

ABSTRACT

Title of thesis: INCENTIVE COMPATIBLE
 MEDIUM ACCESS CONTROL
 IN WIRELESS NETWORKS

Nassir BenAmmar, Master of Science, 2006

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

The current IEEE 802.11 medium access control standard is being deployed in coffee shops, airports and even across major cities. The terminals accessing these wi-fi access points do not belong to the same entity, as in corporate networks, but are usually individually owned and operated. Entities sharing these network resources have no incentive in following protocol rules other than to optimize their overall utility, usually a function of throughput and delay. In this thesis, we discuss the shortfalls of the current IEEE 802.11 standard in environments where terminals are competing for a common bandwidth resource and we introduce a new MAC protocol designed with the above considerations. Thus the new Incentive Compatible MAC (ICMAC) protocol uses Vickrey auction to allocate time slots and is better suited for these open environments, without compromising the overall network performance.

INCENTIVE COMPATIBLE MEDIUM ACCESS CONTROL
IN WIRELESS NETWORKS

by

Nassir BenAmmar

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2006

Advisory Committee:

Professor John S. Baras, Chair/Advisor
Professor Sennur Ulukus
Professor Richard J. La

© Copyright by
Nassir BenAmmar
2006

DEDICATION

To my parents

ACKNOWLEDGMENTS

I am grateful to my advisor, Dr. John Baras for his guidance and continuous support and for allowing me to work on various interesting projects. I thank Dr. Sennur Ulukus and Dr. Richard La for agreeing to serve on my committee.

I want to thank my friends and my colleagues from the SEIL lab and CSHCN for their fruitful discussions and support.

I gratefully acknowledge the financial support I received for my graduate studies and research work from the Graduate School and from the Institute for Systems Research. This work was partially supported by the ARO under Award No DAAD19-01-01-1-0494 and partially prepared through collaborative participation in the Communications and Networks Consortium sponsored by the ARL under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011 through the University of Maryland, College Park.

TABLE OF CONTENTS

List of Tables	v
List of Figures	vi
1 Introduction	1
1.1 Related Work	3
2 The Distributed Coordination Function of IEEE 802.11	7
2.1 Exponential Backoff Mechanism	8
2.2 Shortfalls of the Random Backoff Time	9
3 Bayesian Games and Protocol Design	14
3.1 Random Access Nash Equilibrium	15
3.2 The Revelation Principle	18
3.3 Truth Telling Second Price Auction	19
3.4 Vickrey Auction and Time Slot Allocation	20
4 Incentive Compatible Medium Access Control	23
4.1 Control Packets	25
4.2 Time Slot Valuation	25
5 ICMAC Performance and Design Parameters	29
5.1 Performance	29
5.1.1 IEEE 802.11 Markov Chain Model	35
5.2 Design	38
5.2.1 Fragment Size	39
5.2.2 Number of Slots per Round	44
6 Simulation Results	47
6.1 ICMAC Design	47
6.2 Multiple Packet Scheduling	48
6.3 ICMAC vs IEEE 802.11	50
6.4 Communication Cost	52
6.4.1 Same User Types	52
6.4.2 Different User Types and Qos	55
7 Conclusion	60
Bibliography	62

LIST OF TABLES

3.1	Vickrey Auction Example	21
5.1	Packet sizes and Parameters	30
5.2	Parameter Definition	37
5.3	Optimal Slot Number and Length Pair	46
6.1	Optimal Fragment Size	48
6.2	Valuation Function Parameters	52
6.3	Valuation Function Parameters	56

LIST OF FIGURES

2.1	IEEE 802.11 Backoff Operation	9
2.2	Cooperation vs Noncooperation in IEEE 802.11	12
3.1	3 Stations Normal Form Game	16
4.1	ICMAC Protocol	23
4.2	Valuation Function	26
4.3	Queue Length Dependent Valuation	27
4.4	Demand Curves	28
5.1	ICMAC Round Duration	31
5.2	ICMAC Throughput	34
5.3	Throughput and Round Duration	35
5.4	ICMAC and IEEE 802.11 throughput	38
5.5	ICMAC Overhead	39
5.6	Optimal Data Fragment Size	42
5.7	Mixed Traffic Optimal Solution	44
6.1	Individual Scheduling	48
6.2	Multiple Scheduling	49
6.3	Multiple Scheduling	50
6.4	IEEE 802.11 and ICMAC Throughput	51
6.5	Self Adjusting Price	53
6.6	Stations Queue Length	54
6.7	Average Price Response to Load	55
6.8	Medium Access Delay	57

6.9 Valuation	58
6.10 Medium Access Delay	59

Chapter 1

Introduction

Most network protocols today are designed with the objective of maximizing performance of the network with respect to a set of network criteria, typically a function of throughput and delay, with the assumption that all participating entities of the network will follow protocol rules. This assumption has not been a major issue in wired networks due to the reliable medium and the abundance of bandwidth. However, this is not the case in wireless networks due to the broadcast nature of the wireless medium and the stringent bandwidth limitation. It has been shown that the de-facto medium access control for wireless networks, in particular the IEEE 802.11 protocol [1], suffers from many security weaknesses. A significant amount of work has been done to improve this MAC protocol. Security issues were of various types: Some involved the mechanism of association and authentication; others were at the message encryption protocol [2]. However, the focus of this paper is on the inherent access control mechanism. Different access techniques have been used in multiuser communication allowing communicating entities to share common bandwidth. Time division multiple access divides the time axis into time slots and assigns individual slots to various users in a round robin fashion. Similarly, frequency division multiple access divides the frequency domain into channels used by various terminals. Both these fixed allocation access schemes are not appropriate

for data traffic as traffic is of bursty nature and results in wasted resources when users are assigned slots but have no traffic to send. Code division multiple access and frequency division multiple access take advantage of both frequency and time domain by means of spreading codes allowing concurrent transmissions. The multi-user communication techniques described above are broadly used in cellular networks, where the network is designed to sustain a given number of users at any given time. Usually these networks are overdesigned and augmented with user demands and new applications. Complexity is mainly at the base station; though, this trend is starting to change as end terminals become more powerful. Cellular networks were initially designed for voice traffic only and to sustain constant bit rate. However new data services are starting to emerge in cell phones, relying on dynamic allocation of resources. Another widely used multiuser access mechanism for wireless data networks is random multiple access. As the name indicates, users access the channel at random. The simplest form of random access is ALOHA, where a node access the channel if it has a data packet to transmit and waits a random number of slots if it experiences a collision. Progressively, more techniques and improvements have been added to prevent collision at the access channel. MACA, MACAW and IEEE 802.11 are examples of protocols incorporating some of these collision avoidance techniques. Physical carrier sensing, virtual carrier sensing and exponential backoff timer are all used in IEEE 802.11 distributed coordination function (DCF) in order to reduce collision rate and to get a better network throughput [1]. Due to the random nature of channel access, stations have an incentive to deviate from protocol rules by altering transmission and backoff probabilities to gain better performance.

Noncooperative behavior in a random access MAC has been previously addressed. In the next section we address some of these studies.

1.1 Related Work

In [3], the authors studied the stability region of a slotted ALOHA system with selfish users for a general multipacket reception model. In the model considered, the users enter the game according to a random process. User arrivals at every slot are independent and identically distributed with expected number of arrivals per slot λ . The users participating in the game have perfect information on the number of competing stations and they all choose a symmetric transmission probability $\sigma(n)$. A user gets a payoff of 1 for a successful transmission and the transmission cost is taken to be c . All users have the same per-slot discount rate of δ . Using Nash equilibrium and condition on the Markov Chain drift, the stability region is shown to be

$$\lambda < e^{-\hat{\gamma}} \sum_{k=1}^{\infty} \frac{\hat{\gamma}^k}{k!} r_k.$$

with $\hat{\gamma}$ solution to

$$e^{-\gamma} \sum_{k=1}^{\infty} \frac{\gamma^{k-1}}{k!} r_k = c$$

r_k is the expected number of successes when k stations transmit. For conventional collision channel it reduces to $\lambda < -c \ln(c)$. For a q-frequency hopping model to $\lambda < -c q \ln(c)$.

In [4, 5], the authors consider a slotted ALOHA network. Each user $i \in \{1, \dots, N\}$ has a utility function U_i , a user mean arrival rate γ_i , a peak instantaneous

arrival rate π_i and a user willingness to pay per packet for the initial γ_i bandwidth m_i . The throughput unit is packets/slot. The utility is a function of throughput θ_i and is described as

$$U_i(\theta_i) = \begin{cases} m_i\theta_i, & \text{if } 0 \leq \theta_i \leq \gamma_i \\ m_i\gamma_i \left(\log \left(\frac{\theta_i}{\gamma_i} \right) + 1 \right), & \text{if } \gamma_i < \theta_i \leq \pi_i \\ m_i\gamma_i \left(\log \left(\frac{\pi_i}{\gamma_i} \right) + 1 \right), & \text{if } \pi_i < \theta_i \end{cases}$$

In this model, the network charges the station $M\$$ for each successfully transmitted packet. At every iteration and when the network reaches equilibrium, the price per packet is updated in order to drive the throughput to $T(M)$ according to

$$M^{k+1} = M^k + \kappa \left(\sum_{i=1}^N \theta_i^*(M^k) - T(M^k) \right)$$

with,

$$\theta_i^*(M) = (U_i')^{-1}(M) = \arg \max_{\theta} (U_i(\theta) - M\theta)$$

the solution of user i payoff maximization. In the game proposed, at every iteration, each user updates its transmission probability according to

$$\begin{aligned} q_i^{\ell+1} &= \arg \max_{0 \leq q \leq 1} U_i \left(q \prod_{j \neq i} (1 - q_j^\ell) \right) - Mq \prod_{j \neq i} (1 - q_j^\ell) \\ &= \min \left\{ (U_i')^{-1}(M) / \prod_{j \neq i} (1 - q_j^\ell), 1 \right\} \end{aligned}$$

Note user i throughput is $\theta_i^\ell = q_i^\ell \prod_{j \neq i} q_j^\ell$. At every iteration, each user i also advertises its transmission probability q_i .

By using a Lyapunov function, the authors prove that under certain condition the transmission probability vector q converges to a Nash equilibrium given by a

solution of

$$q_i(1 - q_i)^{N-1} - y_i \prod_{j \neq i} \left(1 + \left(\frac{y_j}{y_i} - 1 \right) q_i \right) = 0, \text{ for all } i$$

with $y_i = (U'_i)^{-1}(M)$. The convergence of the price strategy has also been considered using another Lyapunov function; however, only users with $y_i > 0$ were taken into account. When all users are considered, we do not expect the solution to be stable as some users will be switching on and off when M^* is near m_i . The model also assumes that all stations are always backlogged and that is known by all stations.

In [6], a cooperative team problem and a noncooperative game were considered for an ALOHA network with finite stations and packet arrival rate of q_a . The authors use Markov chain models and numeric results to solve for the optimum retransmission probabilities for networks with 2, 3 and 4 stations. For the cases considered, with homogenous users, the retransmission probability solution decreases as function of q_a for the team problem leading to an increase in throughput. However, the Nash equilibrium solution for the game lead to an initial increase of throughput but then to a dramatic drop as the arrival rate approached 1. They also studied the effect of the transmission cost associated to the battery power. An additional cost associated to each transmission may bring the game equilibrium throughput to the cooperative team throughput. Note that the additional cost is due to transmission and not delivered packets as in [4, 5]. The transmission cost is usually fixed as it is associated to processing and battery power and the network or AP is not capable of charging by transmission given that it cannot identify the senders during a collision.

The work discussed above focuses on equilibria of ALOHA networks using

different models. In chapter 2, we address the vulnerability of the current IEEE 802.11 standard and the gain from noncooperation. In chapter 3, we use game theory to explain the emergent behavior of rational entities in a random access channel and its effect on throughput. The findings naturally lead to an auction mechanism to alleviate some of the problems associated with random access. In chapters 4 and 5, we introduce a new Incentive Compatible Medium Access Control scheme and discuss performance and design parameters. Finally, we present simulation results pertaining to design and performance.

Chapter 2

The Distributed Coordination Function of IEEE 802.11

The IEEE 802.11 standard [1] has two functions, the Point Coordination Function (PCF) and the Distributed Coordination Function (DCF). The PCF is an optional function of the standard and is not even implemented in many wireless routers. It is a contention free access mode where the AP regulates the access to the channel according to its polling list in a round robin fashion. The polling list is updated when a station gets associated with the AP and requests to be polled. PCF results in wasted bandwidth when polled stations have no traffic to send. The DCF is contention based access mode and uses CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance). DCF is more appropriate in data networks with bursty traffic than its PCF counterpart. Thus, in the remaining of the thesis we focus on the distributed coordination function of IEEE 802.11.

The DCF has two access modes, the RTS/CTS mode and the basic mode. In the RTC/CTS mode, a node with a packet to transmit waits a random number of time slots before it tries to reserve the wireless medium. The medium reservation is done by the exchange of a Request to Send (RTS) and Clear to Send (CTS) messages. With this exchange of messages, the other nodes are notified that the medium will be busy for a duration advertised in the RTS packet and then updated in the CTS packets. Thus terminals in the vicinity of the transmitter as well as those

in the vicinity of the receiver are aware of the transmission (assuming these messages are detected correctly) and update their Network Allocation Vector (NAV). NAV informs a node about an ongoing transmission without continuously sensing the medium. This is referred to as virtual transmission sensing as opposed to physical transmission sensing. For instance, the duration advertised in RTS consists of the time required to transmit the data frame, plus the CTS frame, plus the ACK frame, plus three SIFS intervals. The SIFS interval is the short interframe interval required between the RTS, DATA, CTS, and ACK frame. In the basic mode, a node starts transmitting its data traffic after a random waiting time without the exchange of the RTS and CTS control packets.

2.1 Exponential Backoff Mechanism

During a transmission, a collision can occur for various reasons. It can happen if two nodes attempt to transmit at the same time, or if a node in the vicinity does not detect neither RTS nor CTS packet belonging to the upcoming data transmission and attempts to transmit while another data transmission is ongoing. Also, a loss of an RTS or CTS packet is considered as a collision by the initiating transmitter. The collision detection is unlike that of wired medium access, as nodes are not capable of transmitting and receiving at the same time. In addition to the physical and virtual carrier sensing, an exponential backoff mechanism is in place to reduce collision rate. Before transmitting, each node picks a random waiting time from a uniform distribution between 0 and $CW - 1$. The contention window, CW , follows

an exponential increase with the number of experienced collisions up to a maximum CW_{max} .

$$CW = \begin{cases} 2^i CW_{min}, & \text{if } i < m \\ 2^m CW_{min} = CW_{max}, & \text{if } i \geq m. \end{cases} \quad (2.1)$$

CW_{min} is the starting window size and i is the number of collisions experienced by the packet. Upon successful transmission, the window size CW gets reset to CW_{min} . The random backoff selected corresponds to the number of slots a station needs to wait in addition to the mandatory interframe interval, DIFS, before attempting to transmit. The backoff timer is decremented only when the medium is idle; when the medium becomes busy the backoff timer freezes and resumes once the current transmission finishes. Fig. 2.1 illustrates this mechanism. In this example C freezes its backoff counter as stations A and D transmit to B.

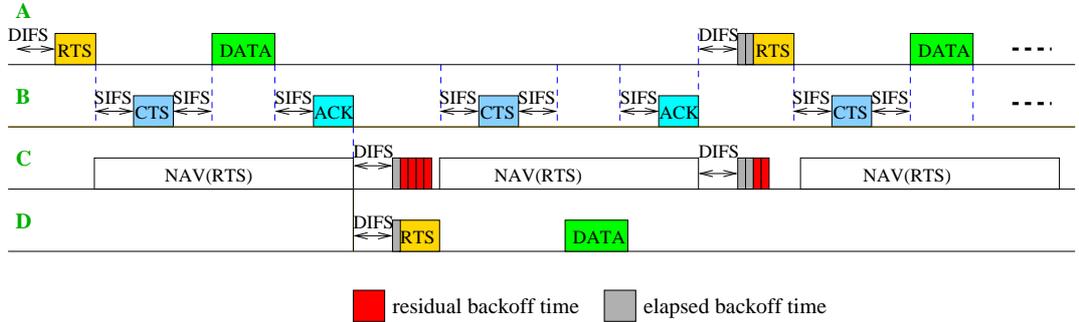


Figure 2.1: IEEE 802.11 Backoff Operation

2.2 Shortfalls of the Random Backoff Time

The protocol was designed for networks where all the entities participating obey the protocol rules. This assumption is valid if the network is owned by the

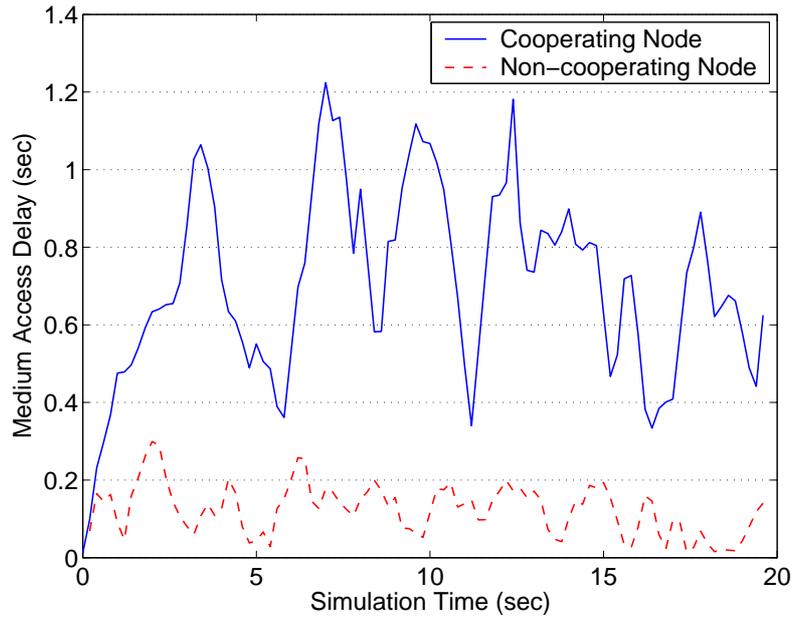
same entity, i.e., company networks and rescue and relief mission networks. However this will not apply in a network where nodes are individually owned and controlled, and are competing for the same network resources. There are many existing networks of this form and more are being deployed. These networks are being deployed in major cities, coffee shops, airports, etc. Some are provided free of charge or as complementary service, with an espresso for instance; others charge users according to time of use, in some airports for example, whether or not traffic is sent. Before we proceed any further, we classify users into three categories from a security standpoint.

1. Well behaved user: A user/station obeying the exact rules of the protocol.
2. Selfish user: A user that might not follow exact protocol rules in order to gain more bandwidth, shorter delay, and a better overall performance.
3. Malicious user: A user that has an objective of disrupting the network operation.

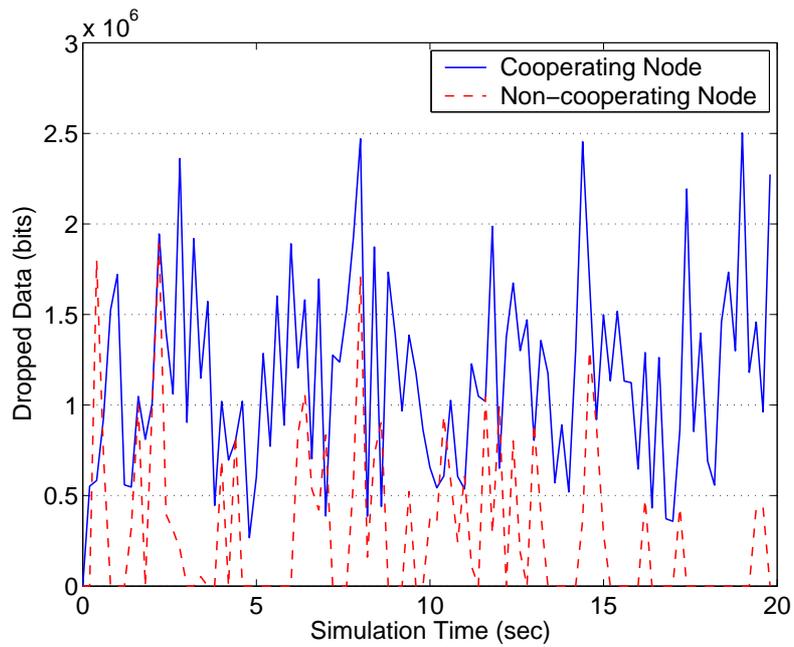
A selfish user might choose a short backoff time after a collision instead of choosing a random backoff time from the uniform distribution as dictated by the protocol. The easiness of protocol parameter modification in some wireless card has been previously addressed in [7, 8]. To show the effect of noncooperation, we simulated a simple 20 second scenario of IEEE 802.11b using OPNET. The load on all the nodes is the same. The packet inter-arrival rate of all nodes is exponential with mean of $0.01sec$ and the packet size is exponentially distributed with mean 2048 bytes. The wireless network consists of 8 nodes transmitting to the same destination.

Direct Sequence Spread Spectrum is chosen at the physical layer with $CW_{min} = 32$ and $CW_{max} = 1024$ according to the standard [1]. The non-cooperating node in this case still chooses from a uniform distribution but with a fixed window size of $CW_{min} = CW_{max} = 24$. A non-cooperating node might still want to randomize to prevent being detected or to avoid constant collisions with another non-cooperating node. We show the MAC delay experienced by one of the cooperating nodes and that of the non-cooperating node in Figure 2.2(a). In Figure 2.2(b), we also show the data dropped due to buffer overflow. Here we have considered a buffer of length $256Kbits$. The non-cooperating node experienced an average data loss of $500Kb/s$, whereas one of the cooperating nodes has a data loss rate of about $1.3Mb/s$. This is a reflection of the difference in the node throughput at about $800kb/s$, very significant considering the goodput of this scenario is less than $4Mb/s$. We have only shown the results for one of the seven cooperating nodes as they all experience similar throughput and delay.

Several papers have addressed detection of protocol noncompliance, specifically with the backoff mechanism [8, 9, 10] and others have proposed some modifications to the backoff mechanism in order to make the detection of non-cooperation easier [10, 11]. DOMINO [8] periodically collects backoff data during a monitoring period, after which it compares the backoff of a node to the nominal of the network with some tolerance parameter. It also keeps a cheating counter for every node. The counter is incremented if a potential non compliance is detected and decremented if the data collected from a node passes the threshold test. If the counter reaches a threshold of K , the node in question is considered cheating. This detection scheme is not



(a) MAC Delay



(b) Packet Drop

Figure 2.2: Cooperation vs Noncooperation in IEEE 802.11

robust against more adaptive cheating mechanisms, as mentioned by the authors.

For example, by knowing the duration of the collection period, a non compliant node

can follow the backoff mechanism of IEEE 802.11 for 3 periods so that its counter gets decremented at least twice, and then can follow a very short backoff during the next monitoring period, which may cause at most an increment of 2 in the cheating counter. Thus, the selfish node keeps the counter within bound and avoids being detected. Another weakness of DOMINO is that no backoff measurements are collected after sensing a collision, thus allowing a selfish user to go undetected when transmitting with short backoff after a channel collision. It is hard to detect non-cooperation of nodes since the backoff times are of random nature and a lot of statistics need to be detected before any assertion can be made. In general, a selfish node can adapt its backoff time to the detection mechanism; thus, a detection mechanism will only limit the extent of non-cooperation. In [10], the authors propose a modification to IEEE 802.11 to ensure the randomness of the backoff values. The protocol takes advantage of a hash function and its binding and hiding property to achieve an agreement on a random backoff between two non-trusted parties. The approach proposed in [11] to alleviate problems relating to the randomness of station backoffs is to assign the receiver the task of choosing backoff values for the senders. The receiver then checks the actual backoff against the assigned one. The authors have only addressed the case of backlogged stations and in reality, it would be cumbersome and difficult for the receiver to track the real backoff of all stations. Stations see different channels and backoff timers freeze when the station senses a busy channel. Collisions also affect their backoff timers and retransmission attempts.

Chapter 3

Bayesian Games and Protocol Design

In the game theory literature, what we have been calling selfish user is considered to be merely a rational user, one who wants to maximize his or her own utility, as one would expect. In our case, for example, the utility of a user can be a function of the throughput and delay.

Before we proceed further, we introduce few definitions, concepts and results that we will need in the subsequent sections. When the payoffs of other players are not well known in advance or depend on the player types, the game is considered to have incomplete information. We thus resort to Bayesian games [12, 13]. A n player Bayesian game can be described with

$$\Gamma = \{S_1, \dots, S_n, T_1, \dots, T_n, p_1, \dots, p_n, U_1, \dots, U_n\}$$

where S_i is the set of strategies of player i . T_i is the set of types of player i . $p_i = p(t_{-i}|t_i)$ is the player belief about other player types t_{-i} given his or her own type t_i . U_i is the player utility and is a function of the player types and their strategies.

Bayesian equilibrium is an extension of the Nash equilibrium in the case of incomplete information. A strategy profile $\sigma = (\sigma_1, \dots, \sigma_n)$ is a Bayesian equilibrium

of Γ if

$$\sum_{t_{-i} \in T_{-i}} p(t_{-i}|t_i) U_i[\sigma(t), t] \geq \sum_{t_{-i} \in T_{-i}} p(t_{-i}|t_i) U_i[\sigma_{-i}(t), s_i, t], \forall i, s_i \in S_i \quad (3.1)$$

σ_i is the plan of action for each possible type.

$$\sigma_i : T_i \rightarrow S_i$$

In other words, and along the Nash equilibrium concept, no player i wants to deviate from $\sigma_i(t_i)$ given his or her belief $p_i(t_{-i}|t_i)$ and that the other players are following the Bayesian equilibrium $\sigma_{-i}(t_{-i})$. We are now ready to revisit the random multiple access problem. For simplicity, let's assume that all users are of the same type, thus the Bayesian equilibrium (3.1) becomes

$$U_i[\sigma(t), t] \geq U_i[\sigma_{-i}(t), s_i, t], \forall i, s_i \in S_i$$

3.1 Random Access Nash Equilibrium

We present the normal form game for three station game along the simple 2 station model presented in [14] and generalize the results to n station game. This will give insight into some of the findings in [3, 4, 5, 15] relying on different models. The station strategies are either Transmit or Wait, $S_i = \{T, W\}$. A successful transmission yields a payoff of u_s , a failed transmission due to collision yields a payoff of u_f and no transmission yields u_i . The payoffs are general but must satisfy $u_f < u_i < u_s$ for obvious reasons. Fig. 3.1 shows the payoffs of the three stations under all strategies sets. We are mainly interested in symmetric equilibria due to fairness requirements. Let x , y and z denote the probability of transmission

	T	W		T	W
T	u_f, u_f, u_f	u_f, u_i, u_f		u_f, u_f, u_i	u_s, u_i, u_i
W	u_i, u_f, u_f	u_i, u_i, u_s		u_i, u_s, u_i	u_i, u_i, u_i
	T			W	

Figure 3.1: 3 Stations Normal Form Game

for station 1, 2 and 3 respectively. In order for station 1 to be willing to mix between transmitting and waiting, it must be indifferent to the payoff it gets from transmitting or from waiting; otherwise it will always choose the one with higher payoff. In other words

$$U_{1|T} = U_{1|W} \tag{3.2}$$

$U_{i|X}$ is the expected utility of station i given it has followed strategy X . Equivalently,

$$yzu_f + (1 - y)zu_f + y(1 - z)u_f + (1 - y)(1 - z)u_s = u_i$$

We get symmetric equations when considering the other users. The solution of these sets of non-linear equations yields all the mixed Nash equilibrium. With $x = y = z$, we get from (3.2)

$$(u_s - u_f)x^2 + 2(u_f - u_s)x + u_s - u_i = 0$$

with unique solution

$$x^* = 1 - \sqrt{\frac{u_i - u_f}{u_s - u_f}}$$

In the general n station case, we get

$$(1-x)^{n-1}u_s + \sum_{k=1}^{n-1} \binom{n-1}{k} x^k (1-x)^{n-1-k} u_f = u_i$$

$$(1-x)^{n-1}u_s + (1-(1-x)^{n-1})u_f = u_i$$

$$\Rightarrow x_n^* = 1 - \left(\frac{u_i - u_f}{u_s - u_f} \right)^{\frac{1}{n-1}}$$

Note that $u_i - u_f = c$ is the cost of transmission and $u_s - u_f = v$ is the payoff due to successful transmission. v can be associated to the valuation of the medium or the transmitted packet. When transmission cost is negligible with respect to medium valuation, the probability of transmission is close to 1. This Nash Equilibrium will bring the network to a crawl, another instance of the tragedy of the common. On the other hand, and as noted in [16], the backoff mechanism of IEEE 802.11 can be viewed as a constant transmission probability in saturated state. This probability is a function of the number of stations and the contention window limits CW_{min} and CW_{max} , and thus the protocol is not in equilibrium for a rational user to follow it. One way to regulate network performance is to add additional cost for transmission. However, the receiver cannot detect who transmits during a collision; therefore, we need to resort to a collision free scheme such as TDMA or FDMA to track and charge for transmissions. We will revisit the transmission costs and the success valuations in chapter 4.

3.2 The Revelation Principle

An important result relating to the Bayesian equilibrium that we will be using for resource allocation is the *revelation principle*.

Assume that $\sigma^*(t)$ is a Bayesian equilibrium of

$$\Gamma = \{S_1, \dots, S_n, T_1, \dots, T_n, p_1, \dots, p_n, U_1, \dots, U_n\}.$$

Then there exists a game

$$\Gamma' = \{S'_1, \dots, S'_n, T_1, \dots, T_n, p_1, \dots, p_n, U'_1, \dots, U'_n\}$$

such that in this game, truthful reporting of type is a Bayesian equilibrium. Now, the strategy set $S'_i = T_i$ and the utility function is $U'_i(s', t) = U_i(\sigma^*(s'), t)$ [12, 13, 17].

A mechanism with the strategy set equal the type set is called a *direct-revelation* mechanism. The user type T_i in our problem corresponds to the user valuation of the time slot and the strategy set S_i could be a probability of medium access. The utility U_i is a function of node's strategies, cost of transmission attempt and payoff. What the revelation principle allows us to do is, instead of solving for the difficult Bayesian Nash equilibrium σ satisfying the set of equations (3.1), we can come up with an intuitive mechanism by setting the proper utility function so as to make users report their true need for the medium.

A direct-revelation mechanism where truthful reporting is the best strategy is called *Incentive Compatible*. Thus, one of our objectives is to design a medium access protocol that is (i) *incentive compatible*. In developing an intuitive mechanism with a suitable utility function, we resort to auction theory as it has been extensively

studied in the allocation of goods [17]. An important difference in our problem is that we are mainly after network performance and not seller (Access Point) utility maximization. The other requirement we have is (ii) *allocation efficiency*, that is, assigning the time slots to those terminals valuing it the most. This constraint also provides quality of service in protocol design.

3.3 Truth Telling Second Price Auction

A clever and simple allocation mechanism where each player/bidder wants to reveal his true valuation is the second-price auction. In the second-price auction, the seller has only one item for sale, and the highest bidder gets the item and only pays the second highest bid of the auction and not his own. Thus winner payment is independent on his bidding price. The bidding price only determines the winner. A bidder i with valuation v_i has a utility

$$U_i(b, v_i) = \begin{cases} v_i - \max_{j \neq i} b_j, & \text{if } b_i > \max_{j \neq i} b_j \\ 0, & \text{if } b_i \leq \max_{j \neq i} b_j \end{cases}$$

b_j is player j bid for $j \in \{1, \dots, n\}$. With this mechanism, every bidder wants to bid his true value.

Proof. Let x_i be user i bid and let $p_i = \max_{j \neq i} b_j$. User i wants to maximize his utility U_i . Let's now consider the case $x_i > v_i$, then we get

$$\begin{aligned} U_i &= P(p_i > x_i > v_i)0 + P(x_i > p_i > v_i)(v_i - p_i) + P(x_i > v_i > p_i)(v_i - p_i) \\ &\leq P(x_i^* = v_i > p_i)(v_i - p_i) \end{aligned}$$

By bidding $x_i^* = v_i$ we eliminate the second term which yields a negative payoff without affecting the rest of the terms. A similar argument holds if user i were to bid $x_i < v_i$ □

Bidding a higher value than the true valuation results in a positive probability that the bidder wins the item and the amount paid result in a negative total payoff. Bidding a lower value than the true valuation results in a positive probability of not winning the item when the bidder could have made profit had he bid his true value. Second price auction is then incentive compatible.

3.4 Vickrey Auction and Time Slot Allocation

The Vickrey auction adopts the idea of second price auction but applies when auctioning multiple items, say K . Each bidder submits his or her demand curve and the seller then calculates the aggregate demand on the goods to be allocated and the K highest winning bids are assigned the items. We use demand curve and bidding vector interchangeably as they are reciprocal of each other. The winning bidders pay only the opportunity cost. The opportunity cost for bidder i refers to the willingness of losing bidders to pay for the items won by i . Formally, with K items to be allocated, each bidder $i \in \{1, \dots, n\}$ submits a bidding vector $\mathbf{b}_i = (b_i^1, b_i^2, \dots, b_i^K)$, where b_i^k is his bid for a k^{th} item. Let $\mathbf{c}_{-i} = (c_{-i}^1, \dots, c_{-i}^K)$ with element c_{-i}^ℓ being the ℓ^{th} largest value among $b_j^k, \forall k \in \{1, \dots, K\}, j \neq i$. The opportunity cost and the payment made by i for k_i items won can be expressed as

$$\sum_{m=1}^{k_i} c_{-i}^{K-k_i+m}.$$

This amount is the total value of the k_i highest losing bids, the opportunity cost. Vickrey auction is also incentive compatible, that is, a node's best strategy is to bid its true valuation for the items. There are some practical problems with the Vickrey auction in certain settings, but is very appropriate for our bandwidth allocation problem. Some variants of the Vickrey auction are very successful in practice. For example, Google AdWords uses it to auction advertisement slots next to search results [18].

As an illustrative example, consider the demand curves depicted in table 3.1. Entry (m, Sta_i) in the table is the willingness of station i to pay for the m^{th} won item. In this example $K = 3$.

Slot	Sta_1	Sta_2	Sta_3
1	6	7	3
2	4	5	2
3	1	1	0

Table 3.1: Vickrey Auction Example

The winning stations are station 1 and station 2 as they have the three highest bids. Station 1 gets one item and station 2 gets two items. The price paid by 1 is 3 and that paid by 2 is $7 = 4 + 3$. These paid prices reflect the opportunity cost.

Recall that our initial design criterion was to develop a medium access control protocol that is robust in an environment where participating stations are individually owned and capable of altering protocol rules. Time slot allocation follows the idea presented in the Vickrey auction and time slots are assigned to the terminals

that value them the most. Terminals participating in this protocol have an incentive to participate in the network and never deviate from reporting their true valuation for the medium. The base station must therefore collect the node valuation before assigning the time slots for transmission. Slot assignment is done in rounds. The number of time slots allocated in every round and the length of each time slot are design parameters and depend on the number of terminals associated with the AP, type of data traffic and supported services. Design parameters will be addressed later on. We can assume that at every round, K number of slots will be allocated to the active users, those who are associated with the receiver.

Chapter 4

Incentive Compatible Medium Access Control

The Incentive Compatible MAC (ICMAC) does not deal with the association and authentication mechanism, but we assume that a secure mechanism is in place. The receiver station has the task of scheduling the transmission of successfully associated stations. Fig. 4 summarizes the protocol operation. At the beginning of

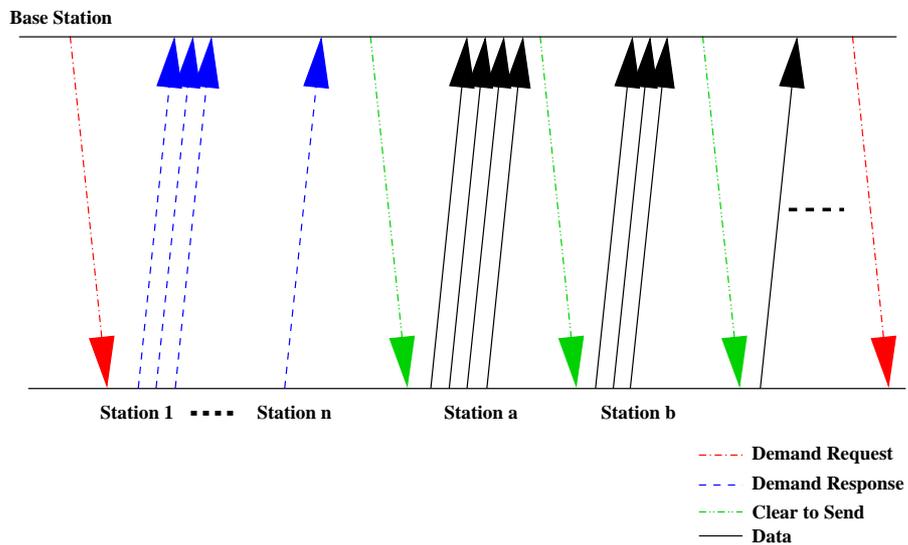


Figure 4.1: ICMAC Protocol

every round, the base station sends a Demand Request (DRQ) packet to inform that it is taking bids for the K next time slots. Upon hearing a DRQ packet, every node responds with a Demand Response (DRS) packet. A DRS packet contains the station address and its bids for each of the K time slots. Attributed to every station is an association ID (AID) and a demand response time slot allowing the stations

to access the medium in a deterministic TDMA fashion with no collision during the bid collection phase. After collecting all the demand curves, the base station aggregates the station demands to determine the winning K bids. Then, sequential Clear To Send (CTS) messages are sent from the AP to the stations, from highest to lowest winning bids, informing them of the time of transmission and number of allocated successive transmissions. Note that once the station is allocated the time slot, it can send its data traffic to any node and not necessarily to the coordination station. Along the CTS message, an optional acknowledgement is sent to the previous transmitting station on the sent data packets. In Fig. 4, station a is one of the n stations associated with the base station having the highest bids for that round. It receives a CTS packet informing it that it gets the next four time slots. After transmitting data for four successive time slots, station a listens for the next CTS packet to get an acknowledgment about its previously transmitted packets. A bit is associated with every previously transmitted packet for acknowledgment. In order to make the acknowledgment mechanism fruitful, the CTS message assigns no more than $MaxSch$ slots at a time. That is, if a station wins more than $MaxSch$, the base station does not schedule all those transmissions in one shot, but breaks them apart, so they get progressively acknowledged.

In addition, the AP needs to have a secure and flexible monetary system in place along with the association in order to charge winning users for the opportunity cost. For example, a station can set a limit on expenditure and get notified when it has reached its limit, or it can check its balance.

4.1 Control Packets

ICMAC control messaging will be exchanged between the transmitters and the receiver to determine who will be transmitting and when. This control overhead must be analyzed thoroughly. The frame formats have been mainly borrowed from IEEE 802.11. The DRQ packet, the DRS packet and the CTS packet are all similar to the CTS of IEEE 802.11. The number of slots per round and the fragment size can be either advertised during association or through the DRQ packet. The DRS packet has an additional field for the bidding vector.

4.2 Time Slot Valuation

A secure monetary or unit system has to be in place to carry out and enforce some of the ideas presented here. Terminals have a private value for the medium access, which is tightly dependent on delay and throughput. For example, the valuation of the time slot depends on packets present in the queue of the transmitter and/or running services such as VoIP. Packets are first categorized according to their type, i.e., data, voice, and video. These packet types have different bandwidth and delay requirements. Packet waiting time also impacts the valuation of the time slot. Three example profiles of packet valuation are presented herein and shown in Fig. 4.2, every user is assumed to have independent valuation of packets. The time slot

valuation is a function of the waiting time and user/packet type.

$$\begin{aligned}
 Y_\ell^1(t) &= c_\ell \\
 Y_\ell^2(t) &= \begin{cases} a_\ell e^{b_\ell t} + c_\ell, & \text{if } t \in [0, t_\ell^{max}] \\ 0, & \text{otherwise} \end{cases} \\
 Y_\ell^3(t) &= c_\ell \left(\frac{1}{1 + e^{-a_\ell(t-b_\ell)}} \right) + d_\ell
 \end{aligned} \tag{4.1}$$

t is the waiting time of the packet in the queue, ℓ is the index of the packet type, a_ℓ , b_ℓ and c_ℓ are type dependent parameters of the valuation function. Note that t_ℓ^{max} is also type defined. Some real-time application might have hard constraints, and packets could be dropped if not transmitted before some expiration time t_ℓ^{max} . Another criterion that may also be considered is the ratio of packets in the queue

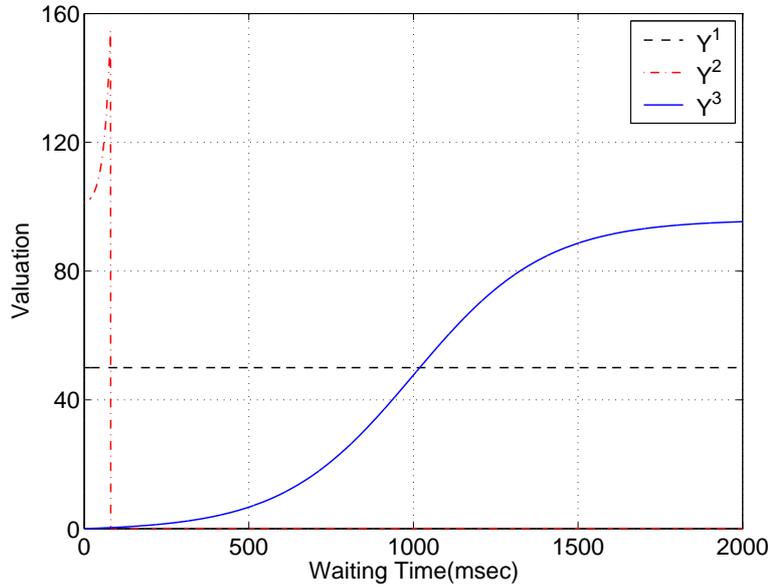


Figure 4.2: Valuation Function

with respect to the buffer size. When the queue size gets large, the new incoming

packets might have to be dropped. In this case, the terminal station can attribute an additional value to the time slot. Consider the following sigmoid valuation function that depends on the queue length L , the buffer size Q_{MAX} , and the packet position p in the queue.

$$W_\ell(p) = c_\ell \left(\frac{1}{1 + e^{-a_\ell(p-b_\ell)}} \right) + d_\ell$$

The parameters c_ℓ and b_ℓ will be functions of $\frac{L}{Q_{MAX}}$. They are both increasing functions of $\frac{L}{Q_{MAX}}$. The parameter c_ℓ determine the maximum increase in valuation of the time slot. b_ℓ can be viewed as the limiting point of the affected packets. The longer the queue the more packets we want to send leading to increase in valuation. The function $W_\ell(p)$ decreases with the position of the packet in the queue. In Fig. 4.3, we show the additional valuation that is associated with the packet position for various queue lengths L for $Q_{MAX}=100$. Therefore, the overall valuation of the time

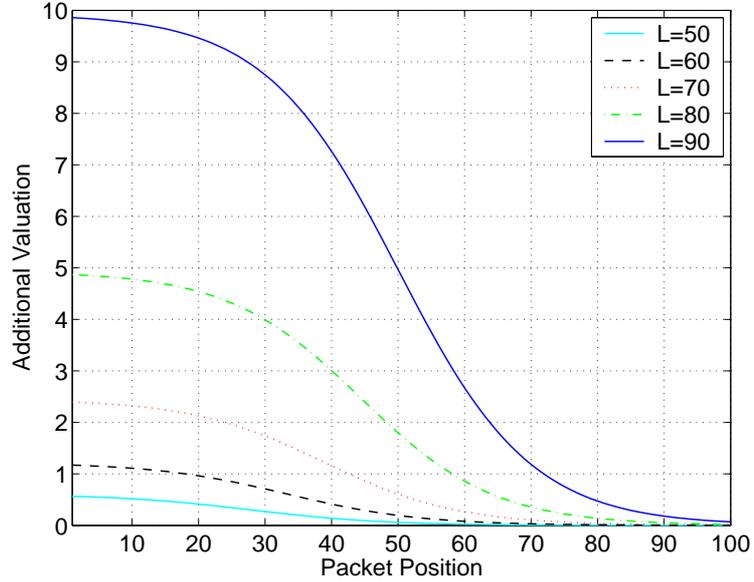


Figure 4.3: Queue Length Dependent Valuation

slot is a function of the packet waiting time, the packet position and the length of the queue. We are assuming that there are different queue types holding different packet types.

$$V_\ell(t, p) = Y_\ell(t) + W_\ell(p)$$

Fig. 4.4 shows the demand curves of two terminals using the information present at their queues, or other information they might have about current running services. This information can be simply represented in a vector. Quantization of the demand curve would also be used to shorten transmission of demand curves and to simplify computation and decision making at the receiver. The receiver can calculate the aggregate demand and then allocate the time slots accordingly. In this case the number of time slots being offered is 20. As before, the highest bids determine the winner and the price paid is the opportunity cost.

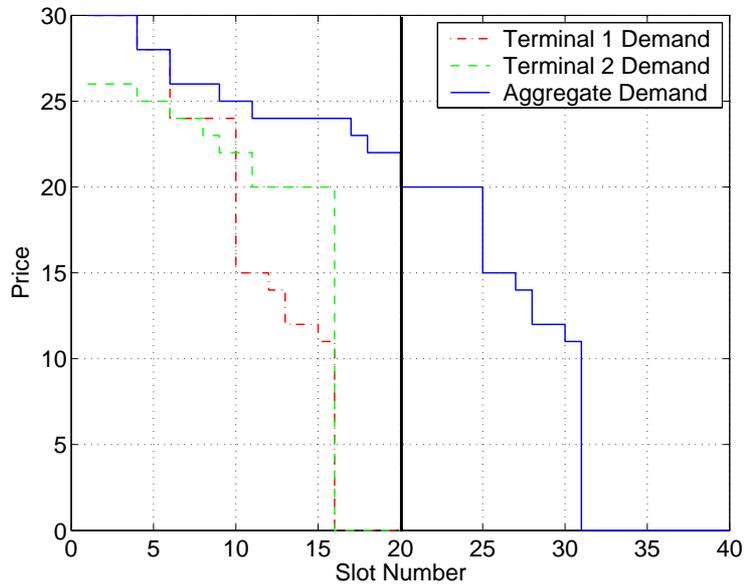


Figure 4.4: Demand Curves

Chapter 5

ICMAC Performance and Design Parameters

5.1 Performance

Before we proceed any further, we define some parameters and tabulate packet sizes and design parameters in Table 5.1. With little abuse of notation *phyhdr* is shown in μs and in *bits* and kept the same for $1Mb/s$ and $11Mb/s$ transmission rates. Design parameters need to be chosen by an administrator based on the type of traffic and user needs. The designated parameters will impact the overall throughput, the delay and the overhead. The control packets, *DRQ*, *DRS*, *CTS* are all sent at control transmission rate of $1Mb/s$. We calculate the throughput of the protocol for what we consider reasonable parameters for some applications. We assume data occupy the whole fragment in this initial calculation. We will revisit performance for the general case after addressin the design parameters.

$$Throughput = \frac{K * DATA}{RoundDuration}$$

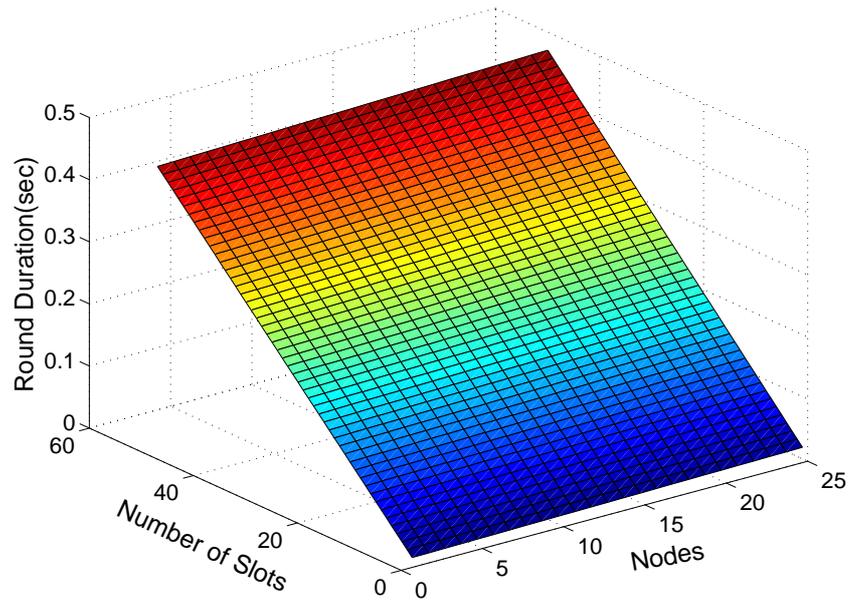
In calculating the round duration we have to consider the transmission rate of the control and data packets.

$$RoundDuration = \frac{DRQ}{CtrlRate} + SIFS + n \left(\frac{DRS}{CtrlRate} + SIFS \right) + K \left(\frac{CTS}{CtrlRate} + SIFS + phyhdr \frac{DATA}{DataRate} + SIFS \right)$$

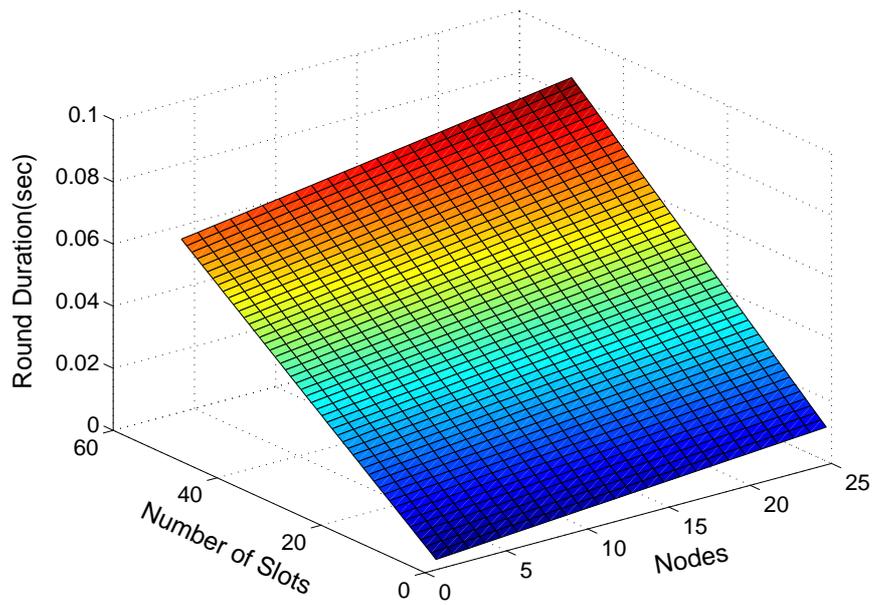
Parameter	Value	Unit
Inter frame duration	$SIFS = 10$	μs
Physical layer delay	$phyhdr = 192$	μs
MAC header	$machdr = 272$	bits
Number of slots per round	K (design parameter)	slots
Fragment length	$FLength$ (design parameter)	bits
Value representation in bits	$BidRep$	bits
DRQ packet length	$160 + phyhdr$	bits
DRS packet length	$160 + phyhdr + K * BidRep$	bits
Maximum packets scheduled	$MaxSch$	n/a
CTS packet length	$160 + phyhdr + MaxSch$	bits
DATA packet length	$272 + FLength$	bits

Table 5.1: Packet sizes and Parameters

In general, we show results for 1Mb/s and 11Mb/s data transmission rates while keeping the control rate fixed at 1Mb/s. The control rate might be kept at a lower transmission rate to give it greater protection and better success rate. After the proper scaling, the results for 1Mb/s for both control and data transmission rate can also be used as a good estimate - some headers are independent of rate - on the performance of other systems that keep control and data rate the same. We now show the round duration in Fig. 5.1 as a function of the number of nodes and the number of slot per round for both 1Mb/s and 11Mb/s data rates while keeping the control rate at 1Mb/s. The packet sizes are equal to 1024 bytes and constant.



(a) 1Mb/s Data Transmission



(b) 11Mb/s Data Transmission

Figure 5.1: ICMAC Round Duration

Using the round duration, we can get an estimate of the MAC delay, not including any queueing delay. Recall that stations submit bids only when they have traffic to send or some services running, such as VoIP. In the case where a station only requests slots when it has packets to send, a new incoming packet of highest type arriving after the DRQ transmission has to wait for the remaining time until the next DRQ plus the new bid collection time. The elapsed time between the station DRQs is the Round Duration(RD). If we consider, as an example, a poisson arrival and the memoryless property of the exponential distribution, we can express the probability density of arrival time given a packet arrives during a RD time as

$$f(t|t \leq RD) = \frac{\lambda \exp(-\lambda t)}{1 - \exp(-\lambda RD)}.$$

with expected arrival time

$$E[t|t \leq RD] = \frac{1}{\lambda} - \frac{RD \exp(-\lambda RD)}{1 - \exp(-\lambda RD)}$$

yielding an expected MAC delay of

$$E[\text{MAC Delay}] = RD - \left(\frac{1}{\lambda} - \frac{RD \exp(-\lambda RD)}{1 - \exp(-\lambda RD)} \right) + (N - AID)(SIFS + DRS) \quad (5.1)$$

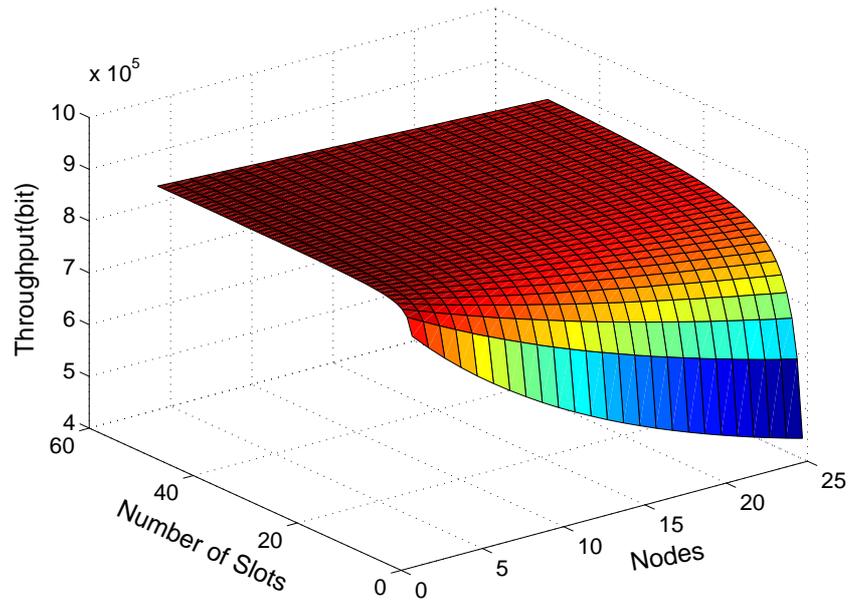
N is the number of stations and AID is the association ID. Thus stations with larger association ID experience a slightly better MAC delay; however this won't be an issue in a practical scenario with a reasonable size of stations.

In Fig. 5.2, we plot the throughput as a function of the number of nodes and the number of slot per round for both 1Mb/s and 11Mb/s data rates while keeping the control rate at 1Mb/s. The packet sizes are again assumed to be constant of

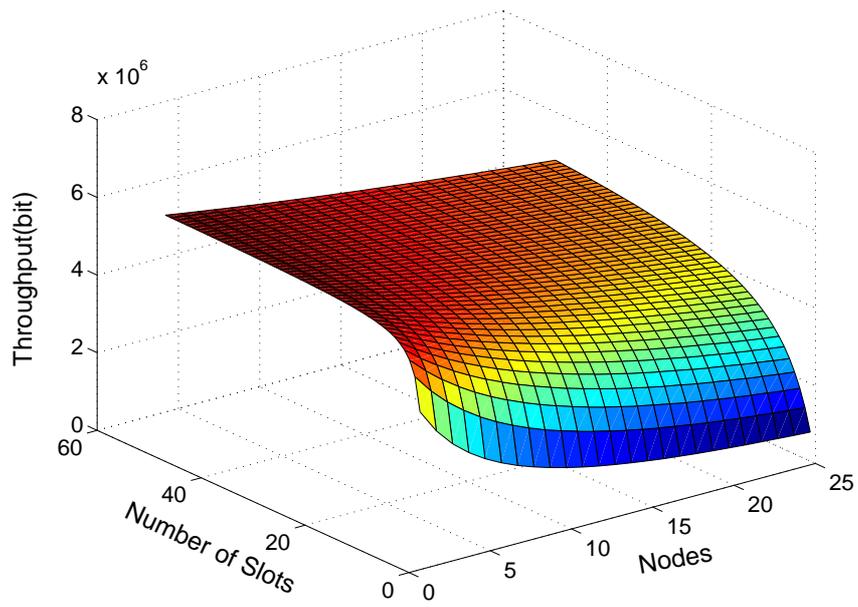
length 1024 bytes. $BidRep=8$ bits are used for value representation. It is also important to note that in the calculation, each packet is individually scheduled and fully occupy the data fragment in the time slot. Multiple scheduling is more appropriate when consecutive winning bids belong to the same station. The benefit of this will be highlighted later.

The performance drops with the number of stations due to bid collection at every round, especially if the control packet transmission rate is lower than the of data transmission rate. In order to reduce the overhead incurred from this bid collection, the network designer can increase the number of slots allocated at every round. The other alternative is to auction multiple rounds at a time. The later option is also appropriate in situations where the services running in the network require sustainable throughput over multiple rounds. In Fig. 5.3 we show the potential throughput gain from auctioning multiple rounds at a time for the case $n=20$, $K=50$, fragment of 1024 bytes and data transmission rate of 11Mbps. The drawback from auctioning many rounds at a time is that some slots may be wasted as the winning stations may have no packets to transmit at later rounds. The extreme case of allocating slots over multiple rounds becomes a fixed TDMA scheme which is not appropriate in data networks. We also plotted the impact on the round duration.

As an initial comparison with the performance of IEEE 802.11, we rely on the Markov chain model in [16] that we briefly summarize.



(a) 1Mb/s Data Transmission



(b) 11Mb/s Data Transmission

Figure 5.2: ICMAC Throughput

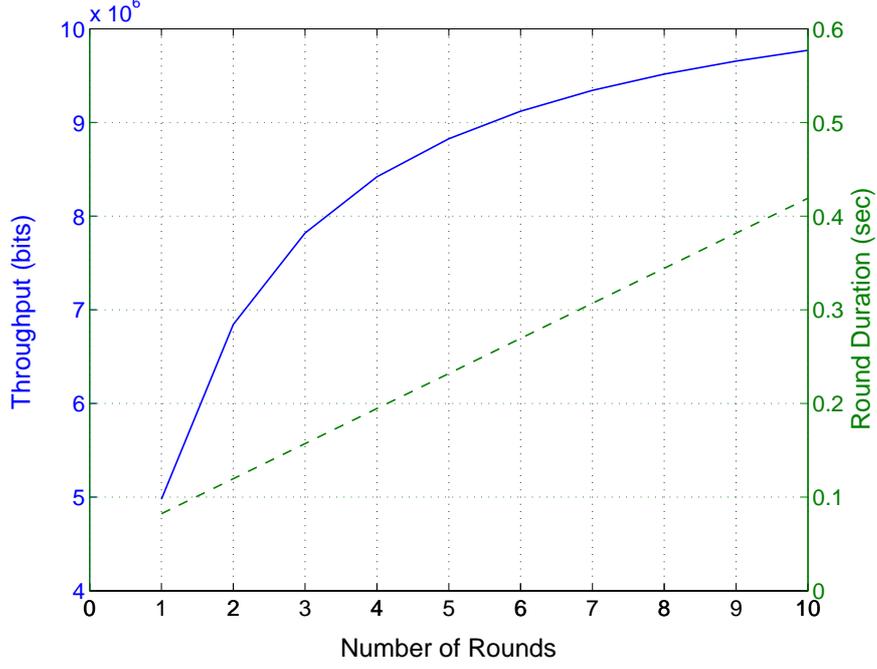


Figure 5.3: Throughput and Round Duration

5.1.1 IEEE 802.11 Markov Chain Model

The model assumes that all the nodes participating are in a saturation and uses a two state Markov chain. The first state being the backoff stage and the second state is the backoff counter. The backoff stage is i in equation (2.1) and reflect the number of collisions already experienced by the packet. With p being the probability of collision, the transition probabilities are as follows

$$\left\{ \begin{array}{l} P\{i, k|i, k+1\} = 1 \quad k \in (0, W_i - 2) \quad i \in (0, m) \\ P\{0, k|i, 0\} = \frac{1-p}{W_0} \quad k \in (0, W_0 - 1) \quad i \in (0, m) \\ P\{i, k|i-1, 0\} = \frac{p}{W_i} \quad k \in (0, W_i - 1) \quad i \in (1, m) \\ P\{m, k|m, 0\} = \frac{p}{W_m} \quad k \in (0, W_m - 1) \end{array} \right.$$

From the steady state distribution of the Markov chain, the probability of a node transmitting during an idle slot is

$$\tau = \frac{2(1-2p)}{(1-2p)(W+1) + pW(1-(2p)^m)}$$

τ is obtained from the steady state probabilities of being at states with backoff counter of 0. To solve for p and τ , another relation is used yielding a unique solution for p and τ

$$p = 1 - (1 - \tau)^{n-1}$$

Next, we will focus on the throughput and not on the utilization; therefore, we need to take into consideration the control and data transmission rates.

$$T = \frac{P_s P_{tr} E[P]}{(1 - P_{tr})\sigma + P_{tr} P_s T_s + P_{tr} (1 - P_s) T_c} \quad (5.2)$$

T_s and T_c are the success duration and collision duration of a transmission respectively and the parameters not previously defined are tabulated in Table 5.2

$$T_s = \frac{phyhdr}{ctrlrate} + \frac{machdr}{datarate} + \frac{E[P]}{datarate} + SIFS + \delta + \frac{ACK}{ctrlrate} + DIFS + \delta$$

$$T_c = \frac{phyhdr}{ctrlrate} + \frac{machdr}{datarate} + \frac{E[P]}{datarate} + DIFS + \delta$$

P_{tr} is the probability that at least one node is transmitting and P_s is the probability of success.

$$P_{tr} = 1 - (1 - \tau)^n$$

$$P_s = \frac{n\tau(1 - \tau)^{n-1}}{P_{tr}}$$

In the RTS/CTS mode, T_s and T_c become

$DIFS$	DCF interframe space
δ	propagation delay
$E[P]$	expected payload
RTS	request to send packet length
CTS	clear to send packet length
ACK	ack packet length
σ	slot duration used for backoff counter

Table 5.2: Parameter Definition

$$T_s = \frac{RTS}{ctrlrate} + 3SIFS + 4\delta + \frac{CTS}{ctrlrate} + \frac{phyhdr}{ctrlrate} + \frac{machdr}{datarate} + \frac{E[P]}{datarate} + \frac{ACK}{ctrlrate} + DIFS$$

$$T_c = \frac{RTS}{ctrlrate} + DIFS + \delta$$

For more in depth discussion of the results above refer to [16]. Recall that the throughput performance depends on a node transmission probability τ which itself depends on CW_{min} and CW_{max} in addition to n . The physical layer used is the Direct Sequence Spread Spectrum physical, with $CW_{min} = 32$ and $CW_{max} = 1024$. Using the model discussed above we plot in Fig. 5.4 the throughput of both ICMAC and IEEE 802.11. The message sizes are kept constant at 1024 bytes. With the parameters above ICMAC performs better than both basic and RTS/CTS mode for reasonable size of local neighbors. It does however degrade at a much faster rate than RTS/CTS mode.

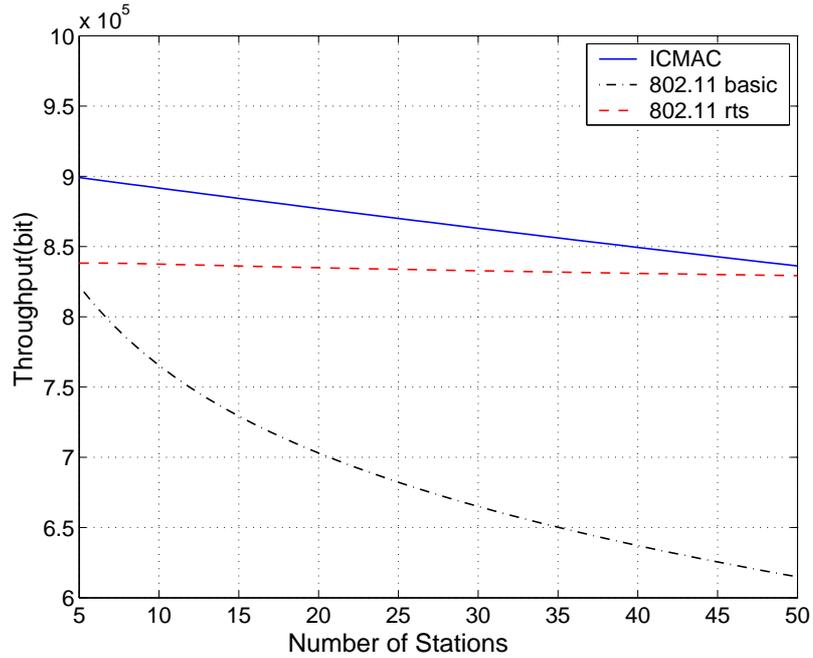


Figure 5.4: ICMAC and IEEE 802.11 throughput

5.2 Design

As ICMAC is a TDMA based access control and the slot size is fixed, the network designer has to properly choose the slot length and the number of slots auctioned at each round. We consider a time slot to contain a CTS control message, the data packet, their headers and all the interframe durations. Refer to Fig. 5.5 for better understanding. The overhead of a time slot is

$$h = 2SIFS + \frac{CTS}{CtrlRate} + phyhdr + \frac{machdr}{DataRate}$$

$h = 788\mu s$ and $h = 584.4\mu s$ for a transmission data rate of $1Mb/s$ and $11Mb/s$ respectively. We also denote by H the round overhead associated with bid collection.

It can be expressed as

$$H = SIFS + \frac{DRQ}{ctrlRate} + n \left(SIFS + \frac{DRS}{CtrlRate} \right)$$

Recall that DRS size depends on K , the number of slots allocated per round, and $BidRep$, the number of bits representing a bid. For $n = 10$, $K = 50$ and $BidRep = 8$, $H = 7.982ms$

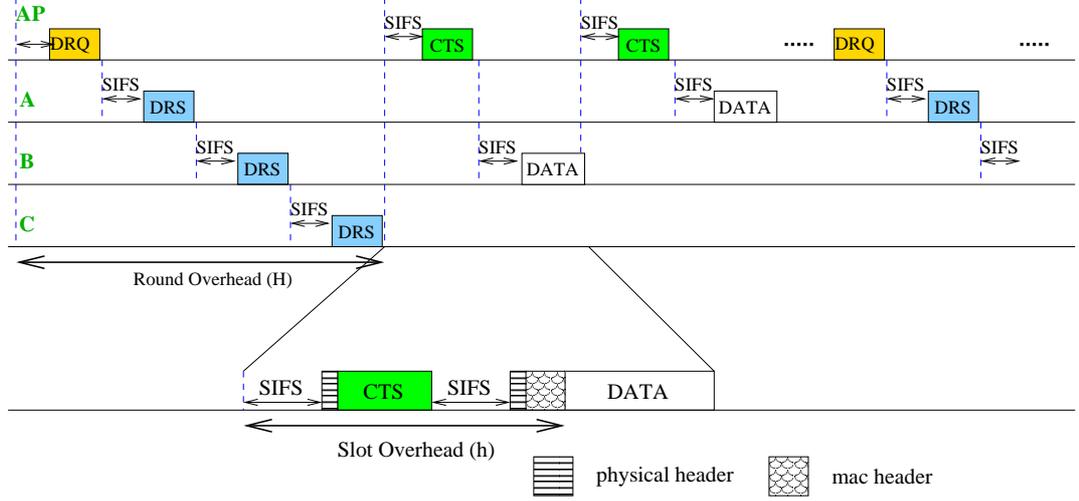


Figure 5.5: ICMAC Overhead

5.2.1 Fragment Size

As messages might be sent over multiple slots and data might not fully occupy the allocated fragment, we need to optimize the transmission efficiency with respect to time slot duration. Clearly the optimum slot duration will be a function of the message length and the overheads h and H . We assume that the data size is distributed according to $f(x)$. The problem of using the fixed slot size efficiently becomes:

$$\begin{aligned} & \min_Y \int_0^\infty \left(Y + \frac{H}{K} \right) \left[\frac{x}{Y - h} \right] f(x) dx \\ & \Leftrightarrow \min_Y \left(Y + \frac{H}{K} \right) \int_0^\infty \left[\frac{x}{Y - h} \right] f(x) dx. \end{aligned} \quad (5.3)$$

In (5.3), Y is the slot duration, $\lceil \frac{x}{Y-h} \rceil$ is the number of slots required by a message of length x , $\frac{H}{K}$ represents the per unit overhead attributed to one slot due to the round overhead H . with $Z = Y + \frac{H}{K}$ and $h' = \frac{H}{K} + h$, the optimization (5.3) can be rewritten as

$$\min_Z Z \int_0^\infty \left\lceil \frac{x}{Z-h'} \right\rceil f(x) dx \quad (5.4)$$

Now consider only the integral term of equation (5.4):

$$\begin{aligned} & \int_0^\infty \left\lceil \frac{x}{Z-h'} \right\rceil f(x) dx \\ &= \sum_{k=1}^\infty \int_{(k-1)(Z-h')}^{k(Z-h')} k f(x) dx \\ &= \sum_{k=1}^\infty k \int_{(k-1)(Z-h')}^{k(Z-h')} f(x) dx \\ &= \sum_{k=1}^\infty k \mathbb{P}((k-1)(Z-h') < X \leq k(Z-h')) \end{aligned} \quad (5.5)$$

$\mathbb{P}((k-1)(Z-h') < X \leq k(Z-h'))$ is the probability that a message m requires k time slots. As an example, we first look at an exponential distribution for packet length, and then exponential packet size distribution mixed with a constant packet size.

Exponential Distribution

With message length exponentially distributed with mean \bar{m} , (5.5) can be expressed as

$$\begin{aligned}
& \sum_{k=1}^{\infty} k \left(\exp\left(-\frac{(k-1)(Z-h')}{\bar{m}}\right) - \exp\left(-\frac{k(Z-h')}{\bar{m}}\right) \right) \\
&= 1 - \exp\left(-\frac{1(Z-h')}{\bar{m}}\right) + \\
& \quad 2\exp\left(-\frac{1(Z-h')}{\bar{m}}\right) - 2\exp\left(-\frac{2(Z-h')}{\bar{m}}\right) + \\
& \quad \dots \\
& \quad \ell \exp\left(-\frac{(\ell-1)(Z-h')}{\bar{m}}\right) - \ell \exp\left(-\frac{\ell(Z-h')}{\bar{m}}\right) + \\
& \quad \dots \\
&= \sum_{k=0}^{\infty} \exp\left(-\frac{k(Z-h')}{\bar{m}}\right) \\
&= \frac{1}{1 - \exp\left(-\frac{(Z-h')}{\bar{m}}\right)}
\end{aligned}$$

The second equality is due to telescoping the terms. The minimization (5.4) becomes

$$\min_Z \frac{Z}{1 - \exp\left(-\frac{(Z-h')}{\bar{m}}\right)} \quad (5.6)$$

with the solution satisfying

$$\exp\left(\frac{Z-h'}{\bar{m}}\right) - \left(1 + \frac{Z}{\bar{m}}\right) = 0. \quad (5.7)$$

Equation (5.7) has a unique solution for $Z > h'$ that can be easily found numerically.

The solution Z^* corresponds to a time duration which can be translated to data fragment size of $frag^* = (Z^* - h') * DataRate$.

In the case where all packets belonging to the same message are scheduled with one CTS due to the same valuation, the transmission efficiency problem stays the same, but now

$$h = SIFS + phyhdr + \frac{machdr}{DataRate}.$$

The main drawback of ICMAC is the overhead that comes from bid collection as it depends on the number of nodes and the value representation of the K values.

However, one way to reduce it is to collect station bids for multiple rounds. The winning station gets the same number of slots for multiple rounds. We can still use the optimum fragment solution as before but now, the per slot overhead due to H , $\frac{H}{K}$, is divided further by the number of rounds auctioned R .

We plot in Fig. 5.6 the fragment size solution with respect to mean packet size \bar{m} for fixed control transmission rate of $1Mb/s$ and data transmission rates of $11Mb/s$ and $1Mb/s$ for $n=10$ and $K=50$. We have included results on both individual packet scheduling and multiple packet scheduling. The solution for optimum packet size is smaller for multiple scheduling than individual scheduling since the fragmentation penalty is less significant.

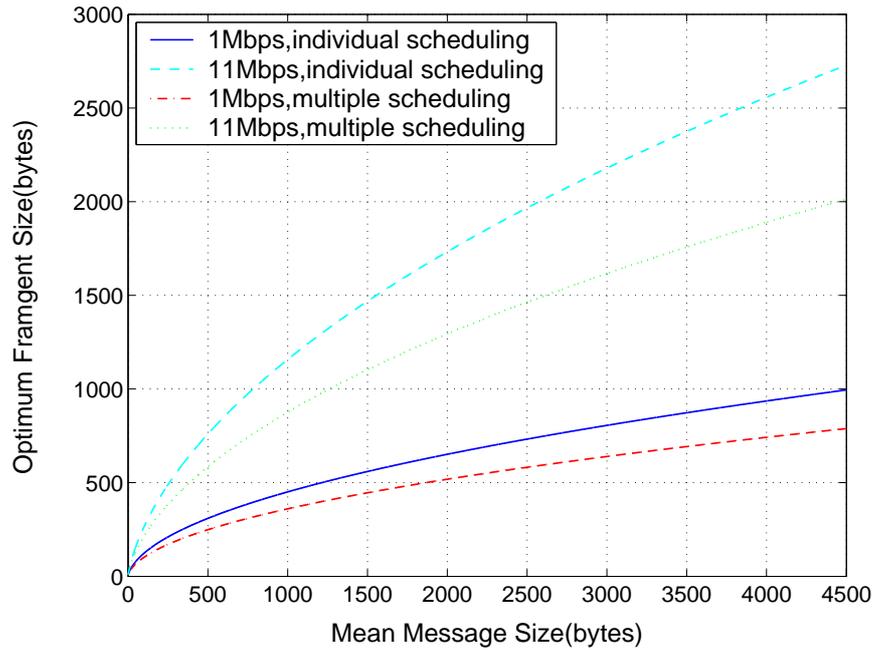


Figure 5.6: Optimal Data Fragment Size

Mixed Exponential and Constant Size Messages

We assume that traffic with exponentially distributed message size is sent with probability p and traffic with constant message size is sent with probability $1 - p$. \bar{m} is the mean of the exponential distribution and \bar{v} is the constant message size. The problem becomes

$$\min_Z Z \left(p \frac{1}{1 - \exp\left(-\frac{(Z-h')}{\bar{m}}\right)} + (1 - p) \left\lceil \frac{\bar{v}}{Z - h'} \right\rceil \right) \quad (5.8)$$

The above problem is not convex; however, we can find a solution bound using (5.9).

$$\begin{aligned} Z \left(p \frac{1}{1 - \exp\left(-\frac{(Z-h')}{\bar{m}}\right)} + (1 - p) \frac{\bar{v}}{Z - h'} \right) &\leq \\ Z \left(p \frac{1}{1 - \exp\left(-\frac{(Z-h')}{\bar{m}}\right)} + (1 - p) \left\lceil \frac{\bar{v}}{Z - h'} \right\rceil \right) &\leq \\ Z \left(p \frac{1}{1 - \exp\left(-\frac{(Z-h')}{\bar{m}}\right)} + (1 - p) \left(\frac{\bar{v}}{Z - h'} + 1 \right) \right) &\end{aligned} \quad (5.9)$$

The minimum of the upper bound function in (5.9), is an upper bound on the minimum of the solution. Now using the lower bound in (5.9) we can limit the range of the solution. We depict all three functions to show the process with which we find a solution bound in Fig. 5.7. The figure shows the case of $p=0.5$, $\bar{v}=(160*8/11E6)s$ and $\bar{m}=(1024*8/11E6)s$. The solution in this case is $614\mu s$ for the slot duration translating to $319bytes$ for the data fragment size. For $p=0.75$, we get $480bytes$ for the data fragment size, as more messages are distributed according to the exponential distribution.

In addition to the message distribution, another important constraint that the designer needs to keep in mind is that of the physical medium. The longer the

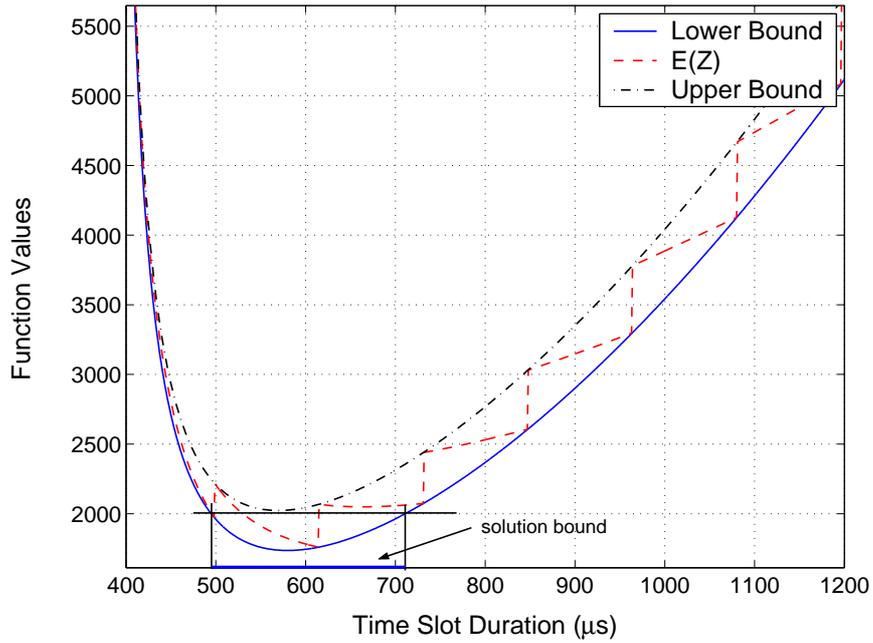


Figure 5.7: Mixed Traffic Optimal Solution

fragment, the more susceptible it is to errors. Thus there are different limits to the fragment length in different environments.

5.2.2 Number of Slots per Round

Recall that we expressed round duration as

$$RoundDuration = H + K * \left(h + \frac{frag}{DataRate} \right) = KZ$$

The round duration time must be appropriate for the traffic supported on the network. For example in delay sensitive application with constant bit rate, stations rather have the slots spread through the round instead of getting all the transmissions in one shot. In the current protocol, the AP schedules only according to the aggregate demand, thus limiting the round duration would allow interleaving between station transmissions. The round duration also affects the MAC delay as

shown in (5.1). With an additional constraint C on the round duration , the problem becomes

$$\begin{aligned} \min_{Y,K} \int_0^\infty \left(Y + \frac{H}{K} \right) \left[\frac{x}{Y-h} \right] f(x) dx & \quad (5.10) \\ \text{s.t. } Y & \geq h \\ K & \in \mathbb{N} \\ H + KY & \leq C \end{aligned}$$

Equivalently, with $Z = Y + \frac{H}{K}$

$$\begin{aligned} \min_{Z,K} \int_0^\infty Z \left[\frac{x}{Z - \frac{H}{K} - h} \right] f(x) dx & \quad (5.11) \\ \text{s.t. } Z & \geq \frac{H}{K} + h \\ K & \in \mathbb{N} \\ KZ & \leq C \end{aligned}$$

For an exponentially distributed message size, (5.11) becomes

$$\begin{aligned} \min_{Z,K} \frac{Z}{1 - \exp\left(-\frac{Z - \frac{H}{K} - h}{\bar{m}}\right)} & \quad (5.12) \\ \text{s.t. } Z & \geq \frac{H}{K} + h \\ K & \in \mathbb{N} \\ KZ & \leq C \end{aligned}$$

As the number of slots K is an integer, we resort to numerical solution. From the first constraint we have $Z \geq h$ and when combining with the third constraint we get $K \leq \lfloor \frac{C}{h} \rfloor$. For a given $K \leq \lfloor \frac{C}{h} \rfloor$, the optimal solution Z_K^* can be obtained from (5.7) or $Z_K^* = \frac{C}{K}$ if (5.7) solution does not satisfy $KZ \leq C$. We show the

optimal pair for 10 stations with 11Mb/s data transmission rate and 1Mb/s control transmission rate. The results are for a round duration constraint of 100ms. Again the optimal slot size is translated to the fragment size as before.

message mean (bytes)	K	fragment (bytes)
512	78	633
1024	63	981
2048	49	1493
4096	37	2248

Table 5.3: Optimal Slot Number and Length Pair

Chapter 6

Simulation Results

We now present some simulation results in order to verify analytical solutions obtained in previous chapters and to illustrate the self adjusting price to network load and the inherent quality of service of ICMAC. We also present comparative performance figures between ICMAC and the IEEE 802.11 DCF a for few simple scenarios. We have used OPNET for simulation and each point corresponds to multiple runs. The scenarios are all the same and that is n nodes sending to one AP. The control transmission rate is kept at 1Mb/s throughout the simulations.

6.1 ICMAC Design

We have addressed design issues relating to optimal fragment size and number of slots per round in chapter 5. We now show results for 10 nodes with 11Mb/s data rate and exponentially distributed message for 4 different means of 512 bytes, 1024 bytes, 2048 bytes and 4096 bytes. Packets are individually scheduled, $Maxsch=1$. We plot in Fig. 6.1 the throughput for different fragment sizes and the network performance peaks are in agreement with the analytical optimum fragment size in Table 6.1. Simulation results for optimal fragment size in the case of multiple packet scheduling are also in agreement with the analytical solution and shown in Fig. 6.2. The simulations results have a 90% confidence interval below 2%.

message mean(bytes)		512	1024	2048	4096
optimum fragment length (bytes)	individual	771	1174	1756	2591
	mult. scheduling	593	888	1311	1914

Table 6.1: Optimal Fragment Size

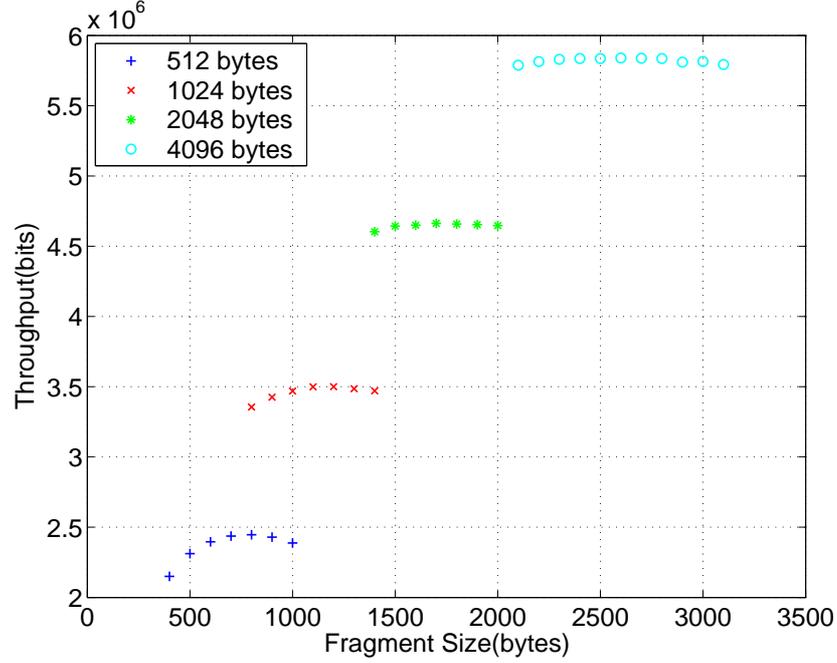


Figure 6.1: Individual Scheduling

6.2 Multiple Packet Scheduling

As previously mentioned, there is an advantage for scheduling multiple transmissions for the same station with only one CTS packet. This benefit is highlighted in Fig. 6.3 again for $n=10$ and $K=50$ for exponentially distributed messages with different message means. Recall that $MaxSch$ is the maximum number of packet that can be scheduled at a time. We show the throughput as $MaxSch$ is increased and disabled. The optimum fragment size changes with $MaxSch$ as discussed in

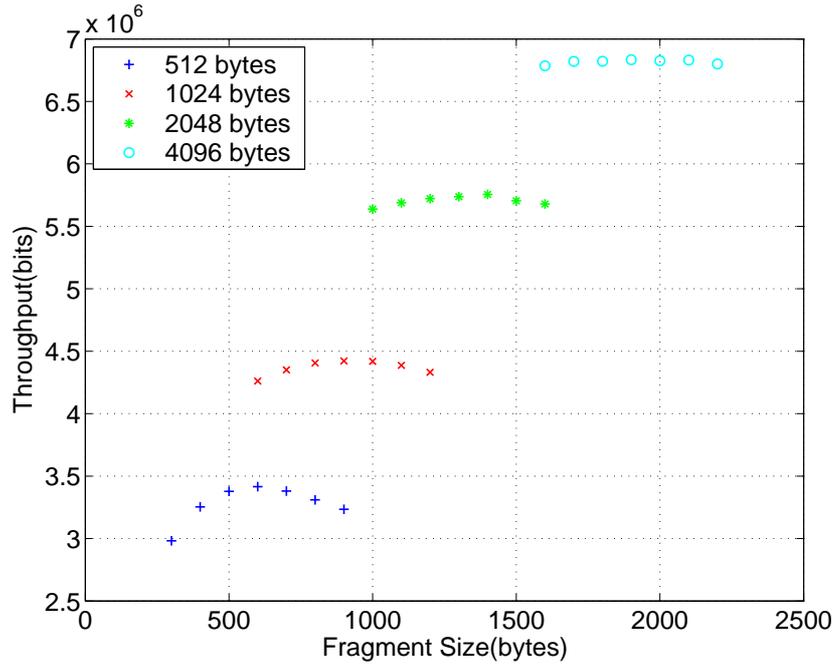


Figure 6.2: Multiple Scheduling

chapter 5. However in this experiment we kept the fragment size fixed corresponding to the optimum for individual scheduling. The throughput gain from individual packet scheduling to completely disabling *Maxsch* is between 15% to 25%. This significant gain is due to the added overhead of scheduling each packet separately and especially as the control rate is kept at $1Mb/s$. The relative gain drops as *MaxSch* is increased. In previous work [19], we have shown a lower gain when *MaxSch* is disabled. This is due to the way bidding ties are handled. In the original work, individual bid value are handled separately, but in the new scheme, bids coming from the same user with same prices are grouped together. In other words, when adding these bids to the aggregate bids, same value bids from the same user are never interleaved with other user bids. In case of ties, user/stations are chosen randomly.

message size.

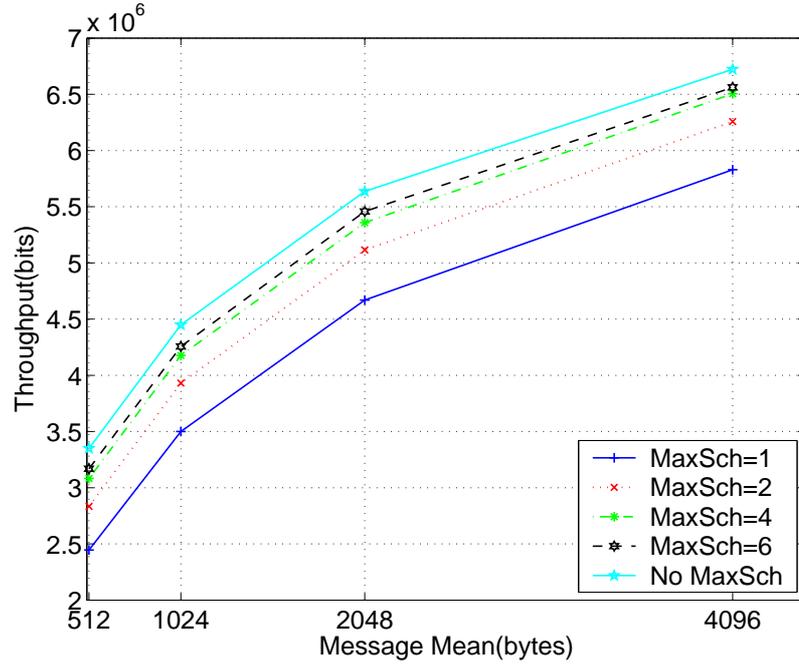


Figure 6.3: Multiple Scheduling

6.3 ICMAC vs IEEE 802.11

Finally, we show a throughput comparison between ICMAC and IEEE 802.11 in Fig. 6.4. We have simulated 5 runs of 30 seconds for each point yielding a 90% confidence interval less than 1%. For IEEE 802.11, we used a simple collision model, that is a transmission is lost only if two or more concurrent transmissions start at the same time. We set the RTS/CTS threshold at 512 bytes, that is packet larger than 512 bytes exchange control messages before data transmission. Packets larger than 2304 bytes get fragmented. Data and control transmission rates are at $11Mb/s$ and $1Mb/s$ respectively for both protocols. All nodes are in saturation mode and

message sizes are again exponentially distributed with varying means as indicated in the figure. Note that we are not favoring ICMAC here with exponential message size since the fragments are of fixed size and some bandwidth is wasted when data does not fully occupy the slot. We show the results for ICMAC with 1 round scheduling and $K=50$. Both protocols perform better with larger message means. As one would expect, the throughput of ICMAC drops with the number of nodes due to the initial bid collection at the beginning of every round. IEEE 802.11 shows a similar trend with respect to the number of nodes but at a slower rate. Note that we use a simple physical model here and IEEE 802.11 does not suffer from hidden terminals as all transmissions are simultaneously and correctly detected, but in a real scenario this would induce degradations in performance.

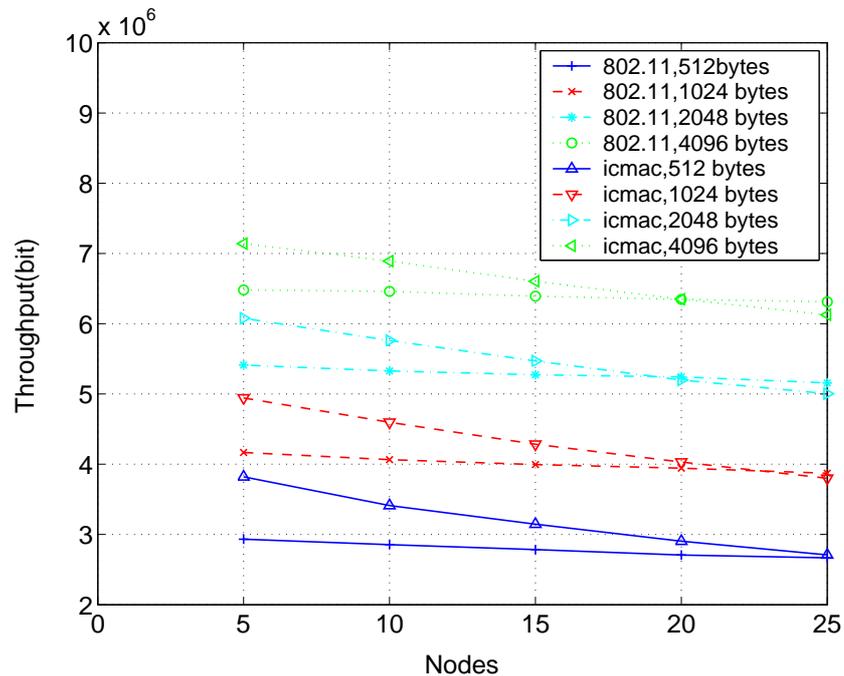


Figure 6.4: IEEE 802.11 and ICMAC Throughput

6.4 Communication Cost

We now show how communication cost gets adjusted with traffic demand.

6.4.1 Same User Types

The network consists of 12 nodes communicating to an AP. In this scenario, all nodes are of the same type and value the slot length and their data packets the same. Their slot valuation is solely based on the packet waiting time. We have used sigmoid function described by the third expression in (4.1) with the parameters in table 6.2. The parameters *range* and *offset* are more intuitive. *offset* is the function value at 0. *range* is the maximum value of the function minus the offset. The relations between c , d , *offset* and *range* are

$$c = \frac{range}{1 - \frac{1}{1 + \exp(a*b)}} \quad (6.1)$$

$$d = offset - \frac{c}{1 + \exp(a * b)} \quad (6.2)$$

Parameters	Value
a	5
b	1
<i>range</i>	256
<i>offset</i>	0

Table 6.2: Valuation Function Parameters

All stations generate data packets according to a poisson process with mean

interarrival time of $0.02s$. Packet sizes are exponentially distributed with mean of 1024 bytes. 9 nodes are always generating traffic corresponding to a total load of $4.096Mb/s$. The three remaining stations 10, 11 and 12 generate traffic only at 20s and 60s for a duration of 20 seconds each time. When all 12 nodes are transmitting, the network load becomes $4.915Mb/s$.

In Fig. 6.5 we show the average slot price at every round for 3 different simulation runs. The slot price starts increasing as a reflection of the buffers getting filled as shown in Fig. 6.6. We have only shown the queue length of stations 1 and 10 of the first run. In this example, 50 slots are assigned at every round and the slot length corresponds to a packet size of 900 bytes. When stations 10, 11 and 12 are

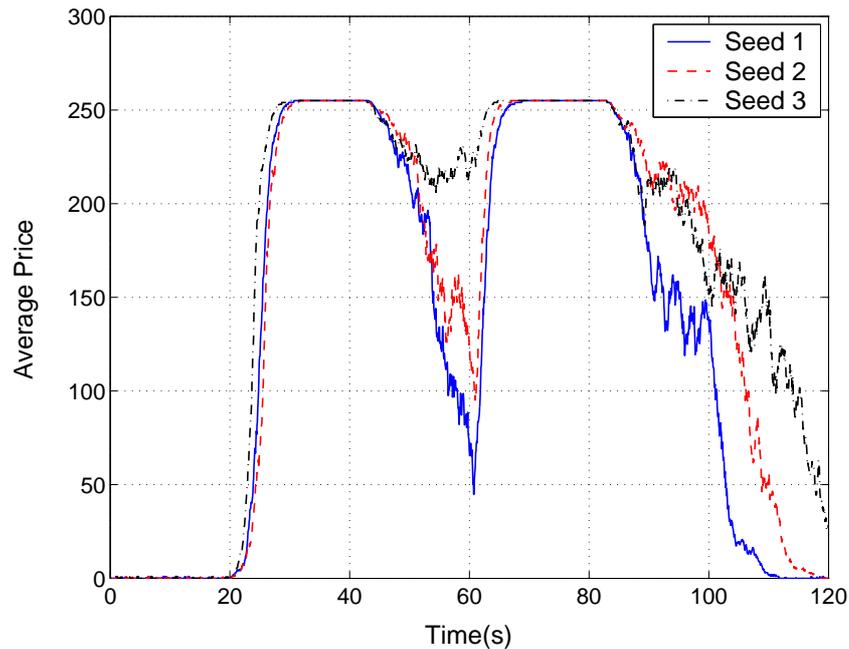


Figure 6.5: Self Adjusting Price

inactive, the system is stable but becomes saturated when they become active. We plot the network load and the average price response in Fig. 6.7 for the first run.

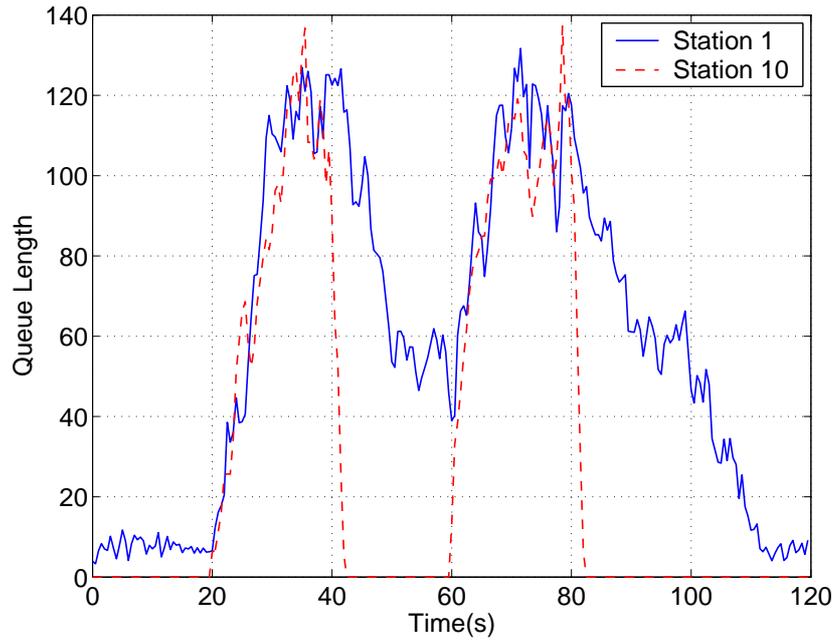


Figure 6.6: Stations Queue Length

Recall that the round duration is a function of the number of stations associated with the AP, the slot length, and the number slots allocated in a round. The round duration for this scenario in the case of individual scheduling is 76.2 ms. However, in the simulation we allow for multiple packet scheduling when consecutive winning bids correspond to the same station and this results in a shorter duration as some of the CTS message are eliminated. Note that the MAC delay profile in Fig. 6.8 follows that of the price. Due to the valuation function in this example, the AP services the station with the longest waiting time and the overall system can be viewed as first come first serve.

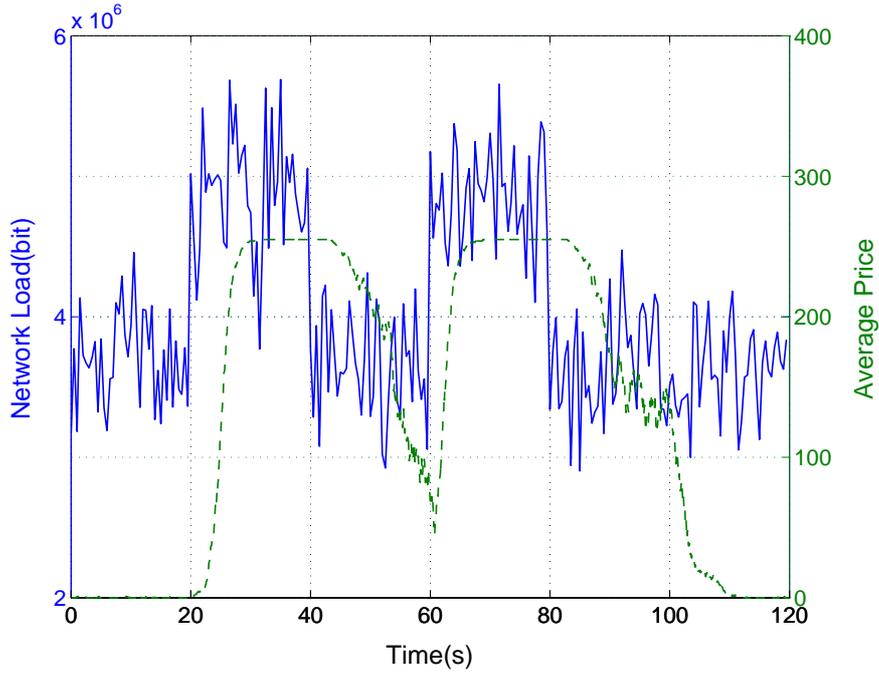


Figure 6.7: Average Price Response to Load

6.4.2 Different User Types and Qos

We now take a look at network with three different types of users with three different valuation functions. Again the functions are taken to be sigmoid function with parameters shown in Table 6.3. We have also plotted the three function in Fig. 6.9. The first valuation function can be associated with normal traffic and the other two functions can be associated with higher priority data traffics.

In this scenario, the first 10 stations generate packets according to a poisson process with mean packet interarrival time of $0.022s$ and the message size are exponentially distributed with mean of 1024 bytes. Stations 11 and 12 start generating traffic at time 25s and 50s respectively until 100s. Stations are divided into three categories. Stations 1-4 generate normal data type/priority. Stations 5-8 have high

Type	Parameters	Value
Normal	a	5
	b	1
	$range$	63
	$offset$	0
High	a	10
	b	0.1
	$range$	63
	$offset$	64
Highest	a	20
	b	0.05
	$range$	63
	$offset$	128

Table 6.3: Valuation Function Parameters

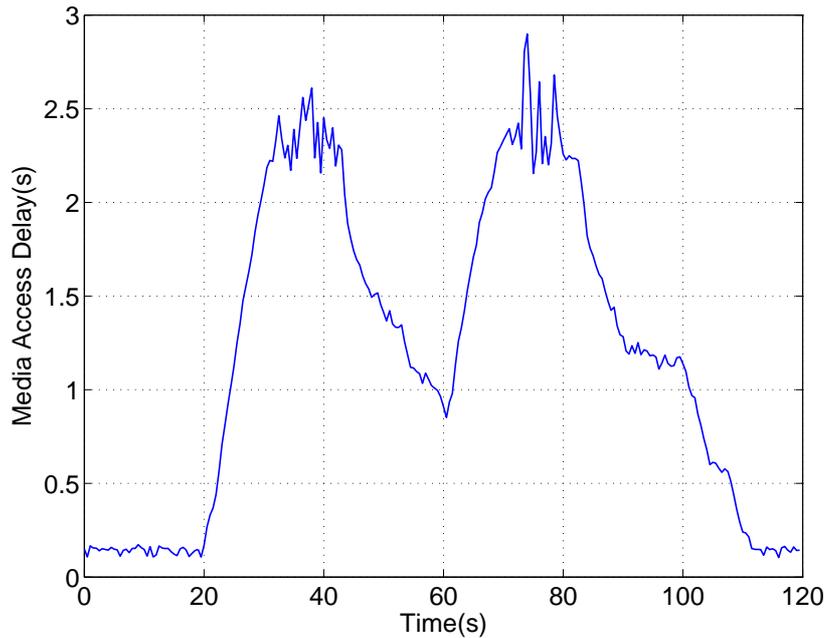


Figure 6.8: Medium Access Delay

data type and the rest of the stations generate the highest data types. When only 10 stations are transmitting, the network is still stable. When station 11 and 12 start generating data traffic, the network becomes overloaded. However, due to station/user types and their respective valuation function, different type stations see different performances. In Fig. 6.10, we show the media access delay of the three different types. The delay experienced by the different stations is ranked according to their types and data priorities. Normal priority stations start seeing an increased delay as station 11 and 12 start generating traffic. High and highest priority stations experience little variation as stations 11 and 12 become active. Stations of high priority experience an average MAC delay of 70ms and station of highest type experience a delay of 50ms even when the network is overloaded. It is important to note that low delay experienced by the stations even when their valuation and

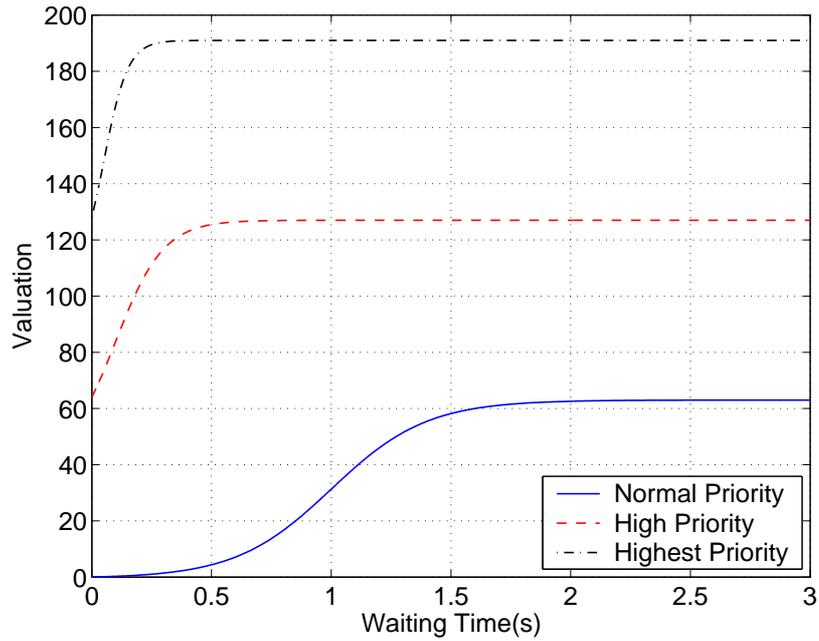


Figure 6.9: Valuation

transmission request solely depends on data packet present in the queue. Transmission request could depend on running services even in the absence of packets in the queue during demand request collection phase.

Only normal priority stations experience buffer overflow. After the buffers get full, normal type stations start dropping new incoming data traffic at a rate of 75kb/s. The transmission price gets adjusted and reflects only the opportunity cost. Initially slot prices are very low, but they get a slight jump at 25 s with the arrival of station 11 and further increase after 50 s when the network becomes overloaded. The average price of the slot becomes 63 corresponding to the price of normal type stations and their willingness to pay for the slots when experiencing long waiting times. The price unit must be set by the administrator. The price needs to provide enough differentiation between the types and take into account the

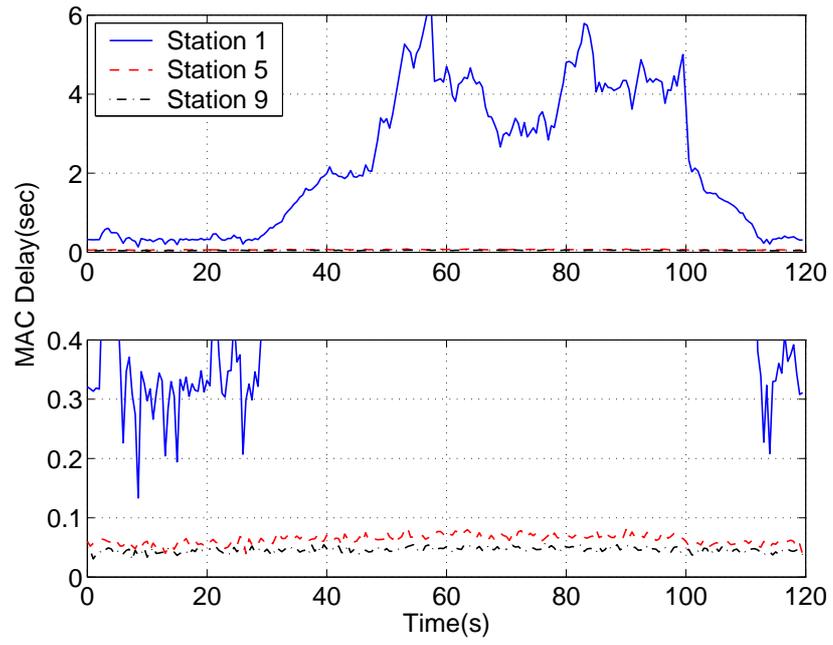


Figure 6.10: Medium Access Delay

BidRep bit representation of values.

Chapter 7

Conclusion

We have introduced a new Incentive Compatible Medium Access Control that takes into account the independence of the entities participating in a wireless network. The stations share a common bandwidth resource and have a utility maximization objective. As previously noted by others and in the simple normal form game discussed previously, the emergent Nash equilibrium yields a low throughput due to the low cost to payoff ratio of transmission. The initial objective of this work has been to design a new MAC protocol where participating stations have an incentive to follow protocol rules and the Nash equilibrium reflects true user types and access valuation even in environment of incomplete information. We have resorted to auction theory to allocate bandwidth in a non-cooperative environment. The new Incentive Compatible MAC is based on the Vickrey auction. At every round, bids are collected for K time slots from the various stations, and then transmission time slots are assigned to the various stations according to the highest bids. The price paid by the winning stations reflect the opportunity cost. The benefit of using Vickrey auction is two fold. First, it keeps the best bidding strategy simple even in a incomplete information setting, as nodes only know their type and not that of the other competing terminals. The other important feature of Vickrey auction relates to setting up the appropriate transmission cost, the transmission cost is self

adjusting and set by the competing users according to the network load and demand curves. The low transmission cost was the network deteriorating factor in random access. No administrator is required to adjust the usage price according to the load. The wireless network usage becomes free under light traffic load and those who do not wish to pay for bandwidth can still use it then. Through the time slot length, number of slot per round, and rounds per bid collection, the network designer has great flexibility to tailor medium access according to traffic demand. In addition to being robust to greedy behavior, ICMAC shows no degradation in performance with respect to IEEE 802.11 for realistic neighborhood size. ICMAC shows great potential as we have not fully explored other potential improvements. We have tried to keep many parameters similar to IEEE 802.11 for comparative reasons. For instance, note that after bid collection, there is no need to individually send a CTS, the AP can broadcast all at once the slot allocation to all the associated nodes. In this scenario, acknowledgment would be left to higher layers as CTS no longer transmit acknowledgment bits for the previously transmitted packets.

BIBLIOGRAPHY

- [1] *IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE 802.11 Std., 1999.
- [2] N. Borisov, I. Goldberg, and D. Wagner, “Intercepting mobile communications: the insecurity of 802.11,” in *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, 2001, pp. 180–189.
- [3] A. B. Mackenzie and S. B. Wicker, “Stability of multipacket slotted Aloha with selfish users and perfect information,” in *Proc. IEEE Infocom*, vol. 3, 2003, pp. 1583–1590.
- [4] Y. Jin and G. Kesidis, “A pricing strategy for an ALOHA network of heterogeneous users with inelastic bandwidth requirements,” in *CISS*, March 2002.
- [5] —, “Equilibria of a noncooperative game for heterogeneous users of an ALOHA network,” *IEEE Comm. Letters*, vol. 6, no. 7, pp. 282–284, 2002.
- [6] E. Altman, R. E. Azouzi, and T. Jiménez, “Slotted Aloha as a stochastic game with partial information,” in *Proc. Wiopt*, 2003.
- [7] J. Bellardo and S. Savage, “802.11 denial-of-service attacks: Real vulnerabilities and practical solutions,” in *Proc. USENIX Security Symposium*, August 2003.
- [8] M. Raya, J.-P. Hubaux, and I. Aad, “DOMINO: A system to detect greedy behavior in IEEE 802.11 hotspots,” in *Proc. MobiSys*, 2004, pp. 84–97.
- [9] S. Radosavac, J. Baras, and I. Koutsopopoulos, “A framework for MAC protocol misbehavior detection in wireless networks,” in *Proc. ACM Workshop on Wireless Security (Wise05)*, 2005, pp. 33–42.
- [10] A. Cardenas, S. Radosavac, and J. Baras, “Detection and prevention of MAC layer misbehavior for ad hoc networks,” in *Proc. Workshop on Security of AD Hoc and Sensor Networks (SASN)*, 2004, pp. 17–22.
- [11] P. Kyasanur and N. Vaidya, “Detection and handling of MAC layer misbehavior in wireless networks,” in *Proc. International Conference on Dependable Systems and Networks*, 2003, pp. 173–182.
- [12] D. Fudenberg and J. Tirole, *Game Theory*. MIT Press, 1991.
- [13] P. Cramton, “Advanced microeconomics lecture notes,” 2005.
- [14] Y. Xiao, X. Shan, and Y. Ren, “Game theory models for IEEE 802.11 DCF in wireless ad hoc networks,” *IEEE Radio Communications*, pp. 22–26, March 2005.

- [15] J. Konorski, “Playing CSMA/CA game to deter backoff attacks in ad hoc wireless LANs,” in *Proc. ADHOC-NOW*, 2005, pp. 127–140.
- [16] G. Bianchi, “Performance analysis of the IEEE 802.11 distributed coordination function,” *IEEE journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, March 2000.
- [17] V. Krishna, *Auction Theory*. Academic Press, An Elsevier Science Imprint, 2002.
- [18] Adwords learning center. Google. [Online]. Available: <http://www.google.com/adwords/learningcenter/>
- [19] N. BenAmmar and J. S. Baras, “Incentive compatible medium access control in wireless networks,” in *Proc. Workshop on Game Theory for Networks (Gamenets)*, 2006, to appear.