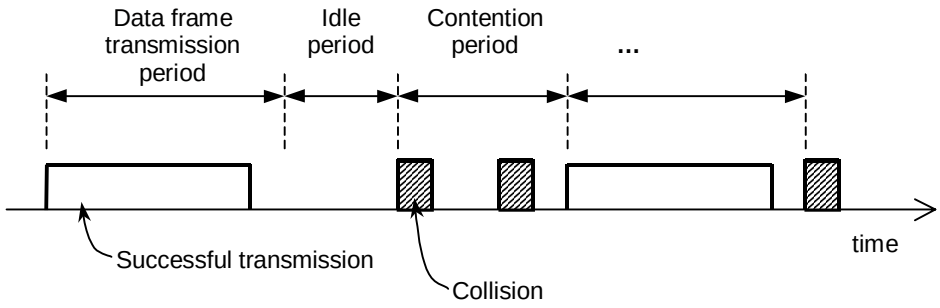


Figure 3.1: A snapshot of the IEEE 802.3 broadcast channel under the saturation scenario

$T_w(m)$ be the overhead period within the data frame transmission. The sum of $T_u(m)$ and $T_w(m)$ is equal to $T(m)$.

As usual, define the channel throughput to be the fraction of time that useful information is carried on the channel. Given m saturated stations, the saturation throughput, S, is

$$S = \frac{\mathrm{E}[T_u(m)]}{\mathrm{E}[I(m) + C(m) + T_u(m) + T_w(m)]}. \tag{3-1}$$

Under the saturation scenario, after a successful data frame transmission by a station, a new data frame is immediately generated for transmission. Thus, $I(m)$=0 since no idle period can be expected.

For the contention period, according to the 1-persistent CSMA/CD protocol in the IEEE 802.3 standard, it is expected that all the saturated stations will participate in the contention period and execute the BEB retransmission algorithm if a retransmission attempt is unsuccessful. Hence the duration of the contention period depends on the collision resolution time of BEB given m stations.

The analysis of the BEB retransmission algorithm has been performed by Molle [Moll94] and reviewed in Section 2.7.1. The result of the analysis is given in Equation (2-57). The analysis measures the average number of slots (including the successful transmission slot), $L(m)$, required for the channel to obtain a successful transmission among m stations after a collision has

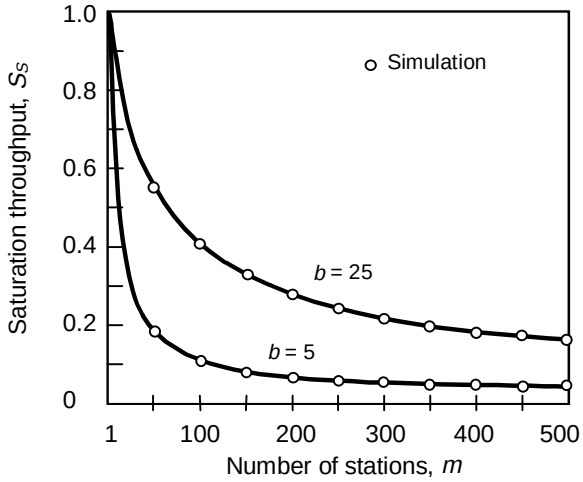


Figure 3.2: Saturation throughput for IEEE 802.3 for two different data frame sizes

occurred. Hence the mean contention period, E[$C(m)$], is $L(m)$ minus the successful transmission slot, that is

$$\mathrm{E}[C(m)] = L(m) - 1. \tag{3-2}$$

For the data frame transmission period, we assume that the data frame itself is the payload, the overhead due to the header transmission is ignored. The data frame size follows a certain distribution function of which the mean transmission time of the data frame is b units of time expressed in slots. Moreover, due to the signal propagation delay, all stations detect an idle channel τ units of time (or 0.5 slot time) after the end of the data frame transmission. Hence

$$\begin{aligned} \mathrm{E}[T_u(m)] &= b \\ \mathrm{E}[T_w(m)] &= 0.5. \end{aligned} \tag{3-3}$$

Since $I(m)$, $C(m)$, $T_u(m)$ and $T_w(m)$ random variables are statistical independent, then Equation (3-1) becomes

$$S = \frac{\mathrm{E}[T_u(m)]}{\mathrm{E}[I(m)] + \mathrm{E}[C(m)] + \mathrm{E}[T_u(m)] + \mathrm{E}[T_w(m)]}. \tag{3-4}$$

With all the developed results, the saturation throughput is readily obtainable. Figure 3.2 plots the saturation throughput of the IEEE 802.3 MAC protocol versus the number of saturated stations for different data frame size. The numerical results (shown in lines) are verified by simulation results (shown in symbols). The parameters used in the numerical computation and computer simulation are summarized in Table 3.1.

The saturation throughput curves indicate the worst performance of IEEE 802.3 given a number of stations in a LAN. In a normal operation, the average throughput of IEEE 802.3 should remain in the region above the saturation throughput curve.

Comparing the saturation throughput of the protocol with the two different data frame sizes, it is evident that the protocol performs better if a larger data frame size is used. This result is consistent with the classical analyses assuming Poisson traffic given in the previous chapter. The figure also shows that the performance of IEEE 802.3 drops as the number of stations increases. This performance degradation is more significant for the case of a shorter data frame.

To compute the mean transmissions delay experienced by a station in the saturation scenario, we notice that at any given time, each of the *m* stations is either in the waiting state or in the data frame transmission state. In other

Table 3.1: Summary of the protocol parameters for the IEEE 802.3 saturation and disaster analyses

Parameter	Value
Protocol used	slotted CSMA/CD with BEB
Channel bit rate	10 Mb/s
Signal propagation time, τ	25.6 μsec (256 bit time)
Slot time	2τ (512 bit time)
Cost of a collision	one slot
The time required for detecting a transmission end	τ
Number of stations, *m*	1,2,…,500
Data frame or payload size (headers are ignored), *b*	(i) 5 slots (320 bytes) (ii) 25 slots (1600 bytes)

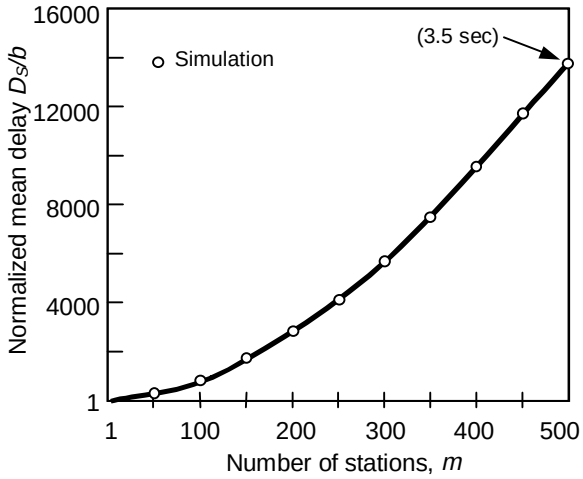


Figure 3.3: Normalized mean transmission delay for IEEE 802.3 under the saturation scenario for b=5

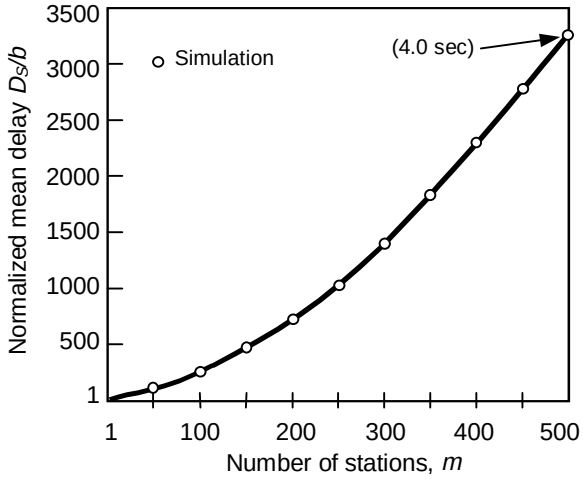


Figure 3.4: Normalized mean transmission delay for IEEE 802.3 under the saturation scenario for b=25

words, there are always m stations in a queue, including the station which is transmitting its data frame. Furthermore, after a successful transmission, the station will immediately generate a new data frame, thus the arrival of a data frame occurs at the same time as the departure of a data frame. Hence, the arrival rate is equal to the departure rate.

According to Little's formula [Litt61], the mean waiting time, that is the time between a data frame is generated and the data frame is successfully transmitted can be computed by

$$\overline{L} = \overline{\lambda} \cdot \overline{W} \tag{3-5}$$

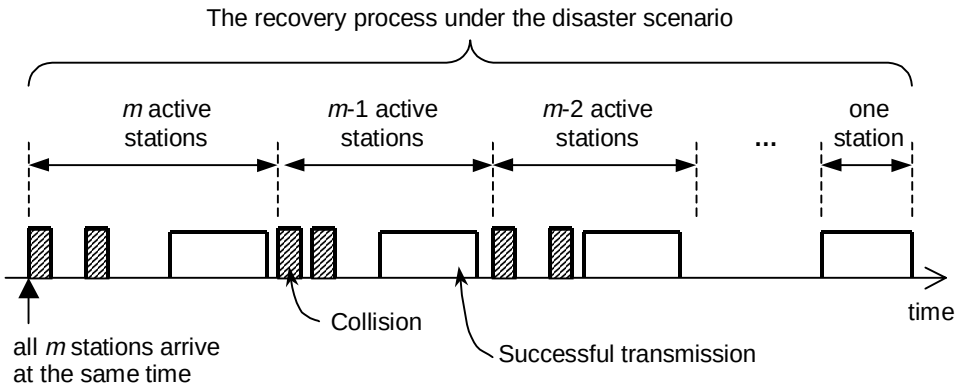


Figure 3.5: A snapshot of the IEEE 802.3 broadcast channel under the disaster scenario

where $\overline{L}$ is the average queue size, $\overline{\lambda}$ is the average arrival rate and $\overline{W}$ is the average waiting time or the mean transmission delay, D_S, experience by a station.

In the saturation scenario, the average queue size is m, the average arrival rate is equal to the service rate of IEEE 802.3. Hence we get

$$\begin{aligned} \overline{L} &= m \\ \overline{\lambda} &= \frac{1}{\mathrm{E}[I(m) + C(m) + T_u(m) + T_w(m)]} . \end{aligned} \tag{3-6}$$

Using Equations (3-5) and (3-6) together with all the developed results, the mean transmission delay can be obtained. Figures 3.3 and 3.4 show the mean transmission delay for IEEE 802.3 normalized to b versus the number of saturated stations with different data frame sizes. The numerical results (drawn in lines) are verified by the simulation results (drawn in symbols) based on the protocol parameters given in Table 3.1. One interesting observation from the results is that the delay curves are non-linear, the mean delay is relatively high under a larger number of saturated stations. This result suggests that the IEEE 802.3 MAC protocol it is not suitable for a network with a large number of active stations.

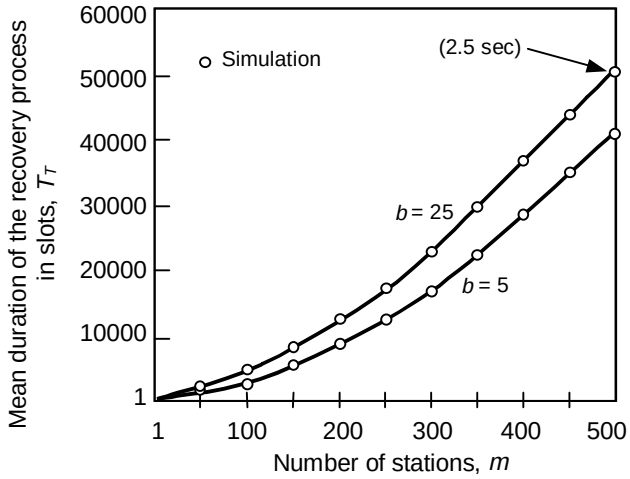


Figure 3.6: Mean duration of the recovery process versus number of stations for IEEE 802.3 under the disaster scenario

3.2.2 The Disaster Analysis

Under the disaster scenario, all stations start to transmit at the same time and each station transmits only one data frame in the entire recovery process. We first focus on the total time required for all m data frames to be transmitted successfully. As illustrated in Figure 3.5, we recognize that the recovery process consists of m cycles of which the first cycle can be seen as the saturation scenario of m stations, the second cycle is that of m-1 stations, and so on. Therefore, the duration of the entire process, T_T, can be expressed by

$$\begin{aligned} T_T &= \sum_{i=1}^{m} \left(\mathrm{E}[I(i) + C(i) + T_u(i) + T_w(i)] \right) \\ &= \sum_{i=1}^{m} \left(\mathrm{E}[I(i)] + \mathrm{E}[C(i)] + \mathrm{E}[T_u(i)] + \mathrm{E}[T_w(i)] \right) . \end{aligned} \tag{3-7}$$

For IEEE 802.3, the sum of the mean of each period during the recovery process can be described by the following equation set.

$$\begin{aligned}
&\sum_{i=1}^{m} \mathrm{E}[I(i)] = 0 \\
&\sum_{i=1}^{m} \mathrm{E}[C(i)] = \sum_{i=1}^{m} \big(L(i) - 1\big) \\
&\sum_{i=1}^{m} \mathrm{E}[T_u(i)] = m \cdot b \\
&\sum_{i=1}^{m} \mathrm{E}[T_w(i)] = \frac{1}{2} \cdot (m-1)
\end{aligned} . \tag{3-8}$$

By substituting Equation (3-8) into Equation (3-7), the duration of the recovery process can be computed. In Figure 3.6, we present the mean duration of the recovery process for IEEE 802.3 under the disaster scenario with two different data frame sizes. When the number of stations is as large as 500, the network requires around two seconds to recovery under the disaster scenario assuming that each station only transmits one data frame. The recovery process may take longer time to complete if a station has more than one data frame to transmit during the process.

The transmission delay a station experienced during the recovery process can be expressed as

$$D_D = \frac{d_1 + d_2 + d_3 + \cdots + d_m}{m} \tag{3-9}$$

where d_k is the mean transmission delay experienced by the k-th station departing the network. According to Figure 3.5, which illustrates the IEEE 802.3 channel activity and timing under the disaster scenario, we have the following recursions

$$\begin{aligned}
&d_1 = (L(m) - 1) + b \\
&d_k = \big(d_{k-1} + ½\big) + \big(L(m-k+1) - 1\big) + b, \quad k = 2,3,\ldots,m .
\end{aligned} \tag{3-10}$$

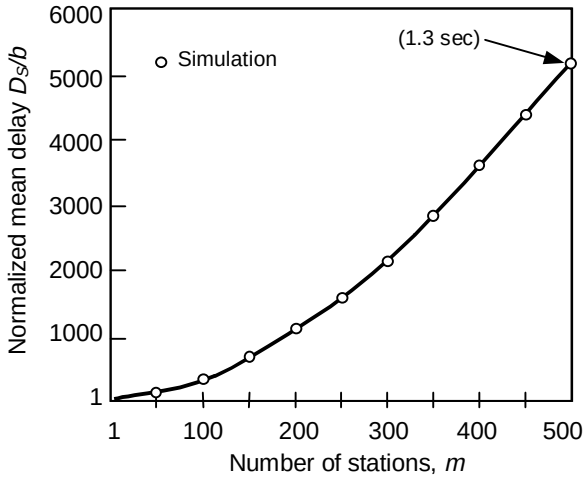


Figure 3.7: Normalized mean transmission delay for IEEE 802.3 under the disaster scenario for b=5

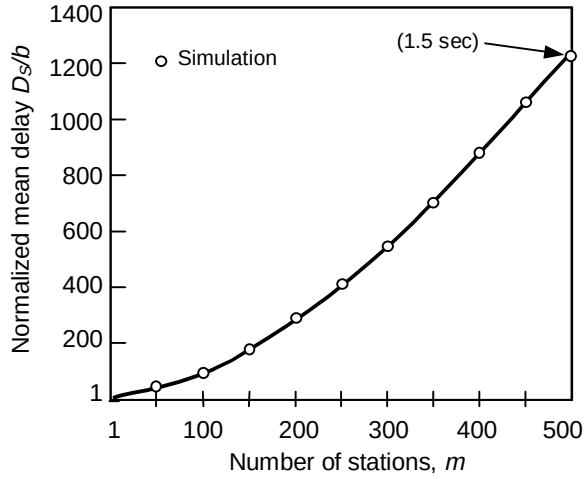


Figure 3.8: Normalized mean transmission delay for IEEE 802.3 under the disaster scenario for b=25

Combining Equations (3-9) and (3-10), we obtain the mean transmission delay a station experienced under the disaster scenario. The mean transmission delay versus the number of stations is plotted in Figures 3.7 and 3.8 based on the protocol parameters given in Table 3.1.

The results again suggest that the IEEE 802.3 MAC protocol performs reasonably well for a small population, but poor for a large population of stations in a LAN.

3.3 IEEE 802.11 under the Saturation and the Disaster Scenarios

The detail operation of the IEEE 802.11 MAC protocol is given in Section 2.9. In this section, we present the performance analyses of IEEE 802.11 under the two realistic scenarios, which are the saturation and the disaster scenarios. We make the following assumptions in the analyses

1. We consider only a fixed data frame size that is smaller than or equal to the maximum fragment size specified in the IEEE 802.11 standard.
2. All stations are stationary within the same radio coverage. The network contains no hidden or exposed stations.
3. The signal propagation between any pair of stations is the same.

3.3.1 The Saturation Analysis

The IEEE 802.11 performance evaluation under the saturation scenario presented in the following is based on Bianchi's work [Bian98, Bian00]. We first consider m saturated stations in a cell implementing the IEEE 802.11 MAC protocol. The broadcast channel is virtually slotted with three types of non-uniform slots, they are the idle slot, the collided slot, and the data frame transmission slot. Define τ to be the transmission probability of each station, which is the probability that a station attempts its transmission in a particular slot. Let p be the probability that an attempt of a transmission is unsuccessful.

Recall the operation of the backoff algorithm described in Section 2.9, before a data frame transmission, each station picks an integer random number from the range [0, CW-1] to decide when the transmission should be initiated. For a new data frame transmission, the value of CW is set to a predefined initial window size, CW_{min}. Each time when the transmission is unsuccessful, the contention window, CW, is doubled, until CW reaches a predefined maximum contention window, CW_{max}, then CW remains at CW_{max}. Let W_i be the contention window at the time a station experienced i

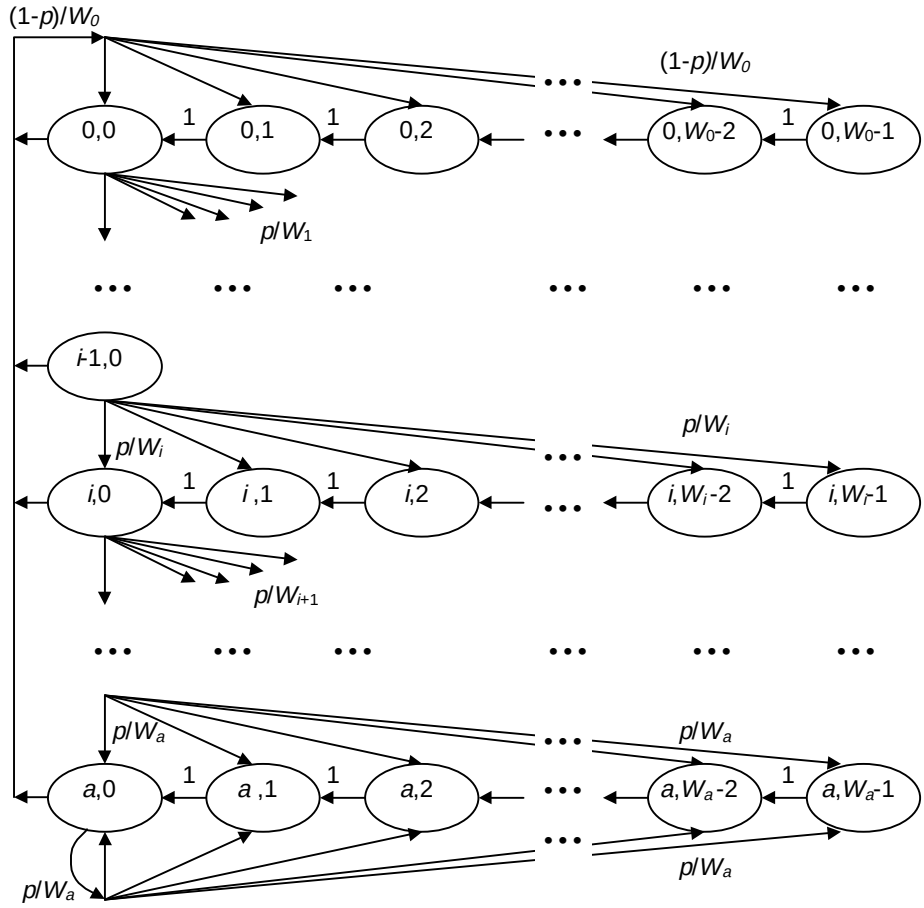


Figure 3.9: Markov chain model for the IEEE 802.11 backoff operation

unsuccessful transmission attempts, then $W_i=2^iW_0$ for $0\le i\le a$ with $W_0=CW_{min}$ and $W_a=CW_{max}$. The backoff operation of a saturated station can be described by a bidimensional discrete Markov chain shown in Figure 3.9.

The one step transition probabilities of the Markov chain from state $\{i_1,k_1\}$ to $\{i_2,k_2\}$, expressed as $\Pr\{i_1,k_1 \mid i_2,k_2\}$, are

$$\begin{aligned}
&\Pr\{i,k \mid i,k+1\}=1, && 0\le k\le W_i-2, \quad 0\le i\le a\\
&\Pr\{0,k \mid i,0\}=\frac{1-p}{W_0}, && 0\le k\le W_0-1, \quad 0\le i\le a\\
&\Pr\{i,k \mid i-1,0\}=\frac{p}{W_i}, && 0\le k\le W_i-1, \quad 1\le i\le a\\
&\Pr\{a,k \mid a,0\}=\frac{p}{W_a}, && 0\le k\le W_a-1.
\end{aligned} \tag{3-11}$$

Let $\pi_{i,k}$ be the stationary probability of state $\{i,k\}$, according to the transition probabilities developed in Equation (3-11), we get

$$\begin{aligned}
&\pi_{i,0} = p^i \pi_{0,0}\,, \qquad 0 < i < a \\
&\pi_{a,0} = \frac{p^m}{1-p}\pi_{0,0} \\
&\pi_{i,k} = \frac{W_i - k}{W_i}\pi_{i,0}\,, \qquad 0 < k < W_i \\
&1 = \sum_{i=0}^{a}\sum_{k=0}^{W_i-1}\pi_{i,k}\,.
\end{aligned} \tag{3-12}$$

From the very last result, we yield

$$\pi_{0,0} = \frac{2(1-2p)(1-p)}{(1-2p)(W_0+1)+pW_0\left(1-(2p)^a\right)}\,. \tag{3-13}$$

A station starts its data frame transmission in a particular slot when its backoff counter is decreased to zero. Hence the transmission probability, τ, can be expressed as

$$\tau = \sum_{i=0}^{a}\pi_{i,0} = \frac{\pi_{0,0}}{1-p} = \frac{2(1-2p)}{(1-2p)(W_0+1)+pW_0\left(1-(2p)^a\right)}\,. \tag{3-14}$$

Assume that the stations are independent, the probability that an attempt of a transmission in a particular slot suffers a collision, p, is when at least one of other stations also chooses to transmit in the same slot. That is

$$p = 1-(1-\tau)^{m-1}\,. \tag{3-15}$$

The two variables τ and p can now be obtained numerically by solving Equations (3-14) and (3-15). Define the channel throughput, S, to be the fraction of time that useful information is carried on the channel, we get

$$S = \frac{P_s P_{tr}\mathrm{E}[P]}{(1-P_{tr})\sigma + P_s P_{tr} T_s + P_{tr}(1-P_s)T_c} \tag{3-16}$$

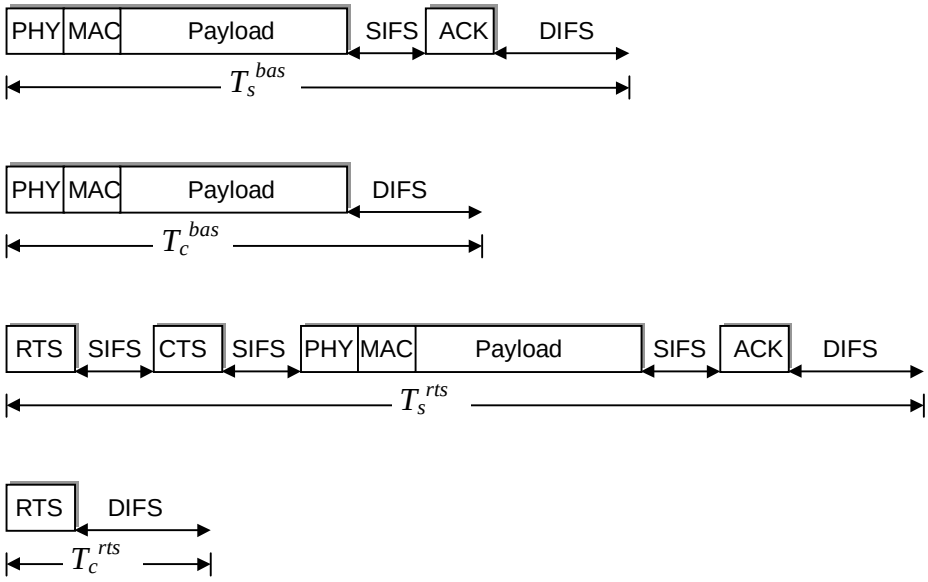


Figure 3.10: Time duration of the data frame transmission and collision for the basic and the four-way handshaking access methods

where P_{tr} is the probability that there is at least one transmission attempt in a particular slot, P_s is the probability that an attempt of a transmission in a slot is successful given that the slot is not an idle slot, E[P] is the mean transmission time of a payload during a data frame transmission, T_s is the average time the channel is sensed busy due to a successful transmission, T_c is the average time the channel is sensed busy due to a collision, and σ is the duration of an idle slot. The values for P_{tr} and P_s can be determined by the following equations

$$\begin{aligned} P_{tr} &= 1-(1-\tau)^m \\ P_s &= \frac{m\tau(1-\tau)^{m-1}}{P_{tr}} \end{aligned} \tag{3-17}$$

Let H denote the transmission time of the header of a data frame, and δ denote the signal propagation time from a station to all other stations. As shown in Figure 3.10, the time duration for a successful transmission, T_s, and a collision, T_c, for both the basic and the four-way handshaking access methods are

$$\begin{cases} T_s^{bas} = H + \mathrm{E}[P] + SIFS + \delta + ACK + DIFS + \delta \\ T_c^{bas} = H + \mathrm{E}[P] + DIFS + \delta \end{cases}$$
$$\begin{cases} T_s^{rts} = RTS + SIFS + \delta + CTS + SIFS + \delta + H \\ \qquad + \mathrm{E}[P] + SIFS + \delta + ACK + DIFS + \delta \\ T_c^{rts} = RTS + DIFS + \delta \,. \end{cases} \tag{3-18}$$

Finally, the numerical results are obtained based on the protocol parameters of the frequency hopping spread spectrum (FHSS) physical layer [IEEE99].

Table 3.2: Summary of the protocol parameters for the IEEE 802.11 saturation and disaster analyses

Parameter	Value
Physical layer	FHSS
Channel bit rate	1 Mbit/s (1μsec bit time)
Signal propagation time, δ	1 μsec
Slot time, σ	50 μsec
DIFS time	128 μsec
SIFS time	28 μsec
PHY header size	128 bits
MAC header size	272 bits
Payload size	8184 bits (constantly distributed)
RTS frame size	160 bits + PHY header
CTS frame size	112 bits + PHY header
ACK frame size	112 bits + PHY header
Data frame size	PHY header + MAC header + Payload
Number of station, m	1,2,3,…,50

Table 3.3: Time duration of a collision and a data frame transmission in IEEE 802.11

Access Method	Variable	Value
Basic access method	T_s^{bas}	8982 μsec
	T_c^{bas}	8713 μsec
Four-way handshaking access method	T_s^{rts}	9568 μsec
	T_c^{rts}	417 μsec

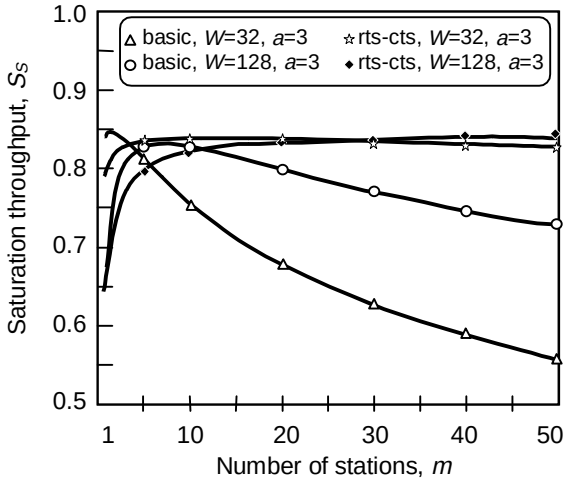


Figure 3.11: Saturation throughput for IEEE 802.11 for different protocol parameters

Table 3.2 summarizes values of some important protocol parameters related to the FHSS physical layer of IEEE 802.11. The time duration of a collision and a data frame transmission is provided in Table 3.3.

Figure 3.11 shows the saturation throughput of IEEE 802.11 versus the number of saturated stations. From the saturation throughput curves, it is evident that with a good selection of *CW* parameters, the IEEE 802.11 MAC protocol performs efficiently for a large population of active stations. The saturation throughput remains over 80% for the case of CW_{min}=32 and CW_{max}=256 for as many as 50 saturated stations.

3.3.2 The Disaster Analysis

Under the disaster scenario, there are *m* stations ready for new data frame transmissions at the same time. Each station has only one data frame to transmit. In this analysis, we concentrate on the duration of the recovery process.

The analysis requires two steps: (i) The computation of the attempt probability, P_n, of a particular station, and (ii) The mean time for all *m* stations to obtain a successful transmission each based on the probability P_n.

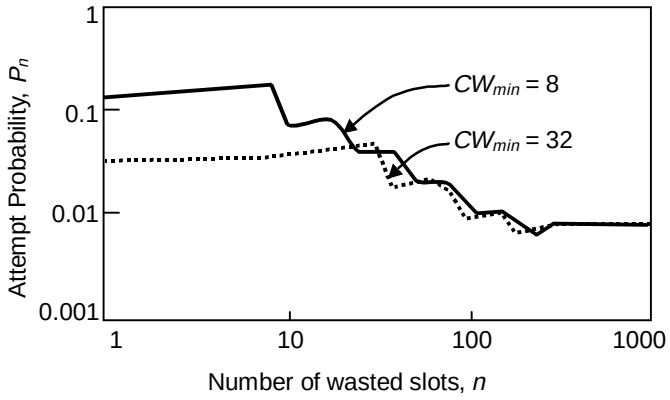


Figure 3.12: Attempt probability for an IEEE 802.11 station under the disaster scenario

Let n be the number of wasted slots (i.e. idle or collided slots) from the time when a station is ready for transmission. After n wasted slots, a station attempts to transmit its data frame in the next slot only if the station experienced an unsuccessful attempt previously between slot n-min(CW_{max},$2^c CW_{min}$) and slot n-1 where c is the number of collisions experienced by that station. Thus the probability, P_n, that a particular station will attempt its data frame transmission after n number of wasted slots can be computed by the following equation

$$P_n = \sum_{c=0}^{n} P(n,c) \qquad ,n = 0,1,2,... \tag{3-19}$$

where

$$\begin{aligned} P(0,c) &= 0 && ,c \geq 1 \\ P(n,0) &= \begin{cases} 0 & ,n \geq CW_{min} \\ 1/CW_{min} & ,\text{otherwise} \end{cases} \\ P(n,c) &= \sum_{k=\max(0,n-bf)}^{n-1} \frac{P(k,c-1)}{bf} && ,n \geq 1, c \geq 1 \\ bf &= \min(CW_{max}, 2^c CW_{min}). \end{aligned} \tag{3-20}$$

Figure 3.12 presents the attempt probability, P_n, of a particular station versus the number of wasted slots, n. In the figure, two attempt probability curves are shown, each with CW_{max}=256 and different values of CW_{min}.

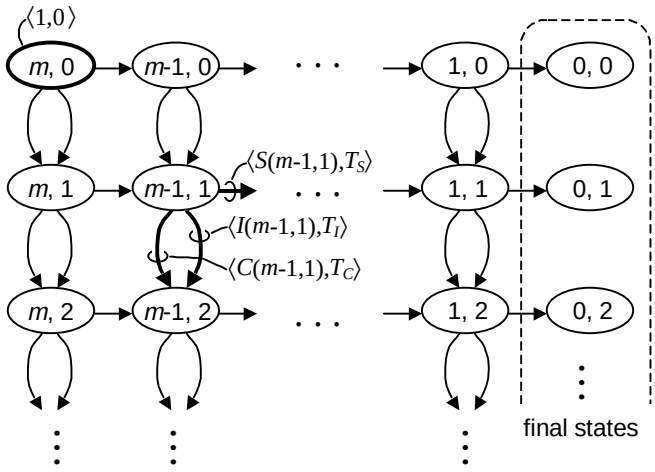


Figure 3.13: Bidimensional state space for the recovery process for IEEE 802.11 under the disaster scenario

To model the disaster scenario, we first assume that after n number of wasted slots counting from the beginning of the recovery process, all stations attempt their data frame transmissions in the next slot according to the attempt probability given in Equation (3-19) independently. With this assumption, as shown in Figure 3.13, we consider a bidimensional state space $\{D_m, D_n\}$, where D_m describes the number of stations and D_n describes the number of wasted slots occurred on the channel since the beginning of the recovery process.

In the model, the recovery process of m stations starts from state $\{m, 0\}$, and ends at the rightmost states with D_m=0. Transitions are only legal rightwards, henceforth called *rightward transitions*, and downwards, henceforth called *downward transitions*. It is because at any slot, either the number of stations is reduced by one as a result of a successful transmission, or a slot is wasted so that the number of wasted slots is increased by one. While a rightward transition can occur only in one way when exactly one station accesses the channel yielding a successful transmission, a downward transition may occur in either one of the two cases: (i) no station accesses the channel resulting an idle period, or (ii) two or more stations access the channel causing a collision. This is shown in Figure 3.13 by two arrows for downward transitions and a single arrow for rightward transitions from any

states excluding the final states. Therefore at any of those states, there are three one-step transitions. Note that for any of the states {1, D_n}, a collision occurs with probability zero.

Define *transition duration* as the time duration for a one-step transition either rightwards or downwards between any pair of adjacent states. Thus with each one-step transition, we associate two values: (i) a transition probability, and (ii) transition duration. Given that the system is in state {k, n}, let $S(k,n)$, $C(k,n)$, and $I(k,n)$ be the probabilities that the channel will obtain a successful transmission, a collision, and an idle period respectively.

The probability, $S(k,n)$, that a successful transmission will occur after n number of wasted slots with k remaining stations, can be viewed as the probability that only one out of k stations accesses the channel, thus

$$S(k,n) = k \cdot P_n \cdot (1-P_n)^{k-1} \tag{3-21}$$

where P_n is the attempt probability. The probability that the channel will be idle, $I(k,n)$, is the probability that all remaining k stations choose not to attempt, which can be expressed as

$$I(k,n) = (1-P_n)^k \ . \tag{3-22}$$

The probability of the appearance of a collision on the channel, $C(k,n)$ is

$$C(k,n) = 1-[S(k,n) + I(k,n)]. \tag{3-23}$$

While the transition probabilities are functions of the state {k, n}, the transition duration is state independent, but it is different when the channel obtains a successful transmission, a collision, and an idle period. The duration of a successful transmission is equal to a data frame transmission duration, denoted T_S. The duration of a collision is denoted T_C, and let T_I be the duration of an idle period.

Our aim is to obtain the mean duration of the recovery process, T_T, that is the total time duration required to transit from state {m, 0} to any final state.

Since the number of states is infinite and the total number of possible paths is huge, a brute force evaluation of the probability of each possible path is not practical.

Our technique used here is based on the key idea that the number of possible paths can be significantly reduced by aggregating paths leading from state $\{m, 0\}$ to any state $\{k, n\}$, which have the same total duration. In other words, if there are j paths leading from the initial state $\{m, 0\}$ to a particular state $\{k, n\}$, and all j paths have duration, say d, we can add up their probabilities to obtain the total probability. There is no need for remembering the particular details of all j paths.

To apply our technique, every state is associated with a set of ordered pairs $\{\langle p_1, d_1\rangle, \langle p_2, d_2\rangle, \ldots, \langle p_j, d_j\rangle\}$, where each d_i represents the time duration that the process will transit into that particular state from the initial state, and p_i represents the probability of that transition. A convenient way to describe the technique is by the use of polynomials. The set of ordered pairs described above can be expressed by a polynomial shown as follows

$$P_x(\{m,n\}) = p_1 x^{d_1} + p_2 x^{d_2} + \cdots + p_j x^{d_j}. \tag{3-24}$$

The initial state $\{m, 0\}$ is assigned with an ordered pair $\langle 1, 0\rangle$. It is because state $\{m, 0\}$ occurred with probability one and the time it takes to reach this state from the beginning of the recovery process is zero. Thus the initial state can be expressed as $P_x(\{m,0\}) = 1 \cdot x^0 = 1$.

As mentioned earlier each of the one-step transitions is already associated with a transition probability, p_t, and transition duration, d_t, hence it can also be expressed in a polynomial form as $p_t \cdot x^{d_t}$.

The polynomials of subsequent states can be obtained by summing all the products of the polynomials of the previous states and the transition polynomials

$$P_x(\{k,n\}) = \begin{cases} P_x(\{k+1,n\})\cdot\left(S(k+1,n)\cdot x^{T_S}\right), & k=0 \\ P_x(\{k+1,n\})\cdot\left(S(k+1,n)\cdot x^{T_S}\right), & 1\le k\le m-1,\ n=0 \\ P_x(\{k,n-1\})\cdot\left(I(k,n-1)\cdot x^{T_I}+C(k,n-1)\cdot x^{T_C}\right), & k=m,\ n=1,2,\ldots \\ P_x(\{k,n-1\})\cdot\left(I(k,n-1)\cdot x^{T_I}+C(k,n-1)\cdot x^{T_C}\right) & \\ \quad + P_x(\{k+1,n\})\cdot\left(S(k+1,n)\cdot x^{T_S}\right), & 1\le k\le m-1,\ n=1,2,\ldots. \end{cases} \quad (3\text{-}25)$$

The numerical computation can be performed by obtaining each $P_x(\{k,n\})$ from left to right, and then top to bottom starting from the initial state $\{m, 0\}$. Then the polynomial for all the final states can be obtained by

$$P_x(\text{all the final states}) = \sum_{j=0}^{\infty} P_x(\{0,j\}) = \sum_{\text{all terms}} p_j \cdot x^{d_j} \quad (3\text{-}26)$$

where the sum of all obtained probabilities from the final state is one. That is

$$\sum_{\substack{\text{all terms in}\\ \text{the final states}}} (p_j) = 1. \quad (3\text{-}27)$$

The mean duration for the recovery process, T_T, is thus

$$T_T = \sum_{\substack{\text{all terms in}\\ \text{the final states}}} (p_j \cdot d_j). \quad (3\text{-}28)$$

One key approximation in this model is we assume that at any state, say $\{k, n\}$, all k stations have experienced the same number of wasted slots and hence they attempt in the next slot with the same probability, P_n. In Figure

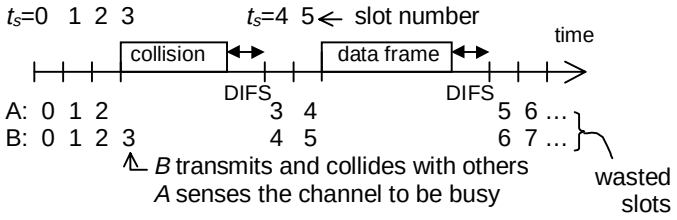


Figure 3.14: Illustration of the impact of collisions on the performance analysis under the disaster scenario

3.14, we illustrate an error this assumption may introduce. The error occurs only when there is a collision. As in Figure 3.14, we consider that station *B* has attempted its data frame transmission at slot t_s=3 and a collision is assumed. While station *A* did not attempt at slot t_s=3 but it detected this collision, it then stopped its backoff timer during the collision period. After the collision is clear, at slot t_s=*4*, station *A* will attempt its data frame transmission with the attempt probability P_n with n=3, whereas station *B* in this case will attempt with the attempt probability P_n with n=4 because station *B* counts the previous collided transmission as a wasted slot. This error will be significant only if collisions are often and the values of attempt probabilities of n and its neighbors are significantly different. Nevertheless, it can be seen from Figure 3.12 that the value of attempt probabilities of a given n and the neighboring points of n are similar, especially when n is large. Therefore, the error is not significant.

Figure 3.15 presents the mean duration of the recovery process for IEEE 802.11 under the disaster scenario. The good agreement between the numerical (shown in solid lines) and the simulation (shown in symbols) results validates the approximation employed in our model. Our numerical and simulation results are based on the protocol parameters of FHSS physical layer given in Table 3.2. The values for T_c and T_s are given in Table 3.3. Each of the numerical results is obtained until the final probability reaches over 0.99, that is $T_T = \sum_{i=0}^{k} (p_i \cdot d_i)$ where k satisfies the condition $\sum_{i=0}^{k} p_i \geq 0.99$.

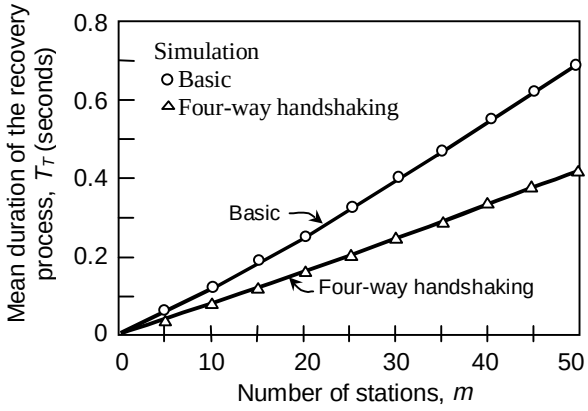


Figure 3.15: Mean duration of the recovery process for IEEE 802.11 under the disaster scenario for CW_{min}=8 and CW_{max}=256

The results shown in Figure 3.15 again confirm the efficiency of the IEEE 802.11 MAC protocol. For as many as 50 saturated stations, the recovery process lasts below one second. Even better performance results are achieved when the four-way handshaking access method is employed.

3.4 A Markovian Framework for Performance Analysis

In this section, we present a Markovian framework for performance analysis of MAC protocols. By employing system approximation technique, we model a MAC protocol into a continuous time Markov chain single server queue (CTMC-SSQ). The "arrival" process of this CTMC-SSQ represents the process of idle stations becoming active. The "service" time is the time it takes for a data frame to be successfully transmitted.

Consider k stations sharing a transmission medium. They are fed by a certain arrival process and will try to access the medium according to a given protocol. Recalling the saturation throughput analysis of a protocol, we assume that there are only i stations sharing a transmission medium, $1 \le i \le k$, and all i stations are saturated so that whenever a station among the i stations has successfully transmitted its data frame, the station is ready for another new data frame transmission immediately. In other words, all i

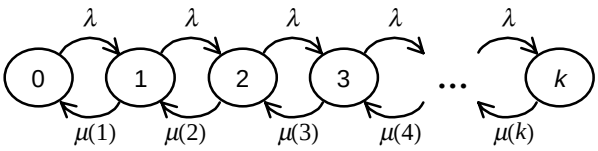


Figure 3.16: The state dependent M/M/1/k queue for the performance analysis of a MAC protocol

stations are always active. The main output from the analysis is the saturation throughput given i saturated stations, $S(i)$.

From the saturation throughput, the mean service rate of a protocol given i active stations, $\mu(i)$, can be obtained by

$$\mu(i) = \frac{S(i)}{t_d} \tag{3-29}$$

where t_d is the mean duration of a payload transmission. In other words, $1/\mu(i)$ is the mean time between two successive successful data frame transmissions given i saturated stations. It is the sum of (i) the channel assignment time, and (ii) the mean duration of a data frame transmission which carries the payload.

Under a certain statistical traffic arrival process, at a particular time, there may be i out of k stations that are active. At that particular instance, we observe that the protocol enters a temporary saturation condition with i active stations. Hence the saturation throughput results can be exploited in this situation.

Under statistical traffic, to analyze a protocol, we first construct a CTMC-SSQ with a certain arrival and service processes. For the purpose of illustrating the approach, we consider a simple example of a CTMC-SSQ, the *state dependent* M/M/1/k (SD-M/M/1/k) queue. The transition state diagram is shown in Figure 3.16. Under this model, there are k stations in a network. The *state* of the queue represents the number of active stations in the network. The number of active stations increases according to a Poisson

process with parameter λ, and it decreases based on the state dependent service process. Both the arrival and service processes are memoryless in this case.

One implication of the use of the SD-M/M/1/k system for performance analysis of a protocol is that the service time under a saturation condition for any number of saturated stations is exponentially distributed. In many cases, this assumption may not be accurate. In such a case, a fitting of the statistical characteristics of the service time under a saturation condition for the service time of the CTMC-SSQ model is required, for each i where $1 \leq i \leq k$. To maintain the Markovian property, a Phase-Type (PH) distribution [Neut81], for example an Erlang distribution, could be used.

For more realistic performance evaluation, it may be required to analyze a protocol with an arrival process that is more bursty than a Poisson process. Any arrival process that falls under the versatile *Markovian Arrival Process* (MAP) [LuMN90] can be used here. One such example is the *Markov Modulated Poisson Process* (MMPP).

In the following subsections, we introduce several traffic arrival models with different data frame size distributions for the performance evaluation using the proposed Makovian framework. We assume that the service time of a constant size data frame of the analyzed protocol can be described by an Erlang distribution with parameter j, or E_j in short. The mean service time is given by the saturation throughput analysis.

3.4.1 Bernoulli Station Arrival Model with Constant Size Data Frames

In the Bernoulli station arrival model, we consider a finite population of stations in a network. The stations are statistically identical and independent, switched between active and idle. The arrival process at each station is modeled as a burst process whereby each station generates a data frame of a certain size. The inter-burst time period of a station, that is, the

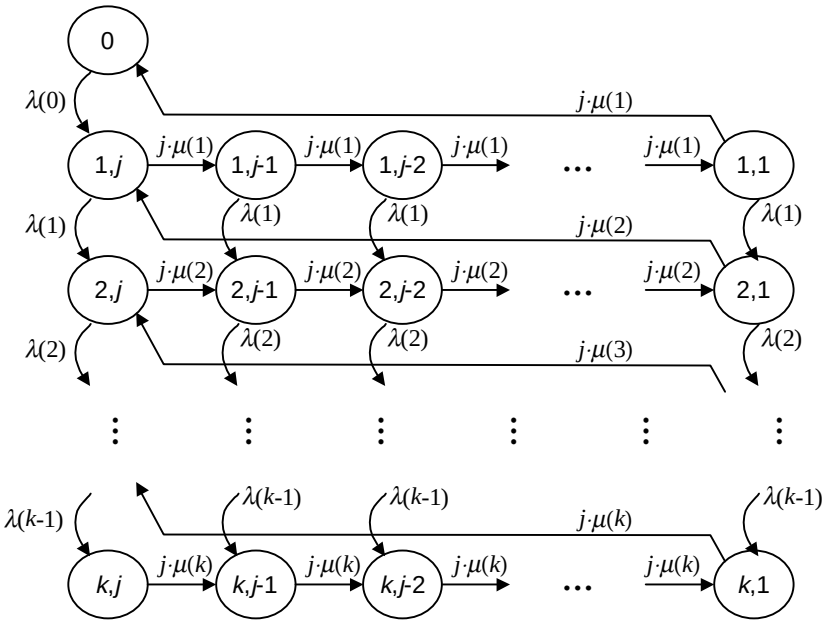


Figure 3.17: The state transition diagram of an $M/E_j/1/k$ system

time from when that station becomes idle upon a successful transmission of a data frame, until it becomes active again due to a transmission of a new data frame, is exponentially distributed. This assumption models the situation where after a user has successfully transmitted a message from a station, the time it takes for the reply to arrive, for instance the reply of the requested web document, plus the time for the user to digest the received information before initiating a new transmission, is exponentially distributed.

By modeling the service process of a protocol into a memoryless process of which the service time has an Erlang distribution, we construct a CTMC-SSQ system, which is an $M/E_j/1/k$ system of which the state transition diagram is presented in Figure 3.17. The input and output parameters of the system are listed in Table 3.4.

Recall that the system state corresponds to the number of active stations in a network, then under the arrival model, the arrival rate at state n, $\lambda(n)$, is the arrival rate aggregated from all idle stations, that is

$$\lambda(n) = \lambda_{ind} \cdot (k - n), \quad n = 0,1,\ldots,k\,. \tag{3-30}$$

The mean arrival rate observed by the system is given by

$$\overline{\lambda} = \sum_{n=0}^{k} (\lambda(n) \cdot p_n). \tag{3-31}$$

The throughput of the system is

$$\rho = \overline{\lambda} \cdot d_t \tag{3-32}$$

and the average transmission delay, D, experienced by a station is computed by using Little's formula [Litt61]. It is

$$D = \frac{1}{\overline{\lambda}} \cdot \sum_{n=0}^{k} (n \cdot p_n). \tag{3-33}$$

The steady state balance equations for this $M/E_j/1/k$ system are

$$\begin{aligned}
0 &= -\lambda(0) p_0 + j\mu(1) p_{1,1}, \\
0 &= -(\lambda(1) + j\mu(1)) p_{1,j} + j\mu(2) p_{2,1} + \lambda(0) p_0, \\
0 &= -(\lambda(1) + j\mu(1)) p_{1,i} + j\mu(1) p_{1,i+1}, \qquad (i = 1,2,\ldots, j-1) \\
0 &= -(\lambda(n) + j\mu(n)) p_{n,j} + j\mu(n+1) p_{n+1,1} + \lambda(n-1) p_{n-1,j}, \\
&\qquad\qquad (n = 2,3,\ldots,k-1) \\
0 &= -(\lambda(n) + j\mu(n)) p_{n,i} + j\mu(n) p_{n,i+1} + \lambda(n-1) p_{n-1,i}, \\
&\qquad\qquad (n = 2,3,\ldots,k-1;\ i = 1,2,\ldots, j-1) \\
0 &= -j\mu(k) p_{k,j} + \lambda(k-1) p_{k-1,j}, \\
0 &= -j\mu(k) p_{k,i} + j\mu(k) p_{k,i+1} + \lambda(k-1) p_{k-1,i}, \quad (i = 1,2,\ldots, j-1)
\end{aligned} \tag{3-34}$$

Table 3.4: Parameter descriptions for an $M/E_j/1/k$ system

Variable		Description
Input	k	The number of Bernoulli stations in the network
	λ_{ind}	The arrival rate of the individual station
	$S(i)$	The saturation throughput of the analyzed protocol given i saturated stations, $1 \le i \le k$
	d_t	The average transmission time of the payload under the saturation scenario
Output	p_n	The steady state probability of the system, $0 \le n \le k$
	$\overline{\lambda}$	The mean arrival rate observed by the system
	ρ	The throughput of the system
	D	The average transmission delay

with

$$p_n = \sum_{i=1}^{j} p_{n,i}, \quad n = 1,2,\ldots,k$$
$$\sum_{n=0}^{k} p_n = 1. \tag{3-35}$$

The steady state probabilities can be determined numerically by using Successive Over-Relaxation (SOR) [GrHa98].

3.4.2 Bernoulli Station Arrival Model with Dual Size Data Frames

In this subsection, we consider the same arrival model as the one described in the previous subsection. However, we replace the constant size data frame assumption with a more realistic dual size data frame assumption [ShHu79]. Under the dual size data frame assumption, there is a mixture of two different sizes of data frames – short and long data frames. The short data frame is usually a control packet of a protocol residing in the upper layer in a protocol stack, whereas the long data frame is normally a fragmented data packet.

Based on the proposed Markovian framework, an equivalent CTMC-SSQ, which is an M/PH/1/k system, is constructed for the dual size data frame case. The PH service process is a hyper-Erlangian process. To see how a hyper-Erlangian process can be used for the dual size data frame

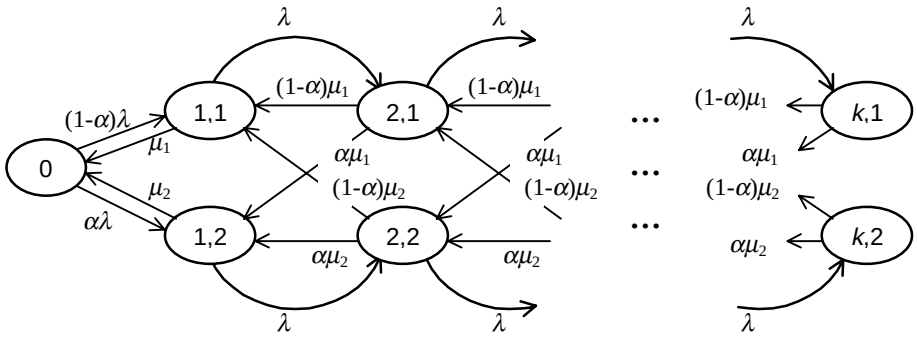


Figure 3.18: State transition diagram of an M/M/1/k queue with dual service rates

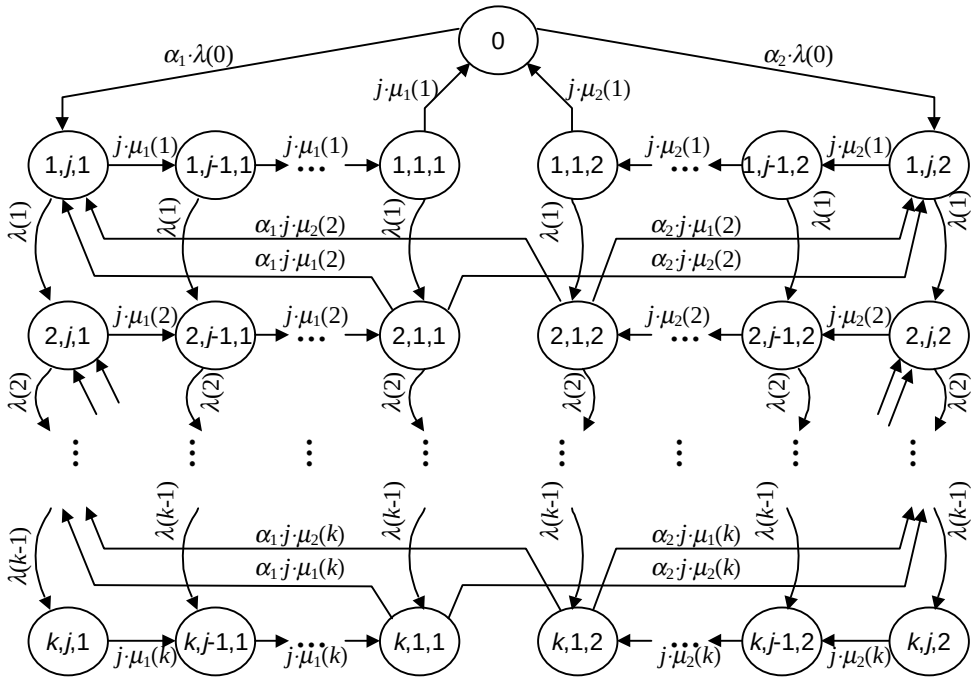


Figure 3.19: State transition diagram of an M/PH/1/k system

assumption, we illustrate with an example using a simple CTMC-SSQ where the server may choose between a fast service rate, μ_2, with probability α or a slow service rate, μ_1, with probability 1-α to serve an arrival. For simplicity, Markovian arrival and service processes are considered. The state transition diagram of this simple example is illustrated in Figure 3.18.

It is easy to see that the service of dual size data frames is similar to the provided example because different data frame sizes requires different service time periods. Since the service time distribution of a constant data frame is not exponential, rather, it can be described by an Erlang distribution with parameter *j*, a hyper-Erlangian process is necessary to model the service process with the dual size data frame assumption.

By extending the state transition diagram drawn in Figure 3.18 to include an Erlangian service process, we obtain the state transition diagram of the M/PH/1/k system as shown in Figure 3.19.

The steady state balance equations for the M/PH/1/k system are

$$
\begin{aligned}
0 &= -\lambda(0)\,p_0 + j\mu_1(1)\,p_{1,1,1} + j\mu_2(1)\,p_{1,1,2}, \\
0 &= -(\lambda(1) + j\mu_b(1))\,p_{1,j,b} + \alpha_b\lambda(0)\,p_0 \\
&\qquad + \alpha_b\, j\mu_1(2)\,p_{2,1,1} + \alpha_b\, j\mu_2(2)\,p_{2,1,2}, \qquad (b=1,2) \\
0 &= -(\lambda(1) + j\mu_b(1))\,p_{1,i,b} + j\mu_b(1)\,p_{1,i+1,b}, \quad (1 \le i \le j-1,\ b=1,2) \\
0 &= -(\lambda(n) + j\mu_b(n))\,p_{n,j,b} + \lambda(n-1)\,p_{n-1,j,b} \\
&\qquad + \alpha_b\, j\mu_1(n+1)\,p_{n+1,1,1} + \alpha_b\, j\mu_2(n+1)\,p_{n+1,1,2}, \\
&\qquad\qquad (2 \le n \le k-1,\ b=1,2) \\
0 &= -(\lambda(n) + j\mu_b(n))\,p_{n,i,b} + \lambda(n-1)\,p_{n-1,i,b} + j\mu_b(n)\,p_{n,i+1,b}, \\
&\qquad\qquad (2 \le n \le k-1,\ 1 \le i \le j-1,\ b=1,2) \\
0 &= -j\mu_b(k)\,p_{k,j,b} + \lambda(k-1)\,p_{k-1,j,b}, \qquad (b=1,2) \\
0 &= -j\mu_b(k)\,p_{k,i,b} + \lambda(k-1)\,p_{k-1,i,b} + j\mu_b(k)\,p_{k,i+1,b}, \\
&\qquad\qquad (1 \le i \le j-1;\ b=1,2)
\end{aligned}
\tag{3-36}
$$

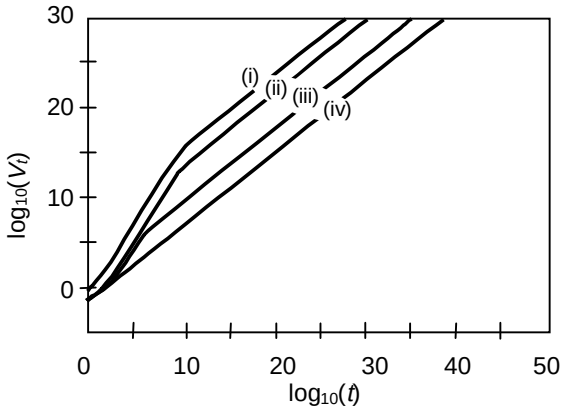
and

$$
\begin{aligned}
& p_n = \sum_{i=1}^{j} \left(p_{n,i,0} + p_{n,i,1}\right), \quad n = 1,2,\ldots,k \\
& \sum_{n=0}^{k} p_n = 1
\end{aligned}
\tag{3-37}
$$

with $\mu_b(i)$ being the service rate of the analyzed protocol given i active stations. The probability that the service rate $\mu_b(i)$ will be chosen for an arrival is α_b. The value b=1,2 is used for the dual size data frame assumption. The description of other variables is given in Table 3.4.

3.4.3 Aggregated MMPP Arrival Model with Constant Size Data Frames

A study of Ethernet LAN traffic by Leland et al. [LTWW94] reveals that Ethernet traffic exhibits a long range dependent (LRD) property. Hence it is important to study the performance of a LAN with a certain MAC protocol under such traffic. In this subsection, we model the arrival process as a MMPP. It is shown that MMPP can be used to produce bursty traffic [HeLu86].



	λ_0	λ_1	r_0	r_1
(i)	0.01	1	10^{-8}	$2\cdot10^{-9}$
(ii)	0.01	0.1	10^{-8}	$2\cdot10^{-9}$
(iii)	0.01	0.05	10^{-5}	10^{-6}
(iv)	0.01	0.05	0.5	0.005

Figure 3.20: The Variance-Time curves of MMPP sources

A MMPP is an alternating Markovian process with two arrival states, where the arrival process in arrival state m is a Poisson process with rate λ_m for m=0,1. The sojourn time in each arrival state is exponentially distributed with the mean sojourn time in arrival state 0 and 1 being r_0^{-1} and r_1^{-1} respectively. The values of the mean and the variance of MMPP traffic are given in [HeLu86]. Let N_t be the amount of work arriving during time interval (0, t). For MMPP, the mean of N_t, denoted λ, is

$$\lambda = \frac{\lambda_0 r_1 + \lambda_1 r_0}{r_0 + r_1} \cdot t \tag{3-38}$$

and the variance of N_t, denoted V_t, is

$$V_t = \lambda \cdot \left(1 + \frac{2(\lambda_0 - \lambda_1)^2 r_0 r_1}{(r_0 + r_1)^2 (\lambda_0 r_1 + \lambda_1 r_0)} - \frac{2(\lambda_0 - \lambda_1)^2 r_0 r_1}{(r_0 + r_1)^3 (\lambda_0 r_1 + \lambda_1 r_0) \cdot t} \left(1 - e^{-(r_0 + r_1)t} \right) \right). \tag{3-39}$$

Here, we further demonstrate the short range dependent (SRD) property of the MMPP. For this purpose, we plot the variance time curve of the MMPP in Figure 3.20. Four variance time curves with different parameter sets of

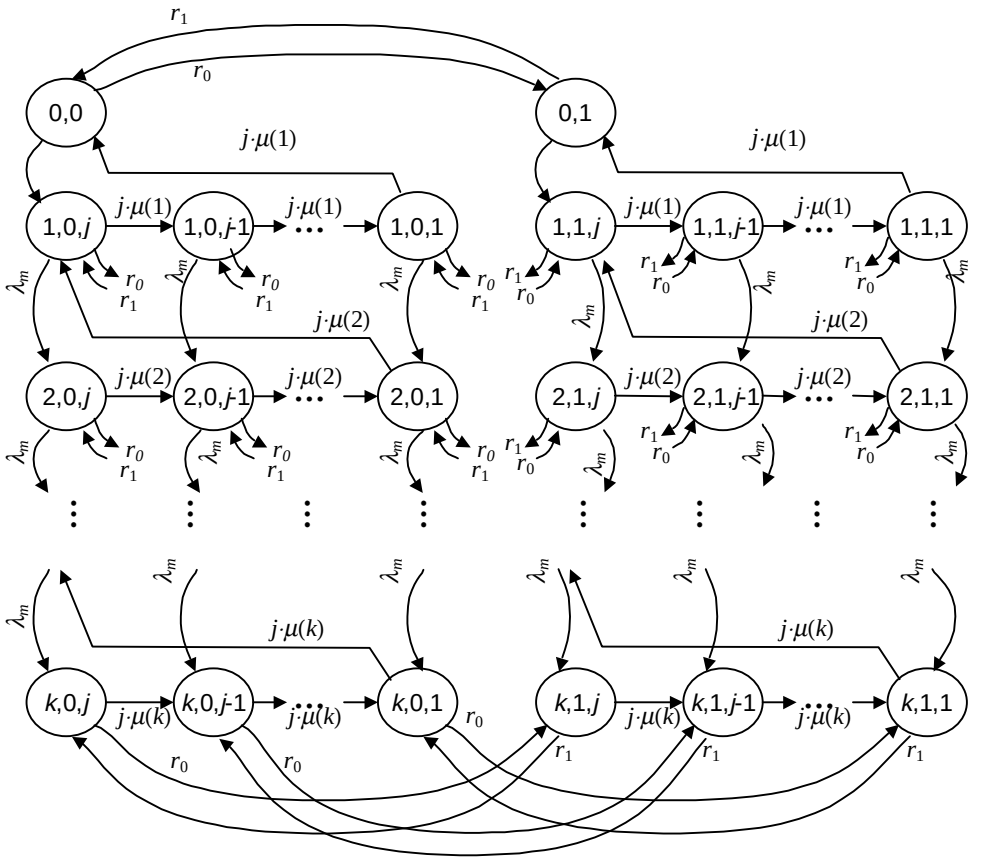


Figure 3.21: State transition diagram of an MMPP/E_j/1/k system

the MMPP are shown in the figure. By comparing the four curves, the critical time interval of the traffic and the slopes of the curve within the critical interval can be controlled by the parameters of the MMPP. For curves (i), (ii) and (iii), we obtained SRD traffic with different critical time intervals.

Similar to the previous subsections, under the constant size data frame assumption with MMPP arrival process, we construct an equivalent CTMC-SSQ for the purpose of performance evaluation for a MAC protocol. The system used here is an MMPP/E_j/1/k system. The MMPP we use here is a truncated MMPP because in a LAN, the number of data frames competing for the broadcast channel is always limited and it is related to the number of stations in the LAN. Hence we limit the number of buffer size of the queue to *k*, which is the number of stations in a LAN.

The state transition diagram is provided in Figure 3.21. The input and output parameters for the system is described in Table 3.5. The steady state balance equations are given as follows

$$
\begin{aligned}
0 &= -(\lambda_m + r_m)\,p_{0,m} + j\mu(1)\,p_{1,m,1} + r_{m+1}\,p_{0,m+1}, \\
0 &= -(\lambda_m + j\mu(1) + r_m)\,p_{1,m,j} + \lambda_m\,p_{0,m} + j\mu(2)\,p_{2,m,1} \\
&\quad + r_{m+1}\,p_{1,m+1,j}, \\
0 &= -(\lambda_m + j\mu(1) + r_m)\,p_{1,m,i} + j\mu(1)\,p_{1,m,i+1} + r_{m+1}\,p_{1,m+1,i}, \\
&\qquad\qquad (1 \le i \le j-1) \\
0 &= -(\lambda_m + j\mu(n) + r_m)\,p_{n,m,j} + j\mu(n+1)\,p_{n+1,m,1} \\
&\quad + \lambda_m\,p_{n-1,m,j} + r_{m+1}\,p_{n,m+1,j}, \qquad (2 \le n \le k-1) \\
0 &= -(\lambda_m + j\mu(n) + r_m)\,p_{n,m,i} + j\mu(n)\,p_{n,m,i+1} + \lambda_m\,p_{n-1,m,i} \\
&\quad + r_{m+1}\,p_{n,m+1,i}, \qquad (2 \le n \le k-1, 1 \le i \le j-1) \\
0 &= -(j\mu(k) + r_m)\,p_{k,m,j} + \lambda_m\,p_{k-1,m,j} + r_m\,p_{k,m+1,j}, \\
&\qquad\qquad (2 \le n \le k-1) \\
0 &= -(j\mu(k) + r_m)\,p_{k,m,i} + j\mu(k)\,p_{k,m,i+1} + \lambda_m\,p_{k-1,m,i} \\
&\quad + r_{m+1}\,p_{k,m+1,i}, \qquad (1 \le i \le j-1)
\end{aligned}
\tag{3-40}
$$

with

$$
\begin{aligned}
m+1 &= \begin{cases} 1, & if\ m = 0 \\ 0, & if\ m = 1 \end{cases} \\
p_0 &= p_{0,0} + p_{0,1} \\
p_n &= \sum_{i=1}^{j} \left(p_{n,0,i} + p_{n,1,i}\right), \quad n = 1,2,\ldots,k
\end{aligned}
\tag{3-41}
$$

and

$$
\sum_{n=0}^{k} p_n = 1. \tag{3-42}
$$

Table 3.5: Parameter descriptions for an MMPP/E_j/1/k system

Variable		Description
Input	k	The number of Bernoulli stations in the network
	λ_m	The arrival rate of MMPP at state m, m=0,1
	r_m	The mode switching rate of MMPP at state m, m=0,1
	$S(i)$	The saturation throughput of the analyzed protocol given i saturated stations, $1 \le i \le k$
	d_t	The average transmission time of the payload under the saturation scenario
Output	p_n	The steady state probability of the system, $0 \le n \le k$
	$\overline{\lambda}$	The mean arrival rate observed by the system
	ρ	The throughput of the system
	D	The average transmission delay

We solve the balance equations by using SOR. Having obtained the steady state probabilities of the system, the mean arrival rate observed by the system is

$$\overline{\lambda} = \sum_{i=0}^{k-1} (\lambda \cdot p_i). \tag{3-43}$$

The throughput of the system is

$$\rho = \overline{\lambda} \cdot d_t \tag{3-44}$$

and finally the average transmission delay, D, experienced by a station is computed by using Little's formula [Litt61]. It is

$$D = \frac{1}{\overline{\lambda}} \cdot \sum_{n=0}^{k} (n \cdot p_n). \tag{3-45}$$

3.5 Performance Analyses for IEEE 802.3 under Statistical Traffic Conditions

In order to use the previously described Markovian framework for performance evaluation of IEEE 802.3, we first need to obtain the saturation throughput of IEEE 802.3. The saturation throughput of IEEE 802.3 is given in Equations (3-1)-(3-4) in Section 3.2. The saturation throughput is

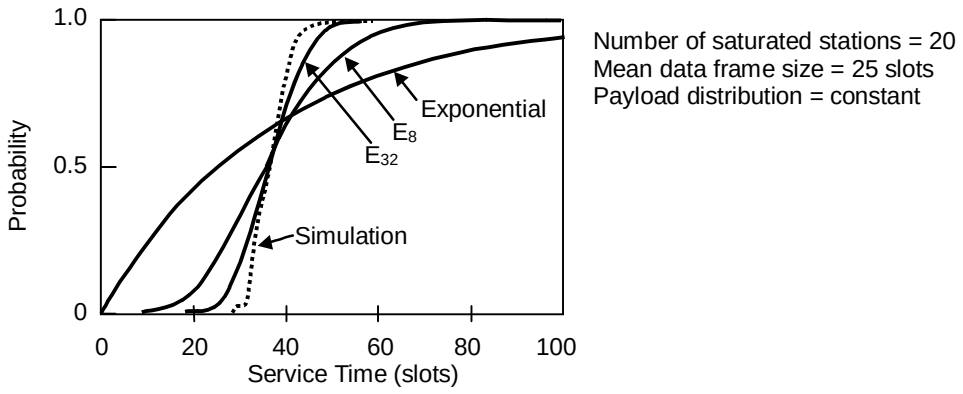


Figure 3.22: A CDF comparison of an exponential, Erlang random variables and the service time distribution of IEEE 802.3

obtained based on the assumptions and protocol parameters listed in Table 3.1.

Using Equations (3-1)-(3-4) and (3-29), the mean service rate of IEEE 802.3 for a given number of saturated stations is known. However, the saturation throughput analysis does not provide the distribution of the service process of IEEE 802.3 under the saturation scenario, which is needed for the construction of a CTMC-SSQ for performance evaluation.

For the purpose of constructing a CTMC-SSQ, we first study the service process of IEEE 802.3 under the saturation scenario. In Figure 3.22, using computer simulation, we obtain the Cumulative Distribution Function (CDF) of a typical service time of 802.3 under the saturation scenario. The number of saturated station in this example is 20. Figure 3.22 also compares that CDF with the CDFs of exponential, Erlang with parameters 8 (E_8) and Erlang with parameter 32 (E_{32}) random variables, all with the same mean value. Comparing the CDFs in Figure 3.22, we see that the Erlang distribution may be used to model the service time process of IEEE 802.3.

Based on the queueing results for an M/G/1 system, the mean queue size of the system, Q, can be expressed as

$$Q = \frac{2\rho - \rho^2 + \rho^2\left(\frac{\sigma_s}{s}\right)^2}{2(1-\rho)} \tag{3-46}$$

where ρ is the utilization, σ_s is standard deviation of the service time distribution and s is the mean service time. From the above result, we notice that if σ_s is much smaller than s, the term (σ_s/s) will produce a small value which will not significantly affect the mean queue size. As from Figure 3.22, the simulation results suggest a small value of (σ_s/s) for the IEEE 802.3 service time distribution. Similarly, an Erlang distribution function also produces a small value of (σ_s/s). Hence an Erlang distribution may provide a good approximation for the service time distribution of IEEE 802.3.

Comparing E_8 and E_{32}, we found that both of them have small values for (σ_s/s), which are 0.353 and 0.176 respectively. Because the difference between their (σ_s/s) values is small, thus the choice of E_8 and E_{32} does not significantly affect the queueing performance, especially when ρ is not large. Therefore, E_8 and E_{32} are equally good in modeling the service time distribution function of the IEEE 802.3 MAC protocol.

3.5.1 Bernoulli Station Arrival Model with Constant Size Data Frames

The performance evaluation of the IEEE 802.3 MAC protocol under Bernoulli station arrival model with constant size data frames can be achieve by analyzing the equivalent state dependent $M/E_j/1/k$ system presented in subsection 3.4.1.

Table 3.6 provides the protocol parameters for performance evaluation of IEEE 802.3 under the assumed arrival model. We use E_8, that is Erlang random variable with parameter j=8, to model the service time distribution

Table 3.6: Protocol parameters used to generate results shown in Figures 3.23 and 3.24

Parameter	Value
Protocol used	slotted CSMA/CD with BEB
Channel bit rate	10 Mb/s
Signal propagation time, τ	25.6 μsec (256 bit time)
Slot time	2τ (512 bit time)
Cost of a collision	one slot
The time required for detecting a transmission end	τ
Number of Bernoulli stations, k	50
Data frame or payload size (headers are ignored), b (or d_t)	(i) 5 slots (320 bytes) (ii) 25 slots (1600 bytes)

of IEEE 802.3. The IEEE 802.3 mean service rate, $\mu(n)$, given n active stations for n=1,2,…,k is computed by Equations (3-1)-(3-4) and (3-29) based on parameters given in Table 3.6. The time unit used for numerical computation is slot time.

Figure 3.23 plots the numerical (shown in lines) and simulation (shown in symbols) results for the delay performance of IEEE 802.3. The figure also compares the delay performance of two different data frame sizes. We first notice the small discrepancy between the numerical and simulation results. It indicates the robustness of the model and confirms the use of Erlang distribution as the distribution of the IEEE 802.3 service time under the assumption of constant size data frames.

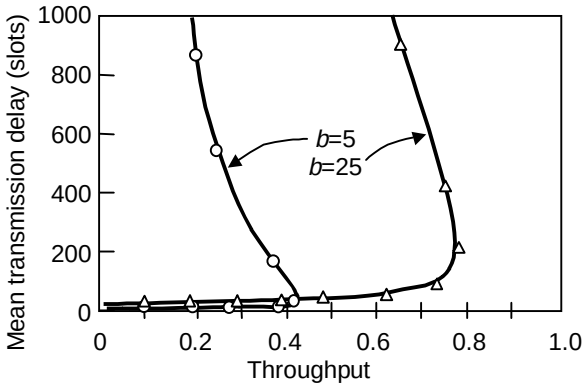


Figure 3.23: Mean transmission delay versus throughput for IEEE 802.3 under the constant size data frame assumption

We also observe that similar to the saturation and the disaster scenarios, the performance of the IEEE 802.3 MAC protocol under the Bernoulli station arrival model is better when longer data frames are used for transmission. At the throughput level just above 40%, the mean transmission delay for IEEE 802.3 with b=5 increases sharply due to a steep increase in the collision resolution time as a result of a heavy load. This increase in the collision resolution time causes the channel throughput to drop and the mean transmission delay to rise. This effect is demonstrated in the figure where when the channel throughput reaches its maximum level, as the load increases further, the channel throughput drops and the mean transmission delay rises.

For the IEEE 802.3 MAC protocol with b=25, a much higher throughput level is achievable. Its maximum throughput reaches just below 80% compared to 40% with 5 slot data frames.

3.5.2 Bernoulli Station Arrival Model with Dual Size Data Frames

The constant size data frame assumption used in the previous subsection is replaced with the realistic dual size data frame assumption. By modeling the service time distribution of IEEE 802.3 into E_8, we construct a M/PH/1/k system presented in subsection 3.4.2 for the performance analysis for IEEE

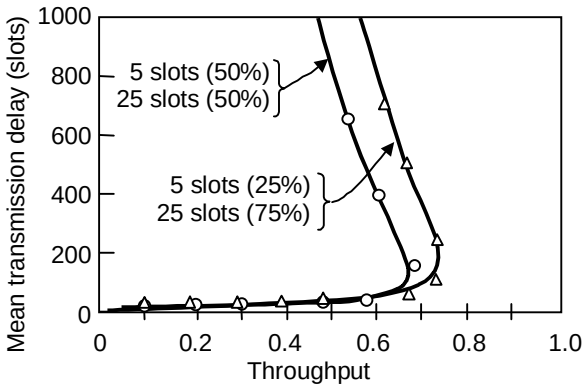


Figure 3.24: Mean transmission delay versus throughput for IEEE 802.3 under the dual size data frame assumption

802.3. The IEEE 802.3 MAC protocol parameters are given in Table 3.6.

Using the developed results presented in subsection 3.4.2, we obtain the delay performance of IEEE 802.3. The results are shown in Figure 3.24. In the figure, the good match between the numerical (shown in lines) and simulation (shown in symbols) results confirms the robustness and versatility of the analytical approach where the model is not limited to a certain data frame size distribution.

3.6 Performance Analyses for IEEE 802.11 under Statistical Traffic Conditions

In this section, we turn our focus to the performance analysis for IEEE 802.11 MAC protocol under statistical traffic conditions. To conduct the analysis using our proposed approach, we first need the saturation throughput of the protocol. The saturation throughput of IEEE 802.11 has been performed by Bianchi and presented in subsection 3.3.1. Having obtained the saturation throughput, we can easily compute the mean service time of IEEE 802.11 given a number of active stations.

Similar to the IEEE 802.3 saturation throughput analysis, the IEEE 802.11 saturation throughput analysis provides only the mean service time of IEEE 802.11. The IEEE 802.11 mean service rate, $\mu(n)$, given n active stations for

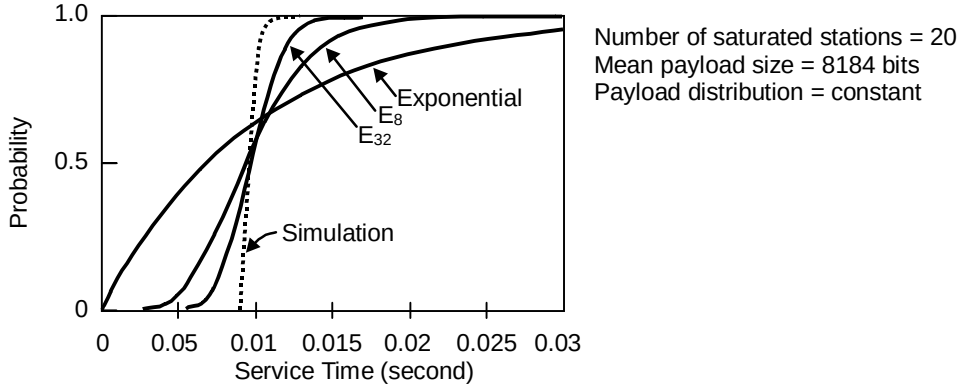


Figure 3.25: A CDF comparison of an exponential, Erlang random variables and the service time of IEEE 802.11

n=1,2,…,k is computed by using Equations (3-14)-(3-18) and (3-29).

Using computer simulation, in Figure 3.25, we demonstrate a typical service time CDF of IEEE 802.11 under the saturation scenario assuming constant size data frames. We further compare it with the CDFs of an exponential and Erlang random variables. As can been seen, an Erlang distribution may be used to describe the service time distribution of IEEE 802.11.

In the following subsections, we present the delay performance analysis for IEEE 802.11 under various arrival models.

3.6.1 Bernoulli Station Arrival Model with Constant Size Data Frames

To perform the performance analysis for IEEE 802.11 under Bernoulli station arrival model with constant size data frames, we construct a state dependent $M/E_8/1/k$ system where the mean service rate of the system follows the mean service rate of IEEE 802.11 under the saturation scenario. We assume the FHSS physical layer for the IEEE 802.11 network. The FHSS physical layer parameters are given in Table 3.2. Further parameters related to this performance analysis are summarized in Table 3.7.

Using the results developed in subsection 3.4.1, we obtain the average transmission delay versus throughput for the four-way handshaking access method of IEEE 802.11 with different data frame sizes. The results are presented in Figure 3.26. The very close agreement between the numerical and simulation results presented in the figure confirms the use of our

Table 3.7: Parameters used to generate results shown in Figures 3.26 to 3.28

Parameter	Value
DCF Method	Basic, Four-way handshaking
Minimum backoff window, CW_{min}	8
Maximum backoff window, CW_{max}	256
Maximum payload size in a data frame	8184 bits
Payload size, d_t	8184, 4348, 2430, 512 bits
Number of stations, k	50

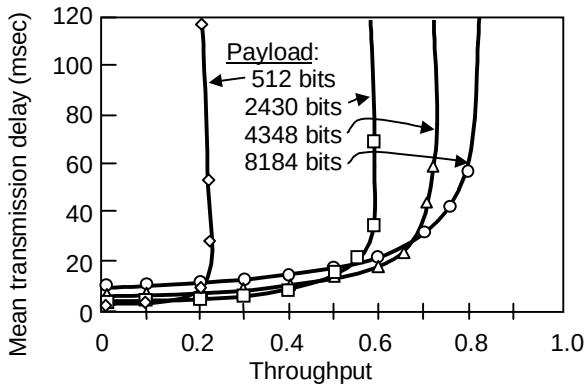


Figure 3.26: Mean transmission delay for four-way handshaking access method of IEEE 802.11 with different payload sizes

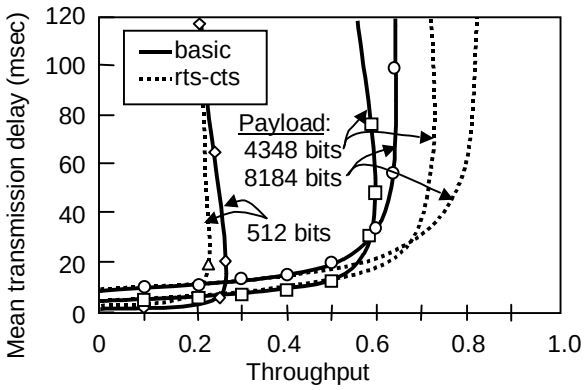


Figure 3.27: Mean transmission delay for IEEE 802.11: Comparison of the two access methods

approach for the performance analysis of IEEE 802.11. Moreover, the effect of the data frame size on the delay performance is demonstrated. The benefit of using a larger data frame is clearly presented as the protocol offers a higher achievable throughput level for a larger data frame size.

In Figure 3.27, we demonstrate the basic access method of IEEE 802.11, and compare the results with that of the four-way handshaking access method. We see from the figure that IEEE 802.11 with the basic access method suffers longer delay than the four-way handshaking access method if long data frames are considered. In contrast, for short data frames, the transmission delay is slightly lower if the basic access method is chosen. This result provides evident justifying the use of the basic access method for short data frame transmissions, and the four-way handshaking access

method for long data frame transmissions specified in the IEEE 802.11 standard [IEEE99].

3.6.2 Bernoulli Station Arrival Model with Dual Size Data Frames

To further investigate the effect of the data frame size distribution on the delay performance, we now consider a dual size data frames under Bernoulli station arrival model. In particular, we consider two possible payload sizes: 512 bits and 8184 bits. Two cases are assumed here. In the first, 75% of the frames are of 512-bit and 25% are of 8184-bit payload sizes. In the second case, half of the data frames are of 512-bit and the other half are of 8184-bit payload sizes.

With the developed balance equations for the M/PH/1/k system provided in subsection 3.4.2, and all the parameters summarized in Tables 3.2 and 3.7, the mean transmission delay of IEEE 802.11 is readily obtainable.

The delay performance of IEEE 802.11 with the four-way handshaking access method and dual size data frames is presented in Figure 3.28. The solid lines and symbols represent the numerical and simulation results respectively. Moreover, we include the numerical results taken from Figure 3.26 (shown in dotted lines) to compare the delay performance between the

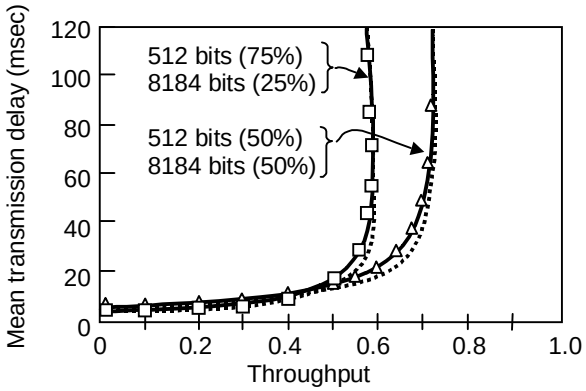


Figure 3.28: Mean transmission delay for IEEE 802.11: Effect of data frame size distribution on delay performance

constant size data frame and the dual size data frame assumptions that have the same mean value.

As in Figure 3.28, an excellent agreement between the numerical and simulation results has again been achieved. Our interest here is also the investigation of the effect of the data frame distribution on the delay performance. Surprisingly, the increase in variance in data frame distribution due to the dual size data frames has only little effect on the mean transmission delay. The average delay of the dual size data frame situation appears to be just slightly higher than that of the constant size data frame situation.

3.6.3 Bernoulli Station Arrival Model with Long Messages

In this subsection, we study the effect of long message transmissions on the average transmission delay. According to the IEEE 802.11 FHSS physical specification, the longest specified data frame carries a payload of 8184 bits [IEEE99]. To transmit a message longer than that, the fragmentation operation described in Section 2.9 must be employed.

To perform the analysis, we first notice that the fragmentation operation is not modeled in [Bian98, Bian00]. To include the fragmentation operation, for the four-way handshaking access method, the variable T_s^{rts} in [Bian98, Bian00], which is the time duration for a successful transmission, must be modified. The new successful transmission time duration, $T_s^{rts^*}$, must include the additional overhead due to the fragmentation depicted in Figure 2.37. It can be expressed by

$$T_s^{rts^*} = T_s^{rts} + \left\lceil \frac{\mathrm{E}[P]}{8184} - 1 \right\rceil \cdot \left(2 \cdot SIFS + 2\delta + ACK + H\right) \qquad (3\text{-}47)$$

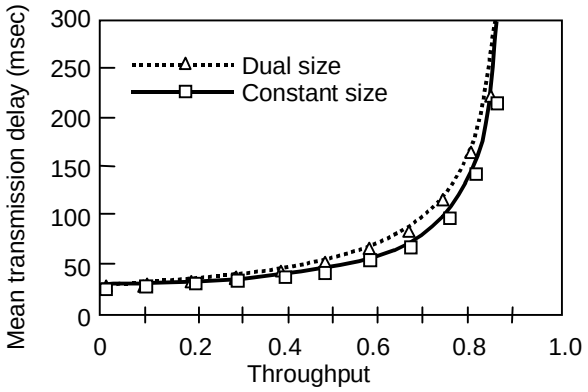


Figure 3.29: Mean transmission delay for IEEE 802.11 with long message transmissions

where the variables E[*P*], *SIFS*, *ACK*, δ and *H* are explained in subsection 3.3.1. In the equation, $\lceil x \rceil$ represents the smallest integer value larger than *x*. Note that this modification is also applicable to the basic access method.

With the appropriate modification, we can now obtain the service rate of IEEE 802.11 under the saturation scenario assuming that each station transmits long messages that will be fragmented to several data frames before the transmissions. We focus on the delay performance analysis of IEEE 802.11 with the four-way handshaking access method.

Table 3.2 provides the protocol parameters used in the numerical and simulation computations. Further MAC parameters used for this analysis are

Table 3.8: Parameters used to generate results shown in Figure 3.29

Parameter	Value
DCF Method	Four-way handshaking
Minimum backoff window, CW_{min}	8
Maximum backoff window, CW_{max}	256
Number of stations, *k*	50
Case 1: Message size distribution = Constant Mean message size (excluding headers) = 24576 bits	
Case 2: Message size distribution = Dual size (36% short + 64% long) Short message size (excluding headers) = 512 bits Long message size (excluding headers) = 38112 bits	

summarized in Table 3.8. We consider both the constant size messages and the dual size messages. The $M/E_8/1/k$ system given in subsection 3.4.1 with k=50 is used here. The delay curves are presented in Figure 3.29.

The results shown in Figure 3.29 again suggest the accuracy and versatility of the analytical approach used here, where the numerical (drawn in lines) results match the simulation (drawn in symbols) results. We also notice that the mean transmission delay a station experienced is not significantly affected even we increase the variance of the message size.

3.6.4 Aggregated MMPP Arrival Model with Constant Size Data Frames

In this subsection, we evaluate IEEE 802.11 under bursty traffic. We consider that the aggregated traffic from all stations follows MMPP. Constant size data frames are assumed. Details of this arrival model are given in subsection 3.4.3.

Since constant size data frame assumption is considered, an $MMPP/E_8/1/k$ system is used for performance evaluation. Tables 3.2 and 3.9 summarize all necessary parameters used for this analysis as well as the simulation. The MMPP parameters are chosen similar to the third parameter set presented in Figure 3.20 which generates SRD traffic. The results are presented in Figure 3.30.

The insignificant difference between the numerical (shown in lines) and

Table 3.9: Parameter used to generate results shown in Figure 3.30

Parameter	Value
DCF Method	Four-way handshaking
Minimum backoff window, CW_{min}	8
Maximum backoff window, CW_{max}	256
Payload size, d_t	8184 bits
Number of stations, k	50
MMPP mean arrival rate in model 0, λ_0	0.005 to 0.022 arrivals/msec
MMPP mean arrival rate in model 1, λ_1	$5 \cdot \lambda_0$
MMPP mode switching rate in model 0, r_0	0.00008 /msec
MMPP mode switching rate in model 1, r_1	0.00002 /msec

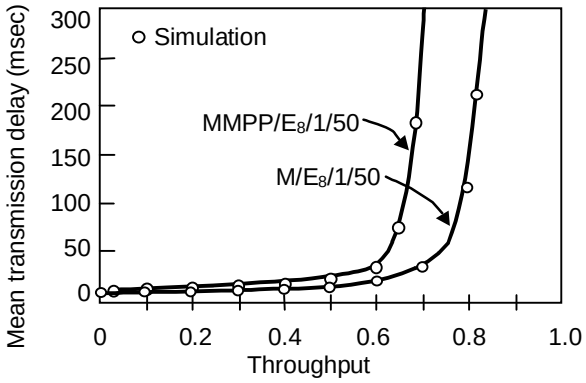


Figure 3.30: Mean transmission delay for IEEE 802.11 under Poisson and MMPP traffic assumption

simulation (shown in symbols) results again confirms the accuracy of the employed approximations in this performance evaluation. The figure also demonstrates the effect of the traffic burstiness on the delay performance of IEEE 802.11. The numerical results under Poisson arrival model is obtained by using the fact that as both the mode switching rates of the MMPP is high, the MMPP becomes a Poisson process [CMRW97].

As can be observed from the figure, the IEEE 802.11 MAC protocol performs quite differently under bursty traffic. This result suggests that the past studies of IEEE 802.11 under Poisson traffic assumption significantly overestimate the performance of IEEE 802.11.

3.7 Summary

In this chapter, we have studied the IEEE 802.3 and the IEEE 802.11 MAC protocols under realistic traffic conditions. In contrast to the analyses performed in past presented in Chapter 2, the analyses conducted in this chapter focus on the performance of the protocols under realistic scenarios and assumptions. The following list summarizes the analyses performed in this chapter.

- The throughput and delay performance of the IEEE 802.3 MAC protocol under the saturation scenario.

- The performance of the IEEE 802.3 MAC protocol under the disaster scenario.
- The performance of the IEEE 802.11 MAC protocol under the saturation scenario by Bianchi.
- The performance of the IEEE 802.11 MAC protocol under the disaster scenario.
- The delay performance of the IEEE 802.3 MAC protocol under Bernoulli station arrival model with constant size data frames.
- The delay performance of the IEEE 802.3 MAC protocol under Bernoulli station arrival model with dual size data frames.
- The delay performance of the IEEE 802.11 MAC protocol under Bernoulli station arrival model with constant size data frames.
- The delay performance of the IEEE 802.11 MAC protocol under Bernoulli station arrival model with dual size data frames.
- The delay performance of the IEEE 802.11 MAC protocol under Bernoulli station arrival model with long messages.
- The delay performance of the IEEE 802.11 MAC protocol under the aggregated Poisson and MMPP arrival models with constant size data frames.

Based on the results of the analyses presented in this chapter, we found that

- As presented in subsection 3.2.1, the performance of the IEEE 802.3 MAC protocol is unattractive in a network consisting a large number of active stations, which is the original design of the Ethernet protocol [MeBo76].
- The IEEE 802.11 MAC protocol performs reasonably well under a certain load condition with a good selection of the protocol parameters as presented in subsection 3.3.1. The results also suggest that the performance of IEEE 802.11 may be further enhanced if a dynamical alternation of parameters according to the load condition based on a certain mechanism can be achieved.

- The effect of the data frame distribution on the mean transmission delay of the IEEE 802.11 MAC protocol is not significant as shown in subsections 3.6.1-3.6.3.
- According to our analysis performed in subsection 3.6.4, the Poisson traffic assumption significantly overestimates the performance of the IEEE 802.11 MAC protocol under an actual operation with the arrival traffic that is usually bursty.

4 Carrier Sense Multiple Access with Reservations by Interruptions

Ethernet is probably one of the most widely used network in today's LANs. Ethernet is based on the Carrier Sense Multiple Access with Collision Detection (CSMA/CD) protocol. When collisions of transmissions occur, they are resolved by using the Binary Exponential Backoff retransmission algorithm. This protocol is standardized as the IEEE 802.3 MAC protocol.

The CSMA/CD protocol is a random access protocol, an extension of the Aloha protocol. Its performance has been studied under various traffic conditions in the previous chapter. Our finding as well as other studies of Ethernet performance, such as one studied by Molle [Moll94], indicate that Ethernet does not perform well in a network with a large population of active stations. Moreover, the CSMA/CD protocol does not have the capability to differentiate and prioritize traffic. As more real-time multimedia traffic is involved in today's network, the lack of the capability of handling real-time traffic in a network has discouraged the future development of the protocol.

Many attempts have been proposed in the past to enhance the CSMA and the CSMA/CD protocols. Some examples are: the CSMA/CD with Dynamic Priorities (CSMA/CD-DP) introduced by Kiesel et al. [KiKu83], the CSMA/CD with Deterministic Contention Resolution (CSMA/CD-DCR) protocol proposed by Takagi et al. [TaYS83], and the Virtual Time CSMA (VT-CSMA) protocol introduced by Molle et al. [MoKl85].

The CSMA/CD-DP protocol [KiKu83] uses additional delay to reduce collision probability. This proposal requires a complicated protocol management scheme to ensure the proper operation. A newly joined station

must realize the total number of stations in the network and the state of each station, which is difficult in a distributed system. As for the CSMA/CD-DCR protocol [TaYS83], it must be implemented in a nonstandard two unidirection bus topology which increases the network complexity and decreases the choices of network topologies. Finally, for the VT-CSMA protocol [MoKl85], an additional clock is used to schedule a new arrival in a way that two new arrivals that arrived in different time during a data frame transmission will be transmitted in different future time after the completion of that data frame transmission. This operation essentially adds a delay in each transmission to reduce the chances of collisions. However, the clock rate of the secondary clock plays an important role in the performance and requires constant adjustments based on traffic conditions to achieve an optimum result which may be complicated.

Despite numerous proposals in enhancing the CSMA protocol, many of the proposed protocols either introduces a delay in each data frame transmission to reduce the chances of collisions, or increases the complexity of the transceivers to achieve a better channel utilization. In this chapter, we propose a totally different approach to improve the CSMA/CD protocol, by **breaking the most sacred rule**. We make a significant improvement for CSMA/CD by allowing stations to interrupt a successful transmission by another station, and in this way make a reservation. We call our protocol CSMA with Reservations by Interruptions, or CSMA/RI in short.

In the following section, the CSMA/RI protocol will be described. In Section 4.2, we analyze CSMA/RI under the saturation and the disaster scenarios. The performance of CSMA/RI under various statistical traffic conditions is provided in Section 4.3. A performance comparison of several MAC protocols implemented in today's LANs is given in Section 4.4. In Section 4.5, we study the stability of the CSMA/RI protocol. Some issues of the CSMA/RI extensions and implementations are finally discussed in Section 4.6.

4.1 The CSMA/RI Protocol

Consider a slotted channel whereby time is divided into fixed length intervals. The purpose of the slotted channel assumption is to simplify the explanation of the CSMA/RI protocol. The implementation of CSMA/RI is not limited to a slotted channel.

Let T be the duration of a slot. All stations in the network are synchronized so that a data frame transmission is always commenced at the beginning of a slot. Let τ denote the maximum propagation delay between any two stations in the network. As in [Lam80], in order to use common mechanisms for detecting collisions and aborting collided transmissions in a slotted channel, the minimum duration of a slot is $T=2\tau$. The channel is sensed idle by all stations τ units of time after the end of a successful transmission, and a successful transmission is detected τ units of time after it is started. Figure 4.1 shows a snapshot of the CSMA/RI broadcast channel. Notice the analogy between this figure and Figure 2.33 in Chapter 2 which describes CSMA/CD.

The rules of the 1-persistent CSMA/CD protocol implemented in Ethernet can be summarized as follows.

(R1) If the broadcast channel is sensed idle, a ready station transmits its data frame immediately. It is required to monitor the channel status for the collision detection purpose.

(R2) If the broadcast channel is sensed busy, a ready station keeps monitoring the channel status. As soon as the channel becomes idle, the ready station transmits its data frame in the next slot with probability one.

(R3) Upon the detection of a successful transmission, each station reads the data from the ongoing data frame transmission into its local buffer. Only the station of which the data frame is addressing to may use the data, others should discard the data.

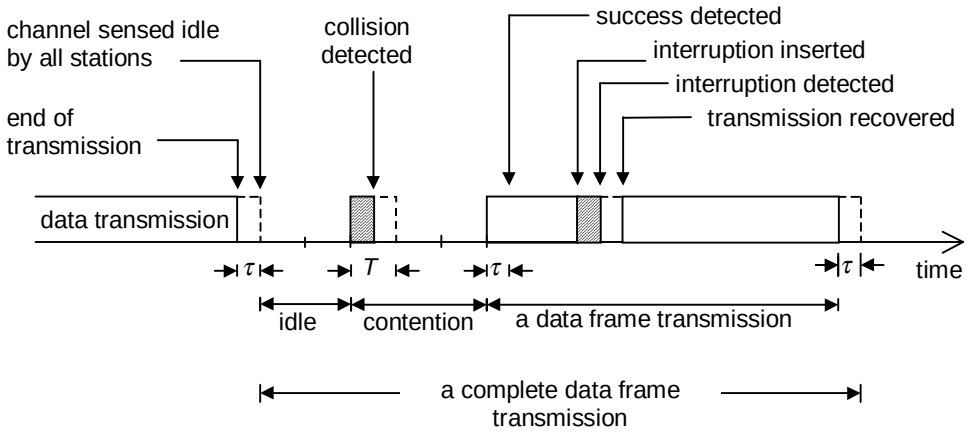


Figure 4.1: A snapshot of the CSMA/RI broadcast channel

(R4) If a collision is detected, each ready station reschedules the retransmission individually to some later time based on the BEB retransmission algorithm.

CSMA/RI enhances CSMA/CD by adding the reservations by interruptions operation. The reservation is performed during an ongoing data frame transmission. It is done by interrupting an ongoing data frame transmission with a short pseudo-noise transmission broadcasting to all stations. The ongoing data frame transmission is then resumed right after the interruption. Upon the completion of the entire data frame transmission, only the stations that have performed the reservation (by interruption), henceforth called *RI stations*, are allowed to access the channel. The rules respectively corresponding to the above CSMA/CD rules are described as follows. The CSMA/RI transmitter operation is described in Figure 4.2.

(R1*) Same as (R1).

(R2*) If the channel is sensed busy, in the case where a successful transmission is detected earlier, i.e. the current busy channel is due to a data frame transmission, (R2a*) applies to the ready station, otherwise the busy channel is due to a collision, and (R2b*) applies.

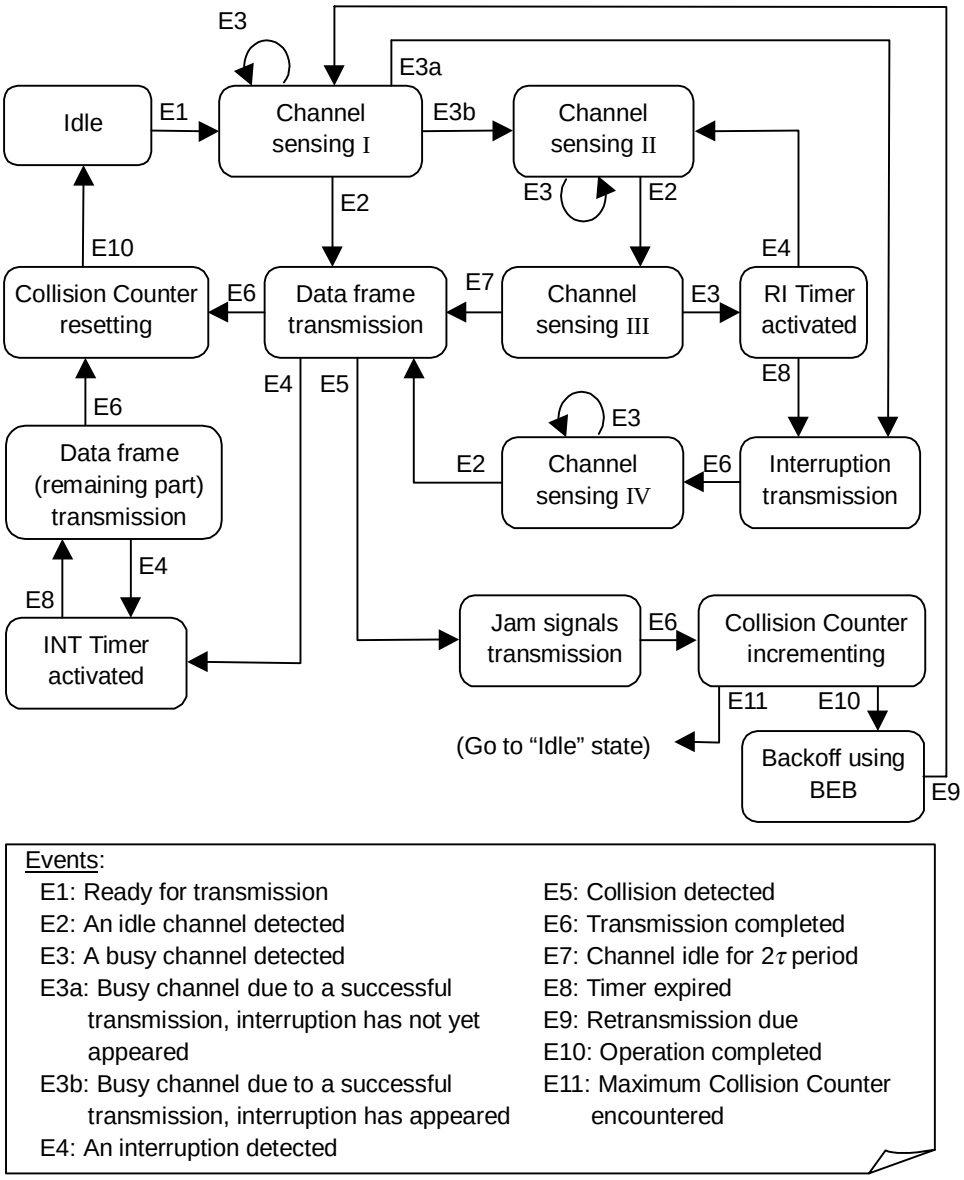


Figure 4.2: The finite state machine of a CSMA/RI transceiver

(R2a*) A ready station may interrupt the data frame transmission to make a reservation and to become an RI station only if the reservation has not already been performed during that ongoing data frame transmission. Otherwise, it becomes a backlogged station. Upon the completion of the successful data frame transmission, only RI stations are allowed to transmit in the next idle slot. Backlogged stations remain silent and continue to monitor the broadcast channel.

(R2b*) In this case, if a ready station becomes ready during a collision, it transmits in the next idle slot after the collision as in (R2).

(R3*) Upon the detection of a successful transmission, in addition to rule (R3), each station, either an RI which failed to obtain the channel transmission right or a backlogged station, waits for a randomly chosen waiting time that is not longer than the data frame transmission time. During the waiting time, the station is required to monitor the channel to detect if other stations make reservations. If such reservation is made by other stations, the station aborts its reservation attempt and becomes a backlogged station. Otherwise, if no one else has made a reservation during the station's waiting time, then by the time its waiting time expires, the station performs the reservation (by interruption). Notice that it is possible for more than one station to perform the reservation at the same time.

(R4*) Same as (R4).

(R5*) For a backlogged station, following a completion of a successful data frame transmission, if the channel remains idle for at least a slot, it becomes a ready station and transmits its data frame into the idle channel immediately as in (R1*).

Notice that in (R2a*), a station that becomes ready during a data frame transmission does not go through the process of choosing a random waiting time to perform the reservation. Instead, they make the reservation immediately if the reservation has not been carried out by the time they are ready. This way, they are making the random choice by the statistical nature of their arrivals so that a statistically similar access right can be maintained among all stations.

By (R3*), the reservation operation is carried out immediately after detecting a successful data frame transmission. Then, each station waits for a random number of slots to initiate the reservation procedure. This random number has a discrete uniform distribution based on the data frame size. For example, if the data frame size is equal to 11 slots, the waiting time will be

equal to any of the numbers: 2, 3, …, 11, each with probability 1/10. Recall that the first slot in a data frame transmission cannot be used due to the detection time requirement. If two or more stations pick the same waiting time, this leads to an interruption collision which in turn leads to a data frame collision. In this case, the collision will be resolved by the BEB retransmission algorithm.

The reservation procedure is performed by interrupting the ongoing transmission with a short period of pseudo-noise for τ units of time. This pseudo-noise is broadcast to all stations. Upon detecting the pseudo-noise, the sender ceases the data frame transmission, all other stations abort their reservation procedures and mark themselves as backlogged stations. Because the pseudo-noise only lasts for τ, it vanishes within the same slot that ends the reservation procedure. The sender then continues the transmission of the same data frame from the point where it was interrupted by the pseudo-noise. Since we assume that the channel is slotted, the fragmented data frames can be recovered if the receiver is equipped with memory to buffer the receiving data frame slot by slot. In practice, the recovery can be achieved by adding necessary information in the beginning of each data frame fragment so that the receiver can paste the fragmented data frame back together. This will be discussed in more details in Section 4.6 that deals with the implementation issues.

After the completion of a data frame transmission, only the RI stations can participate in the next data frame transmission. Backlogged stations remain silent until the next successful transmission is detected. This way, CSMA/RI divides the ready stations into two groups: the RI stations and the backlogged stations. This division reduces the number of stations participating in the contention, reducing the contention resolution time.

Notice that the duration of a data frame transmission is an important factor which affects the performance of CSMA/RI. Let n be the size of a data frame in slots. Since reservations are based on randomly choosing a number among the numbers: 2, 3, …, n, each with probability of $1/(n\text{-}1)$, the larger

the value of n is, the lower the interruption collision probability is, and hence the better is the performance.

In a noisy channel environment, it is not critical to distinguish between pseudo-noise and real noise. If noise occurred during an ongoing data frame transmission, in CSMA/CD, the sender will abort the transmission and schedule the retransmission of the entire data frame which is inefficient. By comparison, in CSMA/RI, the sender will correct it which saves the retransmission overhead. This is an additional benefit of CSMA/RI. Since noise is treated as pseudo-noise that represents a reservation, RI stations may not exist in this case. However, the absence of RI stations will not lead to deadlock because an idle period after a data frame transmission will lead to the normal CSMA/CD collision resolution that described in (R5*).

4.2 CSMA/RI under the Saturation and the Disaster Scenarios

We follow the assumptions made for CSMA/CD in [Lam80] for the analyses of CSMA/RI. We consider that

1. Stations are arranged in a star topology connected to a multiport repeater. The distance between any pair of stations is fixed. The signal propagation time from a station to all other stations is τ units of time.
2. The channel is slotted so that each station can only start its transmission in the beginning of a slot.
3. The collision detection time is ignored. The jam signal transmission is not included.
4. To use a common mechanism for detecting a collision and aborting collided transmissions in a slotted channel, the minimum duration of a slot is 2τ. Here we assume that the slot time is 2τ.
5. Under the perfect slotted channel assumption, the minimum cost of an interruption for reservation is 2τ.

Table 4.1: Summary of the protocol parameters of CSMA/RI for numerical computation and computer simulation

Parameter	Value
Protocol used	slotted CSMA/RI with BEB
Network topology	star network
Channel bit rate	10 Mb/s
Signal propagation time, τ	25.6 μsec (256 bit time)
Slot time	2τ (512 bit time)
Cost of a collision	one slot
Cost of an interruption	one slot
The time required for detecting a transmission end	τ
Number of stations, m	1,2,…,500
Data frame or payload size (headers are ignored), b	(i) 5 slots (320 bytes) (ii) 25 slots (1600 bytes)

6. The BEB retransmission algorithm is used for resolving collisions. All stations reset their Collision Counters upon the detection of a successful transmission.

The protocol parameters used for numerical and simulation computation are listed in Table 4.1.

4.2.1 The Saturation Analysis

Similar to the saturation analysis of CSMA/CD, we observe that the broadcast channel of CSMA/RI is repeating a cycle that consists of an idle period, a contention period and a data frame transmission period as illustrated in Figure 4.1. All cycles are statistically identical. Let $I(m)$, $C(m)$, $T(m)$ denote the random variables of the idle, contention, data frame transmission periods respectively for the case of m saturated stations, $m \geq 1$, and they are expressed in slots. Furthermore, let $T_u(m)$ be the period within a data frame transmission that the channel carries useful information, and $T_w(m)$ be the overhead period within the data frame transmission. The sum of $T_u(m)$ and $T_w(m)$ is equal to $T(m)$.

The throughput of CSMA/RI given m saturated stations in a network, S, can be expressed as

$$S = \frac{\mathrm{E}[T_u(m)]}{\mathrm{E}[I(m) + C(m) + T_u(m) + T_w(m)]}. \tag{4-1}$$

Under the saturation scenario, after a successful data frame transmission by a station, a new data frame is immediately generated for transmission. Thus no idle period will occur. That is $I(m)=0$.

Due to the reservations by interruptions mechanism in CSMA/RI, not all m saturated stations participate during the contention period, instead, only RI stations are contending for the channel access right. Recall that the number of RI stations is the number of stations which make the first interruption during the previous data frame transmission. In a slotted channel, for a data frame size of b slots, we recognize that the first slot of the data frame is not interruptible as it is used for the detection of a successful transmission. Thus, the interruption can only appear between the second slot and the last slot (inclusive) of the ongoing data frame transmission. Given r ready stations in the beginning of a data frame transmission, the probability, $P_{RI}(x,r)$ that x RI stations will be resulted is the probability that exactly x out of r stations choose a particular slot, say the i-th slot of the data frame transmission, within the b-1 data frame transmission slots to perform the reservation (by interruption) and all other r-x stations choose a slot after the i-th slot. It can be expressed as

$$P_{RI}(x,r) = \sum_{i=1}^{b-2} \binom{r}{x} \left(\frac{1}{b-1}\right)^x \left(1 - \frac{i}{b-1}\right)^{r-x}, \quad 1 \le x \le r-1 \tag{4-2}$$

with $\binom{r}{x} = \frac{r!}{x!(r-x)!}$, the binomial coefficient.

The case where all ready stations become RI stations, that is x=r, happens when all r stations choose the same slot to perform the reservation. The probability of the occurrence of this situation is

$$P_{RI}(x,r) = (b-1)\left(\frac{1}{b-1}\right)^{x}, \quad x = r \,. \tag{4-3}$$

Therefore, the contention period immediately after a data frame transmission of b slots with m saturated stations can be computed by conditioning and unconditioning on the number of RI stations, that is

$$\mathrm{E}[C(m)] = \sum_{i=1}^{m-1}\left[\left(L(i)-1\right)\cdot P_{RI}(i,m-1)\right] \tag{4-4}$$

where $L(i)$ is the average number of slots (including the successful transmission slot) required for the broadcast channel to obtain a successful transmission given i stations contending for the channel transmission right using BEB. $L(i)$ is given in Equation (2-57). In Equation (4-4), we consider that the station which performs a data frame transmission is unable to interrupt its own transmission so that only m-1 stations can interrupt the transmission.

Finally, for the data frame transmission period, we have

$$\begin{aligned} &\mathrm{E}[T_u(m)] = b \\ &\mathrm{E}[T_w(m)] = \begin{cases} 1+½ & ,m>1 \\ ½ & ,m=1. \end{cases} \end{aligned} \tag{4-5}$$

We assume that the size of data frames is constant, and the data frame transmission time is b expressed in slots, hence $T_u(m)$=b. The transmission overhead, $T_w(m)$, includes the overhead for the recovery of the interrupted slot (one slot) and the overhead for the detection of the end of a data frame transmission (half slot). Notice that if there is only one station in the network, no interruption will appear.

Combining Equations (4-1) to (4-5), the saturation throughput can be computed numerically.

Next, we derive the average transmission delay of a saturated station implementing the CSMA/RI protocol. Similar to the mean delay analysis for the CSMA/CD protocol under the saturation scenario, the transmission delay for CSMA/RI can be obtained by applying the Little's formula [Litt61]. Here, the mean queue size, $\overline{L}$, is equal to the number of saturated stations, and the arrival rate, $\overline{\lambda}$, is equal to the service rate. That is

$$\begin{aligned} \overline{L} &= m \\ \overline{\lambda} &= \frac{1}{\mathrm{E}[I(m) + C(m) + T_u(m) + T_w(m)]} . \end{aligned} \tag{4-6}$$

By applying the Little's formula, the transmission delay, D_S, for CSMA/RI can be computed by $\overline{L}\left(\overline{\lambda}\right)^{-1}$.

4.2.2 The Disaster Analysis

Given m stations in a network implementing the CSMA/RI protocol, under the disaster scenario, all stations start to transmit at the same time and each station transmits only one data frame in the entire recovery process. As in the disaster analysis for CSMA/CD, we first compute the duration of the recovery process, T_T. It is determined by

$$\begin{aligned} T_T &= \sum_{i=1}^{m} \left(\mathrm{E}[I(i) + C(i) + T_u(i) + T_w(i)]\right) \\ &= \sum_{i=1}^{m} \left(\mathrm{E}[I(i)] + \mathrm{E}[C(i)] + \mathrm{E}[T_u(i)] + \mathrm{E}[T_w(i)]\right) . \end{aligned} \tag{4-7}$$

For CSMA/RI, we yield

$$\begin{aligned}
&\sum_{i=1}^{m} \mathrm{E}[I(i)] = 0 \\
&\sum_{i=1}^{m} \mathrm{E}[C(i)] = \sum_{i=1}^{m-1}\left(\sum_{j=1}^{i} \left(L(j)-1\right) P_{RI}(j,i) \right) + \left(L(m)-1\right) \\
&\sum_{i=1}^{m} \mathrm{E}[T_u(i)] = m \cdot b \\
&\sum_{i=1}^{m} \mathrm{E}[T_w(i)] = \frac{3}{2} \cdot (m-1)
\end{aligned} \quad . \tag{4-8}$$

Notice that during the first contention period of the recovery process, all stations will participate in the contention resolution hence the duration of the first contention period equals $L(m)$-1 as in CSMA/CD. In the following contention periods, due to the reservations by interruptions operation, only RI stations participate in the contention, thus Equation (4-4) is used to determine the duration of the contention periods.

The transmission overhead for CSMA/RI includes the propagation delay (0.5 slot) plus the interruption recovery time (one slot) for all data frame transmissions except the final data frame transmission.

Applying Equation (4-8) to Equation (4-7), we obtain the duration of the recovery process for CSMA/RI under the disaster scenario.

We next look at the mean transmission delay a station experienced during the recovery process for CSMA/RI. The mean transmission delay, D_D, is computed by

$$D_D = \frac{d_1 + d_2 + d_3 + \cdots + d_m}{m} \tag{4-9}$$

where d_k is the mean transmission delay experienced by the k-th station departing the network. All stations will participate in the first contention period during the recovery process, thus the contention period lasts for $L(m)$-1. The duration of the following contention periods depends on the

number of RI stations, which is decided by Equation (4-4). By applying the similar approach for CSMA/CD in the previous chapter, d_k for CSMA/RI can be expressed recursively as

$$\begin{aligned} d_1 &= \big(L(m)-1\big)+(b+1) \\ d_k &= \big(d_{k-1}+½\big)+\left[\sum_{i=1}^{m-k+1}\big(L(i)-1\big)P_{RI}(i,m-k+1)\right]+(b+1), \\ &\qquad\qquad 2 \le k \le m-1 \\ d_m &= (d_{m-1}+½)+b\,. \end{aligned} \tag{4-10}$$

By substituting Equation (4-10) into Equation (4-9), we obtain the mean transmission delay for CSMA/RI under the disaster scenario.

4.2.3 Performance Results

In this subsection, we present the performance results for CSMA/RI under both the saturation and disaster scenarios. In particular, we plot (i) the saturation throughput; (ii) the mean transmission delay under the saturation scenario; (iii) the duration of the recovery process under the disaster scenario; and (iv) the mean transmission delay under the disaster scenario. The performance results are also compared with the results obtained for CSMA/CD given in the previous chapter.

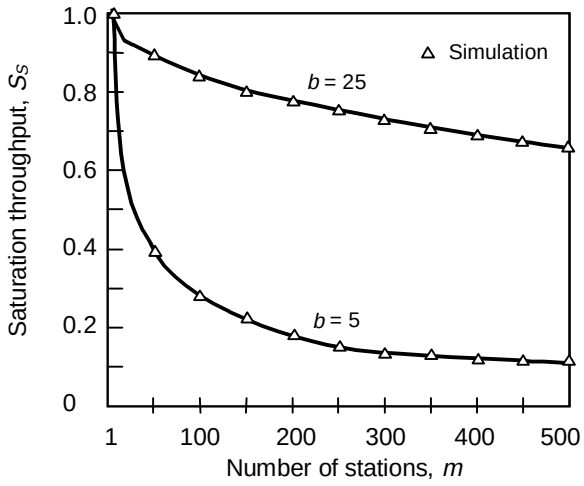


Figure 4.3: Saturation throughput for CSMA/RI for different data frame sizes

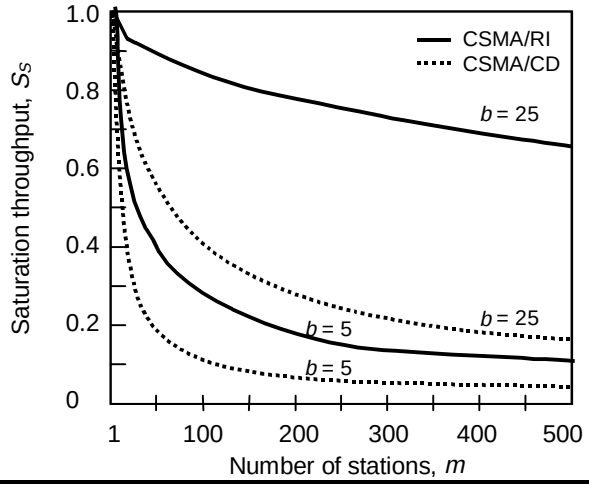


Figure 4.4: The saturation throughput comparison of CSMA/CD and CSMA/RI

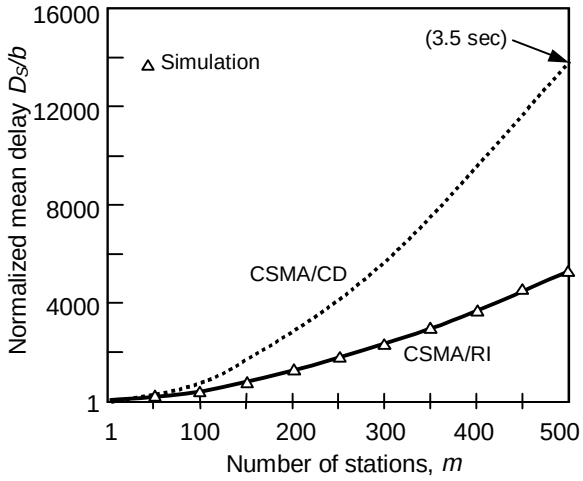


Figure 4.5: Normalized mean transmission delay for CSMA/RI under the saturation scenario for b=5

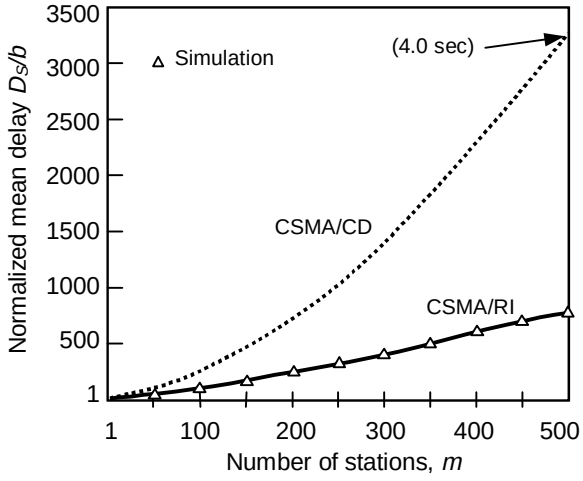


Figure 4.6: Normalized mean transmission delay for CSMA/RI under the saturation scenario for b=25

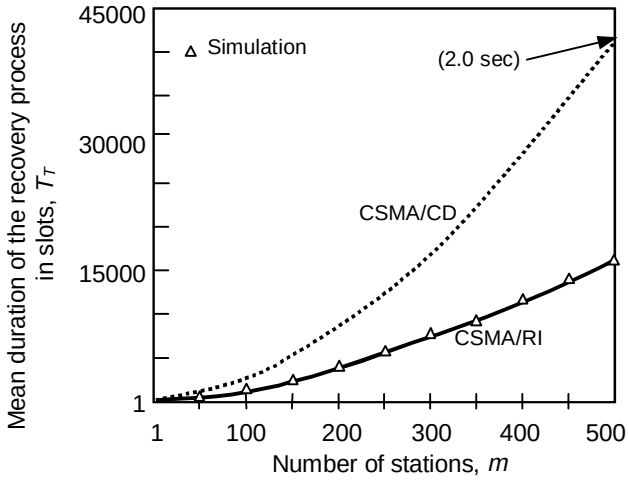


Figure 4.7: Mean duration of the recovery process versus number of stations under the disaster scenario for b=5

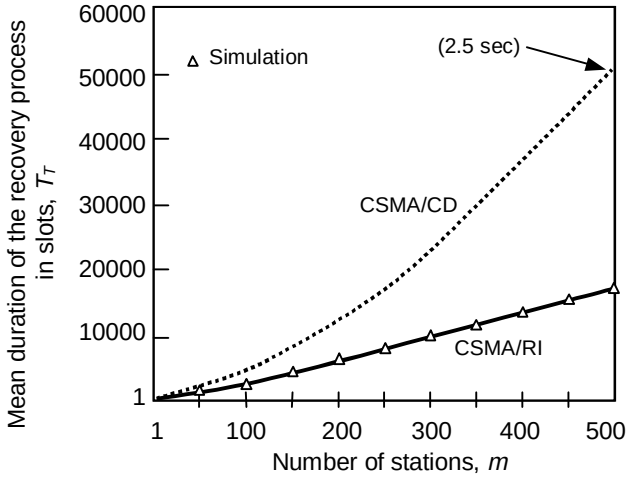


Figure 4.8: Mean duration of the recovery process versus number of stations under the disaster scenario for b=25

We first plot the saturation throughput of CSMA/RI in Figure 4.3 with different data frame sizes. The numerical results (plotted in lines) are verified by the simulation results (plotted in symbols). In Figure 4.4, we further compare the results with that of CSMA/CD (drawn in dotted lines) taken from Figure 3.2.

We see from Figure 4.4 that although there is not much benefit gained for CSMA/RI for short data frames under the saturation scenario, the benefit becomes very significant for long data frames. The saturation throughput of CSMA/RI for b=25 remains above 75% with as many as 200 saturated

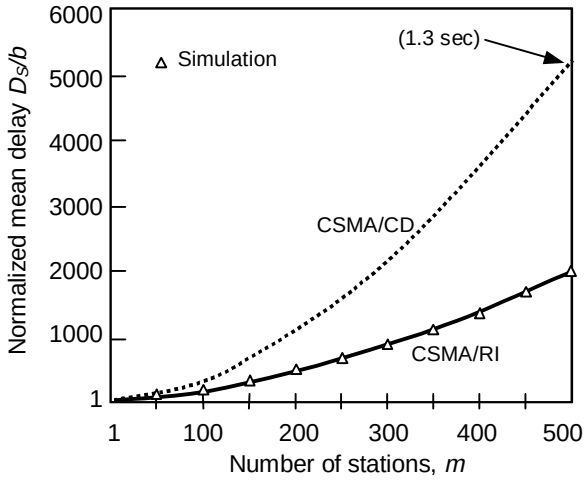


Figure 4.9: Normalized mean transmission delay for CSMA/RI under the disaster scenario for b=5

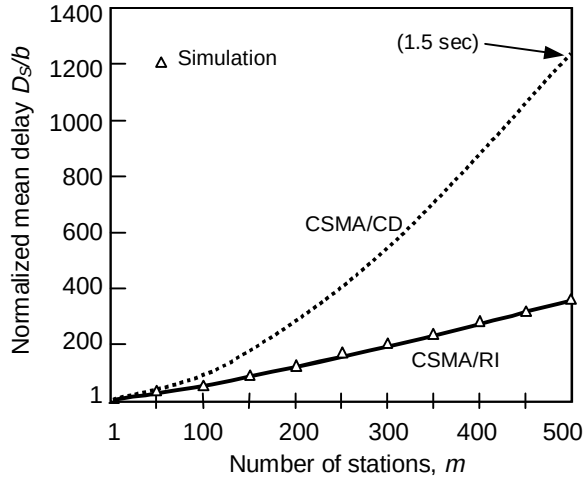


Figure 4.10: Normalized mean transmission delay for CSMA/RI under the disaster scenario for b=25

stations in the network while CSMA/CD achieves only 28% throughput level. Furthermore, CSMA/RI manages to sustain above 65% throughput level for 500 saturated stations. In contrast, only around 15% achieved by CSMA/CD under the same condition.

The mean transmission delay for CSMA/RI under the saturation scenario is presented in Figures 4.5 and 4.6 with two different data frame sizes. The mean transmission delay in seconds shown (in bracket) in the figures is calculated based on the protocol parameters given in Table 4.1. As can be seen, stations using CSMA/CD suffer longer transmission delay than those using CSMA/RI in both cases: average of 3.5 seconds experienced by 500

saturated stations in CSMA/CD with data frame size of 320 bytes, while only 1.3 seconds in CSMA/RI; 4.0 seconds and only around 0.9 seconds of delay experienced by the same number of saturated stations with longer data frames (1.6k bytes) in CSMA/CD and CDMA/RI respectively.

Similar results are found under the disaster scenario presented in Figures 4.7 to 4.10. In both cases of data frame sizes, CSMA/RI offers shorter mean transmission delay and the duration of the recovery process under the disaster scenario compared to that of CSMA/CD.

4.3 CSMA/RI under Statistical Traffic Conditions

Having shown that the CSMA/RI protocol performs better under the two extremely heavy load conditions – the saturation and the disaster scenarios, in this section, we further demonstrate and compare the performance of CSMA/RI with CSMA/CD under several statistical traffic conditions. The technique used here is based on the analytical method developed in Chapter 3, Section 3.4.

One important requirement for the CSMA/RI analysis using the Markovian framework is that the service process of CSMA/RI under the saturation scenario must be available. In Section 4.2.1, we have conducted the saturation throughput analysis for CSMA/RI, hence the saturation throughput results can be ready used here.

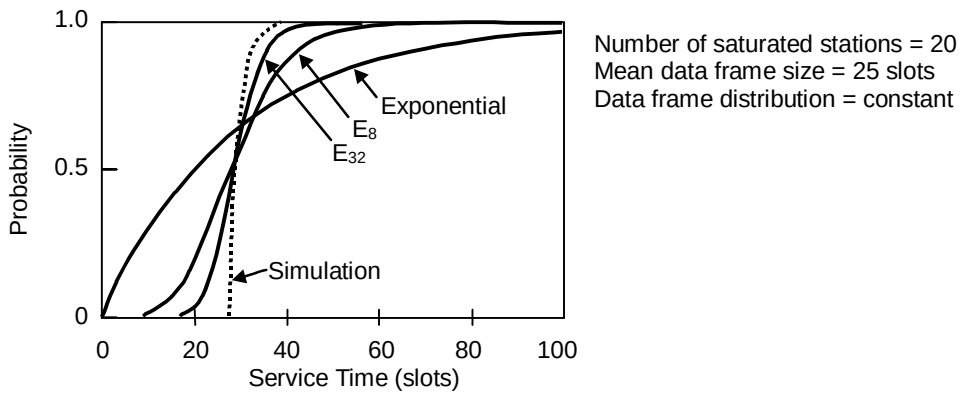


Figure 4.11: A CDF comparison of an exponential, Erlang random variables and the service time of CSMA/RI

Another requirement for the Markovian framework for CSMA/RI performance analysis is the fitting of the service process distribution. Since the saturation throughput analysis does not provide us the distribution of the service process, we use computer simulation to obtain the distribution of the CSMA/RI service process. In Figure 4.11, a typical distribution of the CSMA/RI service process is demonstrated, along with an exponential and two Erlang distribution functions. It is obvious that Erlang random variable has similar distribution characteristics to the CSMA/RI service process, here we use Erlang parameter with parameter eight, E_8, in the analysis.

In the following, the CSMA/RI protocol will be analyzed under the Bernoulli station arrival model with constant and dual size data frame assumptions.

4.3.1 Bernoulli Station Arrival Model with Constant Size Data Frames

With the saturation throughput results of CSMA/RI developed in Section 4.2.1, based on our proposed Markovian framework for the performance analysis, CSMA/RI under the Bernoulli station arrival model with constant size data frame assumption can be modeled into a state dependent $M/E_8/1/k$ system. Table 4.1 summarizes the protocol parameters used in the analysis and simulation.

Using the results developed in subsection 3.4.1, the mean transmission delay for CSMA/RI can be obtained numerically. The analytical results (shown in lines) are plotted in Figure 4.12. The simulation results (shown in symbols) are also presented to verify the numerical results. Furthermore, two different data frame sizes are assumed in the analysis.

It is clear from Figure 4.12 that CSMA/RI performs better with longer data frames. For 25 slot data frames, the maximum throughput of CSMA/RI reaches over 90%, whereas for 5 slot data frames, it is just over 55%. This is mainly due to the reason that for a longer data frame, the number of backlogged stations choosing the same slot for reservations by interruptions

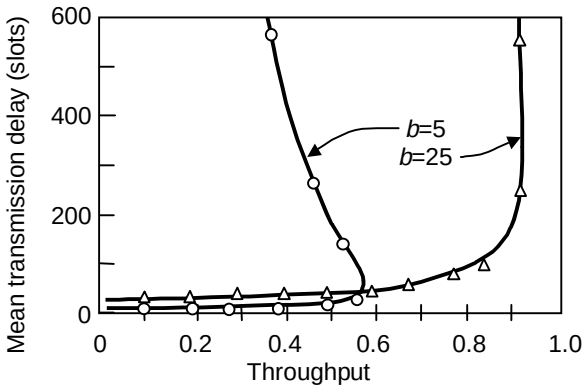


Figure 4.12: Mean transmission delay versus throughput for CSMA/RI under the constant size data frame assumption

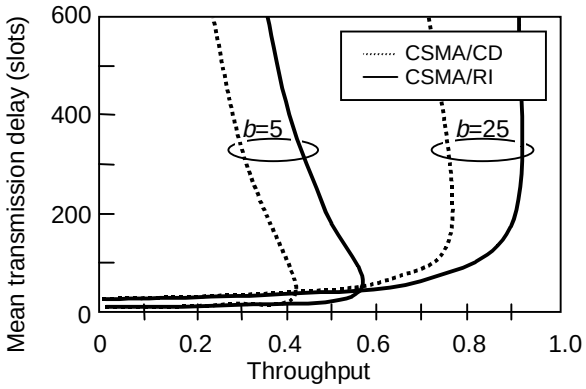


Figure 4.13: A comparison of the mean transmission delay for CSMA/CD and CSMA/RI

is relatively low compared to that of a shorter data frame. This results in a smaller number of RI stations and a shorter contention resolution time period, and hence the throughput for the longer data frames is higher.

To show the benefit of the reservations by interruptions scheme, we compare CSMA/RI with CSMA/CD in Figure 4.13, with two different data frame sizes. They are analyzed under same assumptions. The delay curves for CSMA/CD are taken from Figure 3.23. By comparing the two protocols, we see that CSMA/RI achieves a higher throughput level than CSMA/CD in both the data frame sizes. For 25 slot data frames, CSMA/CD can only achieve 75% throughput, while CSMA/RI reaches over 90% throughput, which is a 15% difference in throughput. This result once again shows the

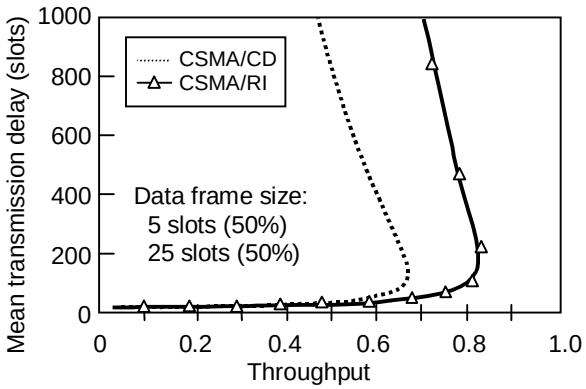


Figure 4.14: Mean transmission delay versus throughput for CSMA/RI under the dual size data frame assumption

benefit of CSMA/RI and confirms the performance advantage of CSMA/RI under this statistical traffic condition.

4.3.2 Bernoulli Station Arrival Model with Dual Size Data Frames

In this subsection, we further investigate the delay performance for CSMA/RI under the same assumptions as the previous subsection, except that the data frames distribution is now a dual size data frame distribution. This distribution is generally realistic than a constant size data frame distribution.

We consider two data frame sizes: 5 slots and 25 slots. A station generates either a 5 slot data frame or a 25 slot data frame with 0.5 probability. Based on the analysis given in subsection 3.4.2, an M/PH/1/k system is first constructed. Using the developed steady state balance equations, we can obtain the analytical results numerically by using SOR. Further protocol parameters are listed in Table 4.1.

We plot the analytical results in Figure 4.14. The results (shown in solid lines) are verified by the simulation results (shown in symbols). We also include the delay performance for CSMA/CD (shown in dotted lines) to compare the performance of the two protocols. As can be seen in the figure, the use of CSMA/RI leads to a higher achievable throughput level. In this

particular dual size data frame case, CSMA/RI offers as high as 80% channel throughput before the mean transmission delay increases exponentially, while CSMA/CD can only reach around 68% under the same arrival model.

4.4 Performance Comparison by Simulation

So far in this chapter, we have demonstrated the performance advantage of CSMA/RI over CSMA/CD under various traffic conditions. In this section, we further compare CSMA/RI to other important protocols. One of which is the token ring protocol that used in a token ring LAN.

The main difference between CSMA/RI and other collision-free MAC protocol is that even though CSMA/RI introduces a reservation scheme in its operation, the reservation scheme is executed in a collision based fashion, that is a reservation by a station does not necessary lead to a successful channel access. If two or more stations have reserved the broadcast channel at the same time, they have to compete for the channel access right by executing a collision resolution protocol. In contrast, for a collision-free MAC protocol such as the token ring protocol, once a station obtains a token, it is guaranteed to receive the channel access right.

In terms of performance, the token ring protocol generally offers a better performance than the CSMA/CD protocol [Bux81] due to its near perfect collision-free medium access scheme in the expense of a complicated and expensive transceiver. It is interesting here to include CSMA/RI in the performance comparison. In the following subsections, we first present the performance comparison of the CSMA/CD, the CSMA/RI and the token ring protocols under the dual size data frame assumption. We then compare CSMA/CD and CSMA/RI with a perfect scheduler (G/D/1) under LRD traffic to further demonstrate the benefit of CSMA/RI.

4.4.1 CSMA/CD, CSMA/RI and the Token Ring Protocol under Poisson Traffic

For the arrival model, we assume that data frames are generated from 100 identical stations according to a Poisson process with unlimited local buffer size. The signal propagation time in all cases is assumed to be 10μs. The ring latency of the token ring network is assumed to be 1 bit/station. A slotted broadcast channel is considered for all protocols. We further assume the limited-1 service discipline for the token ring protocol to achieve a fair comparison with CSMA/CD and CSMA/RI as the stations of CSMA/CD or CSMA/RI transmit no more than one data frame when they seize the broadcast channel. Moreover, we consider the realistic scenario of a dual size data frames. Let b_1 (b_2) be the size of short (long) data frames. The proportion of the short (long) data frames is 70% (30%). We consider two cases for b_1 and b_2 values. In the first case, we consider a 10Mbit/s bit rate

Table 4.2: Summary of the protocol parameters used for generating simulation results presented in Figures 4.15 and 4.16

Parameter	Value
Protocol used	slotted CSMA/CD with BEB, slotted CSMA/RI with BEB and the token ring protocol
Network topology for CSMA/CD and CSMA/RI	star network
End-to-end signal propagation time, τ	10 μsec
Slot time	2τ
Number of stations	100 (Poisson arrival process)
Data frame distribution	Dual size data frames with 70% short and 30% long data frames
Cost of a collision (for CSMA/CD and CSMA/RI)	one slot
Ring latency (for token ring)	1 bit/station
Service discipline (for token ring)	Limited-1
Case 1: Channel bit rate = 10 Mb/s Short data frame size, b_1 = 10 slots (250 bytes) Long data frame size, b_2 = 50 slots (1250 bytes)	
Case 2: Channel bit rate = 16 Mb/s Short data frame size, b_1 = 10 slots (400 bytes) Long data frame size, b_2 = 200 slots (8000 bytes)	

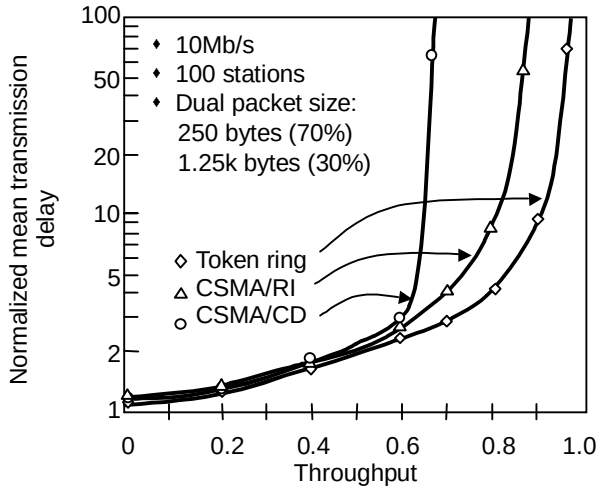


Figure 4.15: Performance comparison of the CSMA/CD, the CSMA/RI and the token ring protocols at 10Mb/s data rate

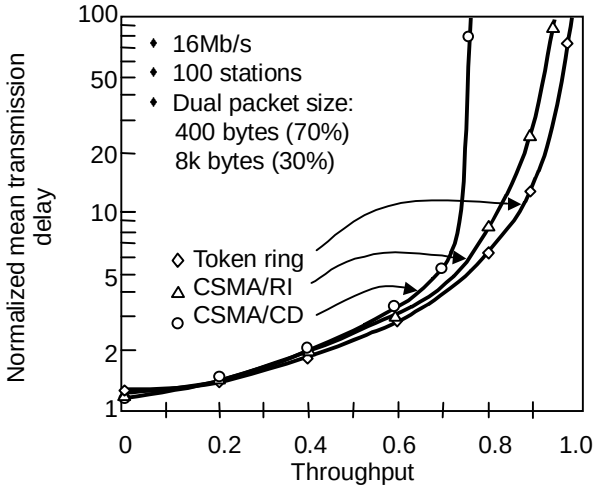


Figure 4.16: Performance comparison of the CSMA/CD, the CSMA/RI and the token ring protocols at 16Mb/s data rate

(corresponding to Ethernet), and 16Mbit/s bit rate (corresponding to the token ring) in the second case. The value for b_1 in both cases is set to be equal to 10 slots. The value of b_2 in both cases, however, is different. It is similar to the standard maximum Ethernet data frame size (b_2=50 slots or 1.25k bytes) in the first case and it is similar to the standard maximum token ring data frame size (b_2=200 slots or 8k bytes) in the latter case. Protocol parameters used in the simulation are summarized in Table 4.2.

Figures 4.15 and 4.16 plot the simulation results of the mean transmission delay for the CSMA/CD, the CSMA/RI and the token ring protocols. In both cases, we see a clear benefit of using CSMA/RI over CSMA/CD.

However, the token ring protocol, which behaves almost like a perfect scheduler, performs better than CSMA/RI. As the mean data frame size increases in the second case, we notice that the performance of CSMA/RI becomes closer to that of the token ring protocol. This indicates that CSMA/RI has the potential to perform like a perfect scheduler with a right choice of protocol parameters even it is a collision-based protocol.

4.4.2 CSMA/CD, CSMA/RI and G/D/1 under LRD Traffic

In subsection 3.6.4 in the previous chapter, we have demonstrated that Poisson traffic overestimates the performance of the IEEE 802.11 MAC protocol under real traffic with LRD traffic characteristics. Investigation of the performance of CSMA/RI under LRD traffic is hence important.

In [AdNZ99], it is shown that the M/Pareto can be used as a realistic traffic model for data traffic. M/Pareto traffic is a process composed of a number of overlapping bursts. Bursts arrive according to a Poisson process and have a Pareto distributed duration. It is characterized by four parameters which allow fitting of mean, variance, Hurst parameter, *H*, and the level of aggregation. For simplicity, we fix the mean of the Pareto distributed burst size to one, so that the process can be characterized by its mean, variance

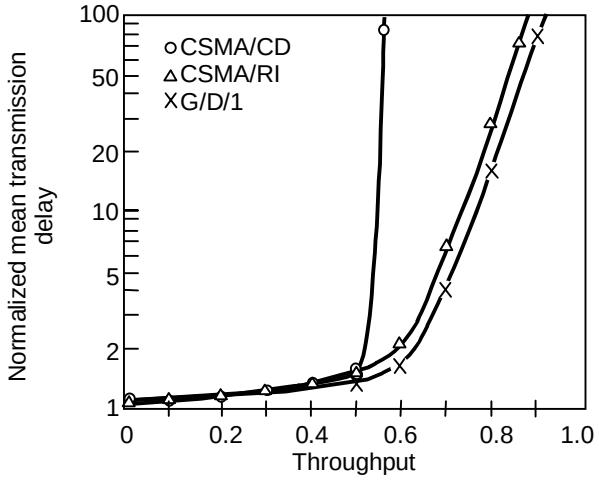


Figure 4.17: Performance of CSMA/CD, CSMA/RI and G/D/1 under M/Pareto traffic for *b*=50 and traffic burstiness = 0.2

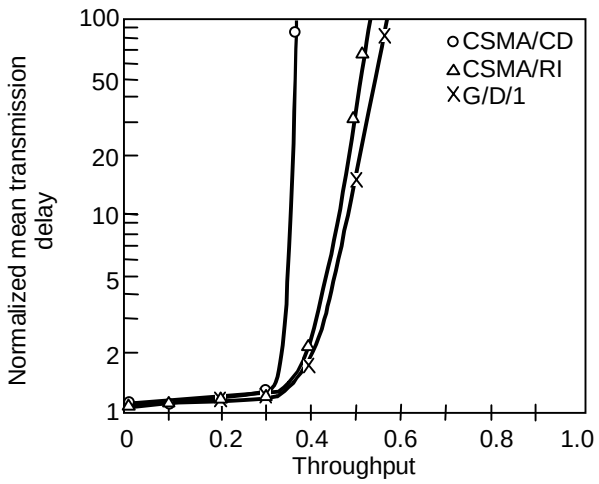


Figure 4.18: Performance of CSMA/CD, CSMA/RI and G/D/1 under M/Pareto traffic for b=50 and traffic burstiness = 0.5

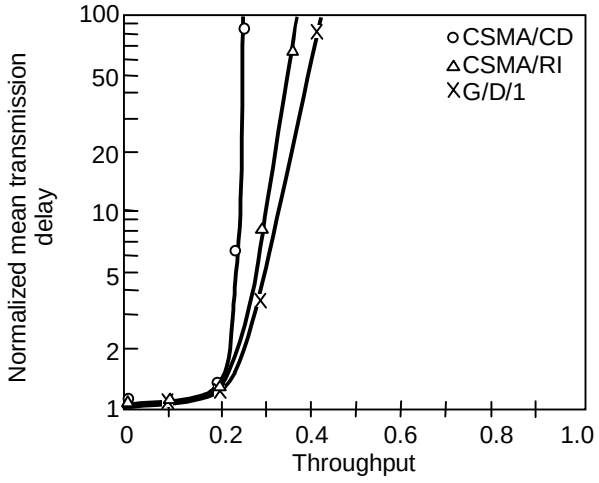


Figure 4.19: Performance of CSMA/CD, CSMA/RI and G/D/1 under M/Pareto traffic for b=50 and traffic burstiness = 0.8

and Hurst parameter.

We first assume a large population of stations and constant size data frames. Define the *burstiness* of LRD traffic as the ratio of the standard deviation to the mean. In Figures 4.17 to 4.20, we provide the delay performance of CSMA/CD and CSMA/RI for the case H=0.9 and with a range of burstiness. The choice of H=0.9 is based on the study of [LTWW94].

To obtain a fair comparison, CSMA/CD and CSMA/RI are fed by the same LRD traffic generated based on the M/Pareto model. We also compare the performance with the work conserving G/D/1 benchmark. The work

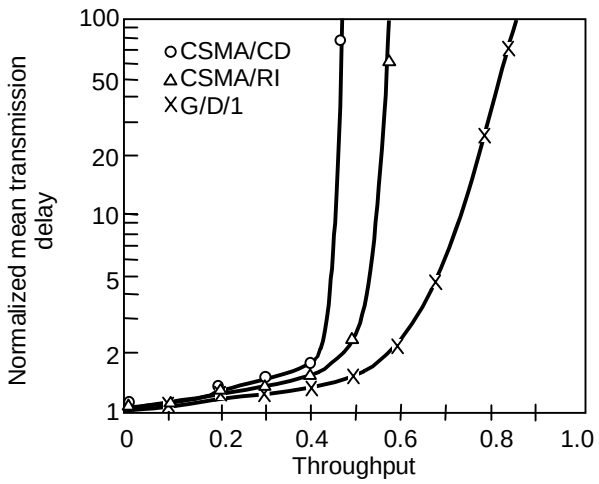


Figure 4.20: Performance of CSMA/CD, CSMA/RI and G/D/1 under M/Pareto traffic for *b*=10 and traffic burstiness = 0.2

conserving G/D/1 benchmark represents the delay performance of a perfect scheduler, which is the best delay performance possible to be achieved by a MAC protocol. In this case, it is important to know the upper limit of delay performance and by how far the delay performance of the CSMA/CD and the CSMA/RI protocols are from this benchmark. As demonstrated in Figures 4.17 to 4.19, for *b*=50, the performance of CSMA/RI is close to this benchmark.

Notice that these good performance results have been achieved for the case of 50 slot data frames. We know that the performance of CSMA/RI degrades if the data frame size is reduced. This is demonstrated in Figure 4.20, where the delay curve of CSMA/RI backs off from the G/D/1 benchmark, and approaches to that of CSMA/CD for *b*=10.

4.5 Stability Study

Throughout Chapter 3, we have demonstrated that the performance of the IEEE 802.3 MAC protocol, which is based on CSMA/CD, is relatively poor in a network that consists of a large number of active stations. This is mainly because as the number of active stations increases, the protocol becomes unstable. To achieve an acceptable level of the protocol performance, the number of active stations in a network may be limited so

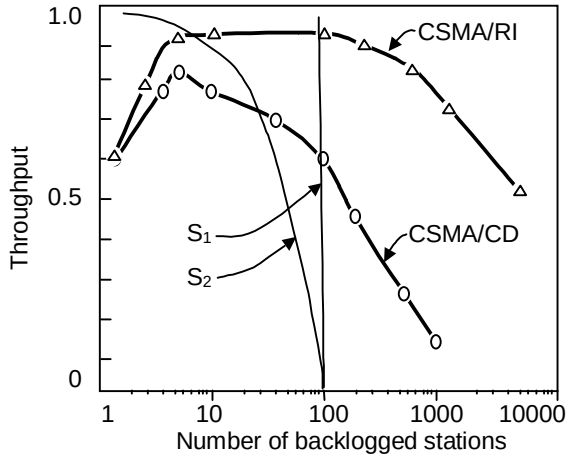


Figure 4.21: Stability of the CSMA/CD and the CSMA/RI broadcast channels

that the stability of the protocol can be maintained, but the cost of network deployments will also be increased in this case.

In this subsection, we study the stability of the IEEE 802.3 MAC protocol, or specifically the CSMA/CD protocol with the BEB retransmission algorithm, in more details. We also study and compare the stability of our proposed CSMA/RI protocol. Our study here is based on Lam's theory [LaKl75] first presented in subsection 2.5.1 in Chapter 2.

We first produce the results of throughput versus the average number of backlogged stations for both CSMA/CD and CSMA/RI by using computer simulation. The results are plotted in Figure 4.21. Each point on the curve indicates the throughput of the protocol given a certain the number of backlogged stations. The protocol parameters of CSMA/RI used for the simulation are summarized in Table 4.1. Equivalent protocol parameters are considered for CSMA/CD. Each result is obtained by adjusting the Poisson load to a specific level so that the simulated protocol is operated at a desired number of backlogged stations with a certain throughput level.

One obvious observation from Figure 4.21 is the number of backlogged stations of the two protocols when they are at their maximum throughput levels. For CSMA/CD, its throughput level peaks at around merely eight backlogged stations, and the throughput level drops quickly as the number

of backlogged stations increases. This indicates that CSMA/CD with BEB is best operated in a network of eight active stations, and more active stations will lower the throughput. On the other hand, the CSMA/RI can be operated at its best performance in a wide range of the number of backlogged stations, which is between around 8 and 100. In other words, a change in backlogged stations due to a change in traffic conditions does not degrade the performance of CSMA/RI provided that the number of backlogged stations is below 100.

Consider a network of 100 stations. In the figure, two channel load lines, S_1 and S_2 are included. Channel load line S_1 represents a busy environment where all stations are saturated. A station is said to be saturated if it generates a new data frame immediately after it has completed its previous data frame transmission. In this case, the access probability, σ, in the channel load line equation (see Equation (2.46)) is set to one. Whereas in the case of the load line S_2, all stations equally share the total bandwidth. In this case, the access probability σ = 1/100 or 0.01.

We notice that both the channel load lines S_1 and S_2 intersect the throughput curve of CSMA/RI at around 0.9, which is its maximum throughput. For CSMA/RI the range between these intersections are both efficient and stable. This result suggests that if the network consists of 100 stations, CSMA/RI can achieve high throughput even under very heavy load conditions (e.g. S_1). On the other hand, CSMA/CD can be operated at its peak throughput level in a narrow range of traffic conditions. Under both S_1 and S_2 load conditions, CSMA/CD is operated at relatively low throughput points, especially under S_1 load condition.

The results presented in the above lead us to conclude that CSMA/RI is stable, even it is operated in a network consisting of a large number of active stations. By comparison, the CSMA/CD channel throughput starts to drops when there is as few as eight active stations.

4.6 Extensions, Implementations and Limitations

4.6.1 Priority Scheme

In a LAN, there often exist different types of traffic. Some of which, such as multimedia traffic, may require more attention than other traffic types. Having a feature that can recognize and handle different types of traffic with different priority will help in meeting a desired QoS in a LAN.

Several proposals had been introduced in the past to include a priority scheme in the CSMA/CD protocols [FrBo80, Toba82b, IiYK83, HuRa85]. However, all of those proposals require additional overhead to perform the task. In other words, the CSMA/CD protocol will be degraded to some degree when a priority scheme is introduced.

In contrast, the introduction of a certain priority scheme in CSMA/RI is relatively straightforward and requires no additional overhead. For different types of traffic, CSMA/RI stations may use different probability distribution functions for choosing the random waiting time during an ongoing data frame transmission for making a reservation by an interruption. This way, traffic with a higher priority may have a higher probability to reserve the channel than those with a lower priority.

One simple example of the priority scheme is that stations which carry a low priority data frame may not interrupt the first few slots of an ongoing transmission, giving stations which carry a high priority data frame the opportunity to interrupt the ongoing transmission earlier and to gain the channel transmission right.

Other schemes such as adaptively adjust the probability distribution functions of the waiting time for interruption according to the load conditions and the statistics of traffic types are also possible. However, the design of an advanced priority scheme is not the focus of this dissertation and it will not be studied here.

4.6.2 Detection of an Interruption

To enable the detection of an interruption, the pseudo-noise signals can be transmitted in different data rate and/or with different signal strength than the signals used for data frame transmissions. It is known that with the existing collision detection feature in the IEEE 802.3 MAC protocol, a sender can pick up interference and detect an interruption. The problem here is how the other passive stations in the network can detect such an interruption.

The IEEE 802.3 MAC protocol uses a self clocked encoded signal for all transmissions. Due to the imperfect hardware, a slight drift in data rate of a signal is possible. One robust scheme to combat this slight drifting problem of signals is by using a device known as *Digital Phase Lock Loop* (DPLL). Such a device provides a mechanism for a receiver to be synchronous precisely to a sender. However, when an interruption occurs, due to the overlapping of two strong signals, the clock of the data frame transmission signals will be altered, and DPLL will attempt to adjust its local clock in order to synchronize itself to the received corrupted signals. In this case, an additional circuit can be introduced to monitor the local clock of DPLL. If the local clock has drifted outside a given threshold range for a period of time, that circuit may raise a signal indicating an occurrence of an interruption. This way, an interruption can be detected by passive stations.

To make the detection of an interruption more robust, we recognize that when interference occurs, the signal strength will also be altered. Hence by monitoring the signal strength, an interruption can be detected by passive stations. Some proposals, such as [Tsyb80, GePa82], have been made to introduce a power detector into Ethernet transceivers for estimating the number of stations participating in a collision since if more stations participate in a collision, the overall signal strength will be higher. The proposed power detector can be introduced here for passive stations to detect an interruption.

4.6.3 Recovery of an Interrupted Transmission

Once an ongoing data frame transmission is interrupted, with the existing collision detection feature of the IEEE 802.3 MAC protocol, a sender will detect the interruption. We further assume that with additional devices described in the previous subsection, all passive stations, including the receiver, will also detect the interruption.

Recall that the interruption is done by interfering an ongoing transmission with a short period of pseudo-noise transmission. After the pseudo noise vanishes from the network, the sender can resume the data frame transmission without competing the channel access with others since all other stations will detect the interruption and are expected to wait for the resumption of the previous ongoing data frame transmission.

Since the data frame transmission has been interrupted, we recognize that the sender should include enough information in the beginning of the transmission after an interruption to ensure that the receiver can paste the fragmented data frame back without error. Some bits that are transmitted just before the detection of an interruption may also be repeated to increase the robustness of the protocol.

4.6.4 The Boundary of Applications

The advantage of CMA/RI comes from the reservations by interruptions operation. With the operation, backlogged stations are allowed to interrupt an ongoing data frame transmission to obtain the channel transmission right. The transmission time of a data frame is an important factor in the performance of CSMA/RI in this case. A longer data frame transmission provides more choices of time slot to perform an interruption, which leads to a lower interruption collision probability and a shorter collision resolution time that follows immediately after the end of the interrupted data frame transmission.

To take the full advantage of CSMA/RI, the data frame transmission time in a network should be long. In many of our numerical and simulation experiments given in this chapter, we assume a 10Mb/s data rate network spanning 2.5km using 1600 bytes of data frame, and the advantage of CSMA/RI is clear. The choice of these parameters is based on the IEEE 802.3 standard.

For a high speed LAN that is operated at 100Mb/s, the data frame transmission time is reduced by 10 times. To maintain the proper operation and performance of CSMA/CD as well as CSMA/RI, the end-to-end signal propagation time must also be reduced by 10 times. This can be achieved easily by reducing the network coverage from 2.5km to 200m. These network parameters are used for Fast Ethernet. With these network parameters, we can also expect a similar performance advantage for CSMA/RI over CSMA/CD at a 100Mb/s LAN.

As the data rate of LANs increases, the data frame transmission time becomes even shorter. At a 1Gb/s data rate LAN, to maintain the proper operation of CSMA/CD and CSMA/RI, the maximum network coverage must be less than 20m, which is considered unrealistic. To retain the maximum network coverage of 200m as in Fast Ethernet, the transmission time of a long data frame will become very short compared to the end-to-end signal propagation time, hence there may only be a few rooms during a data frame transmission for interruptions. As a result, the reservations by interruptions solution discussed in this chapter becomes somewhat unattractive.

However, with some modifications to the proposed reservations by interruptions scheme, the scheme can be further improved so that the boundary of application of this solution can be expanded. One such example is introduced by Sheu et al. [ShWW01]. Based on our reservations by interruptions scheme, Sheu et al. developed the new protocol named CSMA with Adaptive Reservations by Interruptions or CSMA/ARI in short. It is shown that CSMA/ARI performs between than both the CSMA/CD and the

CSMA/RI protocols. Since the description of CSMA/ARI is not the scope of this dissertation, it will not be discussed here.

4.7 Summary

In this chapter, we introduced a new protocol named CSMA/RI to improve the popular CSMA/CD protocol. The CSMA/RI protocol uses a novel reservation scheme to reduce the collisions. In CSMA/RI, we allow a ready station to interrupt an ongoing data frame transmission to reserve capacity. According to our performance analyses presented in Sections 4.2 and 4.3, we have demonstrated that our proposed CSMA/RI protocol performs better than CSMA/CD under various load conditions including the saturation and the disaster scenarios, the Bernoulli station arrival model with fixed and dual size data frames.

Furthermore, we have also conducted simulation experiments in Section 4.4 to compare CSMA/RI with the token ring protocol under Poisson traffic, and to compare CSMA/RI with CSMA/CD and G/D/1 under M/Pareto traffic. Our simulation results suggest that for realistically long data frame, CSMA/RI performs almost as good as the token ring protocol and G/D/1.

Using the stability theory developed by Lam, we studied the stability of CSMA/RI in Section 4.5. The results indicate that CSMA/RI is stable in a network of hundreds of active stations. In contrast, CSMA/CD is only stable in a network with a small number of active stations.

Some issues related to the implementation of CSMA/RI were discussed in Section 4.6. They include the design of a priority scheme for CSMA/RI, the detection of an interruption, the recovery of an interrupted transmission, and the boundary of the application.

Based on our studies given in this chapter, we concluded that our proposed reservation by interruption operation may be a good solution to improve the performance of a MAC protocol that is operated in a network of which the

data frame transmission time is generally longer than the end-to-end signal propagation time such as 10Mb/s and 100Mb/s LANs, or a wireless LAN.

5 Request Contention Multiple Access

The recent advance in fiber optic communication technologies has encouraged the design and deployment of optical LANs. The immediate advantage of optical LANs is a high data rate such as gigabit access for end users. To take the advantage of this high data rate, the IEEE 802.3z working group has been established to develop the Gigabit Ethernet (GE) standard.

The IEEE 802.3z working group has retained the CSMA/CD protocol as the MAC protocol for access arbitration in Gigabit Ethernet [IEEE00, KaCK98]. Due to the gigabit data rate, to achieve backward compatibility and to guarantee the proper operation of CSMA/CD, the IEEE 802.3z working group has introduced the *carrier extension* operation. In brief, if a data frame is too short for the collision detection purpose, senders must append predefined carrier signals to the short data frame for a period of time that is long enough for collision detection. Another modification to the protocol is the slot time parameter. In the shared Gigabit Ethernet, the slot time is increased by almost 10 times from 512-bit time in 10Mb/s or 100Mb/s Ethernet to 4096-bit time. Consequently, each collision in Gigabit Ethernet results in a loss of 10 times more data than in the 10Mb/s or 100Mb/s Ethernet.

When CSMA/CD was introduced a quarter of a century ago, it has been considered one of the first random access MAC protocols that in some sense is a collision based as well as a reservation based protocol (see page 317 in [BeGa92] which views “the first portion of a packet as making a reservation for the rest”). This property increases the efficiency of CSMA/CD compared to other random access MAC protocols. However, interestingly, in gigabit networks, where the transmission time of data frames become “small” relative to the propagation delay, CSMA/CD looses its reservation

"affiliation" and it becomes a pure collision protocol in some cases. Hence the efficiency gained from the reservation property in CSMA/CD vanishes. Significant performance degradation due to the high data rate is expected in the shared Gigabit Ethernet.

In this chapter, we propose a new MAC protocol for Gigabit LANs. We call it the *Request Contention Multiple Access* (RCMA) MAC protocol. RCMA has the following traits: (i) it is based on a distributed control principle, no central controller is required; (ii) it is simple to implement; (iii) it achieves efficient scheduling and fairness with minimum overhead, intelligence and complexity; (iv) it is more efficient than the IEEE 802.3z MAC protocol, and unlike IEEE 802.3z that suffers from efficiency degradation as the number of active stations increases, the performance of RCMA remains stable; and (v) unlike the IEEE 802.3z MAC protocol, RCMA can easily support service differentiation.

In RCMA, we propose that a station which wishes to access the medium, given that the medium is free, will first broadcast a very short request by which it will make a reservation for further data transmission. More importantly, the channel assignment task in RCMA will be performed in a distributive manner without the need for an intelligent central controller. Because the RCMA request is much shorter than an IEEE 802.3 data frame, the probability of collisions is significantly reduced. Also, service differentiation can be achieved by prioritizing the request. For some low priority services, stations can request for the channel access right with a lower request priority number so that delay sensitive services can be served first, thus the introduction of service differentiation in the MAC layer is possible.

Our proposed RCMA protocol is to be operated in a passive optical LAN. There are many proposed MAC protocols designed for a passive optical LAN in the literature [HaKS87, Mukh92, JeUn92, SuKG94, Ashr94, MoBa00]. Unlike RCMA, these protocols must be operated in a passive optical network that employs Wavelength Division Multiplexing (WDM) to

create multiple channels on a single fiber. One of the created channels is dedicated for transmission scheduling with a particular scheme, and other channels are used for data transmissions. Because the scheduling of a transmission is performed in a different channel without affecting other ongoing data transmissions, the performance is generally better than those that has only one shared channel for both scheduling and data transmission purposes. However, the WDM solution for LANs is considered expensive and hence they are unlikely to be used in LANs at this stage.

The chapter is organized as follows. In Section 5.2, we provide the detailed operation of the IEEE 802.3z MAC protocol. The saturation throughput of the protocol will also be studied. In Section 5.3, we describe our proposed MAC protocol, RCMA. We then analyze the RCMA protocol under the saturation scenario. A comparison between the IEEE 802.3z MAC protocol and our proposed RCMA will be demonstrated. Finally in Section 5.3, we discuss some possible extensions and implementation limitations of RCMA.

5.1 The IEEE 802.3z MAC Protocol

5.1.1 Protocol Description

The CSMA/CD protocol has been retained by the IEEE 802.3z working group as the MAC protocol for the shared Gigabit Ethernet. The detail operation of CSMA/CD is given in Section 2.4 in Chapter 2. In the design of the CSMA/CD protocol, to ensure the proper operation of the collision detection feature, the data frame transmission time must always be longer than twice the signal propagation time between any pair of stations in the network. A short data frame must be padded with additional bits so that it is long enough to ensure that the collision detection feature works properly. In the IEEE 802.3 standard, a minimum IEEE 802.3 data frame size is of 64 bytes. This minimum IEEE 802.3 data frame carries 46 bytes payload.

The 10BASE-5 Ethernet operating at 10Mb/s may span over 2.5km under this specification. At 100Mb/s data rate, the 100BASE-T Ethernet must be

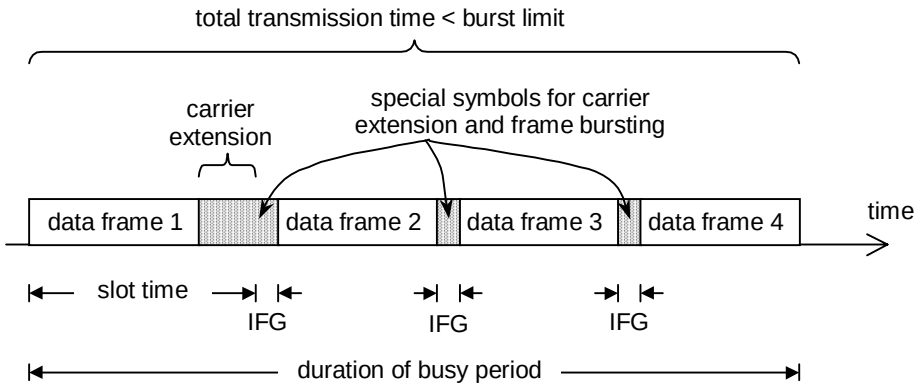


Figure 5.1: Illustration of the frame bursting operation

operated within 200m to ensure the proper operation of the collision detection feature. Since the same data frame format is used for the IEEE 802.3z Gigabit Ethernet MAC protocol, it is expected that the network can only span up to 20m, which is not practical. To overcome this problem, the IEEE 802.3z working group introduces the *carrier extension* operation. The solution is similar to the earlier one where if the transmission time of a data frame is too short for the collision detection purpose, the data frame must be padded. In Gigabit Ethernet, some predefined carrier signals are used to append to a short data frame transmission so that the entire transmission time reaches the minimum transmission time requirement, which is 4096-bit time.

Apart from the carrier extension operation, the slot time is increased from 512-bit time in 10Mb/s or 100Mb/s Ethernet to 4096-bit time. This is mainly due to the gigabit data rate. The increase in slot time in the shared Gigabit Ethernet leads to a loss in 10 times more data than in the 10Mb/s or

Table 5.1: Some selected protocol parameters of the shared Gigabit Ethernet

Parameter	Value
Network topology	star network
Channel bit rate	1Gb/s
Slot time	4096 bit times
Maximum data frame size	1518 bytes
Minimum data frame size	64 bytes
Burst limit	65536 bits

100Mb/s Ethernet.

In an attempt to improve the performance of the shared Gigabit Ethernet, the IEEE 802.3z working group introduces the *frame bursting* operation. After a sender, which carries two or more data frames, has acquired the channel transmission right, it transmits the first data frame using the standard CSMA/CD rules, in addition, the sender is allowed to continue the transmission of the following data frames until a further data frame transmission will cause the total transmission time to exceed a predefined burst limit. Data frames within the same transmission are separated by the carrier signals for a period of a predefined *interframe gap* (IFG). Figure 5.1 illustrates the frame bursting operation of the IEEE 802.3z MAC protocol. Some important protocol parameters are summarized in Table 5.1.

5.1.2 Saturation Throughput Analysis

The saturation throughput analysis of the shared Gigabit Ethernet is based on the saturation throughput analysis of CSMA/CD performed in subsection 3.2.1 in Chapter 3. One important operation of Gigabit Ethernet that is missing in the presented analysis is the carrier extension operation. We do not include the optional frame bursting operation of Gigabit Ethernet in this analysis.

In the saturation throughput analysis given in subsection 3.2.1, the throughput for CSMA/CD protocol implementing the BEB retransmission algorithm, S_{GE}, can be expressed as

$$S_{GE} = \frac{\mathrm{E}[T_u(m)]}{\mathrm{E}[I(m) + C(m) + T_u(m) + T_w(m)]} \tag{5-1}$$

where m is the number of saturated stations, and $I(m)$, $C(m)$, $T_u(m)$, $T_w(m)$ are the random variables of an idle period, a contention period, a useful transmission period, and a transmission overhead period respectively in a statistically identical cycle on the broadcast channel.

According to the results given in subsection 3.2.1, we have the following

$$\begin{aligned} &\mathrm{E}[I(m)] = 0 \\ &\mathrm{E}[C(m)] = \big(L(m)-1\big)\cdot T_{GESLOT} \end{aligned} \tag{5-2}$$

where $L(m)$, given in Equation (2-57), is the mean number of slots required to resolve a collision caused by m stations, and to obtain a successful transmission. T_{GESLOT} is the slot time in Gigabit Ethernet.

Define T_{IFG} to be the duration of the IFG, and τ to be the signal propagation time between the two farthest stations in the network. Given a certain data frame size distribution, let $T_{CARRIER}$, T_{FRAME} and $T_{PAYLOAD}$ be the mean values of random variables representing the duration of the carrier extension, the IEEE 802.3 frame transmission time, and the payload transmission time within a successful transmission respectively. The mean values for the useful as well as the overhead transmission time are

$$\begin{aligned} &\mathrm{E}[T_u(m)] = T_{PAYLOAD} \\ &\mathrm{E}[T_u(m) + T_w(m)] = T_{CARRIER} + T_{FRAME} + T_{IFG} + \tau \end{aligned} \tag{5-3}$$

By substituting Equations (5-2) and (5-3) into Equation (5-1), the saturation throughput of IEEE 802.3z can be found. It is important to note that due to the capture effect in the Ethernet protocol that results in temporary unfairness, a minor modification of the Ethernet protocol has been applied in this saturation throughput analysis to eliminate the transient effect influencing the steady state results of the saturation throughput. The modification is discussed in Section 3.2 in Chapter 3. The results here represent the worst case of the actual Gigabit Ethernet throughput under the saturation scenario.

To complete the analysis, we consider the following data frame distribution. We assume that the data frames are mixes of long and short frames with 35% of the frames being short, and 65% being long. The sizes for long and short data frames are the maximum and the minimum standard sizes of the IEEE 802.3 data frames. Since the transmission time of the short data

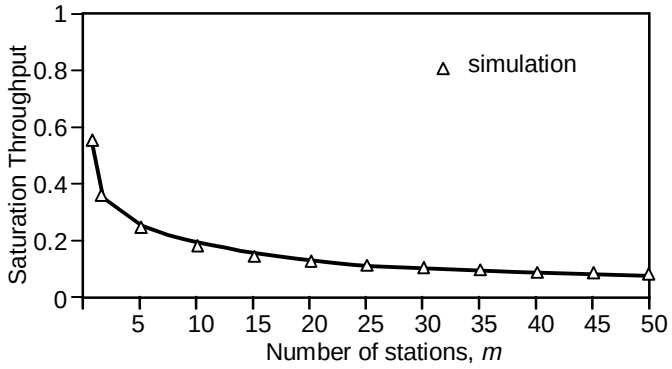


Figure 5.2: Saturation throughput for IEEE 802.3z

frames is below a Gigabit Ethernet slot time, additional carriers will be appended to the short data frames. With this data frame size distribution, we yield

$$\begin{aligned} T_{CARRIER} &= 0.35\times 3.584\mu\,\text{sec} = 1.2544\mu\,\text{sec} \\ T_{PAYLOAD} &= 0.35\times 0.368\mu\,\text{sec} + 0.65\times 12\mu\,\text{sec} = 7.9288\mu\,\text{sec} \\ T_{FRAME} &= 0.35\times 0.576\mu\,\text{sec} + 0.65\times 12.208\mu\,\text{sec} = 8.1368\mu\,\text{sec}\,. \end{aligned} \tag{5-4}$$

The above values are obtained based on the protocol parameters given in

Table 5.2: Summary of the protocol parameters used for generating results presented in Figure 5.2

Parameter	Value
Protocol used	slotted CSMA/CD with BEB
Data rate	1Gb/s
Network topology	star network
Signal propagation time in the network, τ, including the delay of a repeater	2.048μsec
Slot time, T_{GESLOT}	2τ (or 4.096μsec)
The IFG time duration, T_{IFG}	0.096μsec
Cost of a collision	one slot time
Number of stations, m	1,2,…,50
Data frame distribution	Dual size data frames with 35% short and 65% long data frames
The useful transmission time for a short IEEE 802.3 data frame	0.368μsec (46 bytes)
The useful transmission time for a long IEEE 802.3 data frame	12μsec (1500 bytes)
The IEEE 802.3 data frame overhead including preamble and SFD	0.208μsec (26 bytes)

Table 5.2. Having obtained these values, the Gigabit Ethernet saturation throughput can be determined. Figure 5.2 shows the saturation throughput of Gigabit Ethernet for the number of stations between one and 50. As can be seen in the figure, the Gigabit Ethernet is operated below 0.3 channel throughput level for merely five saturated stations with the considered data frame distribution. The saturation throughput continues to drop as the number of saturated stations grows. The results indicate that due to the carrier extension overhead during a transmission and significant bandwidth losses in a collision, the CSMA/CD protocol does not offer a high channel utilization at a gigabit data rate.

5.2 The RCMA Protocol

5.2.1 Network Topology

Our RCMA protocol is proposed to operate in a tree topology with a passive optical repeater similar to the 10BASE-FP Ethernet [IEEE00]. The data rate is expected to be 1Gb/s. The main advantage of this configuration is its low cost characteristics due to the use of passive optical repeaters.

The main difference between the 10BASE-FP Ethernet and other Ethernet variations is the signal repeating mechanism. In 10BASE-FP, two optical fibers are connected to the passive optical repeater from each station, one for incoming, and another for outgoing traffic. When optical signals arrive at one port of the passive repeater, the signals will be repeated to all stations, including the originated station. Since the sender will receive its own transmission during its data frame transmission, collision detection is somehow difficult for the CSMA/CD protocol. Therefore, in 10BASE-FP Ethernet, a special transceiver is designed to allow a sender to detect collisions in the presence of its return signals.

However, in RCMA, a station is required to request for the channel access right before its actual data frame transmission can take place. Once the

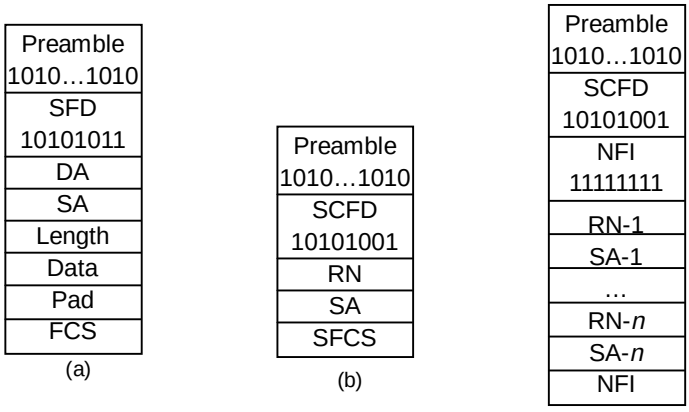
channel is reserved, the data frame is transmitted free of collision, thus collision detection operation is not required.

5.2.2 Protocol Description

The idea of our proposed protocol is that it makes use of the return signals repeated by a passive optical repeater to allow each sender to verify that its earlier transmission was successful. To make the operation more efficient, we introduce the use of a very small request frame for each sender contending for the channel access right to reserve the channel for longer data frame transmissions.

To simplify the explanation of the RCMA protocol, we assume a slotted broadcast channel. Nevertheless, the RCMA protocol is designed to operate in a non-slotted channel.

Let τ be the signal propagation delay between the two farthest stations in the network. When a station is ready for a data transmission, henceforth called a *ready station*, it is required to perform a *request contention* operation. It first



Names of some important fields:
SA: Source Address (6 bytes)
RN: Request Number (1 byte, only 6 bits are used)
NFI: NEXT Frame Indicator (1 byte)
SFD: Start Frame Delimiter (1 byte)
FCS: Frame Check Sequence (4 bytes)
SFCS: Short Frame Check Sequence (1 byte)
SCFD: Start Control Frame Delimiter (1 byte)

Figure 5.3: Frame formats: (a) the IEEE 802.3 data frame; (b) the RCMA request frame; (c) the RCMA NEXT frame

prepares a request frame. The proposed request frame format is depicted in Figure 5.3(b). The station must randomly generate a 6-bit *request number* and store it in the *Request Number* (RN) field of the request frame. Its MAC address is also included. The request frame ends with an 8-bit *short frame check sequence* (SFCS) for error detection.

Before the request frame transmission, the station activates a timer called the *request-waiting timer* (RWTimer). RWTimer = $w \cdot T_s$, where w is a uniformly distributed random integer between zero and k-1, and T_s is the minislot time which is the time required to transmit the entire request frame plus a short guard time. The protocol parameter k can be viewed as the maximum admission delay of a request transmission. A small k value will result in a more aggressive request transmission.

The station waits and monitors the incoming channel after activated RWTimer. Detection of a request or a data frame from the incoming channel generated by another station during that period will cause the station to abort its request frame transmission. This will ensure that the station which has requested the channel earlier has the priority to transmit based on first come first served principle.

When RWTimer expires, if the incoming channel remains idle, the station may transmit its request frame. The station is required to monitor the incoming channel during its request frame transmission. If a carrier is detected on the incoming channel, the request frame transmission must also be aborted immediately. We assume that the cable between a station and the repeater is long enough (in the case of 1Gb/s and a 16-byte request frame, the cable must be at least 19.2m) so that the request frame will not return back to the originated station while transmitting that request frame. However, the request frame transmissions are subject to collisions. If two stations transmit the request almost at the same time such that the request frames meet at the passive repeater, these request frames are corrupted because they overlap each other. Otherwise, the request frames are considered successfully transmitted and can be read correctly by all stations.

After the very last bit of the request frame is transmitted, the station activates another timer called the *request-collection timer* (RCTimer). This timer is set to 2τ plus a short guard time. During this time interval, the station monitors the incoming channel and collects any request frame including its own request frame transmitted earlier. Any incorrect or incomplete request frames are discarded. When RCTimer expires, the station can be sure that all transmitted request frames have arrived and no further request frames are still propagating in the network. The station then compares all the collected requests. One of these requests will be the *winning request*. The winning request is the one with the largest request number among all collected requests. The station that originally sent the winning request will gain the channel access right. This station will henceforth be called the *winner*.

All stations contending for the channel access right, including the winner can identify the winner by comparing the request numbers, and identifying the MAC address of the largest request number. No collision detection is required during the data frame transmission. The choice of data frame format is optional, here we propose to use the IEEE 802.3 data frame [IEEE00] shown in Figure 5.3(a).

It is possible that two or more stations may choose the same request number. There are many ways to break this tie. Under one simple option, the station of a larger numerical MAC address always has the advantage to transmit its data frame first. This will not affect the fairness significantly because this event is very rare.

Since the station, which has given the exclusive right to access the channel, is also aware of other requests while competing for the channel access right, after its data frame transmission and an IFG period similar to its shared Gigabit Ethernet counterpart, it may transmit a special control frame, called the NEXT frame. The proposed frame format for the NEXT frame is given in Figure 5.3(c). The NEXT frame generally contains a list of successful requests that the station collected earlier, sorted by request number.

When the NEXT frame is transmitted, RCMA enters a *non-contention channel assignment* operation. Each station, after receiving the NEXT frame, compares its MAC address with the first MAC address in the NEXT frame. If matched, that station may transmit its data frame after an IFG period. Again, the station removes its record from the NEXT frame and transmits the modified NEXT frame after it has completed its data frame transmission.

When the last station on the list in the NEXT frame completes its data frame transmission, no NEXT frame will follow. After an IFG period, all ready stations enter the request contention operation.

If all request frames collide, no data frame transmission will occur. After discovering that there is no data frame transmission, all ready stations immediately repeat the request contention operation to compete for the channel access right.

A critical aspect of RCMA is its implicit channel assignment property. If the assigned station fails to initiate its transmit, a deadlock situation may occur. To avoid this problem, when the channel is assumed to be assigned to a winner, each station activates a timer called the *idle-channel timer* (ICTimer). The duration of ICTimer must be greater than the duration of RCTimer. ICTimer is reset if incoming channel is sensed busy. However, if the incoming channel remains idle after ICTimer expires, it is assumed that the winner forfeits its transmission right. Each ready station then repeats the request contention operation to compete for the channel access right.

Finally, before a newly started station can join the network, it must wait and monitor the channel for a time period longer than the duration of ICTimer plus T_s. This will ensure that the station does not disrupt any ongoing event.

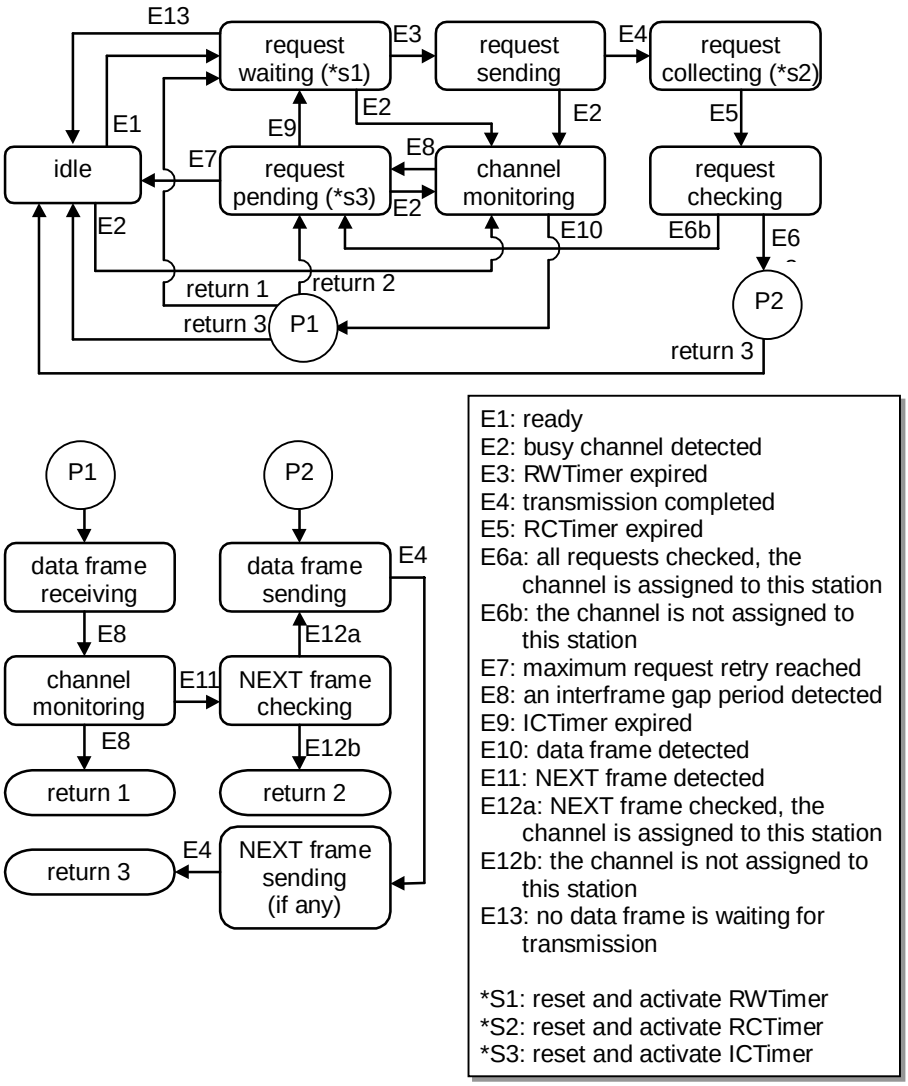


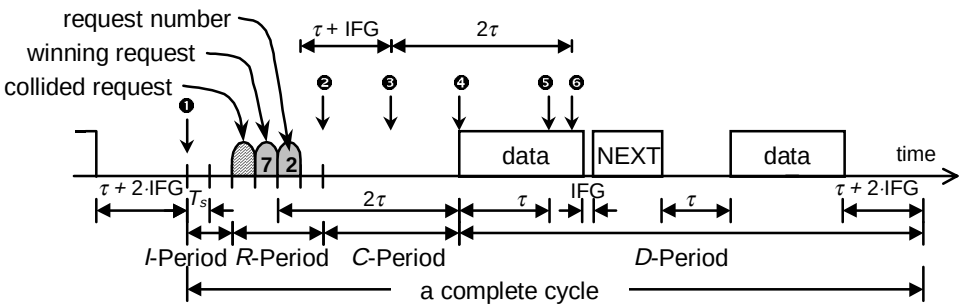
Figure 5.4: The finite state machine for an RCMA transceiver

In Figure 5.4, we construct a finite state machine to describe the detail operation of an RCMA transceiver.

5.2.3 Saturation Throughput Analysis

Consider a network of *m* stations. Each station is saturated so that it always has data to transmit. In other words, the event E1 shown in Figure 5.4 occurs as soon as the station enters “idle” state.

We assume that the distance between any two stations is the same. We consider a data frame size distribution that we used in subsection 5.1.2,



❶ all stations detects the end of a data frame transmission;
❷ all stations detects the first request frame transmission, the stations that have not transmitted their request frames must abort their request frame transmissions;
❸ the channel turns from busy to idle due to the request frame with number '2'. After detecting the channel to be idle for an IFG period, all stations, that do not participate in request contention, reset and activate their ICTimers which will expire at ❻;
❹ the RCTimer of the winner expires, it starts its data frame transmission immediately;
❺ all stations sense the busy incoming channel and will not access the channel at ❻;

Figure 5.5: A snapshot of the RCMA channel

specifically, 35% of the data frames carry 46 bytes of useful information and 65% of the rest carry 1500 bytes of useful information. The data frame sizes are corresponding to the minimum and the maximum sizes of IEEE 802.3 frames [IEEE00].

We first notice that the RCMA broadcast channel is repeating a statistically identical cycle shown in Figure 5.5. Each cycle consists of the following periods: (i) an *I*-period; (ii) an *R*-period; (iii) a *C*-period; and (iv) a *D*-period; representing an idle, request transmission, request collection and data frame transmission periods respectively.

The *I*-period starts as soon as the previous *D*-period ends. According to Figure 5.4, in the *I*-period, all ready stations, including the station just completed a data frame transmission, enter the "request waiting" state. At this state, each station may transmit its request if its RWTimer expires. As soon as the first transmission of the request frame appears on the channel, the *I*-period ends and the *R*-period begins.

During the *R*-period, each station is not aware of any request frames transmitted by other stations, hence whenever its RWTimer expires, that

station transmits its request frame. When the first bit of the first request frame reaches all stations, the *R*-period ends, and the *C*-period begins.

When the *C*-period begins, no further request frame can be transmitted. During this period each station collects the requests to determine the winner. The winner initiates its data frame transmission when its RCTimer expires. The *C*-period ends when the winner starts its data frame transmission.

Due to the non-contention channel assignment operation, during the *D*-period, several data frame transmissions may occur. The *D*-period ends when there is no NEXT frame transmission after a data frame transmission.

Let *B* be the data rate of the network. Given *m* saturated stations, let the random variables *I, R, C, D* be the duration of the *I*-period, the *R*-period, the *C*-period and the *D*-period respectively. Let the random variable *U* be the duration of the actual payload transmission, excluding all IEEE 802.3 data frame overheads during a cycle. Let *H* be the duration of the overhead transmission such that *D=H+U*. Then the RCMA saturation throughput for *m* saturated stations, S_{RCMA}, can be expressed by

$$S_{RCMA} = \frac{E[U]}{E[I+R+C+H+U]} \,. \tag{5-5}$$

As described earlier, RWTimer = $w \cdot T_s$, where *w* is a uniformly distributed random integer between zero and *k*-1, and T_s is the minislot time duration. The *I*-period ends when at least one request frame transmission appears. Hence the probability that the *I*-period lasts for *x* minislots is the probability that any of the *m* stations choose to transmit their request frames given that no request frame transmission appears in previous minislots. The probability density function (pdf) of *I* is thus

$$P\{I = x \cdot T_s\} = \begin{cases} q_x & , x = 0 \\ q_x\left(1 - \sum_{i=0}^{x-1} P\{I = i \cdot T_s\}\right), & x = 1,2,\ldots,k-1 \\ 0 & \text{otherwise} \end{cases} \tag{5-6}$$

where $q_x = 1 - \left(1 - \frac{1}{k-x}\right)^m$ is the probability that any of the m stations choose to transmit its request frame after x idle minislots.

For the *R*-period, since the distance between any two stations is fixed, the duration for a signal to propagate from any station to all stations is constant. Thus

$$R = \tau. \tag{5-7}$$

When the *R*-period ends, no further request frame can be transmitted. Since the *R*-period is a constant, then the number of minislots, *r*, within the *R*-period is also a constant, and it can be obtain by

$$r = \lfloor \tau / T_s \rfloor \tag{5-8}$$

where $\lfloor x \rfloor$ is the floor of x, defined as the largest integer smaller than x.

The duration of the *C*-period depends on the position of the winner within the r minislots. The value of the random variable C is between $\tau+T_s$ and 2τ. Since in LANs, the signal propagation time, τ, is generally small compared to the data frame transmission time (for example in our case, the transmission for a long data frame is about six times larger than τ), hence this random variable has only a small effect on the saturation throughput of RCMA. Therefore, we here consider the worst case where the winner always appears at the last position during the *R*-period, that is

$$C = 2\tau\,. \tag{5-9}$$

Given m saturated stations, r minislots and the k parameter, the pdf of the number of request frames, N, successfully detected by all stations during the *R*-period can be derived recursively to be

$$P\{N = x\} = \frac{N_b(x, r, k, m)}{N_a(k, m)}, x = 0,1,\ldots,r \tag{5-10}$$

where

$$N_b(x,r,k,m)=\binom{m}{0}N_b(x,r,k-1,m)+\binom{m}{1}N_c(x-1,r-1,k-1,m-1)+\sum_{n=2}^{m}\binom{m}{n}N_c(x,r-1,k-1,m-n),$$

$$N_c(x,r,k,m)=\binom{m}{0}N_c(x,r-1,k-1,m)+\binom{m}{1}N_c(x-1,r-1,k-1,m-1)+\sum_{n=2}^{m}\binom{m}{n}N_c(x,r-1,k-1,m-n),$$

$$N_a(k,m)=k^m,$$

with $\binom{m}{n}=\frac{m!}{n!(m-n!)}$ and the following initial conditions,

$$N_b(x=-1,r,k,m)=0;$$
$$N_b(x=0,r,k\neq 1,m=0)=1; N_b(x=0,r,k=1,m=1)=0;$$
$$N_b(x=0,r,k=1,m\neq 1)=1; N_b(x=1,r,k\neq 1,m=0)=0;$$
$$N_b(x=1,r,k=1,m=1)=1; N_b(x=1,r,k=1,m\neq 1)=0;$$
$$N_b(x\geq 2,r,k\neq 1,m=0)=0; N_b(x\geq 2,r,k=1,m)=0;$$

and

$$N_c(x=-1,r,k,m)=0;$$
$$N_c(x=0,r=0,k,m)=N_a(k,m); N_c(x=0,r,k\neq 1,m=0)=1;$$
$$N_c(x=0,r,k=1,m=1)=0; N_c(x=0,r,k=1,m\neq 1)=1;$$
$$N_c(x=1,r=0,k,m)=0; N_c(x=1,r,k\neq 1,m=0)=0;$$
$$N_c(x=1,r,k=1,m=1)=1; N_c(x=1,r,k=1,m\neq 1)=0;$$
$$N_c(x\geq 2,r=0,k,m)=0; N_c(x\geq 2,r,k\neq 1,m=0)=0;$$
$$N_c(x\geq 2,r,k=1,m)=0.$$

$N_c(x,r,k,m)$ is the total number of possible permutations, that x out of r minislots will carry successful requests, given m saturated stations and k value. $N_b(x,r,k,m)$ is similar to $N_c(x,r,k,m)$ but $N_b(x,r,k,m)$ is the number of possible permutations under the assumption that no idle slot appears in any

of the previous minislots. $N_a(k,m)$ is the total number of possible permutations given m and k.

To understand how Equation (5-10) is formulated, we here describe the computation of $N_c(x,r,k,m)$. $N_c(x,r,k,m)$ is the number of possible permutations that given m saturated stations and k value, x out of r minislots will carry successful requests. $N_c(x,r,k,m)$ is computed by conditioning on the first minislot of the r minislots. If the first minislot is idle, then the number of possible permutations for the rest of the r-1 minislots is $N_c(x,r\text{-}1,k\text{-}1,m)$; If the first minislot carries one request, then the number of possible permutations is $N_c(x\text{-}1,r\text{-}1,k\text{-}1,m\text{-}1)$ times the choices of selecting one out of m stations. The parameter m-1 indicates that one station has performed a request, and x-1 indicates that one successful request has been obtained; If there is a collision of n requests in the first minislot, then the number of possible permutations, N_c, for the rest of the r-1 minislots is $N_c(x,r\text{-}1,k\text{-}1,m\text{-}n)$ times the choices of selecting n out of m stations. The same technique is used for computing $N_b(x,r,k,m)$.

The duration of the D-period depends on the number of successful requests that appear in the R-period given in Equation (5-10). If there is no successful request, all ready stations will enter the "request pending" state in Figure 5.4 due to the events E6b or E8. Not all stations discover the failure of channel assignment at the same time, but the difference between the time each station enters the "request pending" state is not significant. Therefore we assume all stations return to the "request pending" state at the same time after the C-period ends. In the case where there is no winner, if ICTimer lasts for 2τ, then it will take 2τ units of time before this cycle ends. With this development, the relationship between the number of successful requests, N, obtained in Equation (5-10) and the duration of the D-period is

$$D = \begin{cases} 2\tau & , N = 0 \\ T_{FRAME} + \tau + 2 \cdot T_{IFG} & , N = 1 \\ \sum_{i=1}^{N-1} \left(T_{FRAME} + T_{IFG} + T_{NEXT}(i) + \tau \right) & \\ \quad + T_{FRAME} + \tau + 2 \cdot T_{IFG} & , N = 2,3,\ldots,r \end{cases} \tag{5-11}$$

with $T_{NEXT}(i) = 8 \cdot (10 + 7i)/B$. T_{FRAME} is the mean transmission time of the IEEE 802.3 data frame, $T_{NEXT}(i)$ is the transmission time of a NEXT frame that carries i requests, and T_{IFG} is the IFG duration. Knowing the distribution of a data frame, the mean value for D can be computed.

The time duration of useful information transmitted during a cycle, U, also depends on N in Equation (5-10). It can be expressed as

$$U = N \cdot T_{PAYLOAD}, \quad N = 0,1,...,r \tag{5-12}$$

where $T_{PAYLOAD}$ is the mean transmission time of the useful information based on a particular data frame distribution. Having obtained the pdf of I, R, C, D, and U, their mean values can be computed, as well as the saturation throughput of RCMA given in Equation (5-5).

Table 5.3 lists the protocol parameters used in the numerical computation. The RCMA saturation throughput curve is plotted in Figure 5.6. We first notice that the numerical results of the RCMA saturation throughput (shown

Table 5.3: Summary of the protocol parameters of RCMA used for numerical computation and computer simulation

Parameter	Value
Protocol used	RCMA
Data rate	1Gb/s
Network topology	passive star network
Signal propagation time in the network, τ	2.048μsec
The IFG time duration, T_{IFG}	0.096μsec
The minislot time duration, T_s	0.128μsec (16 bytes)
The minislot number, r	16
The parameter k	20
Number of stations, m	1,2,…,50
Data frame distribution	Dual size data frames with 35% short and 65% long data frames
The useful transmission time for a short IEEE 802.3 data frame	0.368μsec (46 bytes)
The useful transmission time for a long IEEE 802.3 data frame	12μsec (1500 bytes)
The IEEE 802.3 data frame overhead including preamble and SFD	0.208μsec (26 bytes)

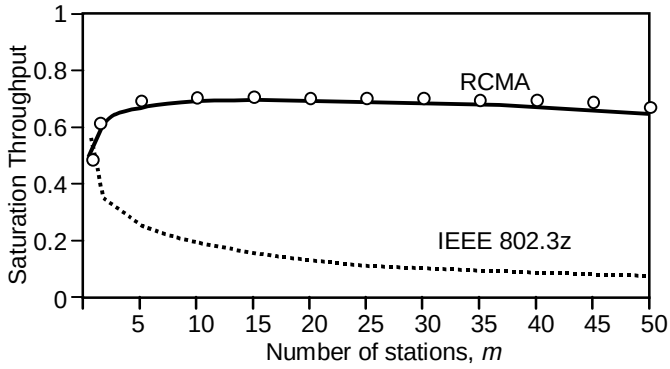


Figure 5.6: Saturation throughput for the RCMA and the IEEE 802.3z MAC protocols

in solid lines) is slightly lower than but generally similar to the simulation results (shown in symbols). This is mainly because we consider the worst case of the *C*-period in our analysis.

Figure 5.6 also compares the saturation throughput curves between the RCMA and the IEEE 802.3z MAC protocols. As can be observed, the saturation throughput of the IEEE 802.3z MAC protocol drops quickly when the number of saturated stations increased from one to five, and its throughput continues to drop as the number of saturated stations increases. The throughput even drops below 10% when there are over 32 saturated stations sharing the 1Gb/s bandwidth. In other words, each saturated station only receives around 3.125Mb/s bandwidth on average under this condition.

On the other hand, the performance of RCMA is stable. It offers over 65% efficiency for up to 50 saturated stations, except when the number of saturated stations is below three. This is because when the number of saturated stations is low, the channel assignment overhead for each data frame transmission is slightly higher due to the need for requests prior to a data frame transmission. However, as the number of saturated stations increases, the non-contention channel assignment operation of RCMA becomes effective, more data frame transmissions can be assigned during a request contention period, thus the channel assignment overhead for each data frame transmission becomes relatively small. In the case of 32 saturated stations, RCMA achieves just below 70% throughput, which is equivalent to

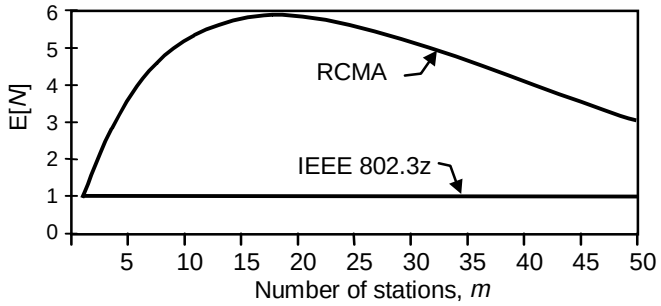


Figure 5.7: The mean number of transmitted data frames within a cycle versus the number of saturated stations for RCMA and IEEE 802.3z

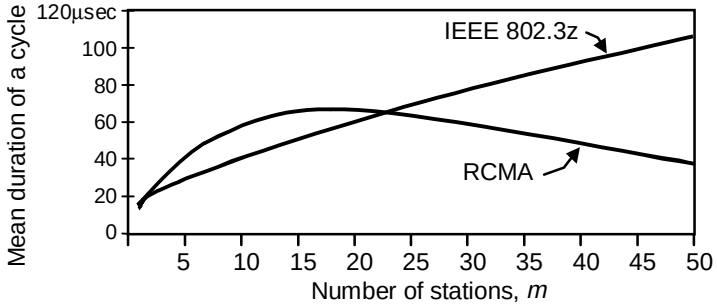


Figure 5.8: The mean duration of a cycle versus the number of saturated stations for RCMA and IEEE 802.3z

21.875Mb/s bandwidth for each station on average, seven times higher than that in the IEEE 802.3z MAC protocol.

The superior performance of RCMA over IEEE 802.3z, as demonstrated in Figure 5.6, is a result of two factors: (i) the number of data frames transmitted successfully within a cycle, and (ii) the mean duration of a cycle. In Figure 5.7, we plot the mean number of data frames transmitted after a contention period during a cycle as a function of the number of saturated stations. In Figure 5.8, we compare the mean duration of a cycle versus the number of saturated stations. While the IEEE 802.3z MAC protocol can only accommodate one data frame transmission in a cycle after a contention period, RCMA allows more than one in average because of the non-contention channel assignment feature.

Comparing the mean duration of a cycle of the two protocols, the mean duration of a cycle of RCMA is mainly dominated by the data frame

transmission period which is useful. This indicates efficiency. Whereas in the case of the IEEE 802.3z MAC protocol, the mean duration of a cycle is dominated by the contention period. Therefore when the number of saturated stations is as many as 50, the mean duration of a cycle of RCMA is not only shorter than that in the IEEE 802.3z MAC protocol, but it also contains a longer period of time that is used to transmit useful information within a cycle.

The only case where the IEEE 802.3z protocol performs better than RCMA is when there is only exactly one saturated station in the network. In this case, no collisions will occur in the IEEE 802.3z MAC protocol and hence it is efficient. As the number of saturated stations increases, collisions become more likely and its throughput drops significantly.

5.3 Extensions, Implementations and Limitations

5.3.1 Priority Scheme

Some classical solutions for the CSMA/CD protocol to implement a certain priority scheme can be found in [FrBo80, Toba82b, IiYK83, HuRa85]. Some degree of performance degradation often results when a priority scheme is implemented in the CSMA/CD protocol. This is mainly due to additional overheads involved in implementing the proposed priority scheme.

In RCMA, however, since all stations are required to transmit a request that contains a random chosen request number, the request number can be used to order a data frame transmission. By using different functions to generate random request numbers for different traffic types, a traffic type that requires higher attention may have a higher priority to be transmitted.

Here we propose to divide all possible request numbers into two groups: (i) the high priority group; and (ii) the low priority group. Recall that in Figure 5.3, six bits are available for the labeling request numbers. We recommend

the range of the request numbers between 0 and 47 to be used for the low priority group, and the range of the request numbers between 48 and 63 to be used for the high priority group. This way, it is guaranteed that higher priority services will gain access before lower priority services if their request frames do not suffer transmission collisions.

It is also possible to enhance this priority scheme so that the request numbers used for the two groups can be adaptively adjusted based on the load conditions and the statistics of traffic types. The design of an advanced priority scheme is out of the focus of this dissertation and it will not be studied here.

5.3.2 RCMA over Copper

It is possible to replace the passive optical repeater with an active electrical repeater so that RCMA can be operated over copper wires. The advantages of RCMA over copper are many. One obvious advantage of an electrical version of RCMA is that the deployment of RCMA to an existing network wired with twisted pairs will be straightforward.

The electrical version of RCMA must feature a full duplex link or two links with one for incoming and one for outgoing traffic between the repeater and each station. The electrical repeater must be operated in the same way as the passive optical repeater where signals received from a port are forwarded to

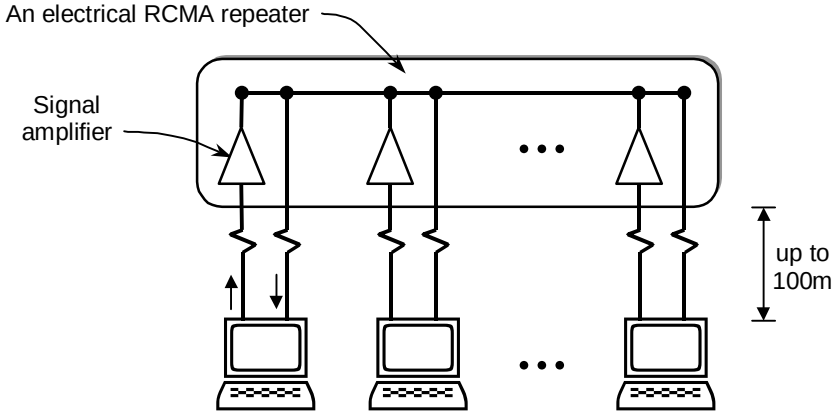


Figure 5.9: A simple electrical RCMA repeater

all other ports including the incoming port. Figure 5.9 shows a simple electrical repeater for RCMA. The repeater features an analogue signal amplifier in each input port, and the output ports share the same broadcast channel. This repeater assumes good quality of cables between the repeater and all other stations. To enhance the robustness of the network, the RCMA repeater may be equipped with digital signal processing devices to regenerate the encoded signals before forwarding to all output ports.

Further performance improvement can be achieved by adding intelligence and buffers into an active electrical RCMA repeater. With added intelligence and buffers, it is possible to turn an electrical RCMA into a *buffer distributor* [KaCK98]. Limited buffers are equipped in each input port in a buffer distributor. The buffer distributor collects and orders the read requests within itself. After then, the buffer distributor broadcasts the processed requests to all stations. Because a buffer distributor collects requests into an individual buffer, no request transmissions will collide even if they arrive at the buffer distributor at the same time. As a result, better performance can be expected. However, this solution is more expensive than a simple electrical repeater.

5.3.3 Network Coverage

Throughout this chapter, we assume that RCMA is operated in a LAN that spans 200m, similar to a shared Gigabit Ethernet LAN. However, unlike the IEEE 802.3z MAC protocol that is limited to a certain cable length, the RCMA protocol may also be operated in a larger LAN with longer cables.

It is known that if network coverage is increased, the signal propagation time between a pair of stations in the network is also increased. According to Equation (5-7), the increase in the signal propagation time will lengthen the *R*-period. This indicates that the overhead of a data frame transmission will be higher. On the other hand, the number of minislots during the *R*-period provided by the Equation (5-8) is higher if the *R*-period is lengthened. A higher number of minislots provides more choices of

Table 5.4: Effect of signal propagation time on the saturation throughput of RCMA

Signal Propagation Time, τ (Estimated Network Span)	Number of Saturated Stations			
	10	30	50	100
2.048μsec (200m)	0.685	0.674	0.597	0.118
4.096μsec (400m)	0.557	0.578	0.574	0.517

minislots for all stations to perform their requests. With more choices, the request collision probability will be reduced. Hence more stations are allowed to be included in a single LAN.

Table 5.4 briefly compares the saturation throughput for RCMA between two different network spans. Due to the low overhead for each data frame transmission in a LAN with shorter cables, its throughput is higher when the number of saturated stations is small. However, as the number of saturated stations increases, the request collision probability increases. This causes a drop in the saturation throughput. On the other hand, because longer cables lengthen the *R*-period which leads to the increase in the number of minislots. Even the number of saturated stations increases significantly, the request collision probability does not increase significantly, hence the saturation throughput only drops slightly as demonstrated in Table 5.4.

Like the IEEE 802.3z MAC protocol, RCMA is also sensitive to the signal propagation time, τ. However, an increase in τ does not always lead to a drop in the saturation throughput. An optimum operation point of RCMA depends on a good combination of the number of active stations in a network and τ. In this chapter, we only focus on the performance of RCMA for a LAN similar to a Gigabit Ethernet LAN that uses short cables and contains a small number of stations. Nevertheless, the application of RCMA is not limited to a LAN. It may be operated in a network with longer cables and a larger number of stations such as a Passive Optical Network in the Residential Area Networks [Lung99, PeKe99]. However, the possibility of this application is not explored here.

5.4 Summary

In this chapter, we have proposed a new MAC protocol for Gigabit LANs called RCMA. RCMA is expected to operate in a gigabit passive optical LAN. In RCMA, a sender uses the return signals repeated by a passive optical repeater as an acknowledgement to determine if its earlier transmission is successful. To make the protocol more efficient, the sender is required to contend for a channel access right for its long data frame transmission using a very short request frame. This leads to a small bandwidth loss due to collisions of the request frames which is significantly small compared with the loss of bandwidth due to collisions of data frame transmissions in the IEEE 802.3z MAC protocol.

To further improve the performance of RCMA, the non-contention channel assignment operation was introduced. Under the non-contention channel assignment operation, the channel assignment information is explicitly passed by a sender to others in order to minimize the overhead of the channel assignment task.

To demonstrate the performance advantage of RCMA over the current IEEE 802.3z MAC protocol, we first studied the saturation throughput of the IEEE 802.3z MAC protocol in Section 5.1. We then analyzed the throughput of RCMA under the saturation scenario in Section 5.2. The comparison of the saturation throughput of the two protocols presented in Figure 5.6 clearly shows a significant benefit of RCMA. The RCMA protocol offers relatively stable and efficient performance whereas the performance of the IEEE 802.3z MAC protocol drops below 20% in the case of merely 10 saturated stations.

A few issues related to the further enhancements and implementations of RCMA were discussed in Section 5.3. They include the design of a priority scheme for RCMA, the possibility of an electrical version for RCMA, as well as the impact of network spans on the performance of RCMA.

Since the performance of RCMA remains stable and efficient even if the number of saturated stations is as many as 50, moreover, it can easy accommodate a priority scheme, we believe that the use of our proposed RCMA protocol in gigabit LANs for the network access from a group of shared end users is not only cost efficient, but also far more reliable than the currently available standards and solutions. The implementation of RCMA may lead to a cost competitive yet efficient solution for the future gigabit LANs. We believe that the use of RCMA will reduce the need for switches in the future deployment of gigabit LANs.

6 Conclusions

This thesis has presented techniques for performance analysis of MAC protocols. In addition, two MAC protocols are proposed to enhance the performance of the current standard MAC protocols for LANs.

We first studied the performance analysis of MAC protocols conducted in the past. We recognized that the analytical models used for performance analysis in most of the previous works are usually simplified to maintain the mathematical tractability, and hence many of the results from the analysis are not applicable to the "real-world".

In an attempt to improve the analytical models, we have demonstrated the use of two realistic scenarios that are likely to occur in a LAN, namely the saturation and the disaster scenarios, for the performance analysis of MAC protocols. The performance results indicate the worst case performance of a protocol given a network configuration. Based on our study of the IEEE 802.3 MAC protocol under the two scenarios, it is found that the protocol performance is unattractive in a network containing a large population of active stations. As for the IEEE 802.11 MAC protocol, it performs reasonably well under the two realistic scenarios with an appropriate choice of protocol parameters.

Moreover, we have proposed a Markovian Framework for the performance analysis of a MAC protocols. The benefits of our proposed framework are many. Firstly, our proposed framework provides a unified model for the analysis of MAC protocols. Secondly, it significantly simplifies the analytical model of a MAC protocol, making the inclusion of a more complex and realistic traffic model possible without compromising the protocol details. Thirdly, vast knowledge is available on the analysis of a continuous time Markov chain, allowing for more insight to the

performance of a MAC protocol. Many examples are given in this thesis to demonstrate the versatility and the accuracy of this technique.

Apart from the performance analysis of MAC protocols, in this thesis, we have proposed two new protocols. We first introduced a novel technique, Reservations by Interruptions, to provide an efficient reservation scheme for CSMA/CD. The resulting protocol is named CSMA with Reservations by Interruptions (CSMA/RI). Extensive studies provided in this thesis, including the performance under the two realistic scenarios and various statistical traffic conditions, the stability of the protocol, as well as the performance comparison of CSMA/RI with the popular MAC protocols, have confirmed the benefit of CSMA/RI. The implementation of the protocol is also addressed.

It is found that the proposed reservations by interruptions scheme has the potential to improve a MAC protocol that is operated in a network of which the data frame transmission time is generally longer than the end-to-end signal propagation time such as 10Mb/s and 100Mb/s LANs, or a wireless LAN. The proposed reservations by interruptions scheme has also inspired a further research by Sheu et al. [ShWW01] whereby a new protocol named CSMA with Adaptive Reservations by Interruptions (CSMA/ARI) was introduced basing on the reservations by interruptions concept.

The second protocol introduced in this thesis is called the Request Contention Multiple Access (RCMA) protocol. RCMA is designed to operate in a passive optical gigabit LAN. The RCMA protocol inherits many attributes of Ethernet that make Ethernet a popular solution in today's LANs. RCMA is simple to implement, and it does not rely on expensive technologies such as WDM or an intelligent central controller. It can also easily accommodate a priority scheme without additional overhead, making service differentiation possible in the MAC layer. According to our performance studies carried out in this thesis, we found that the performance of RCMA under the saturation scenario is significantly better than the existing solution that uses the IEEE 802.3z MAC protocol. In addition, the

performance of RCMA remains stable over a wide range of the number of saturated stations, including the case when the number of saturated stations is large. This result indicates that a RCMA LAN can be used for a large number of end users, leading to a cost competitive yet efficient solution for future gigabit LANs.

References

[Abra70] N. Abramson, "The Aloha System – Another Alternative for Computer Communications", in *Proc. Fall Joint Comput. Conf. AFIPS Conf.*, pp.37, 1970.

[AdNZ99] R. G. Addie, T. D. Neame, and M. Zukerman, "Modelling Superposition of Many Sources Generating Self Similar Traffic," in *Proc. IEEE ICC '99*, June 1999.

[Aldo87] D. J. Aldous, "Ultimate Instability of Exponential Back-Off Protocol for Acknowledgement-Based Transmission Control of Random Access Communication Channels," *IEEE Transactions on Information Theory*, vol. IT-33, no. 2, pp. 219-223, March 1987.

[Ashr94] H. R. van As, "Media Access Techniques: The Evolution Towards Terabit/s LANs and MANs," *Computer Networks and ISDN Systems*, vol. 26, no. 6-8, pp. 603-656, March 1994.

[BDSZ94] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A Media Access Protocol for Wireless LANs," in *Proc. ACM SIGCOMM '94*, pp. 212-225, 1994.

[BeGa92] D. Bertsekas and R. Gallager, "Data Networks (2nd Edition)," *Prentice Hall*, 1992.

[Bian00] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535-547, March 2000.

[Bian98] G. Bianchi, "IEEE 802.11–Saturation Throughput Analysis," *IEEE Communications Letters*, vol. 2, no. 12, pp. 318-320, December 1998.

[Bisd96] C. Bisdikian, "A Review of Random Access Algorithm," *IEEE 802.14 Working Group Document No. IEEE 802.14-96/019*, January 1996.

[BoMK88] D. R. Boggs, J. C. Mogul, and C. A. Kent, "Measured Capacity of an Ethernet: Myths and Reality," in *Proc. ACM SIGCOMM '88*, pp. 222-234, August 1988.

[Bux81] W. Bux, "Local-Area Subnetworks: A Performance Comparison," *IEEE Transactions on Communications*, vol. COM-29, no. 10, pp. 1465-1473, October 1981.

[CaCG00] F. Cali, M. Conti, and E. Gregori, "IEEE 802.11 Protocol: Design and Performance Evaluation of an Adaptive Backoff Mechanism," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 9, pp. 1774-1786, September 2000.

[CaCG98] F. Cali, M. Conti, and E. Gregori, "IEEE 802.11 Wireless LAN: Capacity Analysis and Protocol Enhancement," in *Proc. IEEE Infocom '98*, 1998.

[CaHe75] A. B. Carleial and M. E. Hellman, "Bistable Behavior of ALOHA-Type Systems," *IEEE Transactions on Communications*, vol. COM-23, no. 4, pp. 401-410, April 1975.

[Cape79] J. I. Capetanakis, "Tree Algorithms for Packet Broadcast Channels," *IEEE Transactions on Information Theory*, vol. IT-25, no. 5, pp. 505-515, September 1979.

[CMRW97] G. L. Choudhury, A. Mandelbaum, M. I. Reiman, and W. Whitt, "Fluid and Diffusion Limits for Queues in Slowly Changing Random Environments," *Stochastic Models*, vol. 13, no. 1, pp. 121-146, 1997.

[FLBG74] G. Fayolle, J. Labetoulle, D. Bastin, and E. Gelenbe, "The Stability Problem of Broadcast Packet Switching Computer Networks," *Acta Informatica*, vol. 4, pp. 49-53, 1974.

[FoZu00a] C. H. Foh and M. Zukerman, "Improving the Efficiency of CSMA using Reservations by Interruptions," in *Proc. IEEE ICC '00*, June 2000.

[FoZu00b] C. H. Foh and M. Zukerman, "CSMA with Reservations by Interruptions (CSMA/RI): A Novel Approach to Reduce Collisions in CSMA/CD," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 9, pp. 1572-1580, September 2000.

[FoZu01a] C. H. Foh and M. Zukerman, "Performance Evaluation of IEEE 802.11" in *Proc. IEEE VTC '01*, May 2001.

[FoZu01b] C. H. Foh and M. Zukerman, "Performance Comparison of CSMA/RI and CSMA/CD with BEB," in *Proc. IEEE ICC '01*, June 2001.

[FoZu02a] C. H. Foh and M. Zukerman, "Performance Analysis of the IEEE 802.11 MAC Protocol," in *Proc. European Wireless '02*, February 2002.

[FoZu02b] C. H. Foh and M. Zukerman, "A New Technique for Performance Evaluation of Random Access Protocols," in *Proc. IEEE ICC '02*, April 2002.

[FoZu02c] C. H. Foh and M. Zukerman, "A Novel and Simple MAC Protocol for High Speed Passive Optical LANs," in *Proc. Networking '02*, May 2002.

[FrBo80] W. R. Franta and M. Bilodeau, "Analysis of a Prioritized CSMA Protocol Based on Staggered Delays," *Acta Informatica*, vol. 13, no. 4, pp. 299-324, 1980.

[GePa82] L. Georgiadis and P. Papantoni-Kazakos, "A Collision Resolution Protocol for Random Access Channels with Energy Detectors," *IEEE Transactions on Communications*, vol. COM-30, no. 11, pp. 2413-2420, November 1982.

[GGMM88] J. Goodman, A. G. Greenberg, N. Madras, and P. March, "Stability of Binary Exponential Backoff," *Journal of the ACM*, vol. 35, no 3, pp. 579-602, 1988.

[GrHa98] D. Gross and C. M. Harris, "Fundamentals of Queueing Theory (3rd Edition)," *Wiley-Interscience*, 1998.

[HaKS87] I. M. I. Habbab, M. Kavehrad, and C.-E. W. Sundberg, "Protocols for Very High-Speed Optical Fiber Local Area Networks using a Passive Star Topology," *IEEE Journal on Lightwave Technology*, vol. LT-5, no. 12, pp. 1782-1794, December 1987.

[HaOr88] J. L. Hammond and P. J. P. O'Reilly, "Performance Analysis of Local Computer Networks," *Addison-Wesley*, 1988.

[Haye78] J. F. Hayes, "An Adaptive Technique for Local Distribution," *IEEE Transactions on Communications*, vol. COM-26, no. 8, pp. 1178-1186, August 1978.

[HeLu86] H. Heffes and D. M. Lucantoni, "A Markov Modulated Characterization of Packetized Voice and Data Traffic and Related Statistical Multiplexer Performance," *IEEE Journal on Selected Areas in Communications*, vol. SAC-4, no. 6, pp. 856-868, September 1986.

[HoCh96] T. S. Ho and K. C. Chen, "Performance Analysis of IEEE 802.11 CSMA/CA Medium Access Control Protocol," in *Proc. IEEE PIMRC '96*, pp. 407-411, 1996.

[HuRa85] G. L. Choudhury and S. S. Rappaport, "Priority Access Scheme Using CSMA-CD," *IEEE Transactions on Communications*, vol. COM-33, no. 7, pp. 620-626, July 1985.

[IEEE00] Institute of Electrical and Electronics Engineers, "IEEE Standard for Information Technology – Telecommunications and Information Exchange between Systems – Specific Requirements – Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications," *IEEE Std 802.3-2000 (ISO/IEC 8802-3:2000)*, IEEE, New York, 2000.

[IEEE98] Institute of Electrical and Electronics Engineers, "IEEE Standard for Information Technology – Telecommunications and Information Exchange between Systems – Specific Requirements – Part 5: Token Ring Access Method and Physical Layer Specifications," *IEEE Std 802.5-1998 (ISO/IEC 8802-5:1998)*, IEEE, New York, 1998.

[IEEE99] Institute of Electrical and Electronics Engineers, "IEEE Standard for Information Technology – Telecommunications and Information Exchange between Systems – Specific Requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-1999 (ISO/IEC 8802-11:1999)*, IEEE, New York, 1999.

[IiYK83] I. Iida, Y. Yasuda, and Y. Komachi, "Modified CSMA/CD Local Area network with Message-Based Priority Function," in *Proc. IEEE Infocom '83*, pp. 472-447, 1983.

[JeUn92] H. B. Jeon and C. K. Un, "Contention Based Reservation Protocols in Multiwavelength Optical Networks with a Passive Star Topology," in *Proc. IEEE ICC '92*, 1992.

[KaCK98] J. Kadambi, I. Crayford and M. Kalkunte, "Gigabit Ethernet: Migrating to High-Bandwidth LANs," *Prentice Hall*, 1998.

[Karn90] P. Karn, "MACA – A New Channel Access Method for Packet Radio," *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, pp. 134-140, 1990.

[KiKu83] W. M. Kiesel and P. J. Kuehn, "A New CSMA-CD Protocol for Local Area Networks with Dynamic Priorities and Low Collision Probability," *IEEE Journal on Selected Areas in Communications*, vol. 1, no. 5, pp. 869-876, November 1983.

[KlLa75] L. Kleinrock and S. S. Lam, "Packet Switching in a Multiaccess Broadcast Channel: Performance Evaluation," *IEEE Transactions on Communications*, vol. COM-23, no. 4, pp. 410-423, April 1975.

[KlTo75] L. Kleinrock and F. A. Tobagi, "Packet Switching in Radio Channels: Part I – Carrier Sense Multiple-Access Models and

Their Throughput Delay Characteristics," *IEEE Transactions on Communications*, vol. COM-23, no. 12, pp. 1400-1416, December 1975.

[LaKl75] S. S. Lam and L. Kleinrock, "Packet Switching in a Multiaccess Broadcast Channel: Dynamic Control Procedures," *IEEE Transactions on Communications*, vol. COM-23, no 9, pp. 891-904, September 1975.

[Lam74] S. S. Lam, "Packet Switching in a Multi-Access Broadcast Channel with Application to Satellite Communications in a Computer Network," Ph.D. dissertation, Computer Science Department, University of California, Los Angeles, rep. UCLA-ENG 7429, April 1974.

[Lam80] S. S. Lam, "A Carrier Sense Multiple Access Protocol for Local Networks," *Computer Networks*, vol. 4, pp. 21-32, 1980.

[Litt61] J. D. C. Little, "A Proof of the Queueing Formula, L=λW," *Operation Research*, vol. 9, pp. 383-387, March-April 1961.

[LTWW94] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the Self-Similar Nature of Ethernet Traffic (Extended Version)," *IEEE Transactions on Networking*, vol. 2, no. 1, pp. 1-15, February 1994.

[LuMN90] D. M. Lucantoni, K. S. Meier-Hellstern and M. F. Neuts, "A Single-Server Queue with Server Vacations and a Class of Non-Renewal Arrival Processes," *Advances in Applied Probability*, vol. 22, pp. 676-705, 1990.

[Lung99] B. Lung, "PON Architecture 'Futureproofs' FTTH," *Lightwave*, vol. 16, no. 10, pp. 104-107, September 1999.

[Mass81] J. L. Massey, "Collision Resolution Algorithms and Random Access Communications," *Multiuser Communications Systems*, G. Kingo Ed., Springer-Verlag, pp. 73-137, 1981.

[MeBo76] R. M. Metcalfe and D. R. Boggs, "Ethernet: Distributed Packet Switching for Local Computer Networks," *Communications of the ACM*, vol. 19, no. 7, pp. 395-403, July 1976.

[MoBa00] E. Modiano and R. Barry, "A Novel Medium Access Control Protocol for WDM-Based LAN's and Access Networks Using a Master/Slave Scheduler," *IEEE Journal of Lightwave Technology*, vol. 18, no. 4, April 2000.

[MoKl85] M. L. Molle and L. Kleinrock, "Virtual Time CSMA: Why Two Clocks Are Better than One," *IEEE Transactions on*

Communications, vol. COM-33, no. 9, pp. 919-933, September 1985.

[Moll94] M. L. Molle, "A New Binary Logarithmic Arbitration Method for Ethernet," *Technical Report CSRI-298*, The University of Toronto, April 1994.

[Mukh92] B. Mukherjee, "WDM-based Local Lightwave Networks - Part I: Single-hop Systems," *IEEE Network*, vol. 6, no. 3, pp. 12-27, May 1992.

[Neut81] M. F. Neuts, "Matrix-Geometric Solutions in Stochastic Models – An Algorithmic Approach," *Dover Publications Inc.*, 1981.

[PeKe99] G. Pesavento and M. Kelsey, "PONs for the Broadband Local Loop," *Lightwave*, vol. 16, no. 10, pp. 68-74, September 1999.

[Robe72] L. G. Roberts, "Aloha Packet Systems with and without Slots and Capture," *ASS Note 8*, Stanford Research Institute, Advanced Research Projects Agency, Network Information Center, 1972.

[RoSi90] R. Rom and M. Sidi, "Multiple Access Protocols: Performance and Analysis," *Springer-Verlag*, 1990.

[ShHu79] J. F. Shoch and J. A. Hupp, "Performance of an Ethernet Local Network: A Preliminary Report," in *Proc. of the Local Area Communication Network Symposium*, May 1979.

[ShWW01] S. T. Sheu, C. C. Wu, and P. L. Wu, "A New Contention-based CSMA Protocol for Star Networks," in *Proc. IEEE ICOIN '01*, January 2001.

[SoMV87] K. Sohraby, M. L. Molle, and A. N. Venetsanopoulos, "Comments on 'Throughput Analysis for Persistent CSMA Systems'," *IEEE Transactions on Communications*, vol. COM-35, no. 2, pp. 240-243, February 1987.

[SuKG94] G. N. M. Sudhakar, M. Kavehrad, and N. D. Georganas, "Access Protocols for Passive Optical Star Networks," *Computer Networks and ISDN Systems*, vol. 26, no. 6-8, pp. 913-930, March 1994.

[TaKl85] H. Takagi and L. Kleinrock, "Throughput Analysis for Persistent CSMA Systems," *IEEE Transactions on Communications*, vol. COM-33, no. 7, pp. 627-638, July 1985.

[TaKl87] H. Takagi and L. Kleinrock, "Correct to 'Throughput Analysis for Persistent CSMA Systems'," *IEEE Transactions on*

Communications, vol. COM-35, no. 2, pp. 243-245, February 1987.

[TaYS83] A. Takagi, S. Yamada, and S. Sugawara, "CSMA/CD with Deterministic Contention Resolution," *IEEE Journal on Selected Areas in Communications*, vol. 1, no. 5, pp. 877-884, November 1983.

[Toba82a] F. A. Tobagi, "Distributions of Packet Delay and Interdeparture Time in Slotted ALOHA and Carrier Sense Multiple Access," *Journal of the ACM*, vol. 29, no. 4, pp. 907-927, October 1982.

[Toba82b] F. A. Tobagi, "Carrier Sense Multiple Access with Message-Based Priority Functions," *IEEE Transactions on Communications*, vol. COM-30, no. 1, pp. 185-200, January 1982.

[ToHu80] F. A. Tobagi and V. B. Hunt, "Performance Analysis of Carrier Sense Multiple Access with Collision Detection," *Computer Networks*, vol. 4, pp. 245-259, 1980.

[ToKl77] F. A. Tobagi and L. Kleinrock, "Packet Switching in Radio Channels: Part IV – Stability Considerations and Dynamic Control in Carrier Sense Multiple-Access," *IEEE Transactions on Communications*, vol. COM-25, no. 10, pp. 1103-1119, October 1977.

[TsMi78] B. S. Tsybakov and V. A. Mikhailov, "Free Synchronous Packet Access in a Broadcast Channel with Feedback," *Prob. Information Transmission*, vol. 14, no. 4, pp. 256-280, October-December 1978.

[Tsyb80] B. S. Tsybakov, "Resolution of a Conflict of Known Multiplicity," *Prob. Information Transmission*, vol. 16, no. 2, pp. 134-144, April-June 1980.

[YuMa00] W. Yue and Y. Matsumoto, "An Exact Analysis for CSMA/CA Protocol in Integrated Voice/Data Wireless LANs," in *Proc. IEEE Globecom '00*, pp. 1805-1810, December 2000.

[ZiMo92] Z. Zilic and M. L. Molle, "Modeling the Exponential Backoff Algorithm in CSMA-CD Networks," *Technical Report CSRI-279*, The University of Toronto, October 1992.