การวิเคราะห์เวลาประวิงอัลกอริทึมการแก้ไขปัญหาการชนกันแบบต้นไม้แบบใหม่เมื่อมีการป้อนข้อมูลกลับจำนวนแพ็กเกตที่ชนกัน

นางสาวโรบิโต อันนูร์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2556

DELAY ANALYSIS OF A NEWLY PROPOSED TREE BASED

COLLISION RESOLUTION ALGORITHM WITH KNOWN MULTIPLICITY FEEDBACK

Ms. Robithoh Annur

A Dissertation Submitted in Partial Fulfillment of the Requirements

for the Degree of Doctor of Phylosophy Program in Electrical Engineering

Department of Electrical Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2013

Thesis Title	DELAY ANALYSYS OF A NEWLY PROPOSED TREE BASED COLLISION RESOLUTION ALGORITHM WITH KNOWN MULTIPLICITY FEEDBACK
By	Ms. Robithoh Annur
Field of Study	Electrical Engineering
Thesis Advisor	Associate Professor Lunchakorn Wuttisittikulkij, Ph.D.
Thesis Co-Advisor	Professor Jun-ichi Takada

Accepted by the Faculty of Electrical Engineering, Chulalongkorn University in Partial Fulfillment of the Requirements for the Doctoral Degree

……………………………………… Dean of the Faculty of Engineering
(Professor Bundhit Eua-arporn, Ph.D.)

THESIS COMMITTEE

……………………………………………….. Chairman
(Associate Professor Watit Benjapolakul, D. Eng.)

……………………………………….……. Thesis Advisor
(Associate Professor Lunchakorn Wuttisittikulkij, Ph.D.)

……………………………………………….. Thesis Co-Advisor
(Professor Jun-ichi Takada)

……………………………………………….. Examiner
(Associate Professor Prasit Teekaput, Ph.D.)

……………………………………………….. Examiner
(Assistant Professor Pasu Kaewplung, Ph.D.)

……………………………………………….. External Examiner
(Sathaporn Promwong, Ph.D.)

โรบิโต อันนูร์ : การวิเคราะห์เวลาประวิงอัลกอริทึมการแก้ไขปัญหาการชนกันแบบต้นไม้แบบใหม่เมื่อมีการป้อนข้อมูลกลับจำนวนแพ็กเกตที่ชนกัน

อ. ที่ปรึกษาวิทยานิพนธ์หลัก : รศ.ดร.ลัญฉกร วุฒิสิทธิกุลกิจ

อ.ที่ปรึกษาวิทยานิพนธ์ร่วม : จุนอิจิ ทากะดะ, 112 หน้า.

วิทยานิพนธ์ฉบับนี้ทำการศึกษาสมรรถนะของโพรโทคอลควบคุมการเข้าถึงตัวกลางที่ใช้รูปแบบเฟรมจำนวน 2 โพรโทคอล ดังนี้ โพรโทคอล framed slotted Aloha และอัลกอริทึม tree ที่ใช้ข้อมูลผลป้อนกลับหลายประเภท คือ ผลป้อนกลับแบบ binary, ผลป้อนกลับแบบ ternary และผลป้อนกลับแบบทราบจำนวนผู้ใช้บริการที่เข้าใช้สล็อตสัญญาณใดๆ โดยมีการนำเสนอกลไกพื้นฐานที่ใช้ในการแก้ปัญหาการชนจำนวน 4 รูปแบบเพื่อเป็นรูปแบบพื้นฐานสำหรับโครงสร้างของโพรโทคอลควบคุมการเข้าถึงตัวกลางแบบสุ่มประเภทต่างๆ การประเมินสมรรถนะด้วยการวิเคราะห์ทางคณิตศาสตร์ที่นำเสนอแสดงให้เห็นว่าการใช้ข้อมูลผลป้อนกลับที่มีประสิทธิภาพนั้นจะมีบทบาทสำคัญในการเพิ่มสมรรถนะทางด้านค่าเวลาประวิงให้ดีขึ้น อีกทั้งยังแสดงให้เห็นว่าสมรรถนะทางด้านค่าเวลาประวิงที่รับได้นั้นจะขึ้นอยู่กับข้อมูลผลป้อนกลับที่ใช้ในการแก้ไขการช่วงชิงกัน สำหรับกรณีที่ใช้ผลป้อนกลับแบบทราบจำนวนผู้ใช้บริการที่เข้าใช้สล็อตสัญญาณใดๆ นั้น ค่า MST ที่สามารถรับได้สูงสุดมีค่าเท่ากับ 0.533 ซึ่งได้จากโพรโทคอลการเข้าถึงแบบสุ่มที่นำเสนอโดยใช้กลไกการแบ่ง (split), การปรับเปลี่ยนขนาดของเฟรม, การละทิ้งสล็อตรูปแบบที่ 2 (slot-skipping type II) และความน่าจะเป็นการเข้าถึงแบบไม่เอกรูป (non-uniform) ร่วมกัน เรายังได้นำเสนอการศึกษาแบบจำลองทั่วไปในเบื้องต้น ที่ประกอบด้วยโนดหลายคลาสซึ่งมีความต้องการคุณภาพของการบริการ (quality of service) ที่แตกต่างกัน โดยใช้ slotted Aloha ที่ออกแบบโดยเฉพาะสำหรับ โพรโทคอลชั้นควบคุมการเข้าถึงตัวกลางที่อาศัยการจอง (reservation-based MAC protocol) ที่ยอมให้เราพัฒนาวิธีการจัดลำดับความสำคัญ (prioritization) ของโนดตามอัตราความสำเร็จ (success rate) ในขณะเดียวกันก็ยังรักษาประสิทธิภาพการใช้ช่องสัญญาณ (channel utilization)

ภาควิชา : วิศวกรรมไฟฟ้า ลายมือชื่อนิสิต ....

สาขาวิชา: วิศวกรรมไฟฟ้า ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก ....

ปีการศึกษา 2556 ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์ร่วม ....

5371284121 : MAJOR ELECTRICAL ENGINEERING

KEYWORDS : RANDOM MULTIPLE ACCESS / KNOWN MULTIPLICITY FEEDBACK / TREE ALGORITHM / FRAME SLOTTED ALOHA

ROBITHOH ANNUR : DELAY ANALYSIS OF A NEWLY PROPOSED TREE BASED COLLISION RESOLUTION ALGORITHM WITH KNOWN MULTIPLICITY FEEDBACK. ADVISOR : ASSOC. PROF. LUNCHAKORN WUTTISITTIKULKIJ, Ph.D., CO-ADVISOR : PROF. JUN-ICHI TAKADA, 112 pp.

This thesis investigates the performance of two frame-based MAC protocols namely framed slotted Aloha and the tree algorithms with different types of feedback information; binary, ternary, and known multiplicity. Four fundamental mechanisms for resolving collision are introduced as basic building blocks for the construction of a wide range of random access MAC protocols. The proposed analytical evaluation has shown that the use of feedback information, if used efficiently, plays a vital role in delay performance improvement. The achievable delay performance is shown to be highly dependent upon how the feedback information is used in the contention resolution. For known multiplicity, the maximum achievable MST of 0.533 is obtained by our proposed random access protocol that is derived by the combination of splitting mechanism, adaptive frame size, slot-skipping type II, and non-uniform access probability. We also present a preliminary study of a generic model that serves multi-class nodes with different quality of service requirements using slotted aloha, designed specifically for reservation-based MAC protocol with slotted Aloha that allows us to develop a variety of prioritization schemes, whereby nodes can be prioritized through reservation success rates, while aiming to maintain high efficiency of channel utilization.

Department : Electrical Engineering	Student's Signature ....
Field of Study : Electrical Engineering	Advisor's Signature ....
Academic Year : 2013	Co-advisor's Signature ....

Acknowledgements

This work has been carried out at Telecommunications System Research Laboratory (TSRL), Department of Electrical Engineering, Chulalongkorn University. The financial support by JICA project for AUN/SEED-Net is gratefully acknowledged.

First of all I am grateful to Allah SWT for giving me strength and wisdom throughout all my life and especially to finish this thesis. My sincere thanks goes to my supervisors, Associate ProfessorLunchakorn Wuttisittikulkij and Professor Jun-ichi Takada, for their excellent guidance, encouragement, and insightful comments throughout the period of my thesis.

I greatly appreciate formidable effort and valuable comments of my committee member including Associate Professor. Dr. Watit Benjapolakul , Associate Professor Dr. Prasit Teekaput, Assistant Professor Dr. Pasu Kaewplung, and Dr. Sataporn Promwong. I would like also to express gratitude to all Professors at Department of Electrical Engineering, Chulalongkorn Univerity for providing me valuable knowledge and useful background I also wish to express my special appreciation to Ajarn Suvit Nakpeerayuth for for bringing valuable knowledge to our discussion and a generous advice to my research. The research experience with him became a foundation stone and valuable experience for my future interests in the wireless communication field. Many thanks also goes out to the CU staffs and students especially in Telecommunications lab. for their assistant, friendship and helpful discussion throughout my candidature. Lastly, I am grateful to my parents and my family who have always given me unconditional caring and great support.

Contents

List of Tables

List of Figures

Figure Page

CHAPTER I

Introduction

1.1 Multiple Access Communication

In broadcast or multi-access communication networks, many mobile users (transmitters) are connected to a single receiver through a common communication channel as depicted in Fig. 1.1. The channel here refers to the medium through which all transmitters send their packets. Such a network can be found in a wireless local area network (LAN) and satellite communications. To manage the access among users, a Medium Access Control (MAC) protocol is required to ensure an efficient and fair sharing of the resources [1].

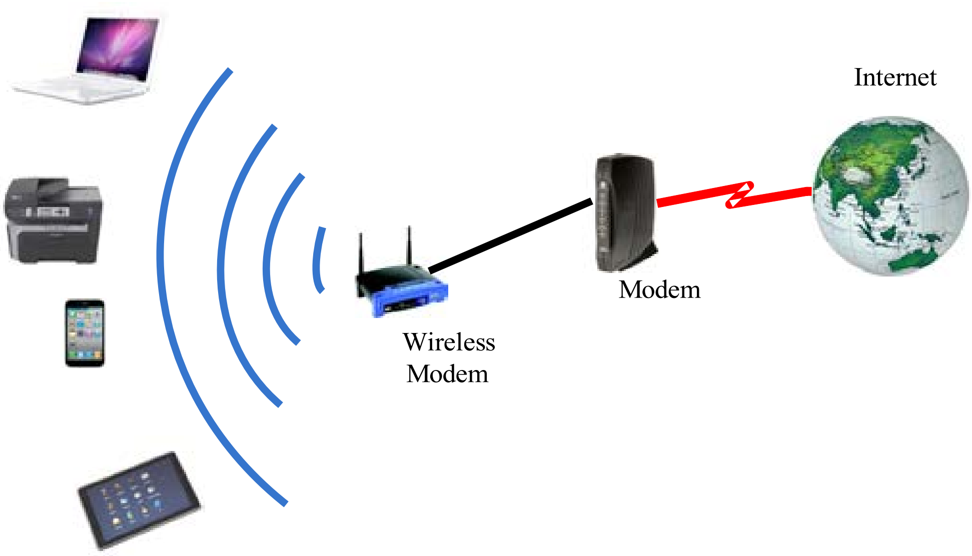


Figure 1.1: An Example of Multiple Access Access Communication System

MAC protocols can be broadly classified into fixed assignment, random assignment, and reservation based protocols [1, 2]. The fixed assignment protocol which is also known as collision free based protocol allows the mobile users to access the shared channel in a predetermined way so that collision will never occurs. Examples of the fixed assignment protocols include Time Division Multi Access (TDMA), Frequency Division Multiple Access (FDMA), and Code Division Multiple Access (CDMA) [1]. The random assignment or contention based protocol allows the users to contend the channel in a random fashion. This includes slotted Aloha, CSMA and splitting protocols [3–5]. In the reservation based MAC protocol, users reserve the channel before transmitting data packet. The reservation can use either collision-free or contention-based protocols. The existing reservation based protocols are Reservation TDMA [13] and Reservation ALOHA [14].

In this thesis we focus on the random access protocols where all users have the same right to access the channel by contention. This is also known as contention-based protocol. Different from the fixed assignment, random access MAC protocol is prone to collisions such that transmission of a packet is not guaranteed to be successful. A collision occurs when at least two users contend for the channel at the same time due to the randomness in nature of contention. In every collision, the collided packets completely loss their information so that retransmission is required until they are successful for each of them. This results in performance degradation of the system such as long delay and low throughput. A lot of retransmission algorithms for random access protocols for minimizing further collisions and hence improve the performance of random access communications have been proposed in literatures.

This line of research began since a simple and elegant protocol called Aloha (Pure Aloha) was introduced by Norman Abramson in the 1970s [7]. Slotted Aloha and frame slotted Aloha came after as the improved model of the pure Aloha. Aloha protocols are widely adapted in a lot of applications because of its simplicity. However, the analysis shows that Aloha has stability problem [9, 30]. Since Aloha does not have real strategy to resolve collision, repeated collision may occur. This will increase the number of collided packets. It means that a very large number of packets are in the system yielding in high probability of collision. This condition will eventually results in zero throughput and the system becomes unstable.

The instability problem of the Aloha protocols gave rise to researchers to develop stable random multiple access algorithms such as Conflict Resolution Algorithm. Resolution Algorithm (CRA) which we now consider has been studied in literatures as another class of protocol for obtaining a stable random multiple access communication. Similar to other MAC protocols [5,7,10,44], CRA defines a set of rules for users to transmit their new packets after generated and retransmit packets upon collisions. However, CRA has a specific rules for resolving collisions till all collided packets are eventually successfully retransmitted. To achieve this condition, users in the system can use the channel history of the system from the feedback information [30]. The details and historical review of CRAs will be given in the next chapter.

1.2 Objective

The main goal of this research is to mathematically analyze and numerically compare the performance of two important classes of random access MAC protocols, namely framed slotted Aloha and tree algorithms with respect to the mean contention resolution interval (CRI) lengths and the maximum stable throughput (MST) under different feedback information, i.e. binary, ternary and multiplicities. It is also aimed to introduce a framework that allows a systematic construction of a wide range of random access protocols based on the combination of four fundamental mechanisms: i) splitting mechanism, ii) adaptive frame size mechanism, iii) slot-skipping mechanism and iv) non-uniform access mechanism. This framework not only serves as an effective means to classify various existing random access protocols but also leads to the ease of new random access protocol designs. Comprehensive investigations on most if not all well known random access protocols as well as newly derived protocols are conducted to determine which protocols are most effective together with their optimal parameter settings for different assumption of feedback information.

1.3 Organization

In this thesis we analyze the delay so called Mean Collision Resolution Interval (CRI) length of framed slotted Aloha and the tree collision resolution under the assumption of binary, ternary and known multiplicity feedback.

In Chapter 2, we present an overview of medium access controls (MAC) protocols for random multiple access communication system. Two conventional random access protocol i.e. Aloha based and tree algorithm are discussed. We also present the common assumption in the study of random access protocols such as slotted system, types of feedback information, and channel access algorithm for the newly generated packet.

In chapter 3, we focus on the mean CRI length of the framed slotted Aloha. With three different types of feedback, we will combine and deploy the two fundamental mechanisms namely slot-skipping and adaptive frame size to the framed slotted Aloha. Mathematical model of the mean CRI length of framed slotted Aloha with the proposed variations are expressed in recursive formula.

In chapter 4, With the focus on known multiplicity feedback, we develop an adaptive and slot skipping tree collision resolution algorithms. Here, all the four aforementioned basic mechanisms are applied. For the analysis, we are interested in two performance metrics; mean CRI length and maximum stable throughput.

In chapter 5, conclusions are drawn based on theoretical analysis and simulation results and recommendations for future work are also presented.

1.4 Contributions

This thesis proposes a framework for systematic study of random access protocols, based on framed slotted Aloha and tree algorithms. This framework enables us to understand deeply into how well each random access protocol performs and identify the key mechanisms that can resolve collisions in the most effective manner.

Through mathematical analysis, all known random access as well as newly proposed protocols are investigated extensively in terms of the mean contention resolution interval (CRI) lengths and the maximum stable throughput (MST) under different feedback information, i.e. binary, ternary and known multiplicity.

We present an insight understanding for the first time that two very different classes of random access protocols, namely framed slotted Aloha and tree algorithms, can be related; that is the framed slotted Aloha can be made exactly identical to the tree algorithms by simply introducing the splitting mechanism. This finding means that any new additional features that can be or have been applied to the tree algorithms can also be used in the framed slotted Aloha protocols, implying that a wide range of random access protocols can be derived with ease.

Numerical results show that our proposed random access protocols with known multiplicity feedback, especially when all four fundamental mechanisms: i) splitting mechanism, ii) adaptive frame size mechanism, iii) slot-skipping mechanism and iv) non-uniform access mechanism are employed, are found most effective to date. The maximum achievable throughput is 0.533.

CHAPTER II

Random Multiple Access Protocols

The random multiple access protocol is also known as the contention based protocol where users in the system need to contend each other to seize the channel for packet transmission. Users generate packets independently and contend the channel in random manner. Moreover, there is no coordination and negotioation between the users to manage who and when to transmit the packet. These make contention based access prone to conflict so called collision i.e when at least two packets are sent at the same time. The collided packets are needed to retransmitted until the get successful transmission. A contention based is then defined to manage the access and minimize resolve the conflicts.

The random multiple access protocol can be classified into two categories namely collision avoidance and collision resolution protocols [6]. Aloha and carrier sensing are example of collision avoidance. Aloha allows the packet to be transmitted as soon as it generated. If a collision occur, the collided packets will be transmitted in the random time later to avoid further collisions. Different from Aloha, carier sensing multiple access (CSMA) which is also known as ”listen before talk” has sensing feature i.e. users sense the channel before transmitting the packet to avoid collision. On the other hand, the collision resolution algorithms (CRA) resolves acollision by deviding a collision into smaller group recursively until all the colided users get succesful transmission. Review of these access protocol is presented in the sequel.

2.1 Pure Aloha

ALOHA or Pure ALOHA is considered as the most conventional medium access control protocol that is proposed by Norman Abramson and his colleagues at the University of Hawaii. It aimed to interconnect a central computer at the university main campus near Honolulu to remote consoles at colleges and research institutes on several islands using UHF radio communications [7]. Two 100 kHz channels at 407.350 MHz and 413.475 MHz are assigned for transmission in each direction, each operating at a bit rate of 24,000 baud. In the ALOHA system, information is transmitted in the form of packets, and all packets are of fixed length, i.e. 88 bytes (8 bytes for header and 80 bytes for data). Therefore, the packet transmission time is about 29 msec and this time becomes 34 msec when information for receiver synchronization is included. The basic idea of the Pure ALOHA protocol is simple, but elegant: each user is allowed to send its packet whenever it has a packet ready for transmission. Since a common channel is shared among user, collision between packets from different users will result when they are sent at nearly the same time. Fig.2.1 shows an example of packet transmissions and possible collisions of four users contending for the same channel. Those packets that are overlapped in time are collided and destroyed. In this example, only two packet transmissions are successful, and the rest of them need to be retransmitted.

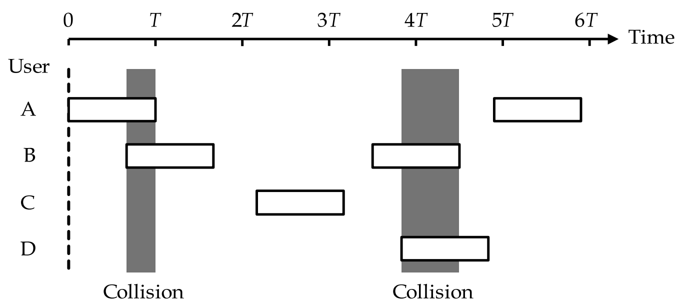


Figure 2.1: Packet transmissions in a Pure ALOHA system.

After a packet transmission, the sending user waits for an acknowledgement from the receiver to indicate successful transmission of the packet. However, if no acknowledgement is returned within a time-out period, the sending user assumes that the packet is destroyed and starts a retransmission procedure. In principle, the time-out period must be set at least equal to the maximum possible round trip delay between two most widely separated users to ensure correct functioning of the protocol. Obviously, if the colliding users try to retransmit their packets immediately, they will collide again. Therefore, each user is required to wait for a random amount of time, called back-off time, before resending the packet. This random back-off mechanism is intended to keep multiple users from trying to transmit at the same time again which helps reduce probability of collisions. The back-off time is randomly chosen from the range $[0, 2^k - 1]$ multiplied by the maximum propagation delay (or alternatively the packet transmission time), where k is the number of previous unsuccessful transmission attempts. This means that the mean value of back-off time is doubled each time the packet is retransmitted. This retransmission is repeated until either the packet is acknowledged or a predetermined number of retransmissions, typically set as 15 attempts, is exceeded.

To see how well such a simple protocol will perform, a throughput analysis for the Pure ALOHA protocol is carried out with the following basic assumptions. There is an infinite number of users that are generating new packets according to a Poisson process with an average of S packets per packet transmission time. All packets are of equal length and the packet transmission time is T seconds. Packets that fail to reach the intended receivers due to collisions are retransmitted. Since retransmitted packets are vulnerable to collisions too, they will also require retransmission again if not successful. Let us define$G$ as the average number of packets both new and retransmitted combined per packet transmission time. Obviously, G is always greater than or equal to S . It is further assumed that generations of these combined packets during one packet transmission time also follows Poisson distribution. The ratio of S to G is essentially the probability of a successful packet, that is

Fig.2.2 shows the vulnerable time of a shaded packet, which starts its transmission at time t and finishes at $t+T$. This shaded packet is successfully transmitted, as long as no other packet is transmitted during the interval $t-T$ to $t+T$, so-called vulnerable period. If another packet begins a transmission within the interval $t-T$ to t, such as packet B, the end of this packet will collide with the start of the shaded packet. If another packet begins a

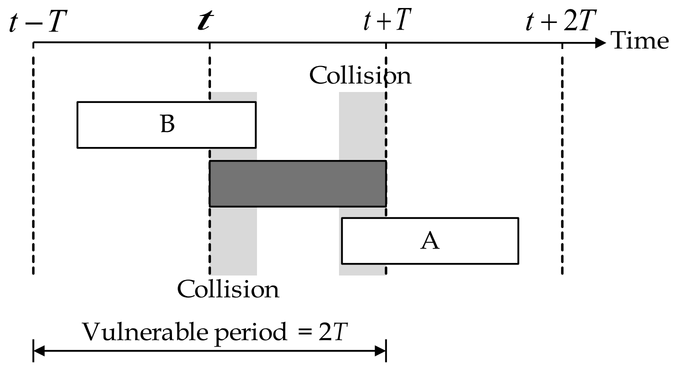


Figure 2.2: Vulnerable time for Pure ALOHA .

transmission within the interval t to $t+T$, such as packet A, the start of this packet will collide with the end of the shaded packet. Based on this observation, it is clear that the shaded packet has a vulnerable period of $2T$, in which if no other packet starts any packet transmission, no collision will occur and the shaded packet will reach the receiver successfully. Therefore, the probability of a successful packet (P_s) in Pure ALOHA is equal to the probability of no generation of packet within $2T$ second. Since the probability of k packets are generated within 2 times the packet transmission time according to the Poisson distribution is given by:

$$\Pr[k] = \frac{(2G)^k e^{-2G}}{k!} \tag{2.1}$$

the probability of no packet generated is

$$\Pr[k=0] = e^{-2G} \tag{2.2}$$

By combining Equations 2.2 and 2.2, we get

$$S = Ge^{-2G} \tag{2.3}$$

This relation between G which represents the total offered traffic on the channel and which represents the throughput of the Pure ALOHA system is plotted in Fig.2.3. It shows that initially at low traffic load throughput increases with increasing offered traffic up to a maximum of $1/2e = 0.184$. occurring at a value of $G = 0.5$. A further increase of traffic leads to a higher collision probability due to more intense contention, causing a reduction of throughput.

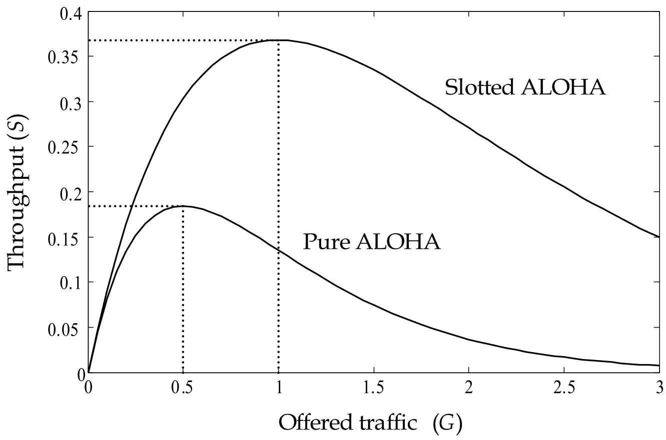


Figure 2.3: Throughput versus offered traffic for Pure and Slotted ALOHA.

2.2 Slotted Aloha

In 1972, a simple modification of Pure ALOHA is introduced to improve the performance by introducing slotted system namely the time is devided into slots [8]. This modification is known as slotted Aloha. The following assumptions are the difference between pure Aloha and slotted Aloha [5]:

- Slotted system

  The length of the slots is equal to the transmission time of data packet. If the user has a packet ready to send, it must wait until the beginning of the next time slot to transmit the packet. When the synchronization is assumed to be perfect, this will finish at the end of the same slot.

- Packet Size

  In idealized slot system, it is assumed that the packet size is equal for all transmitters and it needs one slot to transmit each packet.

- Channel type

The channel which refers to the medium through which all transmitters send their packets including feedback is assumed to be errorless channel. This means that the packet correctly received if the channel slot contains exactly one packet otherwise it is detected as collision.

- Immediate feedback

  The state of the channel is assumed to be broadcasted by the base user to all users at the end of each slot as feedback information which specifies the condition of slot as one of these following possibilities:

  - Idle; when no packet was transmitted
  - Success; when exactly one packet occuppied the slot.
  - Collision; when more than one packet were transmitted in the slot. This result packer error reception where that the collided packets are received incorrectly such that retransmission is needed.

When a user has packet ready to transmit, it will wait for the next slot for transmision. In case of collision, each user involved retransmits its packet in each subsequent slot with probability p until success. Since a packet transmission is confined within the slot boundary, the collided packets will overlap completely. This means that the vulnerable period for Slotted ALOHA is reduced by half compared to Pure ALOHA. Fig. 2.4 shows an example of packet transmissions and possible collisions in the Slotted ALOHA system. Notice that most packets are generated during a slot interval, and they are kept waiting until the start of the next slot before transmitted. Indeed, the traffic pattern is deliberately selected to be the same as in Fig. 2.1 for comparison purpose with Pure ALOHA. Slotted ALOHA appears to reduce collision in this example; only two packets are collided compared to four in case of Pure ALOHA.

Since the throughput of Slotted ALOHA can be analyzed in the same way as Pure ALOHA except that the vulnerable period is now equal to the packet transmission time, the probability of no other packet is sent in the same slot is

$$\Pr[k=0] = e^{-G} \tag{2.4}$$

and thus the relation between throughput and offered traffic for Slotted ALOHA can be obtained as

$$S = Ge^{-G} \tag{2.5}$$

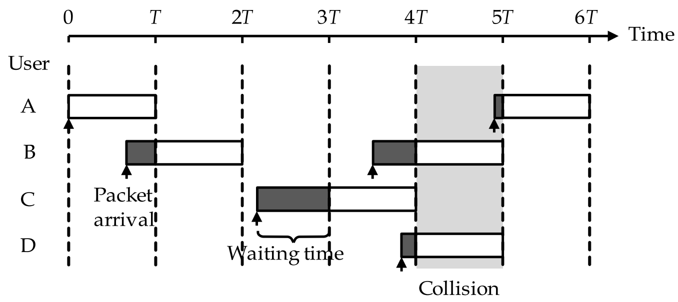


Figure 2.4: Packet transmissions in a Slotted ALOHA system.

Under the assumption of Poisson traffic, some analysis shows that the introduction of slotted system will improve the performance of the pure Aloha namely the maximum throughput of Slotted ALOHA. Fig. 2.3 illustrates the comparison of throughput performance of Pure and Slotted ALOHA. The maximum throughput of Slotted ALOHA is $1/2e = 0.368$, which occurs at $G = 1$; this is doubled of that of Pure ALOHA. As we can see, the efficiency of Pure ALOHA can be improved by the introduced time slot structure. However, time synchronization is required to align stations to the slot structure. One possible solution is to have a central station send a kind of clock signal at a regular interval.

Both Pure and Slotted ALOHA have advantageous features. First, they are highly decentralized and quite simple to implement, especially Pure ALOHA. Second, when there is only one active user, the user can continuously transmit its packets at the maximum channel capacity. These two key features make the ALOHA system particularly useful for large population of users each with light and burst traffic demand. However, due to their simplicity of operation, ALOHA makes inefficient use of channel capacity and is low in throughput performance.

2.2.1 Slotted Aloha with Finite Number of Users

Previously, the throughput Aloha is analyzed under the assumption of Poisson traffic which means that the number of users is infinite. However, real implementation allows the network to have a finite finite number of users. This make the assumption of Poisson traffic is not applicable. In [11, 12], the performance of slotted Aloha with finite number of users is analyzed. It is considered that the slotted Aloha is applied in the system which consist of M number of users, where M is finite. Every users has a single buffer, meaning that users are allowed to have at most one packet. Each user can be in the state ”thinking’ or ”backlogged”. When users have no packet for transmission, they are in the state of ”thinking”. In each slot, they are allowed to generate a new data packet with probability σ and send the packet in the next slot. If the packet transmission is succesfull, the status of a user does not change meaning that the user is allowed to generate a new packet. However, a thinking user can be backlogged user if the transmitted packet collides. This requires the backlogged user to retransmit the collided packet in the future slots with probability v. This backlogged users are also not alllowed to generate a new packet until the status changes into thinking.

Let $N(t)$ denote the number of backlogged users in the beginning of slot t. This can also be used to represent the state of the system. Since $N(t+1)$ depends on N and the state of the users can change from one slot to another slot, a finite Markov chain can be used to model the state transition of the system which represents the changing of the number of backlogged users as depicted in the Fig. 2.5

For the analysis, let first define π_i as 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 which can be obtained

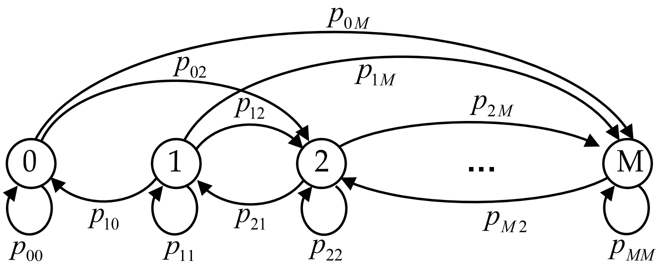


Figure 2.5: Markov model representing the number of backlogged users for slotted Aloha with finite number of users.

by the following expressions [4, 12]:

$$p_{ij} = \begin{cases} 0, & j < i-1 \\ [iv(1-v)^{i-1}](1-\sigma)^{M-i}, & j = i-1 \\ [1-iv(1-v)^{i-1}](1-\sigma)^{M-i} + [(M-i)\sigma(1-\sigma)^{M-i-1}](1-v)^{i}, & j = 1 \\ [(M-i)\sigma(1-\sigma)^{M-i-1}][1-(1-v)^{i}], & j = 1+1 \\ \binom{M-i}{j-i}\sigma^{j-i}(1-\sigma)^{M-j}, & j > i+1 \end{cases} \tag{2.6}$$

To obtain the steady-state probability, it is necessary to solve the following finite set of linear equations

$$\pi = \pi\mathbf{P} \tag{2.7}$$

$$\sum_{i=1}^{M}\pi_i = 1 \tag{2.8}$$

where $\mathbf{P}$ and π is a matrix and a row vector with p_{ij} and π_i as their elements, respectively. Having found the vector π, the values of π_i can be used for the throughput analysis. Fudamentally, throughput denoted by S is defined as the expected number of success slots. This is equal to the probability of successful transmission in a slot, i.e. $S = P_{success}$. For the system described previously whwre users can be either thinking or backlogged. a success slot occurs when there is exactly one packet transmitted by either a thinking or backlogged user. It means that if the one thinking user send a packet, none of the backlogged users sending a packet or if one of the backlogged user sends a packet, none of the thinking users sending a packet For a given i backlogged users from the total M users, the probability of success can be obtained by:

$$\begin{aligned} P_{success}(i) &= \binom{M-i}{1}\sigma(1-\sigma)^{M-i-1}(1-\nu)+\binom{i}{1}\nu(1-\nu)^{i-1}(1-\sigma)^{M-i} \\ &= (M-i)\sigma(1-\sigma)^{M-i-1}(1-\nu)+(i)\nu(1-\nu)^{i-1}(1-\sigma)^{M-i} \end{aligned} \quad (2.9)$$

Therefore, the throughput is

$$S=\sum_{i=0}^{M} P_{success}(i)\pi_i \quad (2.10)$$

For the case when, there is no difference in the trasmission probability of the thinking user and thinking user, i.e. $\sigma = \nu$. Therefore, the probability of success in 2.2.1 is reduced to

$$P_{success}(i) = M\sigma(1-\sigma)^{M-1} \quad (2.11)$$

It is shown that P is no longer function of i. Then the expresion of the throughput becomes:

$$\begin{aligned} S &= \sum_{i=0}^{M} P_{success}(i)\pi_i \\ &= P_{success}(i)\sum_{i=0}^{M}\pi_i \\ &= M\sigma(1-\sigma)^{M-1} \end{aligned} \quad (2.12)$$

With no differentiation between the transmission probability of the thinking and backlogged users, it is clear that in every slot, each of the M users will transmit a packet with probability σ. The average number of packets transnmitted in each slot which is denoted by G is equal to $M\sigma$. By substituting this value into (2.2.1), the system throughput, S, is reduced to

$$S = S\left[1-\frac{G}{M}\right]^{M-1} \quad (2.13)$$

When we bring this to the assupmtion of infinite number of users i.e $M \to \infty$. The throughput equation in 2.2.1 will be exactly equal to throughput of slotted aloha with finite number of users as in the equation (2.2). It is suggested that σ must be set to be much less than v to maintain the system for offering the network load.

2.3 Framed Slotted Aloha

Another variation of Aloha protocol is the framed slotted Aloha (FSA) which divised from the slotted Aloha. A frame structure is introduced where a number of slots are always grouped in a frame [11, 15, 16]. This firstly proposed together with the analysis of the slotted Aloha with finite number of users discussed in the previous section. The mechanism of the frame slotted Aloha is also called as the uniform retransmission randomization scheme since the frame is used for the retansmission of the collided packet. Morover, the collided users choose a slot in the frame with uniform probability. The size of the frame can either be constant (basic framed slotted Aloha) or dynamic (dynamic frame slotted Aloha) [15–21] . This frame structure is proposed for imposing a constraint on retransmission probability by allowing the user to choose one slot for packet retransmission that is useful in maintaining stability. The frame structure may initiated when a collision occurs or it is possible that the frame is applied for the whole system.

The operation of the framed slotted Aloha is explained as follows. Denoted that the frame size is K slots. If there is a collision involving a number of users, each user randomly and independently chooses a single slot within the frame to transmit its packet with probability $1/K$. Therefore, the transmission of the corresponding colided packet occurs in the slots within the frame. The feedback will return at the end of the frame. If they still experience collisions, they will transmit again in the next frames until their packets are successfully retransmitted. This is the feature of the framed slotted Aloha which makes it different from the original slotted Aloha where the retransmission is in the future slots without any boundary.

Fig. 2.6 illustrates the operation of the frame slotted Aloha with $K = 3$ where 5 users

(A, B, C, D, E) collide in Slot 1. It is assummed that there is no new packet comming before this group of users succeed to transmit their packets. The 5 users then transmit in the three consecutive slots in frame 1 at random. Suppose that no user transmit in Slot 2 and 3. Surely, all the collided users transmit in the last slot in this frame which is Slot 4. A new contention frame then starts where user A, D, and E experience a new collision in Slot 5, while B and C get successfully transmit their packet in Slot 6 and 7, consecutively. The resolution continues for resolving the collision involving user A, D and E in a new available frame. User D transmits in Slot 8, Slot 9 is empty and a collision occurs in Slot 10 with user A and E involve in it. These two users then again have to choose one slot in the next frame for their packet transmission. Finally, the collision is resolved where user A and E succeed in Slot 11 and 12. Slot 13 remains idle since all users have succeeded to transmit their packets.

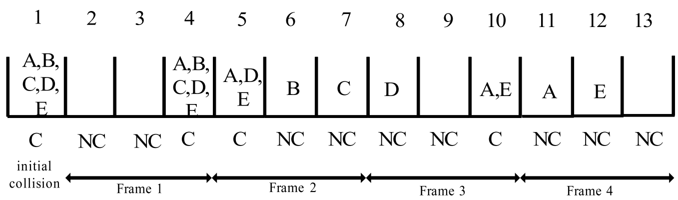


Figure 2.6: Contention resolution in Slotted ALOHA system.

The stability of the framed slotted Aloha with finite number of users can always be maintained when frame size is set into sufficiently large value [11]. This is basically the same nechanisme for the case of original slotted Aloha since the setting of large K has the meaning that the probability of transmission is set to be small which may be smaller than the newly packet geneartion. When K is infinity and the packet retasmission is Poisson traffic, this then will result a system wich is identical to the original slotted Aloha.

2.4 Carrier Sense Multiple Access (CSMA)

Pure ALOHA has a shortcoming in that a user still transmits its packet even if the channel is already occupied by another user. Such collisions can be avoided, if only the sending user senses the channel before using it. This led to the development of an important class of MAC protocols called Carrier Sense Multiple Access (CSMA). A user that wishes to send a packet is required to sense if the channel is busy or idle first. If the channel is sensed busy, the user must wait until the channel becomes idle again before making any transmission. Such a listen before talk strategy helps reduce unnecessary packet collisions, thereby increasing channel efficiency. Fig. 2.7 shows an example of possible packet transmissions in a CSMA system for the same traffic situation as in Fig. 2.4 of Pure ALOHA. As we can see, each packet waits until the channel becomes idle before transmission and in this particular example, no collisions occur at all; all packets are successfully transmitted.

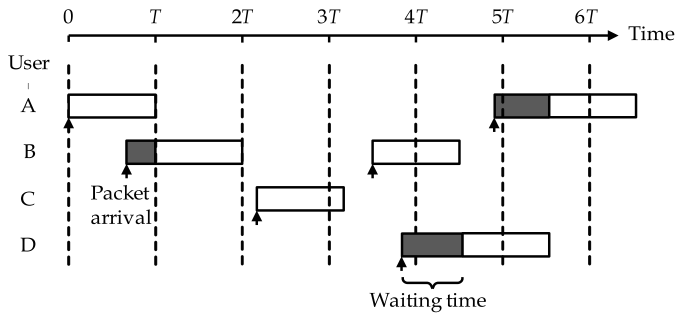


Figure 2.7: Packet transmissions in a CSMA protocol.

2.5 Collision Resolution Algorithm

As aforementioned, the idea of Collision Resolution Algorithm (CRA) is to provide a stable random access protocol since the Slotted Aloha faces the instability problem. By exploiting the feedback information in more sophisticated manner, CRAs aim to resolve collision in more efficient way such that every packet is eventually successfully transmitted with finite delay [4]. The main property of this algorithm is that after a collision, only the users which involve in this particular collision are entitled to contend the channel for retransmission. In addition to that, the other users have to wait for their packet retransmission until the current collision resolved. This approach is also known as splitting algorithm since the basic idea of this algorithm is to resolve collisions by splitting the collided users into smaller subgroup recursively. This algorithm always resolves the first encountered collision completely before starting to resolve the subsequent collisions. The CRA starts when two or more users sent their packet at once and collide. The colliding user are divided into smaller groups which commonly have low probability of collision since these new groups contain lower number of users than that of the initial collision. If further collisions occur, further subdivisions are repeated until result in successful transmission for all users.

The collision resolution protocol was first developed based on binary tree search by Capetanakis [23] and it is known as tree algorithm. Independently, Tsybakov and Mikhailov [24] proposed a protocol so called stack algorithm where a concept that is similiar to the tree algorithm is implemented in a virtual stack system. Resolving collision based on the time arrival of the packets, Gallager [25] described a protocol with the same fashion as in tree or stack algorithm which is also known as First Come First Serve (FCFS) algorithm.

In MAC protocols including Collision resolution algorithms, one must define the rules for transmission of new packets after their generation and re-transmission of the backlogged packets upon collisions. Before discussing further details of the development and classification of collision resolution protocols, it is important to have knowledge regarding the rules of collision resolution algorithm. There are some issues that normally considered in the splitting collision resolution protocols such as Channel Access Algorithm (CAA), Feedback information, User population, Active user model [4,6]. Each of these issues will be discussed in the sequel.

1. Channel Access Algorithm (CAA)

   In general, CRA is defined as a specific part of protocol for resolving collision algorithmically after it arises [31]. Another part of collision resolution protocol is that CAA which determines a rule on the first transmission of new generated packets from the transmitters. Combination of these two algorithms will form a random multi access protocol as a whole. To handle the newly generated packets, CAA in collision resolution protocols can be classified as[Mathys85,Markowski97]:

- Blocked Access

  Blocked access protocols do not allow a new generated packet to join the current CRA before users in the initial collision are all resolved. There are two mechanisms of Blocked access protocols:

  - Obvious blocked access

    After a collision occurs, a CRI starts for resolving the backlogged packets. All users with new generated packets (if so) are blocked to access the channel and do not have right for transmission. The new packets will be stored in a buffer considered as waiting packets. Once the CRI complete, these buffered packets will immediately be transmitted in the first slot of the next CRI. This obvious blocked access is also called as gated access since the channel is gated during CRI and will be opened at the end of CRI. Obvious blocked access splitting algorithm refers to the the tree algorithm which first suggested by Capetanakis and Tsybakov and Mikhailov [23, 28–30, 46, 47].

  - Non obvious blocked access

    In the non obvious blocked access, newly generated packet may join to the in progress collision resolution process with specific rules. The enable packets for transmission is determined based on their arrival time. The algorithm defines an enable interval time and allows the packets generated in this period to be transmitted while other packets including the new arrivals have to wait. If collision occurs, the CRA commences. The length of the enable intervals is varying; it will be shrunken for every collision and will be widened if feedback is either idle or success. Lengthening the enable arrival may allow a new packet to be transmitted before all collided packets solved. This feature is the reason for the name of non obvious blocked access. Changes in the length of the enable intervals can be viewed as a window such that it is also called as window access.

- Free Access

  Different from the blocked access, free access protocols allow users with new arrival packets to directly participate to the in progress collision resolution process. A packet will be transmitted in the right next slot after its generation. The CRA with free access is first presented by Tsybakov [32] which is easily interpreted in the form of stack and hence the Tsybakov model is also called as stack algorithm.

Based on the above explanations, each of these CCA has its own feature. The main feature of blocked access algorithm are: the backlogged packets have higher priority to be resolved and all users in the system with or without packet ready to be transmitted are required to monitor the channel condition. On the other hand, users are not required need to monitor the channel at all time in the free access mode. Only users with backlogged packets are needed to monitor the channel for the retransmission packets until their packet are successfully transmitted.

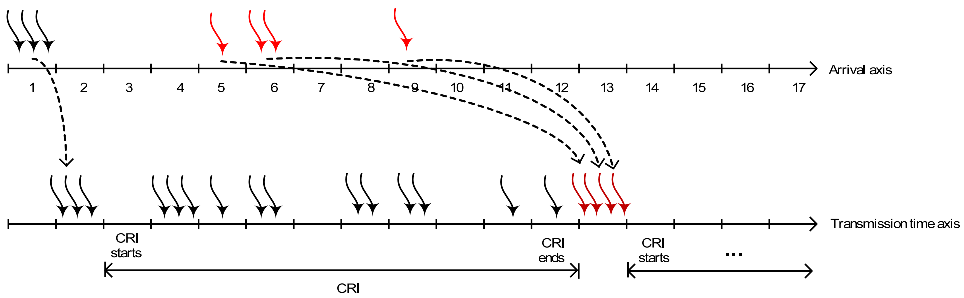


Figure 2.8: Tree Algorithm with Blocked Access

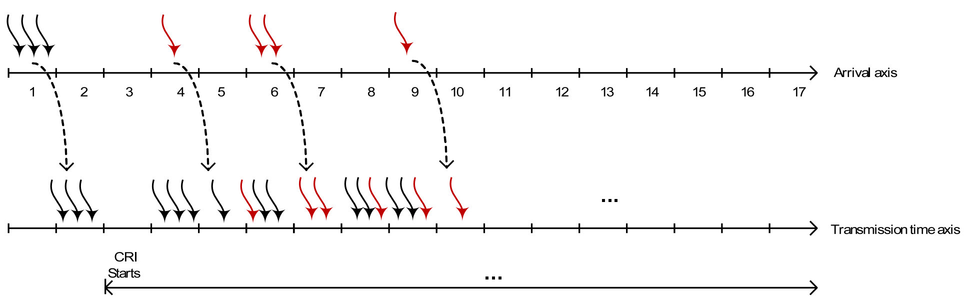


Figure 2.9: Tree Algorithm with Free Access

2. Feedback Information

Transmission of a packet in the random multiple access system may not be unsuccessful due to collisions. To ensure the results of their transmission, channel feedback which contains the channel condition after a transmission is sent to all users by the base station. The channel condition includes:

- Successful transmission if there was exactly one packet in the slot
- Idle if no packet was transmitted
- Error due to collision if at least two users were transmitted simultaneously in a slot.

This feedback can be then exploited to form the access protocol and further be utilized by all users to minimize or avoid further collisions in the next transmissions. The CRAs is kind of random access protocol which relies on the feedback information and hence feedback plays important role in designing a CRA. Here, different feedback information will be presented and classified based on its content and time to return to the users. By exploiting different feedback, different strategies can be exploited to form an effective protocol. The followings are the classification of feedback information based on its content [34, 38]:

(a) Binary feedback [4, 8, 23, 25, 28, 29, 36]

This feedback can be interpreted in three types of channel feedback. It may contain information for the transmitters whether collision or no-collision (C/NC) occurs in the channel. Something or nothing (S/N) is also an interpretation of binary feedback which informs the transmitters that the channel was empty or there is packet transmission in the channel. The last type of binary feedback may contain the information of successful or failure packet transmission which is normally given by success or failure (S/F) feedback.

(b) Ternary feedback [3, 30, 31, 45, 49]

Ternary feedback allows transmitters to clearly distinguish three conditions of the channel i.e. idle (0), or success (S), or collision among transmitters occurs (C).

(c) Known Multiplicity Feedback [**?**, 37, 40–43, 50–54]

This feedback is also called as (N+1)-ary feedback [39]where the system can distinguish N+1 different conditions of the channel. Known multiplicity feedback is basically ternary feedback which can distinguish idle, success and collision, however, in case of collision, this feedback provides the information of multiplicity number or the exact number of users involved in this particular collision. By detecting the energy of the received packets from the channel, the receiver can determine the number of packets on it. This type of feedback is assumed in some earlier studies by [37], [40], [42].

Based on the time of its return, feedback information can be classified into immediate and late feedback. Immediate feedback means that transmission outcome is available at the end of each slot by ignoring the round trip propagation delay. On the other hand, when propagation delay is taken into account, feedback needs more slots to reach at the transmitter side. Since feedback in this environment cannot arrive at the end of each slot, this feedback is categorized as late or delayed feedback.

3. Packet Generation Model

Users become active when they generate packet ready for transmission. In slotted system channel, a packet can be sent in the next slot after its generation. In finite or infinite population, the arrival of packet joining the channel contention can be modeled as Poisson and finite user models as using the same assumption i.e. Poisson arrival. The packets are generated according to a Poisson process with a mean arrival rate (over all users) of λ for which the inter-arrival times are independent and exponentially distributed. In this condition, it is possible that one user can generate many packets while another user may do not have any packet to be transmitted. In order to simplify the model, users are allowed to have at most two packets [28] which cannot involve in the same CRI. It means that if one packet is sent and the result is collision, the other packet cannot be transmitted until successful transmission of the first packet is achieved. This imply that users have buffer which can store one packet before transmitted.

4. Addressing Scheme

Since partition of collided users is the main concern in the splitting algorithm, it is important to define a mechanism for the users to join a subgroup after collisions. This mechanism is referred to as addressing scheme [6, 28] which is classified into deterministic and random addressing.

- Deterministic addressing

  In deterministic addressing, the system has already assigned addresses for every user after each collision. Hence, all users in the network have their own unique address to where they will join group after collision occurs. Since the address is unique, it ensures eventual successful transmission for all users related to the collision. In a system with infinite population, there will be an infinite addresses to be assigned to each users. Capetanakis in [28] mentioned that deterministic addressing performs somewhat better over the random addressing for system with finite population.

- Random addressing

  The sub-grouping of collided user is made on the basis of a random process. In this random addressing, system has already have a fixed number of subgroup for users to choose from. Whenever collision occurs, all collided users join the subgroup by choosing the available subgroup independently with either equal or different probability. A collision is resolved if only if all users already have their own subgroup.

- Arrival time based addressing

  Arrival time based addressing is used in first come first serve collision resolution algorithm. Every packet has its own time stamp indicating the time when it is arrive to the system. The protocol will determine a time interval in which all packets arrive in this interval are eligible to be sent. If there is more than one packet lay in this interval, it means that collision occurs. Subdivision of the collided packets is done by shrunken the time interval until to an interval where only one packet arrive. This addressing scheme, allows the users to transmit their packet if their packet arrival time is in in the enable arrival defined by the system.

In order to have better understanding of splitting protocol, we shall now present some variation of splitting protocols which adopt the aforementioned issues.

2.5.1 Basic Binary Tree Algorithm

Basic tree algorithm usually refers to the tree algorithm which is originally developed by Capetanakis. As aforementioned, stability is achieved by further dividing a group of collided users into small groups until the collision resolved. The basic tree algorithm can achieve maximum stable throughput of 0.347 [23, 28] by exploiting binary feedback type which provides information of ”collision” or ”no collision” (C/ NC) in the slot channel . Originally, basic tree algorithm applies deterministic addressing and follows serial tree searching where two slots are used to retransmit the collided packets successively. The deterministic addressing is represented by binary number (’0’ for first group and ’1’ for second group). Since every collision will result in always two new branches of tree, it is also called as static tree algorithm.

The procedure of resolution of the contentions between users is described as follows. The set of active users for contention are modeled by the assumption of the Poisson arrival. If a collision occurs, the mechanism of packet retransmission commences or in other words, the Collision Resolution Interval (CRI) starts. The users involve in this initial collision will be split into two based on their determined address. Set of users in the first subgroup will retransmit their packet in the next slot. Consecutively, users in the second subgroups send their retransmitted packet in the second slot after the collision. If the both subgroups contain more than one user, new collisions occur. The first subgroup will split again into another two subgroups and the algorithm continues in the same fashion as before. Meanwhile, users in the second subgroup have to defer their transmission until the contention in the first subgroups has all been resolved. The Capetanakis splitting process can be represented in the form of a tree which reflects to its name where two new branches will be generated by each collision [23, 28].

The example in Fig. 2.10 considers a finite population with 8 users(A, B, C, . . . , G) in the network. The binary addresses of the users and the collusion resolution in the slot axis are given in Fig.2.10(a) and Fig.2.10(b), respectively. Supposed there are 6 out of 8 users (A, C, D, E, F, H) collide in a particular slot. The CRA then starts as follows:

- The users in the initial collision are divided into two subgroups according to their defined addresses. A, C, and D are in the first group and have right to send their packet in the next slot (slot 2), while E, F, and H which are in the second group are entitled to transmit their packet in the successive slot (slot 3).

- Since collision occur in both slots, A, C, and D are again split into two subgroups and transmit their packet consecutively in slot 4 and 5. Meanwhile, E, F, and H have to wait until A, C and D succeed. A succeeds in slot 4, whereas C and D collide again in slot 5.

- C and D are further divided resulting in successful transmissions in slot 6 and 7.
- CRA then moves to the group of E, F and H. Splitting process results in collision (E,F) in slot 8 and successful transmission of G in slot 9.
- Eventually, E and F succeed after divided into slot 10 and 11. In this example, the CRA needs 11 to resolve collision among 6 active users.

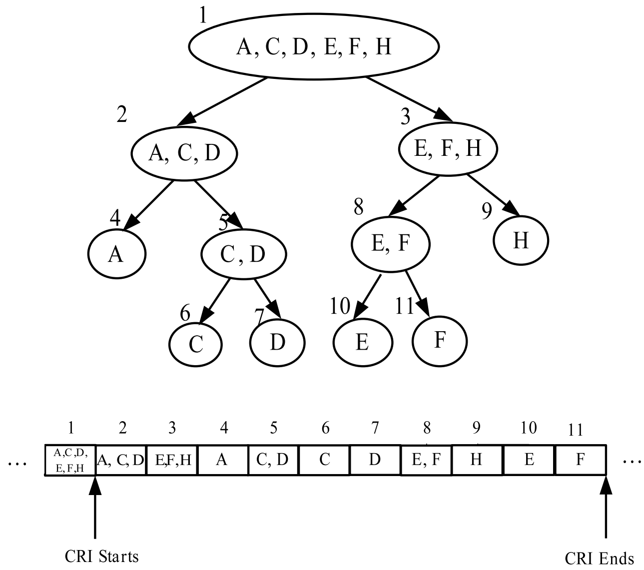


Figure 2.10: Serial Tree 6 active users involve in collision

In his work, Capetanakis also suggested other approach of basic tree algorithm by deploying random addressing scheme and using parallel tree traversal or breadth first search (BFS) tree. As in the serial tree search, the packets in every collision are retransmitted in two consecutive slots. However, the different from the serial tree is that users in the second group are not needed to wait until all users the first subgroup resolved. Instead, the second subgroup can retransmit their packet in the two consecutive slots. The tree algorithm with depth first search (DFS) is another form of collision resolution algorithm which is commonly adopted by researchers nowadays. In every collision, collided users are offered two slots for their retransmission as in the Capetanakis's model. However, the two slots are not paired. Since the feedback is assumed to arrive at the end of slots, the collided users in the first subgroup can directly be subdivided again and transmit in the next slot until all users succeed

Table 2.1: Addressing

User	Address
A	000
B	001
C	010
D	011
E	100
F	101
G	110
H	111
I	111

and then followed by the second group of the initial slot. The iterative algorithm of this DFS tree branching with random addressing proceeds by the following procedures:

1. All users in the collision collision are randomly divided in two subgroups with probability *p* for the first group and 1-*p* for joining the second group.

2. Let a set of users decide to join in the first group. This set of users immediately retransmits their packet in the first slot after the initial collision, while the set of users in the second group becomes inactive and eligible to send their packet in the one slot after.

3. If there is no user the current group, the current slot is idle or if there is one user transmit in the current slot, the transmission will be successful. The collision resolution then moves to the next group and repeat step 3

4. Otherwise, there will be more than one transmitting user. Collision will occur and the splitting process will repeat (i) in the next slot until all the users in the initial collision

succeed to transmit their packet.

The tree representation of BFS with random addressing is depicted in Fig.2.11. In this example, it needs 15 slots to resolve the contention between 6 users in the initial collision.

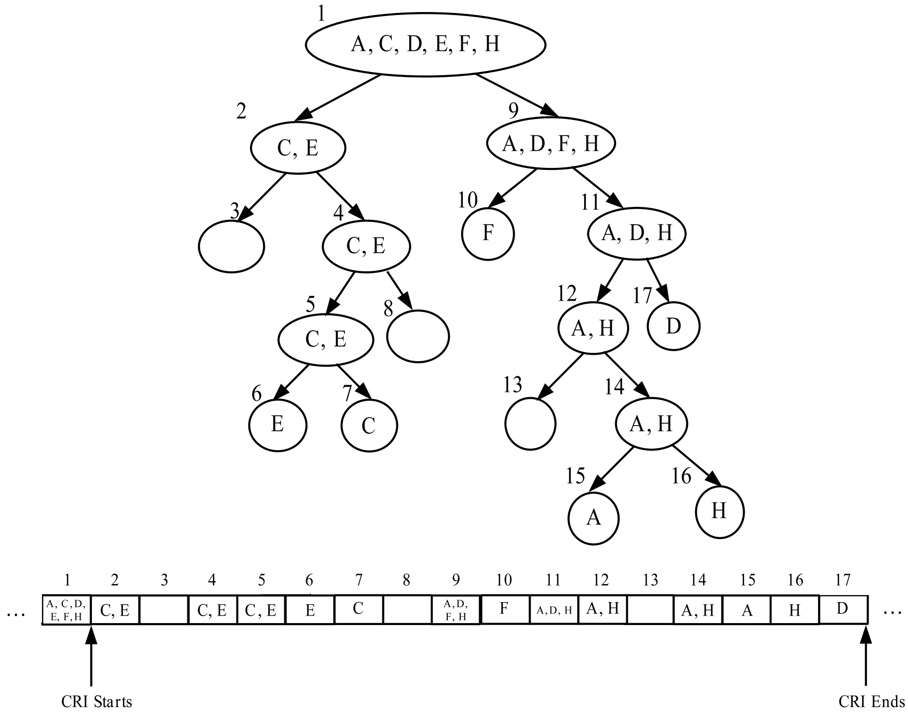


Figure 2.11: Basic DSF tree algorithm with random addressing scheme

The DFS tree algorithm is actually the proposed by Tsybakov and Mikhailov from Russia. They represent the idea of collision resolution algorithm concept in the form of stack such that the CRA of Tsybakov and Mikhailov is usually called as stack algorithm [32, 33]. Fig.2.12 depicts the stack representation of the example in the DSF tree algorithm according the following rules:

1. Users are not allowed to send their newly generated packet during the collision resolution time.

2. Only users in the stack level 0 are allowed to access the channel.

3. A collision occurs if at least two users in stack level 0. The collided users then flip a coin" 0,1" implying the following decision:

- users who choose ”0” will access the channel in the next time slot.
- users who choose ”1” will set their stack level to 1.
- the stack level of all other users will be incremented.

4. Otherwise, if no collision occurs, all users decrease their counter by 1

2.5.2 Dynamic Tree Algorithm

Beside the basic tree algorithm, Capetanakis in [23, 28] also designed the dynamic tree algorithm to minimize the expected access delay the basic tree algorithm based on the traffic in the previous CRI. The idea rose by considering the fact that there will be many users waiting during a CRI in the blocked access scheme. Since the algorithm allows all the waiting users to transmit their packet in the first slot of after the current CRI ends, collision which involves many users will most probably occur. Instead of just allowing them to access that first slot and resulting a collision which only wasting a slot, this step is skipped and directly spilt them into 2^K slots. Collision resolution then continues with the same rules of the binary tree algorithm as previously described. The term of dynamic is used since K may vary from one CRI to other CRIs depending on the traffic condition in the previous CRI. In so doing, the first slot after every CRI is not wasted and the intensity of collision in the beginning of CRI can be reduced. It is proved that the dynamic tree under infinite population with Poisson arrival can achieve maximum average throughput of 0.43 which is much higher than that of the basic tree algorithm. It is found that it is identical to the binary tree algorithm for low traffic load and is equivalent to TDMA in finite population [29].

2.5.3 Modified Binary Tree Algorithm

The modified tree algorithm is considered as a simple modification of the basic tree algorithm which proposed by Massey in [30] and independently by Tsybakov in [24]. The idea of this modification is that saving a timeslot because of definite collision. Considering

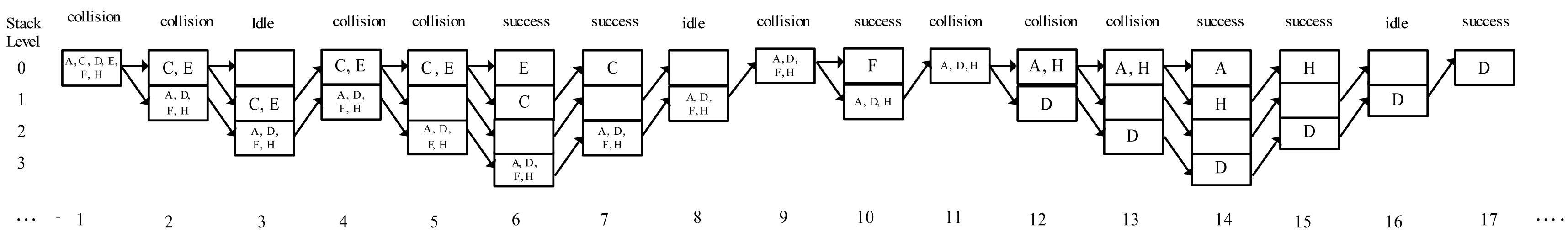


Figure 2.12: Tree algorithm in stack system with 6 active users in the initial collision

the fact that if after a collision, there is no transmission in the first timeslot after the splitting process, it will be sure that all collided users join the second group and their packet retransmissions result in collision in the second slot. Since this collision can be predicted, therefore, there is no purpose to allocate and finally just waste this timeslot. To reduce the slot wastage, the predictable collision the second slot can be skipped by directly splitting the collided users into two new subgroups pretending that the collision just occurs. Hence, this modification is also called as level skipping. In the representation of tree structure, this modification allows the tree algorithm to omit one branch of the tree whenever the event of an idle slot followed by a certain collision occurs in the splitting process.

The detail of this modified algorithm will be presented as follows. From the example in Fig.2.11, it can be noticed that collisions in slot 2 and slot 12 are followed by idle slots in slot 3 and slot 13, respectively. This can be sure that the transmissions in slot 4 and slot 14 results in collisions. Instead of just waste the slots for deterministic collisions, all users involved in collisions can be directly split into two subgroups. The level skipping mechanism can help reducing the number of slots needed to resolve the collision between 6 users from 17 slots to become only 15 slots as shown in Fig.2.13. It is proved that the modified tree algorithm provides better capacity from 0.346 of the basic tree algorithm to 0.375. However, this modification requires more informative feedback i.e. ternary feedback where the system has to be able to distinguish the ”no collision” into idle or successful transmission.

2.5.4 Collision Resolution Algorithm with Additional Information

Having introducing the concept of basic stack (tree) algorithm, Tsybakov continued his investigation on the collision resolution algorithm with new concept of recursive strategy for binary collision resolution algorithm when the multiplicity of every collision is known to all users [32]. Details of the recursive strategy are presented as follows: Suppose collision resolution commences in timeslot 1 since there are N users sending their packet simultaneously. In this protocol, newly generated packets are not allowed to access the channel directly if

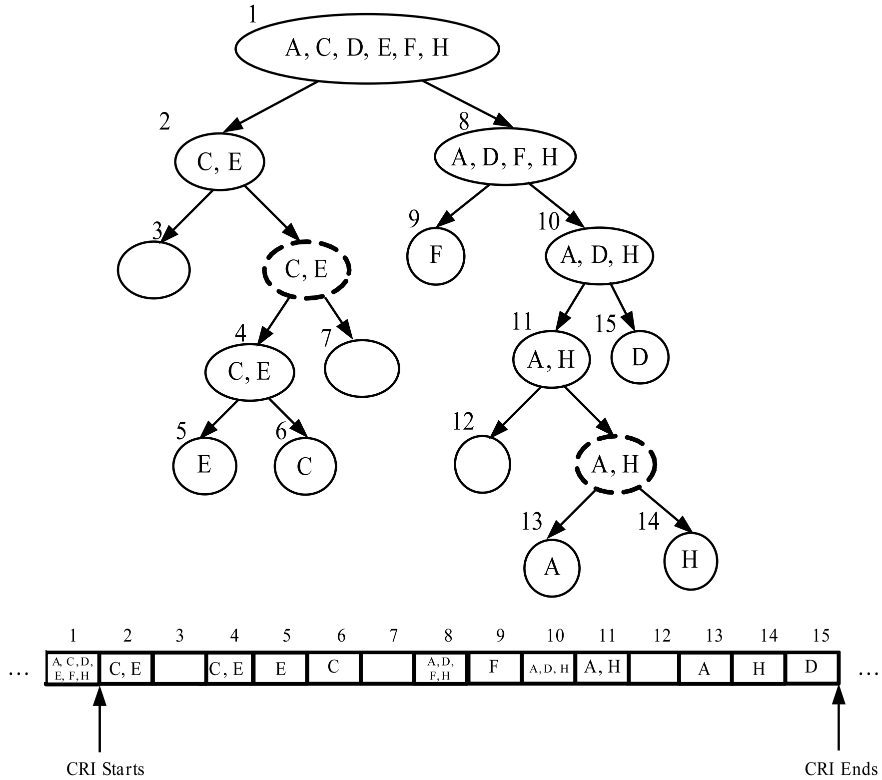


Figure 2.13: Modified tree algorithm with 6 collided users in the initial collision

collision resolution is in progress.

1. Users will retransmit their packets by choosing 1 with probability $p(n)$ or defer their retransmission by choosing 0 with probability 1-p(n).

2. Assume there are m users decide to retransmit their packets.

   - If $m = 0$ or $m = N$, the resolution will go to step 1
   - If $m \neq 0$ or $m \neq N$, this will results in new collision with m users involve in. The system will define a retransmission rule for this m collided users and then continue to step 3

3. Collision between users who postpone their retransmission in timeslot 1 (if any) is a collision of multiplicity $N - m$ and will be resolved by a strategy specified by the protocol. Then the initial collision is resolved.

In this work, Tsybakov also derive recursive expression for any number of N to obtain the average number of slot required to solve the particular number of collided packets (CRI length). If the probability of accessing a slot is set to be 0.5, he can find this average until N tends to infinity. Particularly for $N = 2$ and $N = 3$, it suggested a recursive strategy to find the average of CRI length by optimizing the probability of access depending on the number of packets collided. It is found that the average CRI length is 3 for $N = 2$ with $p(2) = 0.5$ and 4.787 for $N = 3$ with $p(3) = 0.412$. In addition to that with the recursive formula given in this paper, the bound of transmission delay is 1.876 when the number of collided users approaching to infinity.

Not long after Tsybakov work, in 1982 L. Georgadis proposed collision resolution with additional information (CRAI) protocol wherein energy detectors were utilized to provide the known multiplicity feedback [40]. This feedback is the same as considered in Tsybakov's work, however, Georgiadis claimed that his CRAI protocol provides tighter bounds of transmission delay. There are some differences in the rules defined by this CRAI protocol and the protocol of Tsybakov aforementioned. The details of CRAI protocol will be presented in the sequence.

This CRAI protocol has property of first come first serve (FCFS) where the packets generated earlier have priority to access the channel earlier compared to the latter generated packets. However, this FCFS property is only applied for determining the packets to be transmitted, not determining the order of the packets to be resolved in the collision resolution process. In other words, the collision resolution process follows the random addressing. Whenever system starts, the protocol interval parameter as in the Gallager algorithm which is denoted by Δ. This parameter is usually called as enable arrival where only the packets in this interval are allowed to access the channel during a slot instant. If there are N packets in this interval, collision occurs which is denoted by ε_N. By observing this result, the N users involving in the collision will again access the next timeslots with probability σ_N or postpone their retransmission to the one timeslot after the next timeslots with probability $1 - \sigma_N$. Suppose only i users access the next timeslot. At the end of the next timeslot, feedback arrives informing the users of the following possible conditions:

1. If $i = 0$ or $i = N$, the same procedure to resolve ε_N will be repeated in the next timeslot with probability σ_N since no one succeeds,

2. If $i = 1$, it means that 1 user retransmited its packet and succeeded. This also results in an event called as ε_{N-1} which will be resolve in the next timeslot. The remaining $N-1$ users involve in this event will contend the channel with probability σ_{N-1},

3. If $i = N-1$, it means that $N-1$ users attempted to retransmit their packets resulting in event ε_{N-1}. Apart from that, 1 remaining user did not access the current slot. This 1 remaining user will be given the right to access the next timeslot and the procedure of resolving the N-1 users (ε_{N-1}) will be done afterward where the users will retransmit their packet with probability σ_N,

4. For the conditions except in 1,2,3 where there will be two different conditions i.e. ε_i and ε_{N-i}, these two events will be then resolved separately. ε_i will be resolved first with σ_i followed the resolution of ε_{N-i} with σ_{N-i}. Then the initial collision is resolved and the system will allow the packets generated in the next enable arrival time to be transmitted. If collision occurs, the corresponding users have to follow the rules of collision resolution defined in 1-4.

In these collision resolution procedures, Georgadis searched the optimal choice for σ_i with dynamic programming methods. The optimal σ_i is set dynamically according to the number of collided packets such that it offers the least average number of timeslots for resolving collision with i packets. The final goal of this dynamic programming is to minimize the transmission delay by:

- forcing the event of ε_1 where only a packet transmit in the slot after collision,

- avoiding the event of idle slot.

Except for finding optimum access probability, this work also optimized the parameter Δ since it will determine the time interval wherein packets generated to be transmitted. The optimum value of this parameter is intended for resulting s uccessful transmission whenever it is set. If it is not set properly, it may cause severe collisions or just idle timeslots. H owever, the setting of this parameter depends on the packet arrival rate (λ).

In 1985, Eugene Gulko also proposed binary tree protocol with the assumption of known multiplicity feedback. This adopted blocked access mode as in Capetanakis's tree algorithm without defining enable arrival as in [40]. After a collision with N packets, the collided users will attempt to retransmit their packets with bias probability r_N in the first timeslot after collision or defer their transmission to the next timeslot with probability 1-r_N. In order to minimize the average delay and to fasten the users to obtain successful transmission, this protocol adopts depth first search tree. In the collision resolution process, this work presented two approaches called as left-node-first and smallest-multiplicity-first.

- Left Node First

  This tree traversal is exactly the Breadth First Search as explained before where the collision in the first group of subdivision will be resolved earlier. Here, this first group is denoted as the left node.

- Smallest multiplicity first

  After getting feedback from the transmission of the first group, it will be known how many user in the in the first group, as well as the number of users in the second group. The collision resolution in the next slot will be proceed to the group which has smallest number of collided users.

Essentially, this work is identical to that of L. Georgadis in [40] where r_N is set to be optimum and dynamically change depending on the number of users. However, this works offers a close form solution for the optimum probability as $r_N = \frac{2}{2+N}$. The value of r_N is very close to the value of σ_N, however, the average CRI length is not as precise as in [40], there is a small deviation between them. Noted that, for the case when number of collided users is large, this makes the resolution with Smallest Multiplicity First become the same as Left Node First since the bias setting of r_N which aims to obtain in successful transmission in the first group will mostly result in smaller number of users in the first subgroup. These three identical collision resolution protocols can achieve a stable condition if the packet arrival rate is less than 0.532 packet per time slot.

2.5.5 Basic and Modified Q-ary Tree algorithm

The Q-ary tree algorithm is considered as the generalization of tree collision resolution algorithm. This general algorithm was first analyzed by Mathys and Flajolet [45] even though the initial idea was introduced by Tsybakov and Mikhailov in [24] without any analysis. Two access protocols i.e. blocked and free access are also applied to complete the analysis.As already known that the basic tree algorithm split the collided users in each collision into always two subsets. Instead, this algorithm split the collided users into always Q (splitting parameter) subsets where Q is greater than two. With more than two subgroup in the splitting process, this idea aims to reduce the probability of collision such that an improvement can be achieved. Fig.2.14 shows an example of Q-ary tree algorithm where Q=3 (also called ternary tree). In this example, it needs 18 timeslots to resolve 9 users in the initial collision. In their analysis, Mathys and Flajolet included the case of the basic binary tree algorithm i.e. Q=2 to compare the performance and observe how the splitting parameter can affect the system performance. It is found that the optimum performance is achieved when the splitting parameter, Q, is set to 3 or commonly known as ternary tree algorithm for both free access and blocked access modes. The maximum stable throughput offer by this ternary tree algorithm is 0.366 and 0.401 for blocked access and free access, respectively [24, 45].

The idea of level skipping in modified tree algorithm is also applied in this analysis which is called modified Q-ary tree algorithm. As in the basic modified tree algorithm, one slot can be saved due to definite collision. This can occur in every splitting process if there is no users retransmit their packet in the first Q-1 timeslots. This means that all collided users decide to access in time slot Q which can just be skipped. One different aspect from this study that is not be applied in the Q-ary tree algorithm is that the using of bias probability. The level skipping mechanism will allow the system to user bias probability to improve the performance. It is shown that the using of bias probability provides an improvement compared to that of modified Q-ary tree algorithm with equal probability for both blocked and free access protocols. shows the performance of this modified Q-ary tree algorithm achieve MST at 0.4069 and 0.4076 with equal probability and with bias probability, respectively. These practical optimum performance are obtained by ternary modified tree algorithm when free access algorithm is applied. On the other hand, the optimum performance of blocked access algorithm is achieved by binary tree algorithm by obtaining MST of 0.375 with equal probability and 0.381 with bias probability [45].

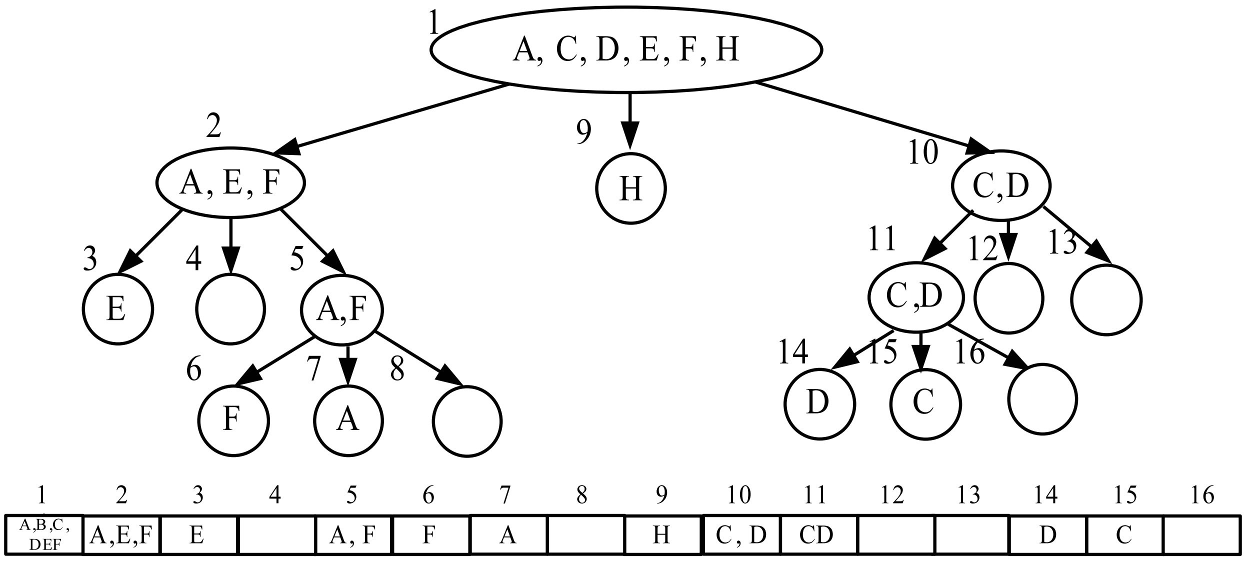


Figure 2.14: Ternary tree algorithm with 6 users in the initial collision

2.5.6 First Come First Serve Splitting Algorithm

The first come first serve splitting algorithm devised by Gallager with the same approach as in the basic tree algorithm i.e. splitting a conflict into smaller groups recursively until all the users succeed [25, 27]. It offers maximum stable throughput of 0.487 which is higher than that of the basic tree algorithm. This algorithm split the collided packet based on the arrival time of the packets with strict first come first serve order. In a given interval time of arrival, a group of users transmit packets in the beginning a slot. If this transmission results in a collision, the retransmission of the colliding packet is described in the following mechanism. The particular interval related to collision is divided into two smaller intervals, say, first and second halves. A set of users in the first half of interval is entitled to transmit packet first and a further division of arrival time by factor of two is applied if collisions still occur. Since this algorithm obtains the maximum stable throughput of 0.487 and the successful retransmission follows the order of first come first serve, this algorithm is called as FCFS 0.487 algorithm. The detail of this algorithm is presented in the sequel. Every packet that joins a contention has its own time stamp of its arrival. The algorithm defines an enable interval (*EI*) where all the packets arrive in this period can be transmitted. Every user has to keep track the current *EI* denoted by x and conflict resolution interval. In the beginning of a contention, *EI* is specified from zero to x, each user who has packet ready for transmission in this interval transmits his packet in a given slot. If the feedback informs that there is no collision, the *EI* is updated to become x to $2x$. However, if the feedback is collision, the algorithm up dates the EI to become half of the previous *EI* (from 0 to $x/2$). The system will then ask the retransmission of the packets whose arrival time lie in the new interval. If it results in successful transmission, the set of packets in the second half interval (between $x/2$ and x) is transmitted in the subsequent slot. In case of further collision, the EI will also be further divided until it contains exactly one packet to obtain a successful transmission. In another case, if there is no packet lying between 0 and $x/2$, it is definitely collision in the second half interval so this interval is then directly divided into two smaller intervals for the next transmission instead of just allocating a slot for assured collision. Some studies had also been conducted based on the FCFS algorithm by employing different types of feedback; variation binary feedback (*S/N, C/NC*) [36]. This study was extended to the case of multiple reception protocol with binary and ternary feedback [38]. Slightly different with the Gallager's, instead of only divide the collided *EI* into new *EI*s, these protocols determine several parameters to shrink or lengthen the enable interval. These parameters can be adjusted according to the traffic condition to obtain the optimum performance.

In the presented collision resolution algorithms, it is assumed that feedback reaches to the users without error. The study of collision resolution algorithms with errors can be found in [57,58].

2.5.7 Mean CRI Lentgh of The Q-ary tree algorithm

In this subsection, the mean access delay of basic Q-ary tree algorithms derived in [27,45] will be presented. The mean access delay we refer to here is the average CRI length (average number of slot required to resolve N packet in the initial collision) which is denoted by $L(N)$. Based on the description of the basic Q-ary tree algorithm, every collision will be subdivided into Q subsets. Each subset will have i_q users, where $q = 1,2,\cdots,Q$ and $\sum_{q=1}^{Q} i_q = N$. If $i_q \geq 2$, it means that a new collision occurs and then will be subdivided again into Q new subgroups. Subgroups after the subdivision process are identically distributed random variables but they are not independent. Denoted $L(N)$ be the expected length of a CRI with N collided users.

Let define

$$\begin{aligned} L(N) &= 1+\sum_{q=1}^{Q}\sum_{i_q=0}^{N} P(i_q|N)L(i_q) \\ &= 1+\sum_{q=1}^{Q}\sum_{i_q=0}^{N} \binom{N}{i_q}(p_q)^{i_q}(1-p_q)^{N-i_q}L(i_q) \end{aligned} \quad (2.14)$$

For the case when equal probability is applied for users to picking a slot where $p_1 = p_2 = \cdots = p_Q = \frac{1}{Q}$, the distribution of users in every slot is the same and the combination of the users accessing every slot is equally likely. Hence equation (2.14) can be reduced to:

$$\begin{aligned} L(N) &= 1+Q\sum_{i_q=0}^{N}\binom{N}{i_q}(\frac{1}{Q})^{i_q}(1-\frac{1}{Q})^{N-i_q}L(i_q) \\ &= 1+Q\sum_{i_q=0}^{N}\binom{N}{i_q}(\frac{1}{Q})^{i_q}(\frac{(Q-1)^{N-i_q}}{Q^{N-i_q}}L(i_q) \\ &= 1+Q\sum_{i_q=0}^{N}\binom{N}{i_q}(\frac{(Q-1)^{N-i_q}}{Q^{N}}L(i_q) \\ &= 1+Q\sum_{i_q=1}^{N-1}\binom{N}{i_q}(\frac{(Q-1)^{N-i_q}}{Q^{N}}L(i_q)+Q\binom{N}{N}(\frac{(Q-1)^{0}}{Q^{N}}L(N) \\ &= 1+Q\sum_{i_q=1}^{N-1}\binom{N}{i_q}(\frac{(Q-1)^{N-i_q}}{Q^{N}}L(i_q)+(\frac{1}{Q^{N-1}})L(N) \\ &= \frac{1}{(1-\frac{1}{Q^{N-1}})}(1+Q\sum_{i_q=1}^{N-1}\binom{N}{i_q}(\frac{(Q-1)^{N-i_q}}{Q^{N}}L(i_q))) \end{aligned}$$

(2.15)

CHAPTER III

Mean CRI length of Framed Slotted Aloha

In this chapter, we consider the modification of the Framed Slotted ALOHA when ternary and known multiplicity feedbacks are available. We also consider that the frame structure will only be initiated whenever a collision occur. Mean CRI length is our main performance metric for the analysis.

3.1 Introduction

As a variation of slotted Aloha, the framed slotted Aloha utilizes slotted channel as described in Chapter 2. We consider the frame slotted aloha which allows the feedback returns to the users at the end of each slots rather than at the end of each frame. Based on the different amount of feedback information, further modification on the retransmission mechanism of the colliding packet will be applied. Other than the binary feedback, it is assumed that another type of feedback such as ternary feedback and known multiplicity feedback is available in the system [40]. With these feedback, we will show the effect of feedback utilization on the performance of frame slotted Aloha.

3.2 System Model

Consider a system in which a number of users contend for access over a shared channel. Each user independently generates packets according to a specific random distribution, such as Poisson. Packet transmission results are returned to each station at the end of each slot in the form of feedback information. Type of feedback that can be assumed includes binary, ternary and multiplicity feedback.

The transmission of new arrival packets is distinguished from the retransmission of the collided packets. Transmission attempts for new arrival packets, namely the packets generated during the resolution of the previous initial collision, are considered to get delayed until the whole resolution process complete. However, if there is no collision resolution process, the new arrival packets are sent immediately in the next slot. This access mechanism is called as Blocked Access [45]. Whenever a new collision occurs, the frame structure which consists of Q slots ($Q \geq 2$) is initiated. Depending on the feedback available in the system, the retransmission rule will change. This results in a classification of the proposed modification of the framed slotted Aloha as follows.

3.3 Access Protocol Description

3.3.1 Framed SLotted Aloha with Ternary Feedback

Considering ternary feedback, system can inform the user whether a slot is either empty slot or success slot when there is no collision . If this feedback is available, a strategy referred to here as skipping strategy type I is applied [45]. During the packet retransmission, a definite collision may occur if there is no packet transmitted in the first $Q-1$ slots of the frame. Each station is aware of the definite collision in Slot Q and hence skips this slot and starts a new contention frame immediately. Fig. 3.1 illustrates the framed slotted Aloha with the skipping strategy I where the frame length,Q, is set to be 3. In this example, 5 users (A, B, C, D, E) collide in Slot 1 which is nomally called as the initial collision. These 5 users then transmit in the three consecutive slots in frame 1 at random. Suppose that no user transmit in Slot 2 and 3. Surely, all the collided users transmit in the last slot in this frame which is Slot 4. A new contention frame then starts where user A, D, and E experience a new collision in Slot 5, while B and C get successfully transmit their packet in Slot 6 and 7, consecutively. The resolution continues for resolving the collision involving user A, D and E in a new available frame. User D transmits in Slot 8, Slot 9 is empty and a collision occurs in Slot 10 with user A and E involve in it. These two users then again have to choose one slot in the next frame for their packet transmission. Finally, the collision is resolved where user A and E succeed in Slot 11 and 12, respectively. Slot 13 remains idle since all users have succeeded to transmit their packets. For the example in 3.1, the definite collision occurs in Slot 4 since Slots 2 and 3 are idle. By skipping this last slot, the resolution time is shortened by one slot from 13 slots to 12 slots.

Figure 3.1: Illustration of the frame Slotted Aloha with skipping strategy type I

3.3.2 Framed Slotted Aloha with Known Multiplicity Feedback

By knowing the number of collision multiplicities as contained in the known multiplicity feedback, it is possible to apply a strategy namely skipping strategy type II and dynamic frame size strategy. Beside definite collisions, definite idle slots can be eliminated to reduce the resolution time. Since the feedback informs the exact number of users accessing each slot, users are able to know that the re is always a collision in the last slot of the frame, when at least two users are waiting for retransmission. This slot is then skipped and the resolution continues in the new frame. In the other case, if all users in the collision access the first view slots in the frame, the users skip the remaining slots and begin a new frame. For the example in Fig. 3.2, Slot 4 and 10 where definite collisions occurred can be eliminated. Also, Slot 13 can be skipped since no user definitely accesses these slots. By saving 3 slots, the resolution time for this collision can be reduced to 10 slots.

As in the dynamic frame slotted Aloha which varies the frame size based on the traffic estimation, the perfect knowledge of the colliison multiplicity allows us to gain the optimum strategy. This mean that maximizing the network throughput by setting the frame size as exactly as the number of collided users given by the known multiplicity feeback. Later, we will then apply the skipping strategy type II when the frame size is set as the same as the number of users. It is expected that this combination will result in a significant improvement to the framed slotted Aloha.

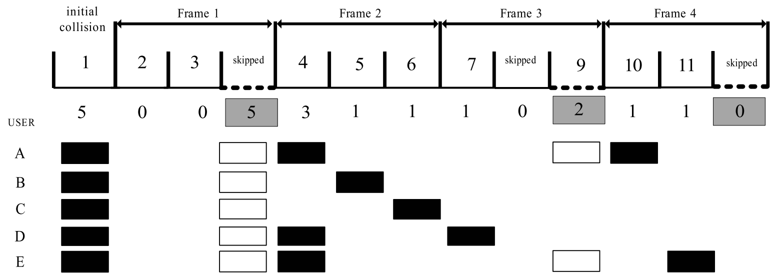


Figure 3.2: Illustration of the frame slotted Aloha with skipping strategy type II

3.4 Analysis

The delay performance of the frame slotted Aloha and its modifications will be presented in this section. We are interested in the mean CRI length which represents the average number of slots required to resolve the contention among a number of users which initially collide [45]. To begin with, we will define the following parameters:

N denotes the number of packets initially collides

Q denotes the number of slots in a frame

i_q are the number of users in Slot q of each frame, where $q = 1,2,..,Q$ and $\sum_{q=1}^{Q} i_q = N$

p_q is the probability of accessing each slot in the frame, where $p_q = 1/Q$

n_f is the random variable of the total number of users who collide after divided into the slots in frame with N users in the initial collision; $n_f = \sum_{q=1, i_q \geq 2}^{Q} i_q$

$L(N)$ is the mean CRI length for N users including the initial collision slot where the frame size is Q

The framed slotted Aloha starts resolving N collided users where these N users is randomly divided into Q subgroups for each frame and the remaining unsuccessful users of this particular frame will be divided again into Q subgroups of the next frame. Now, conditionally on the event Q, for $1 \leq q \leq Q$, an item is sent into the q^{th} subgroup with probability p_q. If I_q is the cardinality of the q^{th} slot, then, conditionally on the event Q, the distribution of number of users in the slots of the frame of the vector $(I_1, ..., I_Q)$ is multinomial with parameter N and $p_1, \cdots p_Q$ which can be given by:

$$Pr(I_i = i_1, I_2 = i_2, \cdots, I_Q = i_Q | N = n) = \frac{N!}{i_1 i_2 i \cdots i_Q} p_1{}^{i_1} p_2{}^{i_2} \cdots p_Q{}^{i_Q} \tag{3.1}$$

The algorithm will stop if there is no more collision i.e $I_q = 0$ or $I_q = 1$. However, if collision occurs in at least one slot i.e $I_q \geq 2$, there will be a new subdivision process which has the same distribution as in 3.1. As previously defined, CRI is the total number of slots for resolving a collision including the initial condition. For example, the CRI is $Q+1$ slots if a collision is resolved in the first frame and $2Q+1$ if it resolved in the second frame, and so forth. It implies that Q slots will be used for each frame when collision can be resolved in one frame. Otherwise, it will need additional slots in the next frame until all the users succeed. It means that to find the CRI length for larger number of users needs to know the mean CRI length of the smaller number of users which is known as recursive form. For the initial condition, let define $L(0) = L(1) = 0$, indicating the condition where there is no packet and exactly one packet transmitted in a slot, respectively. For the case when $N \geq 2$, $L(N)$ can be expressed as:

$$L(N) = 1 + \sum_{i_1, i_2, \ldots, i_Q} \binom{N}{i_1, i_2, \cdots, i_Q} p_1{}^{i_1} p_2{}^{i_2} \cdots p_Q{}^{i_Q} (Q + L(n_f) - 1) \tag{3.2}$$

where the term "1" is the slot where the initial collision occurs. If "1" is replaced by Q, the formulation will result in the same results as that of classical frame Aloha which always has frame structure [22]. and $\sum i_1, i_2, \ldots i_Q \binom{N}{i_1, i_2, \ldots i_Q} = \frac{N!}{i_1! i_2! \cdots, i_Q!}$ is the multinomial coefficient summing up all the possible event in each frame. $Q + L(n_f) - 1$ implies that there are Q slots used for each frame and there will be $L(n_f) - 1$ will be needed if collision still occurs. The folowing are examples of calculation of the mean CRI length for $N = 2$ and 3 where the frame size, Q, is set to 3:

- when there are two users colide

$$
\begin{aligned}
L(2) = 1 &+ \binom{2}{2\,0\,0}\left(\frac{1}{3}\right)^2\left(\frac{1}{3}\right)^0\left(\frac{1}{3}\right)^0(L(2)+L(0)+L(0)) \\
&+ \binom{2}{0\,2\,0}\left(\frac{1}{3}\right)^0\left(\frac{1}{3}\right)^2\left(\frac{1}{3}\right)^0(L(0)+L(2)+L(0)) \\
&+ \binom{2}{0\,0\,2}\left(\frac{1}{3}\right)^0\left(\frac{1}{3}\right)^2\left(\frac{1}{3}\right)^0(L(0)+L(0)+L(2)) \\
&+ \binom{2}{1\,1\,0}\left(\frac{1}{3}\right)^1\left(\frac{1}{3}\right)^1\left(\frac{1}{3}\right)^0(L(1)+L(1)+L(0)) \\
&+ \binom{2}{1\,0\,1}\left(\frac{1}{3}\right)^1\left(\frac{1}{3}\right)^0\left(\frac{1}{3}\right)^1(L(1)+L(0)+L(1)) \\
&+ \binom{2}{0\,1\,1}\left(\frac{1}{3}\right)^0\left(\frac{1}{3}\right)^1\left(\frac{1}{3}\right)^1(L(0)+L(1)+L(1)) \\
&= 1 + \frac{6}{9}3 + \frac{3}{9}(L(2)+2) \\
&= \frac{33}{6} \\
&= 5.5 \text{ slots}
\end{aligned}
$$

$$
\begin{aligned}
L(3) = 1 &+ \binom{3}{1\ 1\ 1}\left(\frac{1}{3}\right)^1\left(\frac{1}{3}\right)^1\left(\frac{1}{3}\right)^1 (L(1)+L(1)+L(1)) \\
&+ \binom{3}{2\ 1\ 0}\left(\frac{1}{3}\right)^2\left(\frac{1}{3}\right)^1\left(\frac{1}{3}\right)^0 (L(2)+L(1)+L(0)) \\
&= 1+\frac{6}{9}3+\frac{3}{9}(L_2+2) \\
&+ \binom{3}{3\ 0\ 0}\left(\frac{1}{3}\right)^3\left(\frac{1}{3}\right)^0\left(\frac{1}{3}\right)^0 (L(3)+L(0)+L(0)) \\
&= 1+\frac{6}{27}3+\frac{18}{27}(L(2)+2)+\frac{3}{27}(L(3)+2) \\
&= \frac{186}{24} \\
&= 7.75 \text{ slots}
\end{aligned}
$$

For the other value of collided users, N and frame size, Q, each modification of the Frame Aloha can be derived in a recursive form as presented in the sequel.

3.4.1 Mean CRI length of framed slotted Aloha with skipping strategy type I

As for the skipping strategy type I [45], the skipped slot is the definite collision when all users access the last slot of the frame. The probability that all users access in slot Q can be expressed as:

$$
\begin{aligned}
\Pr(i_Q = Q) &= \binom{N}{N} {p_Q}^N (1-p_Q)^{(N-N)} \\
&= {p_Q}^N
\end{aligned}
\tag{3.3}
$$

The average number of skipped slot is also ${p_Q}^N$. Then, the mean CRI length is given by:

$$
L(N) = 1 - {p_Q}^N + \sum_{i_1, i_2 \cdots i_Q} \binom{N}{i_1, \cdots i_Q} (Q + L(n_f) - 1) \tag{3.4}
$$

3.4.2 Mean CRI length of framed slotted Aloha with skipping strategy type II

As presented in the previous section, the known multiplicity feedback can be utilized to form skipping strategy type II. The concept of slot skipping is to reduce the cost for solving the collision by eliminating idle and colliding slots in the resolution. However, the conflict to be resolved remains the same. The idle and colliding slots are the slots that are predicted to be idle and collision, respectively. Let denote the average number of predictable collided and idle slots as $C(N)$ and $I(N)$, respectively. The followings are the derivation of $C(N)$ and $I(N)$.

- Expected number of skipped idle slots, $I(N)$.

  When all users have accessed the first j slots (where $j \leq Q-1$), it can be predicted that the last $Q-j$ slots will be idle. It mean that the predictable idle slots can occur in slot 2 to slot Q. We first compute the probability of idle for one slot (denoted by slot j) with the following binomial distribution.

$$\begin{aligned} P[i_j = 0] &= \binom{N}{0} p_j^0 (1-p_j)^N \\ &= (1-p_j)^N \end{aligned} \tag{3.5}$$

  The probability that last j slots from Q slots, where $j = 1, 2, \cdots, Q-1$, are idle is given by:

$$P[i_j = \cdots = i_Q = 0] = (1-p_j)^N - (1-p_{(j-1)})^N \tag{3.6}$$

  The expectation of skipped idle in slot j to slot Q (where $j = 2, 3, , Q$), $I(N)$, can be expressed as:

$$I(N) = 1(1-(p_1)^N - (1-(p_1+p_2))^N + 2((1-(p_1+p_2))^N \tag{3.7}$$

$$- (1-(p_1+p_2+p_3))^N) + \cdots + (Q-1)((1-(p_1+p_2+\cdots p_{Q-1}))^N$$

$$- (1-(p_1+p_2+\cdots p_Q))^N)$$

$$= \sum_{i=1}^{Q-1} \left(\sum_{j=1}^{i} p_j \right)^N \tag{3.8}$$

since $p_1 = p_2 = \cdots = p_Q = \frac{1}{Q}$, equation (3.7) is reduced to:

$$I(N) = \frac{1}{Q^N} \sum_{i=1}^{Q-1} i^N \tag{3.9}$$

- Expected number of skipped collision slots

As aforementioned, the predictable collisions always occur in slot Q. To find the expression of $C(N)$, let $E[C]$ be the probability of collision in slot Q when k out of N users access this particular slot, where $k \geq 2$. This can be given by:

$$E[C] = \binom{N}{k} (p_Q)^k (1-p_Q)^{n-k} \tag{3.10}$$

Then, summation of equation (3.4.2) for all value of k yields to the average number of skipped collision in the resolution interval which can be expressed as:

$$C(N) = \sum_{k=2}^{N} \binom{N}{k} (p_Q)^k (1-p_Q)^{N-k} \tag{3.11}$$

Equation (3.11) can also be expressed by subtracting the average number of all events with the average number of success and idle in slot Q as:

$$C(N) = 1 - (1-p_Q)^{N-1} (1 + p_Q (N-1)) \tag{3.12}$$

Hence, the mean CRI length is given by:

$$\begin{aligned} L(N) = p^N \sum_{i=1}^{N} i^N - ((1-p)^{(N-1)}(1+p(N-1)) \\ + \sum_{i_1,i_2,\cdots,i_Q} \binom{N}{i_1,i_2,\cdots i_Q}(Q+L(n_f)-1) \end{aligned} \quad (3.13)$$

For the case of dynamic frame Aloha i.e. Q is chosen to be equal to N with the skipping strategy type II, the formulation of $L(N)$ in (3.4.2) can be obtained by replacing Q by N.

3.5 Numerical Results and Discussions

Fig.3.3 shows the mean CRI length of blocked access basic framed slotted Aloha versus the number of users in the initial collision, N. We vary the number of collided users from 2 to 30. To investigate the effect of the frame size, we vary the number of slots in the frame to be 2, 4, 6, 8, and 10. It shows that the mean access delay exponentially increases with the number of collided users. This is because a further increase of the number of users will generate more collisions and results in an increase of CRI length. The exponential growth is very clear when the frame used size is small, (e.g., $Q = 2$). Increasing the frame size certainly improves the performance for large number of collided users but the exponential trend remains. However, it is shown that the curves overlap each other for all cases when the number of initial collided users is small. This means that the smaller frame is more effective for resolving small number of users.

When ternary feedback is available, the skipping strategy type I only gives a very small improvement to the basic Frame Aloha. With the same frame size, this skipping strategy can only save not more than one slot on average. For example, for resolving 20 collided users with frame size = 4, the basic modification needs 242.3 slots whereas the modification with skipping strategy type I needs 241.4 slots on average. It means that the improvement is only 0.9 slots on average. These results are very different from the case of tree algorithm where

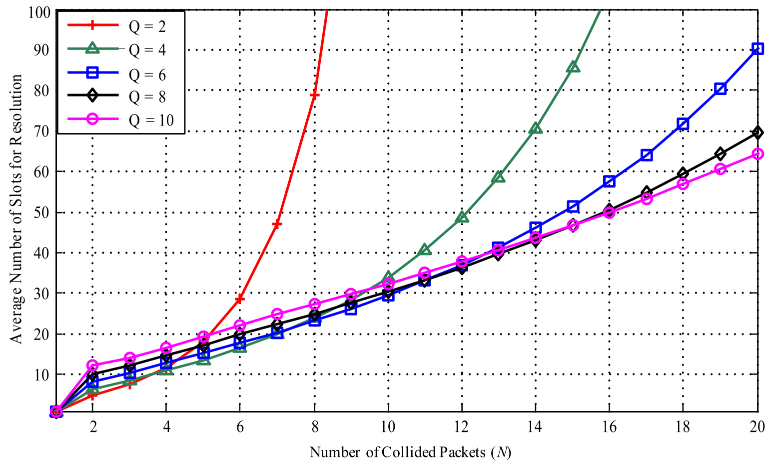


Figure 3.3: The mean CRI lentgh of the basic framed slotted Aloha

the improvement can be seen clearly. This is because of the fact that the frme slotted ALoha does not have feature of splitting the collided users such that the collision resolution time is long and the skipping slots do not give effect on it.

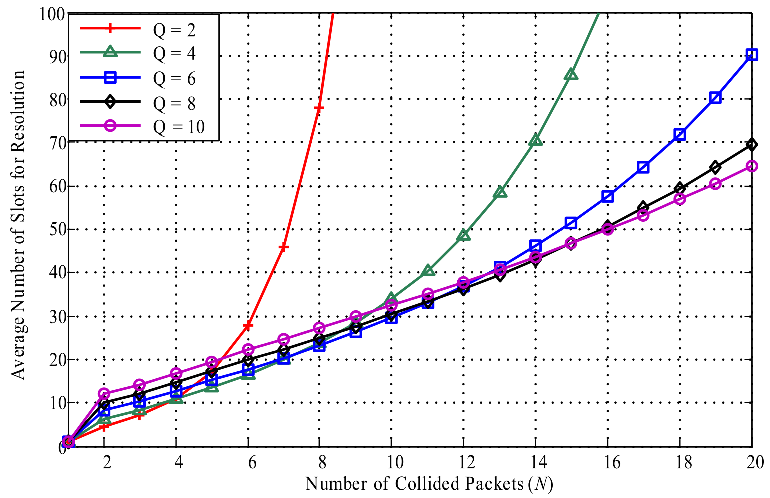


Figure 3.4: The mean CRI lentgh of the framed slotted Aloha with Skipping Type I

Compared to the skipping strategy type I, the skipping strategy type II offers a significant improvement. For the same example ($N = 20$ and $Q = 4$), 186.6 slots are required meaning that this strategy can reduce 55.7 slots for the resolution. When this skipping strategy is applied with the changing the frame size according to the number of users, it needs 47.54 slots on average to resolve 20 users. This means that this strategy is very effective. In addition to that, this strategy can overcome the exponential growth problem when the frame size is set to be constant. These results reveal that many strategies can be created to obtain the fast collision resolution time depending on the type of feedback available by the system. It is shown that the more informative feedback will be useful to improve the delay performance if the feedback is utilized appropriately.

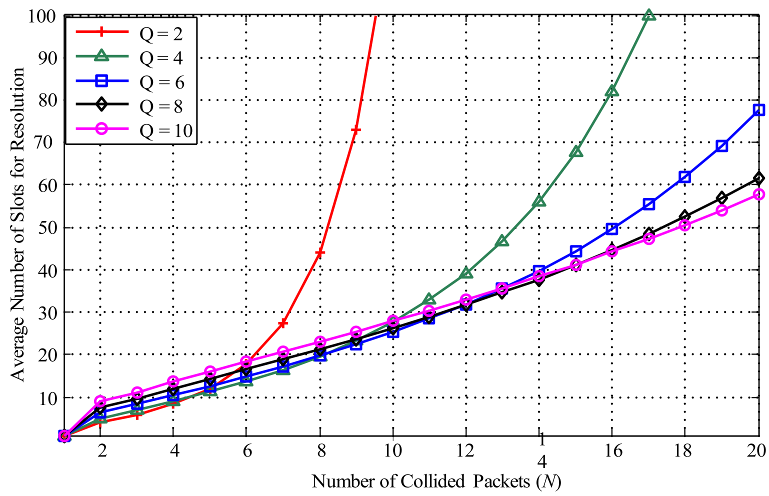


Figure 3.5: The mean CRI lentgh of the framed slotted Aloha with Skipping Type II

As discussed in some other studies [15–21], the dynamic frame Aloha is very effective compared to the fixed value of frame length. With the known multiplicity feedback which offers the exact collision multiplicities (not an estimation), the curve in Fig. 3.6 shows the optimal results for the dynamic frame Aloha. For comparison, with the skipping type I, it can resolve the collision in 52.03 slots for 20 collided users. The effectiveness of the dynamic frame Aloha is clearly seen that it dramatically improves the performance of the traditional frame Aloha and frame Aloha with skipping type I. It can also be seen that it overcomes the exponential growth problem. A more improvement can be further achieved when the dynamic frame strategy is applied together with the skipping strategy type II. The previous setting of frame size i.e. $Q = N$ no longer provides optimum performance of this combination strategy. The better results can be obtained if

$$Q = \begin{cases} 2, & N = 2, 3. \\ N - 1, & N \geq 4. \end{cases} \tag{3.14}$$

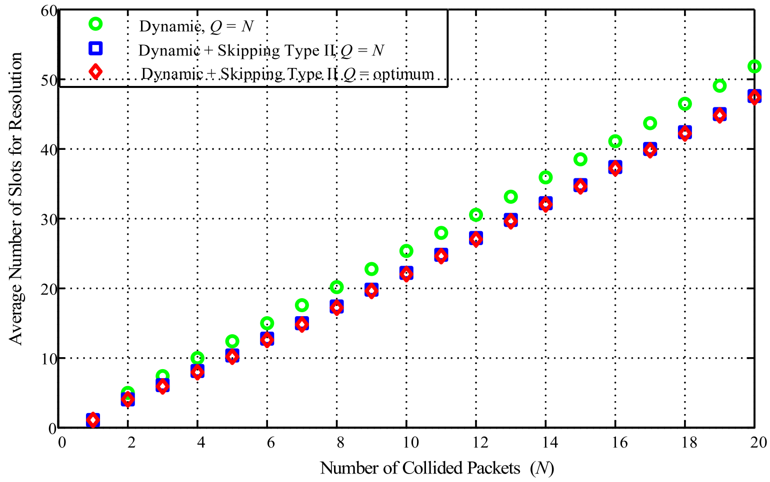


Figure 3.6: The mean CRI lentgh of all the proposed modifications of the Frame Aloha

As already shown that the dynamic frame size alone will reduce the average delay and remove the exponential trend of of the frame Aloha. Adding the skipping type II to the dynamic frame Aloha results in a more improvement as expected since the unnecessary idle and collision slots are reduced.

CHAPTER IV

Mean CRI length and Throughput of Adaptive and Skip Tree Algorithm

In this chapter we first present the proposed scenario for the tree algorithm with the known multiplicity feedback namely adaptive tree algorithm and skip tree algorithm. We then provide the analysis of the delay performance of each algorithm in terms of mean collision resolution interval (CRI) length.

4.1 Introduction

As in the existing tree algorithm, the channel considered here is the idealized slotted channel as presented in Chapter2 [5, 7, 10, 44]. Many variations of the tree algorithm have been comprehensively studied under the assumption of binary feedback and ternary feedback [23, 28–30, 45, 56]. The considered here is the known multiplicity feedback which provides the number of users contending in each slot will return at the end of each slot. There is infinite number of independent users generate their packets which is modeled as Poisson traffic. With this traffic assumption, we apply both blocked access and free access algorithms.

The first mechanism we will introduce is adaptively change the splitting factor for each collision. This is different from the existing tree algorithms which always use static the splitting factor. The second mechanism includes a new concept of level skipping which is more effective than that of the modified tree algorithm. We shall now present the description of the two proposed mechanisms in the sequel.

4.2 Description of the Algorithm

4.2.1 Adaptive Q-ary Tree Algorithm

In the first proposed mechanism, the number of subgroups changes adaptively according to the number of users involved in collisions since the number of users in every collision is detected by the known multiplicity feedback. This mechanism aims to minimize the probability of repeated collisions and hence maximize the probability of successful transmission in each slot. This also means that the number of slots used to resolve collisions can be reduced.

For the following two reasons, this first mechanism becomes our concern. The first reason is that with static Q, probability of collisions is high when Q is set to be a small value for resolving large number of collided users is large. In addition, the number of idle slots is large if a large value of Q is used to resolve a small number of users. The second reason is based on the dynamic tree algorithm by Capetanakis which shows that dynamically change the splitting factor only in the initial collision is more effective compared to the binary splitting [28]. From these facts, we shall show that the fixed splitting factor is not optimum for resolving every number of users. Instead, there is appropriate splitting factor for each number of collided users to minimize the mean number of slots to resolve a collision and improve the throughput. In this thesis, we change the splitting factor depending on the number of collided users which is referred to here as adaptive Q-ary tree algorithm.

In a simplest way, the proposed the number of subgroups of every collision is set to be exactly equal to the number of colliding packets. The operation of the adaptive Q-ary tree algorithm can be explained as follows. As depicted in Fig. 4.1, there are 5 packets initially collided in the first slot, and then corresponding users will be split into 5 subgroups. This results in a new collision in the fourth subgroup with 3 packets which will be again split into 3 new subgroups. Further collision occurs since 2 users pick the second subgroup, then this collision will be resolve in 2 new subgroups. However, it is expected that the probability of idle slots is high when the number of collided users is too large. It implies that there is an optimum splitting factor for every number of collided users. We then suggest another form of adaptive Q-ary tree algorithm where the optimal choice of the number of subgroups denoted as Q_N^o for each collision is set to the value that gives the minimum average number of slots for the resolution. Further investigation will be needed to find this optimum which provides the minimum CRI length for the resolution of each N colliding users.

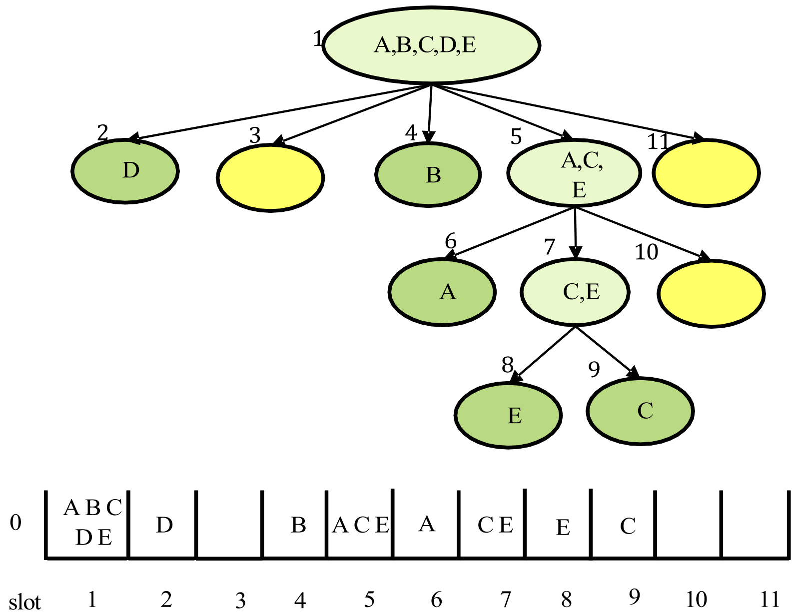


Figure 4.1: Illustration of Adaptive Q-ary tree altoritHm

4.2.2 Skip Q-ary Tree Algorithm

As introduced in the modified tree algorithm by Massey [30] and Mathys [45], an improvement to the Q-ary tree algorithm is obtained by level skipping which utilizing ternary feedback. Inspired by this slot skipping mechanism, the information contained in the known multiplicity feedback can be utilized to form a more effective mechanism of level skipping when it is available. This slot skipping mechanism is expected to much more effective than that of existing modified tree algorithm. This slot skippng mechanism is the same as the slot skipping Type II applied in the frame soltted Aloha in Chapter 3.

As the exact number of users accessing in each slot is known, it is possible for each user to utilize the feedback in order to keep track of the number of the users who have transmitted their packets in each slot. This enables the system to predict definite collision and idle slots which can further be eliminated. Predictable collisions occur when the number of users expected to access the last slot of the frame is more than one. Meanwhile, Q-q slots will be idle and then can be skipped when all users in the initial collision have accessed in the first Q slots of a frame (where $q=1, 2, \ldots, Q-1$). In so doing, it is expected that delay performance of the tree algorithm can be improved. Fig. 3 depicts the collision resolution of skip ternary tree algorithm with 9 collided users. The basic ternary tree algorithm needs 13 slots to resolve them. When ternary tree algorithm with level skipping is applied, only 10 slots are required meaning that it saves 7 slots (indicated by dashed circle) compared to the basic ternary tree algorithm.

With ternary feedback, the modified tree can offer better mean access delay than the conventional tree algorithm by introducing level skipping where a definite collision in slot Q can be skipped due to zero transmission in slot 1 to slot Q-1. When the known multiplicity feedback is available, it is possible to eliminate the wasted slot not only due to predictable collision as in the modified tree algorithm but also due to predictable idle slots. Predictable collisions occur when the number of users expected to access the last slot of the frame is more than one. Meanwhile, $Q-q$ slots will be idle when all collided users in the initial collision have accessed in the first q slots of a frame (where q=1,2,$\cdots$,Q-1). As these collisision and idle slots can be detected, they can be skipped. In so doing, this level skipping mechanism is expected to improve the performance of the modified tree algorithm as depicted in Fig. 4.2. This figure shows that 10 slots are required to resolve collision between 5 users for ternary tree algorithm with level skipping meaning that it saves 3 slots compared to the conventional ternary tree algorithm.

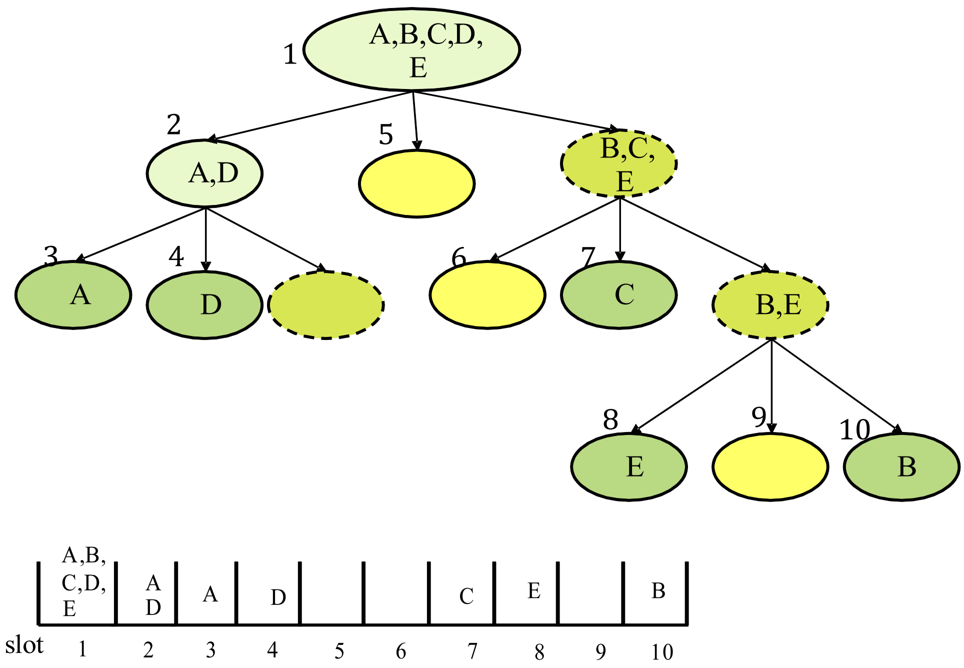


Figure 4.2: Ternary Tree algorithm with level skipping

4.2.3 Skip Adaptive Q-ary Tree Algorithm

Combining the mechanisms in subsections 4.2.1 and 4.2.2 will form skip adaptive Q-ary tree algorithm. To gain more benefit, the adaptive spltting will minimize the probability of collision and the skipping mechnaism will help to reduce the unnecessary wasted slots.

For further study, we will also investigate the condition where the access probability of each slot in the frame is non-uniform due to the skipping policy.

4.3 The Mean CRI Length Analysis for the Blocked Access

In this section, performance analysis of the proposed adaptive and skipped Q-ary tree algorithms with blocked access will be presented. The mean CRI length for adaptive and skipped Q-ary tree algorithms, can be found based on [4, 31, 45]. To begin the analysis, let N and D_N be a variable for the number of users initially collide in a slot and the conditional CRI length namely the number slots needed for the resolutionof a collision with N users including the slot of the initial collision, respectively. Since each collision will randomly be divided into Q group, the number of users in subgroup q is denoted by i_q, where $\sum_{q=1}^{Q} i_q = N$.

In addition, idle and success require one slot, D_N can be expressed as:

$$D_N = \begin{cases} 1 & N=0,1 \\ 1+D_{i_1}+D_{i_2}+...+D_{i_Q} & N\geq 2 \end{cases} \tag{4.1}$$

We then derive the conditional probability generating function (PGF) of the random variable D_N. The mean CRI length can be obtained by computing the first moment of the PGF. The PGF of D_N is defined as

$$G_N(z) \triangleq \sum_{k=0}^{\infty} \Pr\{D_N = k\} z^k = \mathrm{E}\left\{z^{D_N}\right\} \qquad N \geq 2 \tag{4.2}$$

For $N = 0 = 1$, it is clear that $G_0(z) = G_1(z) = 1$. Taking the conditional expectation on the right-hand side (RHS) of 4.2, we get

$$\begin{aligned} G_N(z) &= \mathrm{E}\left\{\mathrm{E}\left\{z^D | i_1, i_2, ..., i_Q\right\} | N\right\} \\ &= \sum_{i_1, i_2, \ ..., \ i_Q} \begin{pmatrix} N \\ i_1, i_2, \ ..., \ i_Q \end{pmatrix} p_1^{i_1} G_{i_1}(z) . p_2^{i_2} G_{i_2}(z) ... p_Q^{i_Q} G_{i_Q}(z) \quad N \geq 2 \end{aligned} \tag{4.3}$$

Where $\sum\limits_{i_1, i_2, \ ..., \ i_Q}^{N}$ is sum of all possible combination of $i_1, i_2, \ ..., \ i_Q$, $\begin{pmatrix} N \\ i_1, i_2, \ ..., \ i_{Q)} \end{pmatrix} = \frac{N!}{i_1! i_2! \ ... \ i_{Q)}!}$ is the multinomial coefficient and $\sum_{q=1}^{Q} p_q = 1$.

A recursive expression of the mean CRI length L_N can be obtained by differentiating 4.3 with respect to z and setting $z = 1$ as expressed as:

$$L_N = 1 + \sum_{q=1}^{Q} \begin{pmatrix} N \\ i_q \end{pmatrix} p_q^{i_q} \left(1 - p_q\right)^{N - i_q} L_{i_q} \quad N \geq 2 \tag{4.4}$$

with the initial values $L_0 = L_1 = 1$, indicating the condition where there is no packet and exactly one packet transmitted in a slot, respectively.

The mathematical expression of L_N for the adaptive and skipped Q-ary tree algorithms will be presented in the sequel.

4.3.1 Adaptive Q-ary tree algorithm

The proposed adaptive Q-ary splitting aims to obtain minimum number of slots for a given number of users initially collide by using the optimum Q. To derive optimum Q, some studies use the concept of system efficiency which is defined as the ratio between the expected number of success slots in one split and the splitting factor as expressed as [21]:

$$
\begin{aligned}
S(N) &= \frac{Ps(N)}{Q} \\
&= \frac{Q\binom{N}{1}\frac{1}{Q}\left(1-\frac{1}{Q}\right)^{N-1}}{Q} \\
&= N\left(\frac{1}{Q}\right)\left(1-\frac{1}{Q}\right)^{N-1}
\end{aligned}
$$

Maximizing system efficiency, *S(N),* has the same meaning as maximizing the probability of success transmission in each slot. This can be obtained by choosing Q to be exactly as N (i.e. Q = N). Then the mean CRI length, $L_{N,N}$[1], can be expressed as:

$$
L_{N,N} = 1 + \sum_{q=1}^{N} \begin{pmatrix} N \\ \\ i_q \end{pmatrix} \left(\frac{1}{N}\right)^{i_q} \left(1-\frac{1}{N}\right)^{N-i_q} L_{i_q,i_q} \qquad N \geq 2 \tag{4.5}
$$

However, we shall show that the maximum system efficiency does not imply minimum number of slots for resolution. With the nature of recursive splitting of the tree algorithm, we search the optimum Q which can result in the best mean CRI length for each N. We provide some example for finding the optimum splitting factor as follows:

- N, it is known from binary tree and ternary tree that $L_{2,2} = 5$ and $L_{2,3} = 5.5$. For greater splitting factors, the mean CRI length is larger than those two values.

- $N = 3$, from binary tree, $L_{3,2}$= 7.75, with the recursive searching, we will obtain the

[1]For an easy notation, we change the notation L_N to be $L_{N,Q}$ which implies that the mean CRI length is also a function of Q

value of $L_{3,3}$ and $L_{3,4}$ as follows:

$$\begin{aligned}
L_{3,3} &= 1 + \sum_{q=1}^{N} \sum_{i_q=0}^{N} \binom{3}{i_q} \left(\frac{1}{3}\right)^{i_q} \left(1 - \frac{1}{3}\right)^{N-i_q} L_{i_q,i_q} \\
&= 1 + 3\binom{3}{0} \left(\frac{1}{3}\right)^{0} \left(\frac{2}{3}\right)^{3} L_0 + 3\binom{3}{1} \left(\frac{1}{3}\right)^{1} \left(\frac{2}{3}\right)^{2} L_1 \\
&+ 3\binom{3}{2} \left(\frac{1}{3}\right)^{2} \left(\frac{2}{3}\right)^{1} L_{2,2} + 3\binom{3}{3} \left(\frac{1}{3}\right)^{3} \left(\frac{2}{3}\right)^{0} L_{3,3} \\
&= 1 + \frac{24}{27}L_0 + \frac{36}{27}L_1 + \frac{18}{27}L_{2,2} + \frac{3}{27}L_{3,3} \\
\frac{24}{27}L_{3,3} &= 1 + \frac{24}{27} + \frac{36}{27} + \frac{90}{27} \\
&= \frac{117}{27}\frac{27}{24} \\
&= 7.375
\end{aligned}$$

$$\begin{aligned}
L_{3,4} &= 1 + \sum_{q=1}^{4} \sum_{i_q=0}^{3} \binom{3}{i_q} \left(\frac{1}{3}\right)^{i_q} \left(1 - \frac{1}{3}\right)^{N-i_q} L_{i_q,i_q} \\
&= 1 + 4\binom{3}{0} \left(\frac{1}{4}\right)^{0} \left(\frac{3}{4}\right)^{3} L_0 + 4\binom{3}{1} \left(\frac{1}{4}\right)^{1} \left(\frac{3}{4}\right)^{2} L_1 \\
&+ 4\binom{3}{2} \left(\frac{1}{4}\right)^{2} \left(\frac{3}{4}\right)^{1} L_{2,2} + 4\binom{3}{3} \left(\frac{1}{4}\right)^{4} \left(\frac{3}{4}\right)^{0} L_{3,3} \\
&= 1 + \frac{108}{64}L_0 + \frac{108}{64}L_1 + \frac{36}{64}L_{2,2} + \frac{4}{64}L_{3,4} \\
\frac{60}{64}L_{3,4} &= 1 + \frac{108}{64} + \frac{108}{64} + \frac{180}{64} \\
&= \frac{460}{64}\frac{64}{60} \\
&= 7.667
\end{aligned} \tag{4.6}$$

- for N = 4, $L_{4,4} = 9.4853$ that is found by:

$$
\begin{aligned}
L_{4,4} &= 1 + \sum_{q=1}^{N} \sum_{i_q=0}^{N} \binom{4}{i_q} \left(\frac{1}{4}\right)^{i_q} \left(1 - \frac{1}{4}\right)^{N-i_q} L_{i_q,i_q} \\
&= 1 + 4\binom{4}{0} \left(\frac{1}{4}\right)^{0} \left(\frac{3}{4}\right)^{4} L_0 + 4\binom{4}{1} \left(\frac{1}{4}\right)^{1} \left(\frac{3}{4}\right)^{3} L_1 \\
&\quad + 4\binom{4}{2} \left(\frac{1}{4}\right)^{2} \left(\frac{3}{4}\right)^{2} L_{2,2} + 4\binom{4}{3} \left(\frac{1}{4}\right)^{3} \left(\frac{3}{4}\right)^{1} L_{3,3} + 4\binom{4}{4} \left(\frac{1}{4}\right)^{4} \left(\frac{3}{4}\right)^{0} L_{4,4} \\
&= 1 + \frac{243}{256} L_0 + \frac{432}{256} L_1 + \frac{216}{256} L_{2,2} + \frac{32}{256} L_{3,3} + \frac{4}{256} L_{4,4} \\
\frac{252}{256} L_{4,4} &= 1 + \frac{32}{27} + \frac{48}{27} + \frac{120}{27} \\
&= \frac{227}{256} \frac{256}{252} \\
&= 9.4853
\end{aligned}
$$

- for N = 5, $L_{5,5} = 12.0242$, and we will find $L_{5,5}$ as follows:

$$
\begin{aligned}
L_{5,4} &= 1 + \sum_{q=1}^{4} \sum_{i_q=0}^{5} \binom{5}{i_q} \left(\frac{1}{4}\right)^{i_q} \left(1 - \frac{1}{4}\right)^{N-i_q} L_{i_q,i_q} \\
&= 1 + 4\binom{5}{0} \left(\frac{1}{4}\right)^{0} \left(\frac{3}{4}\right)^{5} L_0 + 4\binom{5}{1} \left(\frac{1}{4}\right)^{1} \left(\frac{3}{4}\right)^{4} L_1 + 4\binom{5}{2} \left(\frac{1}{4}\right)^{2} \left(\frac{3}{4}\right)^{3} L_{2,2} \\
&\quad + 4\binom{5}{3} \left(\frac{1}{4}\right)^{3} \left(\frac{3}{4}\right)^{2} L_{3,3} + 4\binom{5}{4} \left(\frac{1}{4}\right)^{4} \left(\frac{3}{4}\right)^{1} L_{4,4} + 4\binom{5}{5} \left(\frac{1}{4}\right)^{5} \left(\frac{3}{4}\right)^{0} L_{5,4} \\
&= 1 + \frac{972}{1024} L_0 + \frac{1620}{1024} L_1 + \frac{1080}{1024} L_{2,2} + \frac{180}{1024} L_{3,3} + \frac{20}{1024} L_{4,4} + \frac{4}{1024} L_{5,4} \\
\frac{1020}{1024} L_{5,4} &= 1 + \frac{972}{1024} + \frac{1620}{1024} + \frac{5400}{1024} + \frac{2655}{1024} + \frac{569.118}{1024} \\
&= \frac{12236.118}{1024} \frac{1024}{1020} \\
&= 11.996
\end{aligned}
$$

From the above calculation, we found that Q = 2, 3, 4 are optimum for N = 2, 3, and 4, respectively, meaning that $L_{N,N}$ is optimum. However, for N = 5, Q = 5 is no longer optimum where $L_{5,5}$ = 12.0242 slots. The mathematical computation shows that using Q = 4 is optimum with $L_{5,4}$ = 11.996 slots. It means that Q optimum for N = 5 is 4, i.e. Q_5^o = 4. For higher N, it is expected that there will be an optimum value of Q for resolving each N,

Q_N^o. Then, for the proposed adaptive Q-ary splitting, the mean CRI length as the function of Q_N^o, i.e.L_{N,Q_N^o}, is given by:

$$L_{N,Q_N^o} = 1 + \sum_{q=1}^{Q_N^o} \left(\begin{array}{c} N \\ i_q \end{array} \right) p_q^{i_q} \left(1 - p_q\right)^{N-i_q} L_{i_q,Q_{i_q}^o} \qquad N \geq 2 \tag{4.7}$$

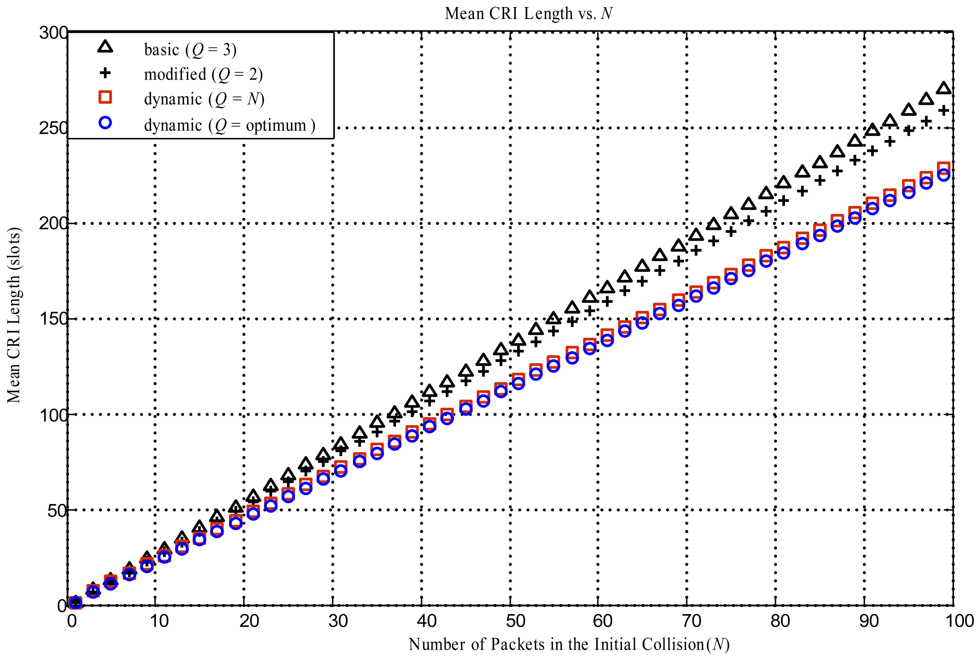


Figure 4.3: Mean CRI length vs Number of users in the initial collision (N) for adaptive Q-ary tree algorithms

In Fig.4.3, the mean CRI length of the proposed adaptive Q-ary tree algorithm, L_{N,Q_N^o}, is plotted as a function of N. For the comparison, $L_{N,2}$ (basic binary splitting), $L_{N,3}$ (basic ternary splitting), $L_{N,N}$are also plotted. It can be seen that $L_{N,N}$ offers sub-optimum performance which is much better than the ternary tree. The optimum performance is achieved by choosing the optimum splitting factor for each N, Q_N^o. We search the value of Q for each N which gives the smallest value of mean CRI length, and the value of $Q_N^o$is given in Table I. These results show that the setting value of Q to a static value does not offer an optimum performance of tree algorithm, instead it can be improved if the value of Q is changed in accordance with the number of packets initially collide. This means that the number of subgroups in the tree algorithm is an important parameter to reduce the probability of colli-

Table 4.1: Optimum value of splitting factor, $Q_{o(N)}$ for adaptive Q-ary tree algorithm

N	Q	N	Q	N	Q	N	Q	N	Q
1	-	21	17	41	33	61	49	81	64
2	2	22	18	42	34	62	49	82	65
3	3	23	19	43	34	63	50	83	66
4	4	24	19	44	35	64	51	84	67
5	4	25	20	45	36	65	52	85	68
6	5	26	21	46	37	66	53	86	68
7	6	27	22	47	38	67	53	87	69
8	7	28	23	48	38	68	54	88	70
9	8	29	23	49	39	69	55	89	71
10	8	30	24	50	40	70	56	90	72
11	9	31	25	51	41	71	57	91	72
12	10	32	26	52	42	72	57	92	73
13	11	33	27	53	42	73	58	93	74
14	12	34	27	54	43	74	59	94	75
15	12	35	28	55	44	75	60	95	75
16	13	36	29	56	45	76	60	96	76
17	14	37	30	57	46	77	61	97	77
18	15	38	31	58	46	78	62	98	78
19	16	39	31	59	47	79	63	99	79
20	16	40	32	60	48	80	64	100	79

sions and idle slots and hence offer an optimum time required for the resolution. These are even much better than the modified tree algorithm which has mechanism to skip predictable collision.

4.3.2 Skip Q-ary tree algorithm

The concept of slot skipping is to reduce the cost of solving the collision by eliminating idle and colliding slots in the resolution. However, the conflict to be resolved remains the same. The idle and colliding slots are the slots that are predicted to be idle and collision, respectively. Let denote the average number of predictable collided and idle slots as $C_{N,Q}$ and $I_{N,Q}$, respectively. The subscript N,Q represents the number of users and the splitting factor as in the notation of mean CRI length,respectively, $L_{N,Q}$.

- Expected number of skipped idle slots, $I_{N,Q}$.

  When all users have accessed the first j slots (where $j \leq Q-1$), it can be predicted that the last $Q-j$ slots will be idle. It means that the predictable idle slots can occur in slot 2 to slot Q. We first compute the probability of idle for one slot (denoted by slot j) with the following binomial distribution.

$$\begin{aligned} P[i_j = 0] &= \binom{N}{0} p_j^0 (1-p_j)^N \\ &= (1-p_j)^N \end{aligned} \tag{4.8}$$

  The probability that last j slots from Q slots, where $j = 1, 2, \cdots, Q-1$, are idle is given by:

$$P[i_j = \cdots = i_Q = 0] = (1-p_j)^N - (1-p_{(j-1)})^N \tag{4.9}$$

  The expectation of skipped idle in slot j to slot Q (where $j = 2, 3, , Q$), $I_{N,Q}$, can be expressed as:

$$\begin{aligned} I_{N,Q} = {} & 1(1-(p_1)^N - (1-(p_1+p_2))^N + 2((1-(p_1+p_2))^N \\ & - (1-(p_1+p_2+p_3))^N) + \cdots + (Q-1)((1-(p_1+p_2+\cdots p_{Q-1}))^N \\ & - (1-(p_1+p_2+\cdots p_Q))^N) \\ & = \sum_{i=1}^{Q-1} \left(\sum_{j=1}^{i} p_j \right)^N \end{aligned}$$

For the case when $p_1 = p_2 = \cdots = p_Q = \frac{1}{Q}$, equation (4.3.2) is reduced to:

$$I_{N,N} = \frac{1}{Q^N} \sum_{i=1}^{Q-1} i^N \tag{4.10}$$

- Expected number of skipped collision slots

  As aforementioned, the predictable collisions always occur in slot Q. To find the expression of $C_{N,Q}$, let $E[C]$ be the probability of of collision in slot Q when k out of N users access this particular slot, where $k \geq 2$. This can be given by:

$$E[C] = \binom{N}{k} (p_Q)^k (1 - p_Q)^{n-k} \tag{4.11}$$

  Then, summation of equation (4.11) for all value of k yields to the average number of skipped collisions in the resolution which can be expressed as:

$$C_{N,Q} = \sum_{k=2}^{N} \begin{pmatrix} N \\ \\ k \end{pmatrix} (p_Q)^k (1 - p_Q)^{N-k} \tag{4.12}$$

Equation (4.12) can also be expressed by subtracting the average number of all events with the average number of success and idle slot Q as:

$$C_{N,Q} = 1 - (1 - p_Q)^{N-1} (1 + p_Q (N-1)) \tag{4.13}$$

Eventually, the mean CRI length of blocked access skip Q-ary tree algorithm can be expressed as:

$$L_{N,Q} = 1 - C_{N,Q} - I_{N,Q} + \sum_{q=1}^{Q} \sum_{i_q=0}^{N} \begin{pmatrix} N \\ \\ i_q \end{pmatrix} p_q^{i_q} \left(1 - p_q\right)^{N-i_q} L_{i_q,Q} \quad N \geq 2 \tag{4.14}$$

We shall now derive the close form expression of the mean CRI length of the skip Q-ary tree algorithm in (4.14) based on [45]. First, let define $L(z) \triangleq \sum_{N=0}^{\infty} L_{N,Q} \frac{z^N}{N!}$. With $L_{N,Q}$ in

(4.14), we find $L(z)$ as follows:

$$L(z)=e^{z}-e^{p_Q z}e^{z\left(1-p_Q\right)}+(1+p_Q z)e^{z(1-p_Q)}-\sum_{i=1}^{Q-1}e^{z\sum_{j=1}^{i}p_j}+(Q-1) \tag{4.15}$$
$$+\sum_{i=1}^{Q-1}\sum_{j=1}^{i}p_j z+e^{z}\sum_{j=1}^{Q}L\left(p_j z\right)\cdot e^{-p_j z}-Q\left(1+z\right)$$

Defining $L^{*}(z)\triangleq\sum_{k=0}^{\infty}L_k^{*}z^{k}\triangleq e^{-z}L(z)$ we then obtain:

$$\sum_{k=0}^{\infty}L_k^{*}z^{k}-\sum_{k=0}^{\infty}\sum_{j=1}^{Q}L_k^{*}.\left(p_j z\right)^{k}=-Q\sum_{k=0}^{\infty}\left(\frac{(-1)^{k}}{k!}+\frac{(-1)^{k-1}}{(k-1)!}\right)z^{k} \tag{4.16}$$
$$+\sum_{k=0}^{\infty}\left(\frac{(-1)^{k}p_Q^{k}}{k!}+p_Q\frac{(-1)^{k-1}p_Q^{k-1}}{(k-1)!}\right)z^{k}$$

by equating the coefficients of z^{k} on both sides, this becomes

$$L_k^{*}\left(1-\sum_{j=1}^{Q}\left(p_j\right)^{k}\right)=Q\left(-1\right)^{k}\frac{k-1}{k!}+\left(-1\right)^{k}p_Q^{k}\frac{(1-k)}{k!} \tag{4.17}$$
$$-\left(-1\right)^{k}\frac{\sum_{i=1}^{Q-1}\left(1-\sum_{j=1}^{i}p_j\right)^{k}-(Q-1)+k\sum_{i=1}^{Q-1}\sum_{j=1}^{i}p_j}{k!} \tag{4.18}$$

Finally, the closed from expression of the skip Q-ary tree algorithm can expressed as:

$$L_{N,Q}=1+\sum_{k=0}^{N}\begin{pmatrix}N\\k\end{pmatrix}\frac{(-1)^{k}\left\{\begin{array}{c}Q\left(k-1\right)+p_Q^{k}\left(1-k\right)-\sum_{i=1}^{Q-1}\left(1-\sum_{j=1}^{i}p_j\right)^{k}\\-\left(Q-1\right)+k\sum_{i=1}^{Q-1}\sum_{j=1}^{i}p_j\end{array}\right\}}{\left(1-\sum_{j=1}^{Q}\left(p_j\right)^{k}\right)} \tag{4.19}$$

The mean CRI length of the proposed skip Q-ary tree algorithms are plotted in Fig 4.4.(a) as function of number of collided packets. As expected, a significant improvement to the ternary tree algorithm can be obtained when the proposed level skipping mechanism is applied to tree algorithm for Q = 2,3 and 4. Further, it can be seen that the skip binary tree offers the most significant improvement implying that the level skipping mechanism is very effective to reduce wasted slots when number of subgroup is 2. The reason of these results is explained as follows. In binary splitting where every collision is divided into only two subgroups, a maximum amount of feedback (50% of the information) can be properly utilized by the users by deciding to access or not to access between the two provided slots. This yield to the condition where it is easy to predict what will occur in the second slot once the feedback of the first subgroup returns. If the second slot is expected to be either collision or idle, the system can directly skip the second slot. On the other hand, for other splitting factor, the information in the feedback of one slot cannot be directly utilized by the users to skip the unnecessary wasted slots. For example, feedback from slot 1 cannot be used to skip the rest of slots except all the collided users access in this slots. To skip colliding slot, it also has to wait for feedback from the first $Q-1$ slots. In the consequence of these facts, there are still many wasted slots due to collisions and zero transmissions causing more cost still to be paid. For the special case of skip binary splitting, the results obtained here are apparently similiar those in [42].

4.3.3 Skip Adaptive Q-ary tree algorithm

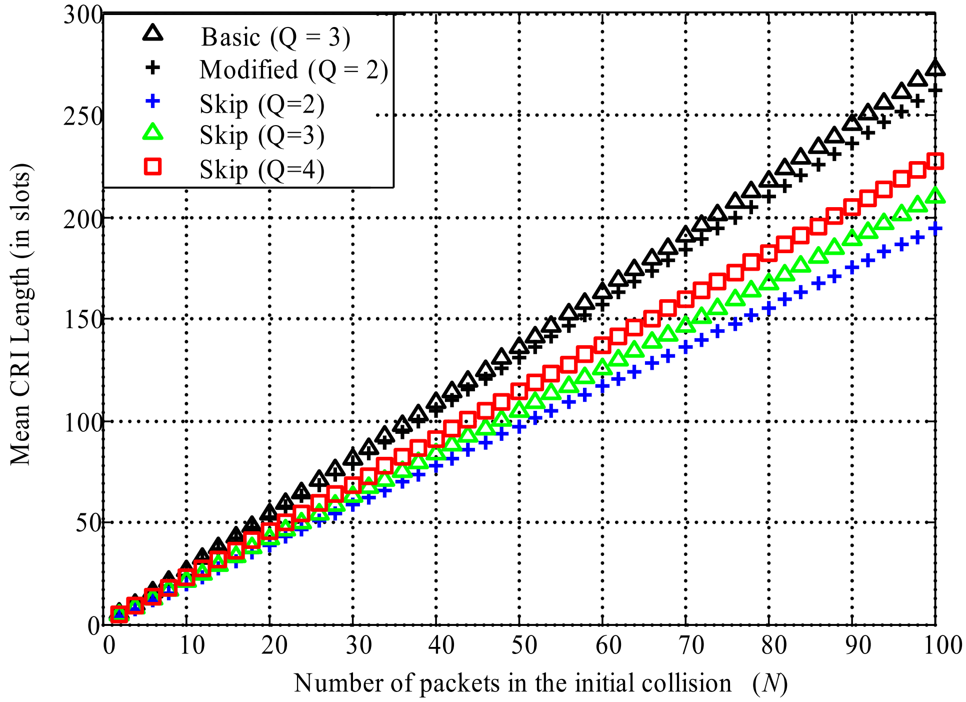
This proposed skipping mechanism will also be applied to the adaptive Q-ary tree algorithm previously discussed. This forms a skip adaptive Q-ary tree algorithm. Since an additional mechanism i.e. skipping mechanism is applied, the optimum splitting factor given in Table 4.1 cannot be used in this adaptive splitting with skipping mechanism. Then, we search the value of Q which gives the minimum mean CRI length for each N as shown in Table 4.1. The mathematical expression of the mean CRI length of this skip adaptive splitting can be obtained by modifiying the expression in (4.14) with changing Q by the value in the table. However, due to the complexity of the derivation, the closed from expression of this skip adaptive Q-ary tree algorithm cannot be delivered.

For the skip adaptive Q-ary tree algorithm where the two proposed mechanisms are combined, further improvement is expected. As suggested in the results of the adaptive Q-ary tree algorithm in III.A, the performance will be optimum if Q is not set to a static number for all number of colliding packets, but rather than dynamically changed accordingly to the number of colliding packets. It is shown in the Fig. 4.4 that when Q is set to be Q_N^o in Table. 4.1, this outperforms of the static skip binary tree algorithm. We can also see again that the choice of Q =N is not optimum since it is inferior to the skip binary tree. Disregarding the skipping mechanism, these results again prove that Q is an important parameter to be set in order to reduce the average number of collisions and idle slots and hence optimum performance of the algorithm.

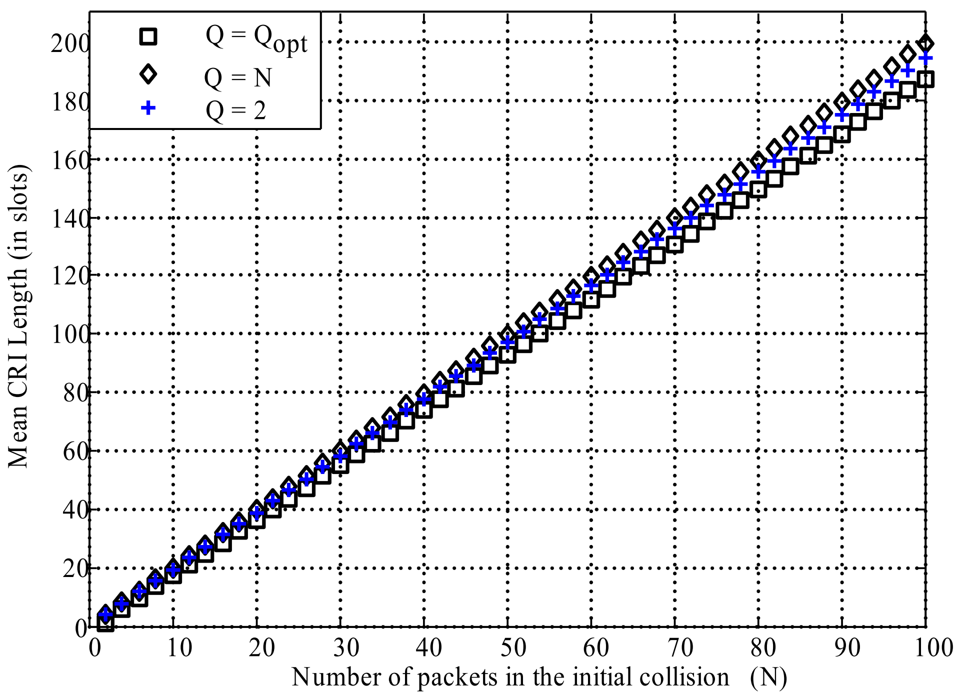
In the previous results, we only consider the case when the collided users join each subgroup with uniform probability. The skipping strategy allows the access probability in each slot to bet non-uniformly. Intuitively, the resolution will require small number of slots if the probability of success slots is high. In addition, we can force high probability of success in the first view slots. Then, the probability of accessing the first view slots should be set to optimum value. The optimum access probability is a function of the number of collided packets; hence the value may differ for each case. Table 4.3 shows the lists of the optimum value of p_1 of the skipping binary tree. The value of p_2 can be calculated by $1-p_1$. For skip ternary tree, the optimum values of $p1$ and p_2 are shown in Table 4.4, where $p_3 = 1-(p_1+p_2)$.

Fig.4.6. shows the mean CRI length of Q-ary tree algorithm with the proposed skipping mechanism and non-uniform access probability. It can be seen that there is an improvement compared to those with uniform probability in Fig. 4.5. This again shows that with static Q, the skip binary splitting is superior. As already explained, this can maximize the utilization of information contained in the feedback. Moreover, the non-uniform probability makes the proposed skipping mechanism very effective. For this binary case, the results is apparently similar to the one proposed in [32, 40, 42]. The value of the optimum probability we found also similar to the one in [40].

When non-uniform probability is applied to the adaptive skip Q-ary tree algorithm,



a. Skip Q-ary splitting



b.Skip adaptive Q-ary splitting

Figure 4.4: Mean CRI length vs Number of packets in the initial collision (N) for skip Q-ary tree algorithms (fixed Q) and skip adaptive Q-ary tree algorithms

this becomes the most complex algorithm since it needs to choose optimum setting for both number of subgroups and the access probability. For the case of adaptive skip Q-ary tree al-

Table 4.2: Optimum value of splitting factor, $L_{Q_o(N)}$ for skip adaptive Q-ary tree algorithm

N	Q	N	Q	N	Q	N	Q	N	Q
1	-	21	12	41	23	61	35	81	46
2	2	22	13	42	24	62	35	82	46
3	2	23	13	43	25	63	36	83	47
4	3	24	14	44	25	64	36	84	47
5	3	25	14	45	26	65	37	85	48
6	4	26	15	46	26	66	37	86	48
7	4	27	16	47	27	67	38	87	49
8	5	28	16	48	27	68	38	88	50
9	6	29	17	49	28	69	39	89	50
10	6	30	17	50	28	70	40	90	51
11	7	31	18	51	29	71	40	91	51
12	7	32	18	52	30	72	41	92	52
13	8	33	19	53	30	73	41	93	52
14	8	34	19	54	31	74	42	94	53
15	9	35	20	55	31	75	42	95	54
16	9	36	21	56	32	76	43	96	54
17	10	37	21	57	32	77	43	97	55
18	11	38	22	58	33	78	44	98	55
19	11	39	22	59	33	79	45	99	56
20	12	40	23	60	34	80	45	100	56

gorithm with non-uniform probability, the non-uniform probability setting makes the mechanism for adaptive changing of the number of subgroup does not work well. The setting of Q=2 is always more effective for resolving collision until the number of collided packets is 39. The ternary subgroup (Q = 3) is getting better for the number of collided packets is greater than 40. To find the possibility for using Q=4 , we seek the optimum probability until N=1000. However, our results shows that Q=*3* is the most effective. We plotted the results in

Table 4.3: Values of optimum p_1

N	p_1	N	p_1	N	p_1	N	p_1	N	p_1
1	-	21	0.08327	41	0.044	61	0.03	81	0.023
2	0.5	22	0.07974	42	0.043	62	0.03	82	0.023
3	0.4119	23	0.07650	42	0.042	63	0.029	83	0.022
4	0.3429	24	0.07351	44	0.041	64	0.029	84	0.022
5	0.2882	25	0.07074	45	0.040	65	0.028	85	0.022
6	0.2493	26	0.06818	46	0.040	66	0.028	86	0.022
7	0.2199	27	0.06580	47	0.039	67	0.027	87	0.021
8	0.1968	28	0.06357	48	0.038	68	0.027	88	0.021
9	0.1780	29	0.06150	49	0.037	69	0.027	89	0.021
10	0.1625	30	0.05955	50	0.036	70	0.026	90	0.021
11	0.1496	31	0.05846	51	0.036	71	0.026	91	0.020
12	0.1385	32	0.05615	52	0.035	72	0.026	92	0.020
13	0.1290	33	0.05501	53	0.034	73	0.025	93	0.020
14	0.1207	34	0.05362	54	0.034	74	0.025	94	0.020
15	0.1134	35	0.05210	55	0.033	75	0.025	95	0.020
16	0.1069	36	0.05000	56	0.033	76	0.024	96	0.019
17	0.1012	37	0.04900	57	0.032	77	0.024	97	0.019
18	0.09603	38	0.04800	58	0.032	78	0.024	98	0.019
19	0.09136	39	0.04600	59	0.031	79	0.023	99	0.019
20	0.08713	40	0.04500	60	0.031	80	0.023	100	0.019

Fig.6.(a). Since we seek the optimum performance until N=1000, the improvement of using non-uniform probability in dynamic skip Q-ary tree algorithm can be seen clearly. We then zoom the results in some points in Fig.6 (b)where the improvement can be seen clearly.

From the mean CRI length, we can also find the expected number of slots required by one users to get successful transmission after its the collision resolution process starts collision resolution time for one station, $D_{N,Q}$. Since $L_{N,Q}$ is the number of slots needed to

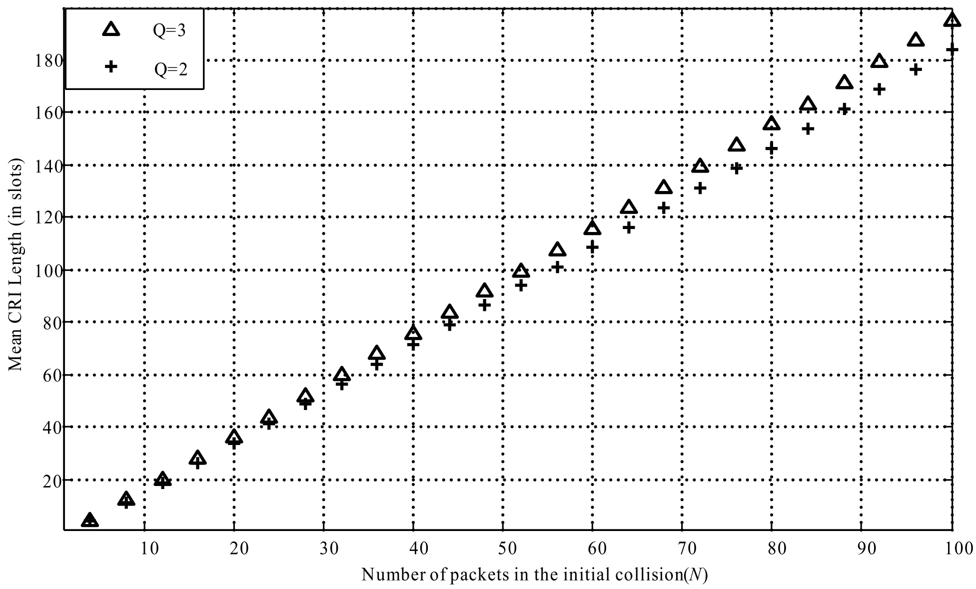
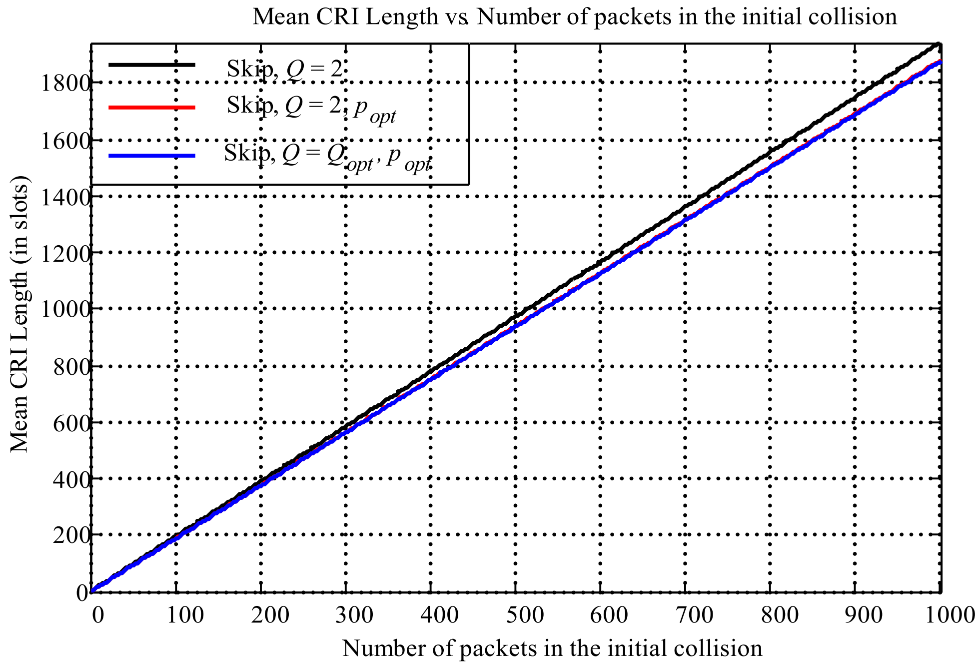


Figure 4.5: Mean CRI length vs Number of packets in the initial collision (N) for skip Q-ary tree algorithms (static Q) with non-uniform probability

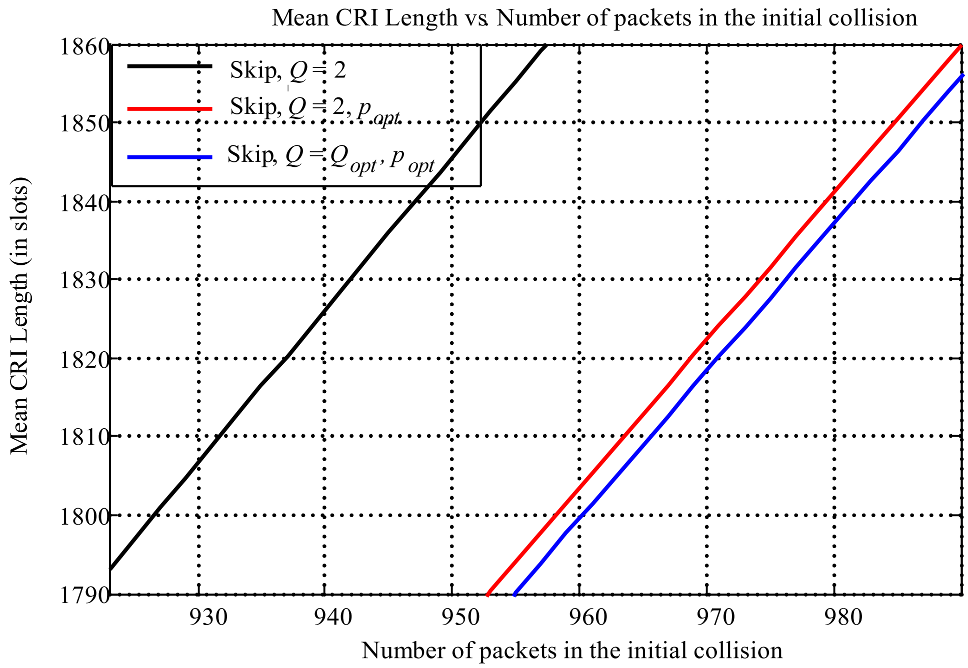
resolve N users initially collided, then $D_{N,Q}$ can be expressed as:

$$D_{N,Q} = \frac{L_{N,Q}}{N} \tag{4.20}$$

To validate the presented analytical results of the mean CRI length, the simulation results by using MATLAB are provided. For the blocked access mode, a collision can be modeled without considering the packet arrival process. Simulations with considering the packet arrival process will be given in 4.4 when the MST is analyzed. We set a case of collision involving N occurs in a slot, then resolve them with the algorithm. We the set the splitting factor/ frame size, Q. Users randomly select a slot in a frame with probability $1/Q$. If more than one users responds to the same slot, there will be a collision. As a result, a new frame will be initiated right after the corresponding collision slot. Consequently, we must shift the access slots for the users waiting for the transmission. The simulation will run until all users has their own slot. Simulations were run for upto 50000 times. We present the result of mean CRI length of 2-100 users. To compare the results of the analysis and the simulation, we plot the mean CRI length of skip binary and ternary tree algorithm with uniform and non-uniform probability as given in 4.7



(a) CRI Length of static and skip adaptive Q -ary tree algorithms



(b) CRI Length of static and skip adaptive Q-ary tree algorithms (zoom)

Figure 4.6: Mean CRI length vs Number of packets in the initial collision (N) for skip adaptive Q-ary tree algorithms with non-uniform probability.

Table 4.4: Values of optimum p_1, p_2

N	p_1	p_2	N	p_1	p_2	N	p_1	p_2	N	p_1	p_2
	-	-	26	0.068	0.069	51	0.035	0.483	76	0.024	0.322
2	0.5	0.142	27	0.066	0.065	52	0.035	0.483	77	0.024	0.322
3	0.366	0.324	28	0.063	0.064	53	0.034	0.483	78	0.023	0.322
4	0.31	0.391	29	0.063	0.061	54	0.034	0.483	79	0.023	0.322
5	0.274	0.455	30	0.063	0.059	55	0.033	0.483	80	0.023	0.322
6	0.242	0.516	31	0.058	0.063	56	0.032	0.484	81	0.023	0.322
7	0.216	0.569	32	0.056	0.063	57	0.032	0.484	82	0.022	0.322
8	0.194	0.612	33	0.054	0.063	58	0.031	0.484	83	0.022	0.322
9	0.176	0.648	34	0.063	0.053	59	0.031	0.484	84	0.022	0.322
10	0.161	0.678	35	0.063	0.051	60	0.03	0.485	85	0.021	0.656
11	0.148	0.148	36	0.475	0.049	61	0.03	0.485	86	0.021	0.323
12	0.138	0.138	37	0.476	0.048	62	0.029	0.485	87	0.021	0.656
13	0.128	0.128	38	0.476	0.047	63	0.029	0.486	88	0.021	0.322
14	0.12	0.12	39	0.477	0.046	64	0.028	0.486	89	0.021	0.322
15	0.113	0.113	40	0.045	0.477	65	0.028	0.486	90	0.02	0.322
16	0.107	0.107	41	0.044	0.478	66	0.028	0.486	91	0.02	0.659
17	0.101	0.101	42	0.043	0.478	67	0.027	0.487	92	0.02	0.32
18	0.096	0.096	43	0.042	0.479	68	0.027	0.487	93	0.02	0.318
19	0.091	0.091	44	0.041	0.479	69	0.026	0.338	94	0.019	0.314
20	0.087	0.087	45	0.04	0.48	70	0.026	0.332	95	0.019	0.308
21	0.083	0.083	46	0.039	0.481	71	0.026	0.328	96	0.019	0.684
22	0.08	0.08	47	0.038	0.481	72	0.025	0.326	97	0.019	0.491
23	0.076	0.076	48	0.038	0.481	73	0.025	0.324	98	0.019	0.491
24	0.073	0.073	49	0.037	0.481	74	0.025	0.323	99	0.018	0.491
25	0.071	0.071	50	0.036	0.482	75	0.024	0.323	100	0.018	0.491

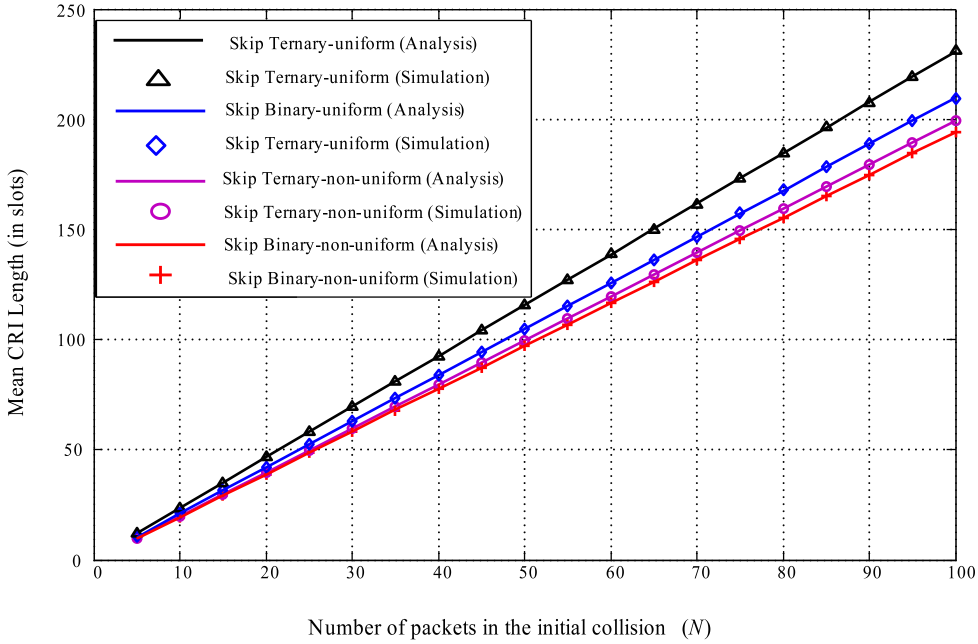


Figure 4.7: Analytical and simulation results of the mean CRI length of skip tree algorithms.

4.4 Throughput Analysis of Adaptive and Skip Tree Algorithm

In this section the throughput analysis of the proposed adaptive and skip tree algorithm will be presented. The throughput performance in terms of maximum stable throughput is the focus of the analysis. We consider that the users generate packets following the Poisson process. To confirm our results, the simulation is conducted for each mechanism.

4.4.1 Blocked Access

The channel throughput can be defined as the ratio between the average successful slot and the time slots elapsed as expressed as:

$$S = \frac{\text{average successful transmission slot}}{\text{average required slots}} \tag{4.21}$$

For the case of tree algorithm with blocked access, throughput can be found from the mean CRI length, $L_{N,Q}$, presented in previous where the mean CRI length represents the average required slots. Since the new arrival packets are not allowed to access during the collision resolution period the current resolution is finished, it is obvious that the successful slots is exactly the number of users in the initial collision. Then, the throughput is the ratio between the number of collided users and the mean CRI length. In [4], this ratio is called as the effective service rate of the system. When the arrival packets is modeled as Poisson distribution with the arrival rate is G, the system throughput can be then defined as the ratio between the arrival rate and the average of $L_{N,Q}$ which can be given by:

$$S = \frac{G}{\sum_{N=0}^{\infty} L_{N,Q} \frac{e^{-G} G^N}{N!}} \tag{4.22}$$

By using this formulation, we find the maximum stable throughput of the adaptive splitting tree algorithm to be 43.75% which is slightly higher than those when the splitting factor Q=N that is 43.33 %. These results are much better than the blocked access tree algorithms presented in [45] which is only 38.1 % with ternary feedback. To validate these

Table 4.5: List of System Effective Service Rate N/L_N

N	$N/L_{N,2}$	$N/L_{N,3}$	$N/L_{N,N}$	$N/L_{N,Q^o}$	$N/L_{N,2}, p^o$	$N/L_{N,3}, p^o$	$N/L_{N,Q^o}, p^0$
2	0.4000	0.3636	0.4000	0.5000	0.5	0.4637	0.5
3	0.3913	0.3871	0.4068	0.5143	0.5183	0.4848	0.5183
4	0.3801	0.3866	0.4121	0.5174	0.5241	0.4930	0.5241
5	0.3726	0.3822	0.4158	0.5202	0.5271	0.4956	0.5271
10	0.3590	0.3726	0.4244	0.5248	0.5319	0.5005	0.5319
15	0.3548	0.3710	0.4276	0.5261	0.5327	0.5016	0.5327
25	0.3514	0.3688	0.4302	0.5271	0.5202	0.5020	0.5330
50	0.3490	0.3677	0.4323	0.5278	0.5328	0.5020	0.5330
75	0.3482	0.3670	0.4330	0.5281	0.5326	0.5019	0.5331
100	0.3478	0.3668	0.4333	0.5282	0.5324	0.5019	0.5332

results, simulations are conducted. The packet generation process is modeled as Poisson traffic with rate G. For each value of G, we simulated the algorithms over 750,000 slots. Fig. 4.8 shows the throughput- packet delay curves of the algorithms based on the simulations. Packet delay is calculated from the slot where the packet is firstly generated until the slot is successfully transmitted. The figure shows that the value of maximum stable throughput of proposed algorithms match with those in the table.

Analysis of free access tree algorithm has been comprehensively presented in [24, 45, 48]. These analysis utilize functional equations and other mathematical theories which make the determination of the MST lengthy. Peeter in [56] comes with a new approach to find the MST of the free access tree algorithms. He views the splitting process in the tree algorithm as a branching process where a branch of the tree which represents a group of collision with i packets ($i \geq 2$) will results in new branches which contain j users. The j packets in the new groups can be from the initial collision and the new arrival. When the value of j is 0 and 1, it means that these particular slots are idle and success, respectively. These then do not produce any new branch. On the other hand, new branches will be generated when

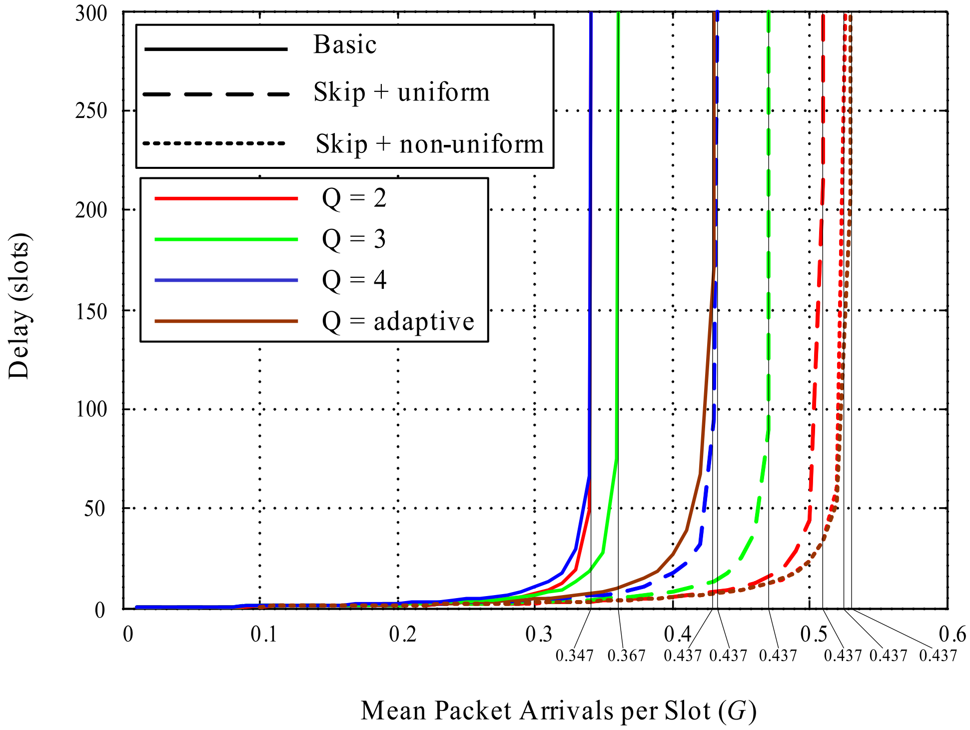


Figure 4.8: Traffic Load vs. Delay.

$j \geq 2$. This process is figured out into a matrix denoted as matrix **M**, in which each row corresponds to the number packets initially collide, whereas each column corresponds to the expected number of packets in a group after splitting. The MST can be determined from this matrix by calculating its Eigen value. The MST exists if the dominant Eigen value is less or equal to one in which this implies that the algorithm is still stable and able to work for a certain value of arrival rate. The detail of matrix construction for the case of adaptive and skip tree algorithms with free access will be explained in the sequel.

4.4.1.1 Free Access Adaptive Q-ary tree algorithm

In the blocked access case, optimum number of subgroups of the proposed adaptive Q-ary tree algorithm can be found as in Table I. However, it is complicated for the case of free access since the CRI length varies depending on the packet arrival rate. Therefore, free access adaptive Q-ary tree algorithm will only consider Q=N.

As reported in [56], finding the MST of the tree algorithm needs two matrices; **A** and

A. Matrix **B** represents the branching process in which the elements are the expected number of slots which contain $0, 1, \ldots, N$ number of users as expressed as:

$$B_{i,j} = \begin{cases} \begin{pmatrix} i \\ j \end{pmatrix} \sum_{q=1}^{Q_i} p_q^i (1-p_q)^{i-j} & i \geq 2, j \geq 2 \\ 0 & \text{otherwise} \end{cases} \tag{4.23}$$

In this matrix, element in row i represents the number of users initially collided ($0 \ldots N\text{-}1$) and column j is the number of users which access the same slot after the splitting. Since this is free access mode, we need to model the new packets arrival. In every row of the matrix, each element represents the expected number of newly arrive packets (from 0 to ∞) joining the resolution process. If all the elements in each row are summed up, it must be equal to 1. Since the new arrived packet is assumed to be Poisson distribution, with arrival rate, λ, the new arrival matrix and its each element can be respectively given by:

$$A = \begin{bmatrix} a_0 & a_1 & a_2 & \ldots & 1-\sum_{k=l-1}^{\infty} a_k \\ 0 & a_0 & a_1 & \ldots & 1-\sum_{k=l-2}^{\infty} a_k \\ 0 & 0 & a_0 & \ldots & 1-\sum_{k=l-3}^{\infty} a_k \\ 0 & 0 & 0 & \ldots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \end{bmatrix} \tag{4.24}$$

Where $a_k = \frac{e^{-\lambda}\lambda^k}{k!}$. Multiplication of matrix B and A will result in matrix M where the new arrival be taken into account to the resolution process.

$$\mathbf{M} = \mathbf{BA} \tag{4.25}$$

Removing the first two rows and columns does not change the end results since these rows and column contain the expected number of idle and success slots which will not generate any new collision. From this matrix, the value of MST can be obtained by finding the dominant value of matrix A. We found that the MST for the free access adaptive Q-ary tree algorithm is 0.3928. This result shows that the adaptive splitting is little inferior to the static ternary splitting with the MST of 0.4015 [45]. From the results of matrix **M**, the adaptive splitting results in higher probability of idle slots compared to those ternary splitting. This make the adaptive splitting is less efficient and has lower MST compared to the ternary splitting.

4.4.1.2 Free Access Skip Q-ary tree algorithm

The proposed skipping mechanism explained in II. b) in the case of blocked access cannot fully be adopted in the case of tree algorithm with free access. The skipping strategy cannot be used to eliminate idle slots. There are two main reasons for this. Even though the number of packets transmitted in each slot can be tracked but the packets cannot be identified whether they are the collided packets or newly arrived packets. Secondly, a new packet may arrive in the expected idle slots so letting these slots will increase the probability of successes. Then, expected collided slots can be eliminated when the different between the number of users in the initial collision and the number of users accessing slot 1 to slot Q-1 is equal or greater than two. This is not necessary to distinguish the users in the initial collision and the new arrival. We define a matrix of probability eliminated slots as **P** where its each element can be expressed as:

$$P_{i,j} = \begin{cases} \begin{pmatrix} i \\ j \end{pmatrix} \sum_{q=1}^{Q} p_Q^j \left(1 - p_Q^{i-j}\right) \prod_1^{Q-1} b_{(q)} & i \geq 3, i \geq j, \left(i - \left(i_Q + \sum_1^{Q-1} b_{(q)}\right)\right) \geq 2 \\ 0 & \text{otherwise} \end{cases} \tag{4.26}$$

Then the MST can be determined from the dominant Eigen-value of matrix **A** as follows:

$$\mathbf{M} = (\mathbf{B} - \mathbf{P})\,\mathbf{A} \tag{4.27}$$

Table 4.6: Maximum stable throughput for tree algorithm with the proposed skipping mechanism

Algorithm	Blocked Access			Free Access		
	Q	p_Q	MST	Q	p_Q	MST
Skip Q-ary (fixed)	2	$1/Q$ (uniform)	0.514	2	$1/Q$ (uniform)	0.3926
	3		0.477	3		0.4082
	4		0.439	4		0.4011
Skip adaptive Q-ary	N	$1/N$(uniform)	0.501	N	$1/N$(uniform)	0.41412
	Q_N^o	$1/Q_N^o$ (uniform)	0.528			
Skip Q-ary (fixed)	2	$p_{opt}(N)$ (non-uniform)	0.532	2	0.6	0.403461
	3		0.501	3	0.37	0.409848
Skip Adaptive$Q$-ary	Q_N^o	$p_{opt}(N)$ (non-uniform)	0.533		NA	

By using this policy we found that the MSTs of skip Q-ary tree algorithm with free access for different values of Q are given is Table 4.6.

For the case of skip adaptive Q-ary tree algorithm with free access, we find its MST is 0.41412. This MST is the higher than those static splitting. The skipping mechanism helps to reduce the number of collision slots and increase the number of success slots as the results of matrix **M** in (4.25).

If we compare the MST of tree algorithms with the proposed mechanism for blocked access and free access modes in this study, the blocked access mode offers much better performance. It is shown that the proposed mechanism in free access mode does not work as effective as in the blocked access one. This is because the blocking policy to the newly arrived packets in the blocked access mode will reduce the probability of collision. Hence, the collision resolution time is decreased. On the other hand, the free access mode which allows newly arrive packets will contribute to the occurrence of new collisions.

At the end of this paper, we summarize the relation between the existing tree algorithm and our proposed mechanisms as depicted in Fig. 4.9. Starting from the STA, the key fea-

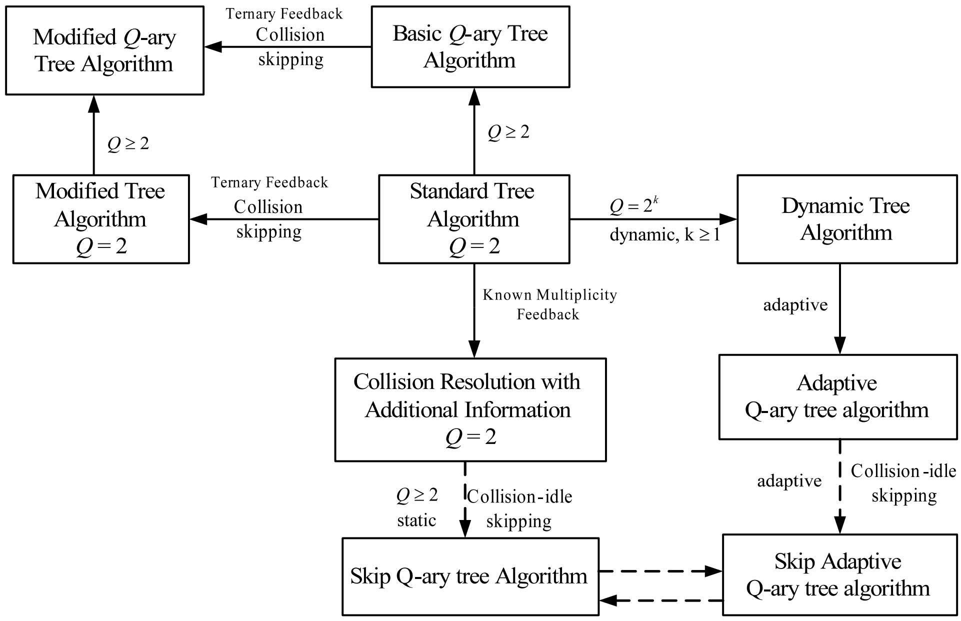


Figure 4.9: Development of framed-based random access protocols.

ture of it is binary splitting which requires only binary feedback. When another feature i.e. collision skipping is applied, the STA evolve to MTA. Ternary feedback is required in this evolution. These STA and MTA can form basic and modified Q-ary tree algorithms, respectively, when the splitting is set to Q ($Q \geq 2$). When known multiplicity is available, three variations of tree algorithms namely adaptive Q-ary, Skip Q-ary and Skip adaptive Q-ary tree algorithms can be formed. The adaptive Q-ary dynamically changes the splitting factor in accordance with the number of collided users, while the skip Q-ary has the feature of collision-idle skipping. The two latest will construct the skip adaptive Q-ary tree algorithm.

CHAPTER V

A Proportional Differentiation Model using Slotted Aloha for Reservation-based MAC Protocols

In this section, we consider prioritization schemes for reservation-based MAC protocols, in which the slotted Aloha [7] is applied during the channel reservation period. Nodes are classified into multiple classes, with class-1 node being the highest priority. Each class has a predefined reservation success rate with respect to that of class-1. This ratio is normally set in relation to the QoS requirement of each class. In this study, the objective of the MAC protocol design is to satisfy this proportional differentiation requirement over all classes, without sacricing much bandwidth utilization efficiency.

5.1 Introduction

Future broadband wireless access networks demand highly efficient MAC protocols that are not only easy to implement at high speed, offer high throughput and low transmission latency, but also can provide different quality of service guarantee. Reservation-based MAC protocols are perceived as a very important approach towards achieving these goals [59]. New emerging wireless standards have adopted this approach as an integrated part, see ECMA-368 [61] for example. Such protocols have separate and alternating reservation and data transmission periods in each frame. A mobile node will first make a channel reservation by transmitting a request packet to the base station during the reservation period. Upon successful reservation, the node is then assigned a data slot or more by the base station for the data packet transmission on a contention-free basis. Since collisions only occur during the reservation periods, very high channel bandwidth efficiency can be achieved by making reservation slots much shorter than data slots, so called minislots. This commonly known concept has been adopted in numerous research studies. As multimedia services have become prevalent, it is very important for MAC protocols to provide different quality of services (QoS) for different classes of nodes. For reservation-based MAC protocols, different priorities of nodes can be accomplished during reservation periods by using the following two approaches. First approach is to design a reservation protocol that can differentiate the reservation success rate between different classes of nodes. A set of rules defined by the reservation-based protocol must guarantee that higher priority classes are more successful in reserving the channel than the lower priority classes in a prescribed criteria. In the second approach, the base station allocates bandwidth to each node in proportion according to their demands; this occurs after the node has made a successful reservation. Note that one may apply the combination of both approaches for achieving greater control and more effective service differentiation. However, in this paper we concentrate only on the first approach. In literature, prioritization schemes have been derived by using different service differentiation policies. Some schemes adopt the policy that is completely discriminated against low priority class; all low priority nodes are not allowed to transmit any packet until all higher priority nodes have already successfully accessed the channel [60]. Other protocols suggest policies that are less discriminatory towards low priority nodes [62, 63]. For instance, low priority nodes are allowed to compete against high priority nodes, but on average they experience longer delay or attain lower throughput than those high priority nodes. This is commonly referred to as proportional differentiation [64, 65].

5.2 System Model Description

To differentiate the QoS between J classes through reservation period, we propose a frame structure of slots that is divided into K blocks with each block having a length of M_k slots as shown in Fig. 5.1. Note that the data transfer period which immediately follows the reservation period are not shown here. Let p_{jk} be the access probability of class-j nodes for making reservation in block k where $k \in \{1,2,\cdots,K\}$. By strategically adjusting the values of these parameters, i.e. M_k and p_{jk}, we can achieve the proportional differentiation requirement over all classes.

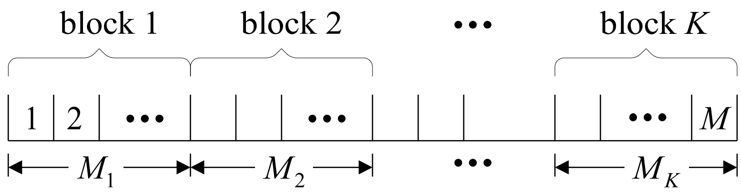


Figure 5.1: The frame structure for one reservation period

This generalized model can be devised into various different prioritization schemes, three of which are presented here to illustrate its applicability and determine the effectiveness of QoS differentiation provisioning. The first scheme (Scheme I) defines only one block, i.e., $K = 1$ and assigns different access probabilities for different classes, i.e. $p_{jk} = p_j$ for $\forall j,k = \{1,2,\cdots,J\}$. To ensure that the higher priority nodes have higher probability of making successful channel reservations than the lower priority nodes, we set $p_1 > p_2 > \cdots > p_J$ As maintaining high overall success rate is of prime importance, parameters p_j must be optimized.

The second scheme (Scheme II) is based on the concept of complete partitioning where a separate portion of slots is allocated for each class. This is achieved by setting $K = J$ and $p_{jk} = 0$ if $j \neq k$ for $\forall j,k = \{1,2,\cdots,J\}$. In this scheme, parameters have to be set to be the optimum values according to the number of class-j nodes and the number of allocated slots in the block j to obtain maximal overall success rates.

The third scheme (Scheme III) gives advantages to higher classes by forcing lower classes to start their first access at later blocks. This is done by setting $K = J$ and $p_{jk} = 0$ if $j > k$ for $\forall j,k = \{1,2,\cdots,J\}$. This means that class-1 nodes are entitled to contend in all available slots while other class-j nodes are required to delay or shift their first attempt until the first slot of block k where $j = k$ This mechanism ensures that the high priority nodes will always have advantages since they are always entitled to contend in more blocks of slots. The access probability of all classes should be optimized simultaneously.

The performance of these three schemes will be compared to another scheme (Scheme IV), which strongly discriminated the lower class nodes. That is the lower priority nodes are allowed to access the slots if only if all higher priority nodes have succeeded. This makes a possibility that there is no slot left for the lower priority nodes. If there are still some slots left, then the lower priority nodes may begin their reservation accesses.

5.3 performance Analysis

In this section, we will present the mathematical analysis of our reservation model with different classes of nodes. To start the mathematical expression, we first present the analysis of the system with only one class. Let N be the number of active nodes competing for reservation in a frame of M slots. Let $S(N,M)$ denote the average number of successful reservations. To compute for $S(N,M)$, we write a recursive formula that expresses large problems in terms of smaller ones as follows:

$$S(n,m) = P_s(1+S(n-1,m-1)) + (1-P_s)S(n,m-1) \tag{5.1}$$

where $n \in \{0,1,2,\cdots,N\}$ and $m \in \{0,1,2,\cdots,M\}$. $P_s = np(1-p)^{(n-1)}$ is the probability of a successful reservation in a slot and $1-P_s$, is the probability of no successful reservation due to either collision or idle. By recursively repeating this relation, eventually we will arrive at very small problems, in which solutions are known and precisely they include $S(0,m) = 0$ and $S(n,0) = 0$.

When there are J classes of nodes in the system, the average number of successful reservations can be derived in a similar fashion as the single class case. Let N_j be the number of class-j nodes, and p_{jk} be the access probability of class-j nodes in block k, where $j \in \{1,2,\cdots,J\}$ and $k \in \{1,2,\cdots,K\}$. It should be noted that the values of p_{jk} essentially depend on m and predefined by each scheme. For example, Scheme II would set the values of p_{1k} to p_1 for $0 < m \leq M$ and set to 0 for $M_1 < m \leq M$ The average number of successful reservations of class-j nodes for any scheme is given by:

$$\begin{aligned}
&S_j(n_1, n_2, \cdots, n_J, m) \\
&= n_j p_{jk}(1-p_{jk})^{(n_j-1)} \prod_{l \neq j}(1-p_{lk})^{(n_l-1)}[1+S_j(n_1, n_2, \cdots, n_j-1, \cdots, n_J, m-1)] \\
&+ \sum_{i \neq j} n_j p_{jk}(1-p_{jk})^{(n_j-1)} \prod_{l \neq j}(1-p_{lk})^{(n_l-1)}[1+S_j(n_1, n_2, \cdots, n_j-1, \cdots, n_J, m-1)] \\
&+ \left[1-\sum_{j=1}^{J}\left(n_j p_{jk}(1-p_{jk})^{(n_j-1)} \prod_{l \neq j}(1-p_{lk})^{(n_l-1)}\right)\right][1+S_j(n_1, n_2, \cdots, n_J, m-1)] \quad (5.2)
\end{aligned}$$

Finally the overall average number of successful reservations of the system with J classes of nodes, , is expressed as:

$$S_T(N_1, N_2, \cdots, N_J, M) = \sum_{j=1}^{J} S_j(N_1, N_2, \cdots, N_J, M)$$

The ratio between the success rates of class-j nodes to that of class-1, r_j , is defined as a measure to indicate the success rate differentiation and can be expressed as:

$$r_j = \frac{S_j(N_1, N_2, \cdots, N_J, M)}{S_1(N_1, N_2, \cdots, N_J, M)}$$

5.4 Numerical Results and Discussion

To illustrate how effective the proposed model can provide proportional differentiation in terms of reservation success rates, while achieving the maximal efficiency of slot utilization, systems with $M = 16$ and $N = 8$ supporting two classes are tested with three different proportions of class-1 and class-2 nodes, i.e. $(N_1 = 2, N_2 = 6)$, $(N_1 = 4, N_2 = 4)$, and $(N_1 = 6, N_2 = 4)$. We summarize the actual mechanism used by each prioritization scheme for differentiating between classes of nodes, and the parameters for optimizing the overall successes in Table 1. For example, Scheme I applies the mechanism that increases the values of p_1 so that class-1 nodes obtain a better chance of success, hence lowering the values of r_2. At the same time, this scheme strategically decreases the values of p_2 such that the maximal value of S_T is accomplished for each increased value of p_1 . In all schemes, parameter settings are carefully selected to ensure that the success rates of class-1 nodes are always higher than that of class-2.

Table 5.1: Details of how parameters are selected and optimized for each scheme.

Scheme	Mechanism to differentiate classes	Parameters to optimize for maximal overall successes
I	Increases p_1	Decreases p_2
II	Varies $M-1$	Optimizes p_1 and p_2 independently
III	Varies $M-1$	Optimizes both p_1 and p_2 simultaneously
IV	Class-1 nodes finish first before Class-2 nodes can begin	Optimizes both and simultaneously19

Fig. 5.2 (a)-(c) illustrate the average number of successful reservations of class-1, class-2 and overall, respectively, when changing the probabilities p_1, p_2 and the number of users in each class. When nodes are not differentiated in classes and probability to make reservation is 0.1814 ($p_1 = p_2$) then we can achieve the maximum success reservation of 5.8148. When two classes are differentiated by increasing p_1 and decreasing p_2 , the success rates of class-2 nodes are always inversely affected, while the success rates of class-1 nodes exhibit various behaviors depending on the proportion of class-1 and class-2. If majority of nodes are from class 1, the success rates of class-1 tend to degrade significantly; this also contributes to significant drop in the overall success rates. However, at certain range of p_1, there is a gain for class-1 nodes, but very slightly. These results show that by adjusting the values of p_1 and p_2, we can control the success rates of each class with fine granularity, at the cost of overall success rates.

Fig. 5.3 shows the average number of successful reservations of class-1, class-2 and overall by adjusting M_1 as in Scheme II. Allocating different proportion of slots to class-1 and class-2 obviously is another means to differentiate the success rates between classes.

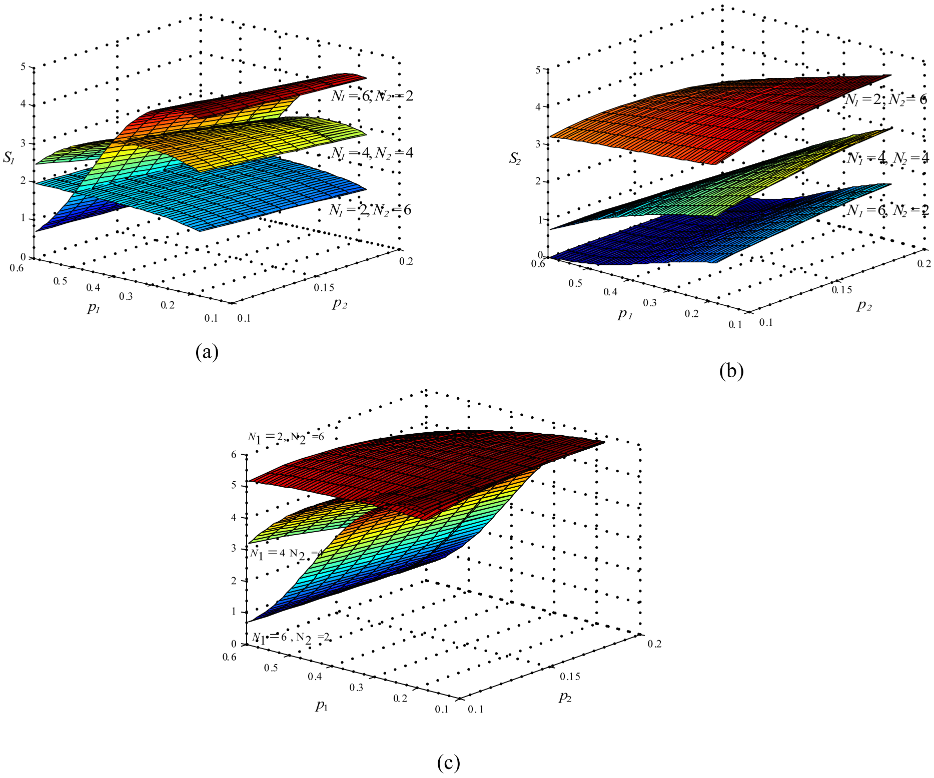


Figure 5.2: The average number of successful reservations of (a) class-1, (b) class-2 and (c) overall by adjusting p_1 and p_2 as in Scheme I.

Unlike changing p_1 and p_2, as in Scheme I, the adjustable parameter M_1 is constrained to only some discrete values. This becomes even more limited, when the majority of nodes are from class-1. Having revealed some useful characteristics of slot allocation and varying probabilities for controlling success rates, we will now directly compare the effectiveness of the three proposed schemes under different success rate differentiations.

Numerical results presented in terms of the normalized overall successful reservations as a function of r_2 for all four schemes are shown in Fig. 5.4. The proposed schemes can provide a control across the entire range of r_2 . This is in contrast to the scheme IV, in which only a single value of r_2 is achieved. Such a feature of our schemes serves well for the requirement of proportional differentiation between classes of nodes. All three schemes offer comparable performance and exhibit similar behavior in terms of success rates with slight variation in scheme II where there are a series of abrupt changes. This is due to the discrete nature of complete partitioning. Interestingly enough, Scheme IV exhibits the

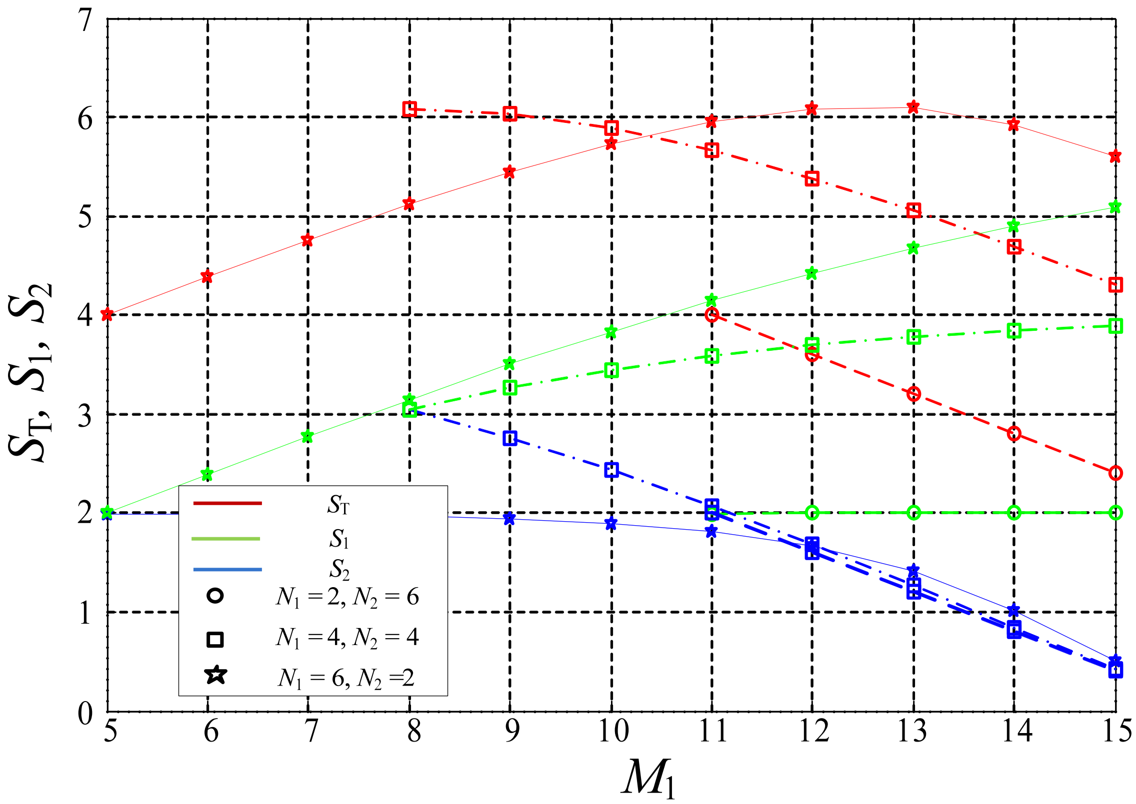


Figure 5.3: The average number of successful reservations of class-1, class-2 and overall by adjusting M_1 as in Scheme II.

highest overall success rates. In general, when prioritization is imposed, the overall success rates will degrade. Two factors that influent this degradation include whether the majority of nodes is of low-class and when the targeted value of r_2 is low. For the worst case, the overall success rate can drop below 50% if two out of eight nodes are of high-class and the target for r_2 is 0.1. These results suggest that there is a tradeoff between controllability and the overall success rates, when selecting prioritization schemes.

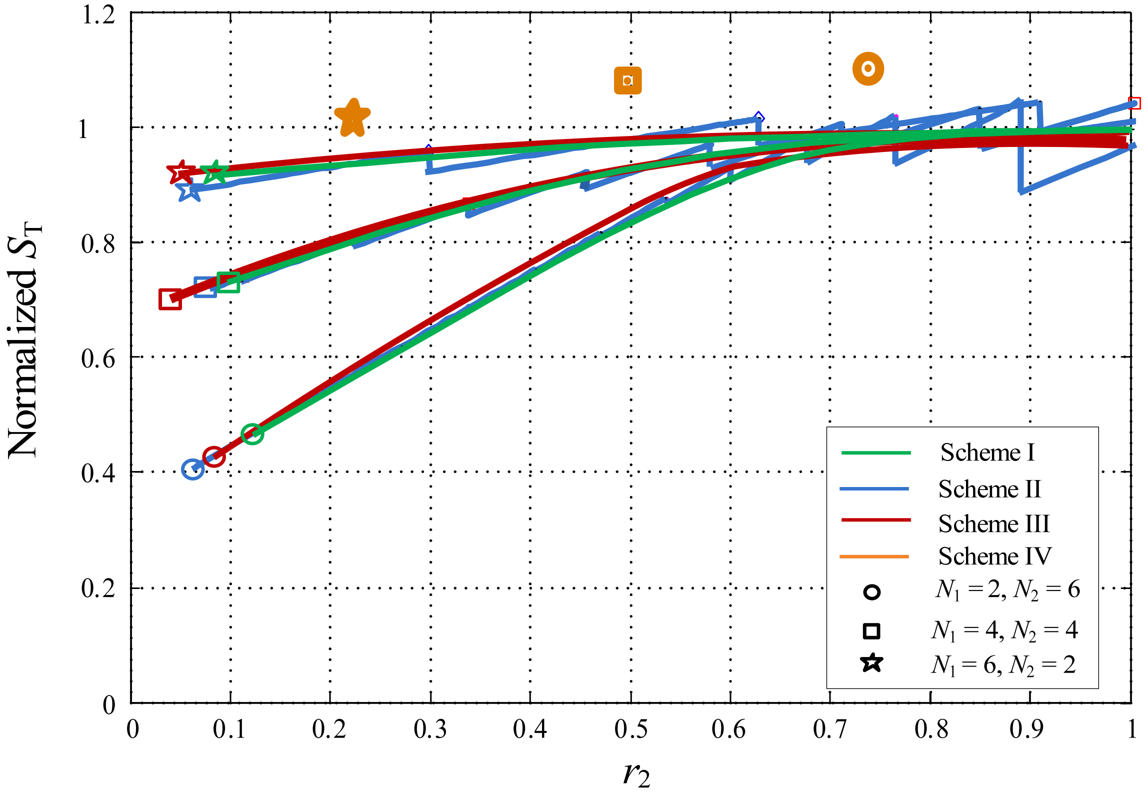


Figure 5.4: The normalized overall successful reservations against r_2 for M=16.

CHAPTER VI

Conclusions

This thesis investigated the performance of framed slotted Aloha and the tree algorithms with different types of feedback information; binary, ternary, and known multiplicity. Four fundamental mechanisms for resolving collision are introduced as basic building blocks for the construction of a wide range of random access MAC protocols. The delay analysis in terms of mean CRI length is conducted when the protocol is applied on the blocked access mode. The analytical model for finding the value of the mean CRI length is expressed in the recursive formula. Especially for the tree algorithm with the proposed skipping mechanism, the closed-form expression can be derived. Throughput of the proposed tree algorithm are analyzed under the blocked access and free access mode.

From the obtained results, the following conclusions can be drawn:

1. The proposed analytical evaluation has shown that the use of feedback information, if used efficiently, plays a vital role in delay performance improvement. For known multiplicity, the maximum achievable MST of 0.533 is obtained by our proposed random access protocol that is derived by the combination of splitting mechanism, adaptive frame size, slot-skipping type II, and non-uniform access. For the ternary feedback,the maximum achievable MST of 0.381 is accomplished by using splitting mechanism, static frame size, slot-skipping type I, and non-uniform access, which was the protocol proposed by [30, 45].

2. In the frame-based access protocols, the frame size is an important parameter to set to minimize the idle and collision slots. It reveals that the adaptive changing of the frame size in accordance to the number of collided packets is more effective compared to the

case when the fixed frame size.

3. The skipping strategy type II which skips predicted collision and idle slots is effective to be applied especially in the blocked access mode. However, it is not fully applicable for the case of free access mode since the idle slot cannot be predicted since it is not possible that new and collided packets can be distinguished.

4. This study also found that the blocked access tree algorithms with skipping type II is more superior compared to those free access algorithms since the contention is less severe by not allowing the newly generated packet to join then in progress collision resolution.

5. In the tree algorithm, slot skipping strategy will be more effective if it is applied with non uniform access probability.

6. For the framed slotted Aloha, a considerable improvement is clearly shown when the adaptive frame size is combined with the skip strategy type II. This can overcome the problem of exponential growth of the mean CRI length of the framed slotted Aloha with fixed frame size. This results in a great improvement of MST where the achievable MST is 0.408.

7. Based on the four fundamental mechanisms: i) splitting mechanism, ii) adaptive frame size mechanism, iii) slot-skipping mechanism and iv) non-uniform access mechanism., we are able to classify the framed-based random access protocols as illustrated in Fig. 6.1.

8. The development of the frame-based random access protocols and the relation between them can be drawn in a systematic diagram as depicted in 6.1. It shows that there are 13 variations of frame-based MAC protocols that can be developed with the aforementioned fundamental mechanisms. All the protocols with shaded boxes are the

contributions of this thesis.

Splitting	Frame Size	Slot Skipping	Access Probability	Rangking Mean CRI Length	MST		References
					Blocked	Free	
Yes	Adaptive	Type II	Uniform	3	0.528	0.4141	
			Non uniform	1	0.533	NA	
		No Skipping	Uniform	5	0.437	0.392	
	Fixed	Type I	Uniform	7	0.375	0.406	[15.30]
			Non Uniform	8	0.381	0.407	[15,30]
		Type II	Uniform	4	0.514	0.408	
			Non Uniform	2	0.532	0.409	
		No Skipping	Uniform	9	0.366	0.401	[12,13,14,15,16, 30,31,32]
No	Adaptive	Type II	Uniform	6	0.406	NA	
		No Skipping	Uniform	NA	NA	NA	[8,9]
	Fixed	Type I	Uniform	NA	NA	NA	
		Type II	Uniform	NA	NA	NA	
		No Skipping	Uniform	NA	NA	NA	[5,6,7]

Figure 6.1: Classification of framed-based random access protocols with the four fundamental mechanisms.

REFERENCES

[1] Peyravi, H. Multiple Access Control Protocols for the Mars Regional Network: A Survey and Assessments. Technical report Dept. of Math and Computer Science Kent State Univ, Sept. 1995.

[2] Giambene, G. Resource Management in Satellite Networks:Optimization and Cross-Layer Design, Springer, 2007.

[3] Gallager, R.G. A perspective on multiaccess channels. IEEE Transactions on Information Theory 31 (March 1985) : 124-142.

[4] Rom, R. and Sidi, M. Multiple Access Protocols: Performance and Analysis (Telecommunication Networks and Computer Systems), 1: Springer, 1990.

[5] Berstekas, D. ,Gallager, R.,Data Networks, 2nd Edition. New Jersey: Prentice-Hall, 1992.

[6] Wu, H. and Pan, Y. Medium Access Control in Wireless Networks (Wireless Networks and Mobile Computing : Nova Publisher, 2008.

[7] Abramson, N. The ALOHA System - Another Alternative for Computer Communications. Proc. Fall Joint Computer Conference. AFIPS Press., pp. 281-285, 1970.

[8] Roberts, L. G. and Lawrence, G. ALOHA Packet System With and Without Slots and Capture. Computer Communications Review 2, 5(April 1975) : 28–42.

[9] Sadhana, V. Stability of multiple access communication networks. Springer India 15(December 1990) : 365-380.

[10] Kleinrock, L. and Tobagi, F.A. Packet Switching in Radio Channels: Part I - Carrier Sense Multiple-Access Modes and Their Throughput Delay Characteristics IEEE Transactions on Communications COM-23(December 1975) : 1400-1416

[11] Kleinrock, L. and Lam, S. S. Packet Switching in a Multiaccess Broadcast Channel: Performance Evaluation.IEEE Transactions on Communications 23 (April 1975) : 410–423.

[12] Lam, S. S. and Kleinrock, L. Packet switching in a multiaccess broadcast channel: Dynamic control procedures. IEEE Transaction on Communications 23 (September 1975) : 891–904.

[13] Frigon, J., Leung, W. C. M., Chan, H., Bun, H. C. Dynamic Reservation TDMA Protocol for Wireless ATM Networks. IEEE Journal on Selected Areas in Communications 19 (2001): 370–383.

[14] Sudhakar, G. N. M., Georganas, N. D., and Kavehrad, M. Slotted Aloha and reservation Aloha protocols for very high-speed optical fiber local area networks using passive star topology. Journal of Lightwave Technology 9 (Oktober 1991): 1411-1422.

[15] Okada, H. Nakanishi, Y. and Igarashi, Y. Analysis of framed ALOHA channel in satellite packet switching networks. in Proceeding Fourth Znt. Conference Computer Communications. pp. 617-622, 1978.

[16] Schoute, F. C. Dynamic frame length ALOHA. IEEE Transaction on Communications COM-31 (April 1983) : 565–568.

[17] Bing, B. Stabilization of the Randomized Slotted ALOHA Protocol Without the Use of Channel Feedback Information. IEEE Communications Letters 4 (August 2000) : 249–251.

[18] Wieselthier, J. E., Michaels,L. A. , and Ephremides, A. An Exact Analysis and Performance Evaluation of Framed ALOHA with Capture and are with the Naval Research. IEEE Transactions on Communications 31 (February 1989): 125–137

[19] Prodanoff, Z. G. Optimal frame size analysis for framed slotted ALOHA based RFID networks. Computer Communications 33 (November 2010): 648653.

[20] Eom, J. B , Lee, T. J. Ronald, R., and Yener, A. An Efficient Framed-Slotted ALOHA Algorithm with Pilot Frame and Binary Selection for Anti-Collision of RFID Tags. IEEE Communications Letters 12 (November 2008): 561 – 563.

[21] Nazir, M.H., Xu, Y., Johansson, A., Optimal Dynamic Frame-Slotted Aloha, International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM), pp.1– 4: IEEE, 2011.

[22] Barletta, L. F.Borgonovo, Cesana, M. Performance of Dynamic-Frame-Aloha protocols : Closing the gap with tree protocols, in Med-Hoc-Net, pp. 9-16, 2011.

[23] Capetanakis, J. I. Tree algorithms for packet broadcast channels. IEEE Transactions on Information Theory IT-25 (September 1979) : 505-515.

[24] Tsybakov, B. S. and Mihailov, V. A. Slotted multiaccess packet broadcasting feedback channel. Problemy Peredachi Informatsii 14 (December 1978) : 32-59.

[25] Gallager, R. G. Conflict resolution in random access broadcast networks, in Proceeding AFOSR Workshop Communications Theory Applications. pp. 74-76, Provinceton, MA, 1978.

[26] Markowski, M.J. and Sethi, A.S. Real-time wireless communication using splitting protocols. Global Telecommunications Conference GLOBECOM 1997 , pp.16211625 : IEEE 1997.

[27] Huang, J. C. and Berger, T. Delay analysis of interval-searching contention resolution algorithms. IEEE Transactions onInformation Theory 31 (March 1985) : 264,273.

[28] Capetanakis, J. I. The Multiple Access Broadcast Channel: Protocol and Capacity Considerations. Ph.D. Dissertation, Department of Electrical Engineering, Massachusetts Institute of Technology, 1977.

[29] Capetanakis, J. I. Generalized TDMA: The Multi-Accessing Tree Protocol. IEEE Transactions on Communications COM-27(October 1979) : 1476-1484.

[30] Massey, J. L. Collision resolution algorithms and random access algorithms. Multi-User Communication Systems ser. CISM Course and Lecture Notes, G. Longo, Ed. New York: Springer Verlag 265 (1981) : 73-137.

[31] Molle, M. L. and Polyzos, G. C. Conflict Resolution Algorithms and their Performance Analysis, Technical Report CS93-300, Department of Computer Science and Engineering, University of California, San Diego, July 1993.

[32] Tsybakov, B. S. and Vvedenskaya, N. D. Stack-algorithm for random multiple access. Problemy Peredachi Informatsii 16(September 1980) : 80-94.

[33] Fayolle, G. and Hofri, M. On the Capacity of a Collision Channel Under Stack-based Collision Resolution Algorithm, Technical Report, March 1982.

[34] Raychaudhuh, D. Announced Retransmission Random Access Protocols. IEEE Transactions on Communications COM-33 (November 1985) : 1183-1190.

[35] Wu, C. S. and Chen, P. N. Improved collision resolution algorithms for multiple access channels with limited number of users, Technical Report, 1997.

[36] Mahravari, N. and Berger, T. Poisson multiple-access contention with binary feedback. IEEE Transactions on Information Theory IT-30 (Sept 1984) : 1745–751.

[37] Tsybakov, B. S. Resolution of a conflict of known multiplicity. Problem Information Transmission translated from Problemy Peredachi Informatsii 16 (June 1980) : 134–144.

[38] N. Mahravari, Random-access communication with multiple reception, IEEE Transactions on Information Theory, 36 (May 1990) : 614-622.

[39] Park, J. and Van der Schaar, M. Medium Access Control Protocols With Memory. *IEEE/ACM Trans. Networking* COM-18(December 2010) : 1921-1934.

[40] Georgiadis, L. and Papantoni-Kazakos, P. A collision resolution protocol for random access channels with energy detectors. IEEE Transactions on Communications COM-30 (November 1982) : 2413-2420.

[41] Pippenger, N. Bounds on performance of protocols for a multiple access broadcast channel. IEEE Transactions on Information Theory IT-27 (March 1981) : 145-151.

[42] Gulko, E. Tree-Based Multiaccess Protocols Where Collision Multiplicities are known. IEEE Transactions on Communications COM-33 (September 1985) : 999–1001.

[43] Panwar, S.S. On Achieving a Throughput of One for a Random Access Channel with Collisions of Known Multiplicity. IEEE International Symposium on Information Theory. Budapest, 1991, pp. 337, Budapest : IEEE, 1991.

[44] Tanenbaum, A. S. Computer Networks. Third Edition: Prentice Hall,1996.

[45] Mathys, P. and Flajolet, P. *Q*-ary collision resolution algorithms in random-access systems with free or blocked channel access. IEEE Transactions on Information Theory IT-31 (March 1985) : 217-243.

[46] Oie, Y., Suda, T., Miyahara, H., and Hasegawa, T., Throughput and delay analysis of free access tree algorithm with minislots. IEEE Transactions on Communications 38 (February 1990) : 137141.

[47] Szpankowski, W. An analysis of a contention resolution algorithm. Another approach. Journal Acta Informatica 24 (April 1987) : 173-190.

[48] Fayolle, G. et al. Analysis of a stack algorithm for random multiple-access communication. IEEE Transactions on Information Theory IT-31 (March 1985) : 244-254.

[49] Janssen, A. J. E. M. and de Jong, M. J. M. Analysis of contention tree algorithms. IEEE Transactions on Information Theory 46 (September 2000) : 2163–2172.

[50] Paris, B. P. and Aazhang, B. Near-optimum control of multiple-access collision channels. IEEE Transactions on Communications 40 (August 1992) : 1298–1309.

[51] Christen, C. A. Contention among three senders. IEEE Transactions on Information Theory IT-34(July 1988) : 851–857.

[52] Mehravi, N. On Poisson contention resolution with feedback based on conflict intensity. IEEE Transactions on Communications 36(April 1988) : 513–516.

[53] Yuang, M. C., Lo, B. C., and Chen, J.Y. Hexanary-Feedback Contention Access With PDF-Based Multiuser Estimation for Wireless Access Networks. IEEE Transactions on Communications 3 (January 2004) : 278–289.

[54] Cidon, I.and Sidi, M. Conflict multiplicity estimation and batch resolution algorithms. IEEE Transactions on Information Theory 34 (January 1998) : 101– 110.

[55] Tsybakov, B. Packet Multiple Access for Channel With Binary Feedback, Capture, and Multiple Reception. IEEE Transactions on Information Theory 50 (June 2004) : 1073-1083.

[56] Peeters, G. T. and Van Houdt, B. On the Maximum Stable Throughput of Tree Algorithms with Free Access. IEEE Transactions on Information Theory 55 (November 2009) : 5087–5099.

[57] Vvedenskaya, N. and Tsybakov, B. S. Random multiple access of packets to a channel with errors. Problemy Peredachi Informatsii 19 (1983) : 69-84, 1983.

[58] Liang, R. M. and Tan, H. H. On the error analysis of single-channel free-access collision resolution algorithms. in Proceeding Conference IEEE Aerospace. Montana, 2000, pp. 129140, Montana: IEEE, 2000.

[59] Vinel,A., Ni, Q., Staehle, D., and Turlikov, A. Capacity Analysis of Reservation-Based Random Access for Broadband Wireless Access Networks. IEEE Journal on Selected Areas in Communications 27 (February 2009): 172-181.

[60] Al-nahari, A. Y., El-Dolil, S. A., Dessouky, M. I., Abd El-Samie, F. E. Reservation-based dynamic admission control scheme for wideband code division multiple access systems. Journal of Central South University 9 (February 2012) : 393-401.

[61] ECMA-368 High Rate Ultra Wideband PHY and MAC Standard, ECMA, 3rd Edition, December 2008.

[62] Papantoni-Kazakos, P. Multiple-access algorithms for a system with mixed traffic: High and low priority. IEEE Transation on Communications 40 (March 1992) : 541555.

[63] Frigon, J. and Leung, V. C. M. A Pseudo-Bayesian Aloha Algorithm with Mixed Priorities Journal of Wireless Networks 7 (1999) : 2001.

[64] Chen, Y., Qiao, C., Hamdi, M. and Tsang, D. H. K. Proportional Differentiation: A Scalable QoS Approach. IEEE Communications Magazine. pp.52-58, June 2003.

[65] Xue, Q., Gong, W. and Ganz, A. Proportional Service Differentiation in Wireless LANs Using Spacing-based Channel Occupancy Regulation. Mobile Networks and Applications 11 (2006): 229240.

APPENDIX

Close Form of Mean CRI length of Skip Tree Algorithm

In this appendix we present the derivation of the mean CRI length of the skip tree algorithm in 4.

We first define the exponential generating function of $L_{N,Q}$ by

$$L(z) \triangleq \sum_{n=0}^{\infty} L_{N,Q} \frac{z^N}{N!} \tag{1}$$

Multiplying the recursive-form of $L_{N,Q}$ in equation 4.14 by $z^N/N!$ and summing both sides for $N \geq 2$ we obtain

$$\sum_{N=2}^{\infty} L_{N,Q} \frac{z^N}{N!} = \sum_{N=2}^{\infty} \frac{z^N}{N!} - \sum_{N=2}^{\infty} \left(C_{N,Q} - I_{N,Q} + \sum_{q=1}^{Q} \sum_{i_q=0}^{N} \binom{N}{i_q} p_q^{i_q} \left(1-p_q\right)^{N-i_q} L_{i_q,Q} \right) \frac{z^N}{N!} \tag{2}$$

Using the expression in (1), we will proceed the derivation as follows

$$\begin{aligned}
L(z) &= L_0 \frac{z^0}{0!} + L_1 \frac{z^1}{1!} + \sum_{N=2}^{\infty} L_{N,Q} \frac{z^N}{N!} \\
L(z) &= 1 + z + \sum_{N=2}^{\infty} L_{N,Q} \frac{z^N}{N!} \\
&= \sum_{N=2}^{\infty} \left(1 - C_{N,Q} - I_{N,Q} + \sum_{q=1}^{Q} \sum_{i_q=0}^{N} \binom{N}{i_q} p_q^{i_q} \left(1-p_q\right)^{N-i_q} L_{i_q,Q} \right) \frac{z^N}{N!}
\end{aligned}$$

$$L(z) = \sum_{N=2}^{\infty}\left(1 - \sum_{i_Q=2}^{N}\binom{N}{i_Q} p_Q^{i_j}(1-p_Q)^{N-i_j} - \sum_{i=1}^{Q-1}\left(\sum_{q=1}^{i} p_j\right)^N\right.$$

$$\left. + \sum_{q=1}^{Q}\sum_{i_q=0}^{N}\binom{N}{i_q} p_q^{i_q}\left(1-p_j\right)^{N-i_q} L_{i_q,Q}\right)\frac{z^N}{N!}$$

$$= \sum_{N=2}^{\infty}\left(\begin{array}{l} 1 - \sum_{i_Q=0}^{N}\binom{N}{i_Q} p_Q^{i_q}(1-p_Q)^{N-i_Q} + (1-p_Q)^N + Np_Q(1-p_Q)^{N-1} \\ -\sum_{i=1}^{Q-1}\left(\sum_{j=1}^{i} p_j\right)^N + \sum_{q=1}^{Q}\Sigma_{i_q=0}^{N}\binom{N}{i_q} p_q^{i_q}\left(1-p_q\right)^{N-i_q} L_{i_q,Q} \end{array}\right)\frac{z^N}{N!}$$

$$= \sum_{N=0}^{\infty}\frac{z^N}{N!} - \sum_{N=0}^{\infty}\sum_{i_Q=0}^{N}\binom{N}{i_q} p_Q^{i_Q}(1-p_Q)^{N-i_Q}\frac{z^N}{N!} + \sum_{N=0}^{\infty}\frac{z^N}{N!}(1-p_Q)^N$$

$$+ \sum_{N=0}^{\infty}\frac{z^N}{N!}Np_Q(1-p_Q)^{N-1} + \sum_{N=0}^{1}\sum_{i_Q=0}^{N}\binom{N}{i_Q} p_Q^{i_Q}(1-p_Q)^{N-i_Q}\frac{z^N}{N!}$$

$$- \sum_{i=1}^{Q-1}\sum_{N=0}^{\infty}\left(\sum_{j=1}^{i} p_j\right)^N \frac{z^N}{N!} + \sum_{i=1}^{Q-1}\sum_{N=0}^{1}\left(\sum_{j=1}^{i} p_j\right)^N \frac{z^N}{N!}$$

$$+ \sum_{q=1}^{Q}\sum_{N=0}^{\infty}\sum_{i_j=0}^{N}\binom{N}{i_q} p_q^{i_q}\left(1-p_q\right)^{N-i_q} L_{i_q}\frac{z^N}{N!} - \sum_{q=1}^{Q}\sum_{N=0}^{1}\sum_{i_q=0}^{N}\binom{N}{i_q} p_q^{i_q}\left(1-p_q\right)^{N-i_q} L_{i_q,Q}\frac{z^N}{N!}$$

$$=\sum_{N=0}^{\infty}\frac{z^N}{N!}-\sum_{N=0}^{\infty}\sum_{i_Q=0}^{N}\begin{pmatrix} N \\ i_Q \end{pmatrix} p_Q^{i_q}(1-p_Q)^{N-i_Q}\frac{z^N}{N!}+\sum_{N=0}^{\infty}\frac{((1-p_Q)z)^N}{N!}$$

$$+\frac{p_Q}{(1-p_Q)}\sum_{N=0}^{\infty}\frac{((1-p_Q)z)^N}{(N-1)!}-\sum_{i=1}^{Q-1}\sum_{N=0}^{\infty}\left(\sum_{j=1}^{i}p_j\right)^N\frac{z^N}{N!}$$

$$+\left((Q-1)+\sum_{i=1}^{Q-1}\sum_{q=1}^{i}p_q z\right)+\sum_{q=1}^{Q}\sum_{N=0}^{\infty}\sum_{i_q=0}^{N}\binom{N}{i_q}p_q^{i_q}\left(1-p_q\right)^{N-i_q}L_{i_q}\frac{z^N}{N!}$$

$$-\left(Q+\sum_{q=1}^{Q}\left(1-p_q\right)z+\sum_{j=1}^{Q}p_q z\right)$$

$$=e^z-\sum_{N=0}^{\infty}\sum_{i_Q=0}^{N}\frac{N!}{i_Q!(N-i_Q)!}p_Q^{i_Q}\left(1-p_Q\right)^{N-i_Q}\frac{z^N}{N!}+\sum_{N=0}^{\infty}\frac{((1-p_Q)z)^N}{N!}$$

$$+\frac{p_Q}{(1-p_Q)}\sum_{N=0}^{\infty}\frac{((1-p_Q)z)^N}{(N-1)!}-\sum_{i=1}^{Q-1}\sum_{N=0}^{\infty}\left(\sum_{j=1}^{i}p_j\right)^N\frac{z^N}{N!}+\left((Q-1)+\sum_{i=1}^{Q-1}\sum_{j=1}^{i}p_j z\right)$$

$$+\sum_{q=1}^{Q}\sum_{N=0}^{\infty}\sum_{i_q=0}^{N}\frac{N!}{i_q!(N-i_q)!}p_q^{i_q}\left(1-p_q\right)^{N-i_q}L_{i_q}\frac{z^N}{N!}-(Q+Qz)$$

$$=e^z-\sum_{N=0}^{\infty}\sum_{i_j=0}^{N}\frac{p_Q^{i_Q}}{i_Q!}(1-p_Q)^{N-i_j}\frac{z^{N+i_Q-i_Q}}{(N-i_Q)!}+e^{z(1-p_Q)}+\frac{p_Q}{(1-p_Q)}z(1-p_Q)e^{z(1-p_Q)}$$

$$-\sum_{i=1}^{Q-1}\sum_{N=0}^{\infty}\left(\sum_{j=1}^{i}p_j\right)^N\frac{z^N}{N!}+\left((Q-1)+\sum_{i=1}^{Q-1}\sum_{j=1}^{i}p_j z\right)$$

$$+\sum_{j=1}^{Q}\sum_{N=0}^{\infty}\sum_{i_j=0}^{N}\frac{p_j^{i_j}}{i_j!}L_{i_j}\left(1-p_j\right)^{N-i_j}\frac{z^{N+i_j-i_j}}{(N-i_j)!}-(Q+Qz)$$

$$=e^z-\sum_{N=0}^{\infty}\sum_{i_Q=0}^{N}\frac{(p_Q z)^{i_Q}}{i_Q!}\frac{(z(1-p_Q))^{N-i_Q}}{(N-i_Q)!}+(1+p_Q z)e^{z(1-p_Q)}-\sum_{i=1}^{Q-1}\sum_{N=0}^{\infty}\frac{\left(z\sum_{j=1}^{i}p_j\right)^N}{N!}$$

$$+\left((Q-1)+\sum_{i=1}^{Q-1}\sum_{j=1}^{i}p_j z\right)+\sum_{j=1}^{Q}\sum_{N=0}^{\infty}\sum_{i_j=0}^{N}\frac{\left(p_q z\right)^{i_q}}{i_q!}L_{i_q}\frac{\left(z\left(1-p_q\right)\right)^{N-i_q}}{(N-i_q)!}-Q\left(1+z\right)$$

$$
\begin{aligned}
=&e^{z}-e^{zp_Q}e^{z(1-p_Q)}+(1+p_Qz)e^{z(1-p_Q)}-\sum_{i=1}^{Q-1}\sum_{N=0}^{\infty}\frac{\left(z\sum_{j=1}^{i}p_j\right)^N}{N!} \qquad (3)\\
&+(Q-1)+\sum_{i=1}^{Q-1}\sum_{j=1}^{i}p_jz+\sum_{j=1}^{Q}L\left(p_qz\right)\cdot e^{z\left(1-p_q\right)}-Q\left(1+z\right)\\
=&e^{z}-e^{p_Qz}e^{z\left(1-p_Q\right)}+(1+p_Qz)e^{z(1-p_Q)}-\sum_{i=1}^{Q-1}e^{z\sum_{j=1}^{i}p_j}+(Q-1)\\
&+\sum_{i=1}^{Q-1}\sum_{j=1}^{i}p_jz+e^{z}\sum_{q=1}^{Q}L\left(p_qz\right)\cdot e^{-p_qz}-Q\left(1+z\right)\\
=&\ e^{z}+e^{z}\sum_{q=1}^{Q}L\left(p_qz\right)\cdot e^{-p_jz}-e^{z}-\sum_{i=1}^{Q-1}e^{z\sum_{j=1}^{i}p_j}+(1+p_Qz)e^{z(1-p_Q)}\\
&+(Q-1)+z\sum_{i=1}^{Q-1}\sum_{j=1}^{i}p_j-Q\left(1+z\right)
\end{aligned}
$$

We then define

$$L^*(z)\triangleq\sum_{k=0}^{\infty}L_k^*z^k\triangleq e^{-z}L(z) \qquad (4)$$

By using (4), we transform (3) to

$$
\begin{aligned}
e^{-z}L(z)=&\sum_{j=1}^{Q}L\left(p_jz\right)\cdot e^{-p_jz}-Q\left(1+z\right)e^{-z}+(1+p_Qz)e^{-zp_Q}\\
&-\sum_{i=1}^{Q-1}e^{-z\left(1-\sum_{j=1}^{i}p_j\right)}+(Q-1)\,e^{-z}+z\sum_{i=1}^{Q-1}\sum_{j=1}^{i}p_je^{-z} \qquad (5)
\end{aligned}
$$

Expanding both sides of (6) to a taylor series around $z=0$ and letting $L^*(z)=\sum_{k=0}^{\infty}B_k{}^*z^k$

and equating the coefficient of z^k we get for $k \geq 2$

$$\sum_{k=0}^{\infty} L_k^* z^k - \sum_{k=0}^{\infty} \sum_{j=1}^{Q} L_{k\cdot}^* \left(p_q z\right)^k = -Q \sum_{k=0}^{\infty} \left(\frac{(-1)^k}{k!} + \frac{(-1)^{k-1}}{(k-1)!} \right) z^k$$
$$+ \sum_{k=0}^{\infty} \left(\frac{(-1)^k p_Q^k}{k!} + p_Q \frac{(-1)^{k-1} p_Q^{k-1}}{(k-1)!} \right) z^k$$
$$- \sum_{k=0}^{\infty} \frac{\sum\limits_{i=1}^{Q-1} \left(1 - \sum\limits_{j=1}^{i} p_j \right)^k (-1)^k z^k}{k!}$$
$$+ (Q-1) \sum_{k=0}^{\infty} \frac{(-1)^k z^k}{k!} + \sum_{k=0}^{\infty} \left(\frac{\sum\limits_{i=1}^{Q-1} \sum\limits_{j=1}^{i} p_j (-1)^{k-1}}{(k-1)!} \right) z^k$$

$$L_k^* \left(1 - \sum_{q=1}^{Q} \left(p_q\right)^k \right) = -Q \left(\frac{(-1)^k (k-1)! + (-1)^{k-1} k!}{k!(k-1)!} \right) + \frac{(-1)^k p_Q^k - (-1)^{k-1} k p_Q^k}{k!}$$
$$- \frac{(-1)^k \sum\limits_{i=1}^{Q-1} \left(1 - \sum\limits_{j=1}^{i} p_j \right)^k - (-1)^k (Q-1) + (-1)^k k \sum\limits_{i=1}^{Q-1} \sum\limits_{j=1}^{i} p_j}{k!}$$
$$= Q (-1)^k \frac{k-1}{k!} + (-1)^k p_Q^k \frac{(1-k)}{k!}$$
$$- (-1)^k \frac{\sum\limits_{i=1}^{Q-1} \left(1 - \sum\limits_{j=1}^{i} p_j \right)^k - (Q-1) + k \sum\limits_{i=1}^{Q-1} \sum\limits_{j=1}^{i} p_j}{k!}$$

$$L_k^* = (-1)^k \frac{Q(k-1) + p_Q{}^k (1-k) - \sum\limits_{i=1}^{Q-1} \left(1 - \sum\limits_{j=1}^{i} p_j \right)^k - (Q-1) + k \sum\limits_{i=1}^{Q-1} \sum\limits_{j=1}^{i} p_j}{\left(1 - \sum\limits_{q=1}^{Q} \left(p_q\right)^k \right) k!}$$

By the definition of (1) and (4) we obtain

$$\sum_{N=0}^{\infty} L_{N,Q}\frac{z^N}{N!}{=}L(z) \triangleq e^Z L(z) = \sum_{l=0}^{\infty}\frac{z^l}{l!}\sum_{j=1}^{Q} L_k^* z^k$$
$$= \sum_{l=0}^{\infty}\sum_{k=0}^{\infty}\frac{L_k^*}{l!} z^{l+k}$$
$$\sum_{N=0}^{\infty} L_{N,Q}\frac{z^N}{N!} = \sum_{N=0}^{\infty}\sum_{k=0}^{N}\frac{L_k^*}{(N-k)!} z^N$$

By equating the coefficient of z^N, we obtain

$$\frac{L_{N,Q}}{N!} = \sum_{k=0}^{N}\frac{L_k^*}{(N-k)!}$$

$$L_{N,Q}{=}1+\sum_{k=0}^{N}\frac{N!\,(-1)^k\left\{Q\,(k-1)+p_Q^k\,(1-k)-\sum\limits_{i=1}^{Q-1}\left(1-\sum\limits_{j=1}^{i} p_j\right)^k-(Q-1)+k\sum\limits_{i=1}^{Q-1}\sum\limits_{j=1}^{i} p_j\right\}}{(N-k)!k!\left(1-\sum\limits_{q=1}^{Q}\left(p_q\right)^k\right)}$$

$$= 1+\sum_{k=0}^{N}\binom{N}{k}\frac{(-1)^k\left\{Q\,(k-1)+p_Q^k\,(1-k)-\sum\limits_{i=1}^{Q-1}\left(1-\sum\limits_{j=1}^{i} p_j\right)^k-(Q-1)+k\sum\limits_{i=1}^{Q-1}\sum\limits_{j=1}^{i} p_j\right\}}{\left(1-\sum\limits_{q=1}^{Q}\left(p_q\right)^k\right)}$$

Biography

Robithoh Annur was born in Klaten, Central Java, Indonesia. She received her B.Eng in electrical engineering from Gadjah Mada University (UGM), Indonesia, in 2006. in 2010, she started her Doctoral degree at Department of Electrical Engineering, Chulalongkorn University, Thailand under the support of AUN/SEED-Net (www.seed-net.org). Her main research interest is on multiple acccess control protocols design.

List of Publications

1. R. Annur, P. Vonlopvisut, K. Wannakong, S. Nakpeerayuth, J-i. Takada, L. Wuttisittikulkij, "A study of contention resolution algorithms with different types of feedback", in the Fifth International Workshop on Ad Hoc, Sensor and P2P Networks in conjuction with International Symposium on Autonomous Decentralized System (ISADS) 2013, Mexico City, March 5-8, 2013

2. N. Wattanamongkhol, W. Srichavengsup, P. Vanichchanunt, R. Annur, J-i Takada, L. Wuttisittikulki, "Analysis of Effect of User Misbehaviours on the Reservation-Based MAC Protocols in Wireless Communication Networks", in IEICE TRANSACTIONS on Communications Vol.E95-B No.9 pp.2794-2806, Sep. 2012

3. R. Annur, L. Wuttisittikulkij, S. Nakpeerayuth, W. Srichavengsup, J.Takada and M. Saadi, "A Study on Energy Efficient MAC Protocols for Wireless Body Area Networks, in EECON 34, Chonburi, Thailand, Nov 30 Dec 2, 2011.

4. R. Annur, W. Srichavengsup, J-i. Takada, L. Wuttisittikulkij, "Improved Performance of the Tree Algorithm for Long Propagation Delay Systems by Using Accumulated Feedback", in International Symposium on Multimedia and Communication Technology 2011, Sapporo, September 1 - 2, 2011

5. R. Annur, N. Wattanamongkhol, S. Nakpeerayuth, L.Wuttisittikulkij, "Applying the Tree Algorithm with Prioritization for Body Area Networks" in the Fourth International Workshop on Ad Hoc, Sensor and P2P Networks in conjuction with International Symposium on Autonomous Decentralized System (ISADS) 2011, Kobe, June 29 July 1, 2011.

6. K. Wannakong, R. Annur, N. Wattanamongkhol, T. Issariyakul, L. Wuttisittikulkij and S. Nakpeerayuth, "Performance Improvement of Tree Algorithm with Additional Feedback Information", in ICSE 2011, Yangon, Myanmar, Dec., 2011.

7. K. Wannakong, N. Wattanamongkhol, R. Annur, L. Wuttisittikulkij, P. Vanichchanunt and W. Srichavengsup, "Performance Comparison of Contention Resolution Schemes for Wireless MAC protocols", in ISMAC 2010, Manila, Philippines, Sep., 2010.