A Novel Collision Probability based Adaptive Contention Windows Adjustment for QoS Fairness on Ad Hoc Wireless Networks†

Hengheng Xie[1], Azzedine Boukerche[1 3], Richard W. Pazzi[2]
[1]PARADISE Research Lab - School of Information Technology and Engineering - University of Ottawa - Canada
[2]Faculty of Business and Information Technology - University of Ontario Institute of Technology - Canada
[3]IMDEA Institute and University of Carlos III of Madrid, Spain
Email: hxie072@site.uottawa.ca, richard.pazzi@uoit.ca, boukerch@site.uottawa.ca

Abstract—**It is crucial to achieve Quality of Service (QoS) on IEEE802.11 in order to provide stable and reliable communication for real time and multimedia applications. Most of the recent QoS techniques for Ad Hoc Networks rely on basic QoS classifications such as Enhanced Distributed Channel Access (EDCA) and Hybrid Coordination Function Channel Access (HCCA) with stationary backoff range for specific services, without considering overall resource consumption and the external correlation among terminals. In this paper, a novel contention window adjustment QoS scheme, as a special case of QoS classification, is proposed for achieving QoS-fairness on Ad Hoc Networks. It applies dynamic automatic QoS assignment with low complexity by considering the restriction among various terminal requirements. Throughput estimation with a proportional relation between throughput and backoff parameters is discussed based on a Markov chain model of a saturated Ad Hoc Networks. We measure the performance of different QoS algorithms theoretically and verify improvements of our algorithm on QoS-fairness with detailed simulations.**

I. Introduction

With the rapid spread of mobile devices such as smartphones, PDAs, laptops, tablets, etc., real time applications with a variety of purposes are being widely supported on these devices. Real time applications have strict constraints in terms of delay, throughput and other QoS parameters in order to meet the operational deadline from event to system response. Due to the lack of QoS techniques on early version of IEEE802.11 [1], a QoS scheme must be implemented on the Ad Hoc Networks in order to provide a stable and practical environment for real time applications. Even though EDAC and HCCA are supported in IEEE802.11e, their stationary parameter has difficulty handling a busy networks with large amount of transmission, because it neglects the external relation among terminal requirements.

Contention window mechanisms are designed to avoid collision on networks. If a terminal attempts to access one resource simultaneously, it will cause collision. When a collision occurs, the contention window mechanism forces the terminal, which has started the transmission, to wait for a random time for retransmission. Thus the retransmission has a smaller opportunity to collide. A suitable contention window range can simultaneously absorb the increased system load and guarantee the maximum throughput.

Previous Contention Window adjustment algorithms focus on maximizing overall throughput and ignore requirement variety. A Determinist Contention Window Algorithm (DCWA) [2] is a mechanism by which to get a more specific contention window range. During the contention stage, it increases both the upper and lower bound of the contention backoff parameter through the predefined contention window range at this stage. The contention backoff is chosen from a lower bound to an upper bound, rather than from 0 to the upper bound. The contention window's decrease process is based on the current networks load and networks history. The study [3] proposed a GDCF mechanism, which is an improvement upon Distributed Coordination Function (DCF) on IEEE802.11, by limiting the contention window recovery only on continuously successful transmissions.

Some other research tries to consider terminal requirement specification based on contention window classification, which assigns the terminals into different priority groups by different contention window ranges. Based on the contention window mechanism, the size of contention window affects the average number of backoff parameters. A non-contiguous contention window range mechanism [4] was proposed in order to provide better relative performance between high priority class and low priority class. This method only support two priority classes and one total contention window range. Usually, there are more than two QoS classes for the differentiating application. The design should be able to support multiple priority classes. In order to guarantee the QoS of real time multimedia application on IEEE802.11, an Adaptive Contention Window Size Adjustment scheme (ACWA) [5] has been developed. A cross layer design was introduce to ACWA, in order to cooperate with the Networks layer and MAC layer. The MAC layer was designed as a multiple priority queue for outgoing packets. The packet scheduler puts packets of a flow in the suitable queue, based on a flow requirements and channel state. A Multiplicative-Increase Linear-Decrease (MILD) [6] based on a Sliding Contention Window (SCW) is proposed,

† This work is partially supported by NSERC DIVA Network, Canada Research Chair Program, MRI/OIT Research Funds and Prof. A. Boukerche's Visiting Professor at UC3M/IMDEA.

which provide a SCW per priority class. The contention windows changing on the contention stage is based on the MILD algorithm, with the aim to get higher performance under heavy system load conditions. This research focuses on how to manage the different QoS constraints among priority groups. The study [7] provides a Call Admission Control (CAC) design in order to satisfy the QoS fairness between high priority and low priority connections. Adaptive Contention Window Adjustment (ACA) is an additional mechanism provided in order to reduce the collision while the system load increases.

The purpose of the contention window mechanism is to avoid collision, when more than one device tries to simultaneously send a packet on a shared media. The contention avoidance mechanism should be more responsive to the interaction among terminals. An improved Distributed Coordination Function (DCF) scheme, DCF-MB, has been proposed in [8]. This research tries to solve the unfairness behavior of the DCF algorithm by multiple contention windows principle. [9] points to a solution to the QoS fairness problem caused by DCF on IEEE802.11. This paper presents a non-linear function by which to measure the difference between the achievable throughput and the corresponding QoS requirements. A Neural Networks will be built based on this non-linear function. After learning the Neural Networks, the non-linear function can adaptively determine the suitable backoff for the specified QoS requirement. Even though most of the current QoS classification algorithms can partially solve the QoS problem on Ad Hoc Networks, they cannot achieve real QoS fairness over the networks for the same reasons as EDCA and HCCA, which ignore the external relation among terminal requirements.

Previous concepts of fairness sought to provide similar resource for one priority group, but terminals in one group may need less bandwidth than the fairly shared bandwidth. Some of the bandwidth is actually wasted, because terminals gain more than what is requested. The real fairness is that each request will receive exact resource that they request. Different priority groups can not represent the specific requirement of a terminal, because there are still requirement variety inside each group. Consideration of specific terminal requirement is needed to minimize the wasted resource to achieve the real fairness. The throughput of one terminal is not only decided by itself, but also affected by the other terminals. In order to solve the problem, we need a analysis on external relation among terminals to calculate specific parameter for each one. The main interruption among terminals on MAC layer is contention. A contention analysis on the WLAN for formulating the relation between throughput and contention window is found in this paper. The analysis of collision probability is able to improve the networks performance by scheduling the packets based on the networks traffic estimation [10].

Markov Chain [11] is able to model the system, whose next state only depends on the current state. It is very useful on statistical modeling. Many researches use Markov Chain for collision probability prediction. Paper [12] create a 2-state discrete time Markov Chain with specified probability to predict the collision probability of direct and staggered collision. In order to investigate the collision probability on saturated IEEE802.11 Ad Hoc Networks, paper [13] makes an analysis about the effect of contention window range to the collision probability. It presents a way to predict that the collision probability is affected by average contention backoff and the number of terminals in the channel. A Markov Chain is also created for modeling the saturated channel. With the Markov Chain, it is able to calculate the probability of getting a idle or busy channel in next step. A performance evaluation is carried out in paper [14] for analysis of Distributed Coordination Function (DCF) on IEEE802.11. The authors first studied the stationary probability of single terminal with Markov Chain in a generic slot time. Then they provide the function description of the basic and RTS/CTS access method with the stationary probability retrieved from the first study.

In this paper, we model the saturated channel as a Markov Chain for measuring the stationary probability. From the Markov Chain, we can model the collision probability and do a analysis on the average throughput on each terminal. Based on the modeling information, we retrieve a ratio relation between throughput and backoff parameters. With this proportion, a contention window adjustment mechanism was proposed. The rest of the paper will be organized as follow: Chapter II discuss the Markov Chain of transmission behavior on a saturated channel, and achieve a analysis on the transmission failure probability and average throughput estimation; Chapter III presents a contention window adjustment algorithm based on the analysis in Chapter II; Chapter IV indicates some simulation to emphasize the performance of our proposed system; finally Chapter V summarize the contribution of this paper.

II. Average Throughput Analysis

In our analysis, we assume that the network is in saturation condition, which means all terminals always have a packet to send and all queues are non-empty. Our analysis attempts to model the transmission failure on a saturated channel. Though the causes of transmission failure vary, they trigger the contention window mechanism every time in IEEE802.11. In order to study the behavior of the saturated channel, we model the system as a discrete time Markov Chain. Figure 1 is the Gilbert Burst model[15], while a terminal is sending a packet on the channel. a and b is the probability corresponding to the transmission of the terminal, which is greater than 0 and smaller than 1. We have two states in our transition model, the *Start* state and the *Transmission* state. If a transmission fails, the model will go to or stay in *Start* state. The transition will go to *Transmission* state, only when a transmission is successful. At each state, there is a probability for it to transit to another state. For example, the transition probability from the *Start* state to *Transmission* state is the probability of a success packet transmission on a saturated channel. In the following sections, we will discuss how to calculate the transmission failure probability with our model, and how to

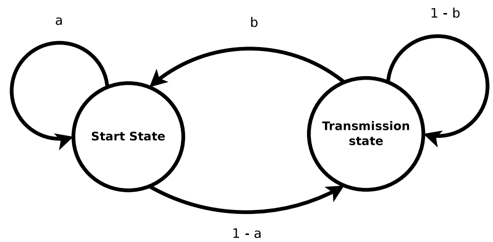

Fig. 1. Gilbert Burst Model of Transmission on a Saturated Channel

estimate the average throughput of one terminal in a saturated channel.

A. Transmission Failure Probability

We make the assumption that the network model is in a saturated condition. We can therefore assume that the network condition is stable at any point during the runtime. The probability of sending a packet successfully is the same during the runtime. The probability of each transmission is shown in the transition matrix **P** in equation (1). Transition matrix $\mathbf{P}^n$ represents the transmission matrix after n transmission times.

$$P = \begin{bmatrix} a & 1-a \\ b & 1-b \end{bmatrix} \tag{1}$$

In a saturated channel, the probability of losing a packet at each transmission is greater than 0 because of the busy traffic. According to the theory of Markov Chain, every transition matrix $\mathbf{P}^n$ of regular Markov Chain can converge to a matrix **W**, which has constant columns. Therefore, the Markov Chain that we create is a regular Markov Chain because all elements in the matrix are positive.

$$W = \begin{bmatrix} w_1 & w_2 \\ w_1 & w_2 \end{bmatrix} \tag{2}$$

Based on the theory of Markov Chain, we can calculate the w_1 and w_2 from the equations derived from equations (3).

$$\begin{cases} w_1 + w_2 = 1 \\ w_1 * a + w_2 * b = w_1 \\ w_1 * (1-a) + w_2 * (1-b) = w_2 \end{cases} \tag{3}$$

We then determine the convergence matrix **W** of $\mathbf{P}^n$ as shown in equation (4), where all rows in matrix **W** are the same vector. When $n \to \infty$, $\mathbf{P}^n$ is equal to **W**. We can predict the probability of the transmission based on matrix **W** after sufficient steps are taken. In the next section, we will discuss the prediction of average throughput of each terminals according to the matrix **W**.

$$W = \begin{bmatrix} \frac{b}{1-a+b} & \frac{1-a}{1-a+b} \\ \frac{b}{1-a+b} & \frac{1-a}{1-a+b} \end{bmatrix} \tag{4}$$

B. Average Throughput Analysis

The contention window mechanism aims to achieve contention avoidance, in order to improve the average bandwidth $E[B]$ of each terminal. Therefore, in our analysis, we seek to know how the contention backoff parameter affects the average bandwidth. Firstly, we create an equation for predicting the average bandwidth in the Markov Chain, as generated in the previous section. $E[B]$ can be formalized as equation (5).

$$E[B] = \frac{TP_{total}}{T_{total}} \tag{5}$$

T_{total} denotes the total measured time, and TP_{total} denotes total throughput within T_{total}. We assume that a terminal transmits **N** packets during T_{total}, while **N** is large enough. In this case, transmission matrix $\mathbf{P}^N$ is closed to matrix **W**. The Markov chain starts at *Start* state, so we only focus on the probability that start at the *Start* state. The transition matrix can multiply to a vector $\begin{bmatrix} 1 & 0 \end{bmatrix}$ in order to eliminate the probability corresponding to the transmission state at the first state. The amount of TP_{total} and T_{total} can be calculate as equations (6).

$$\begin{cases} TP_{total} = \sum_{n=1}^{N} \begin{bmatrix} 1 & 0 \end{bmatrix} * P^n * \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ T_{total} = \\ \sum_{n=1}^{N} \begin{bmatrix} 1 & 0 \end{bmatrix} * P^n * \begin{bmatrix} E[BF] \\ E[T_{tx}] \end{bmatrix} \end{cases} \tag{6}$$

$E[BF]$ denotes average Backoff parameter and $E[T_{tx}]$ denotes average time of packet transmission. The total throughput T_{total} of a terminal is the number of successful transmissions during the runtime. Therefore, it multiplies a vector $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ to choose the number of successful transmissions from the matrix. The total timeslot that the terminal uses should be separated into two situations: 1) When the transmission succeeds, the cost timeslot is the time spent transmitting a packet; 2) When the transmission fails, the cost timeslot should be based on the backoff time chosen by the contention window mechanism. Here we will use the average transmission time and average backoff parameter to replace the exact execution time.

$$E[B] = \frac{\begin{bmatrix} 1 & 0 \end{bmatrix} * ((N+1)W - I) * \begin{bmatrix} 0 \\ 1 \end{bmatrix}}{\begin{bmatrix} 1 & 0 \end{bmatrix} * ((N+1)W - I) * \begin{bmatrix} E[BF] \\ E[T_{tx}] \end{bmatrix}} \tag{7}$$

Based on the theory of the regular Markov Chain, $\mathbf{P}^n$ is predictable, because it converges to **W**. By replacing $\mathbf{P}^n$, we get the equation (7) from the equations (6), when the number **N** is large enough.

$$\lim_{n \to \infty} E[B] \approx \frac{1}{\frac{b}{1-a} * E[BF] + E[T_{tx}]} \tag{8}$$

After simplifying equation (7) by making n sufficiently close to infinite, we get equation (8). In order to avoid further contention after collision occurs, the contention window must not be too small. Thus the average transmission time is usually very small compared to the average backoff time. In a saturated channel, probability *a* and probability *b* are very similar, because because of their stable environment. Based on the research in the study [13], the collision probability a reasonable network size with 30 terminals is within the range of [45%, 55%], while the size of the initial contention window is 16 to 32. In a reasonable network size with about 30 terminals transmitting data simultaneously, *a* and *b* are similar and within the range of [0.45, 0.55]. The factor $\frac{b}{1-a}$ is greater than 80% in the network with a size of 30 terminals. Therefore, $\frac{b}{1-a} * E[BF]$ is still much greater than $E[T_{tx}]$. We can predict the average throughput $E[B]$ is as approximate to as equation (9) in a reasonable network size with 30 terminals.

$$E[B] \approx \frac{1}{\frac{b}{1-a} * E[BF]} \tag{9}$$

III. Proposed Algorithm

The fixed contention window range of a set of terminals cannot satisfy the requirement of real time application in Ad Hoc Networks, because the fixed parameters cannot adjust the throughput according to the various QoS requirements. In this chapter, we propose a novel contention window adjustment algorithm based on the theory put forth in Chapter II, which proposes assigning corresponding contention window ranges to a specific QoS requirement. In the previous chapter, we acquired the estimated average throughput of a terminal from our analysis of transmission probability in a saturated channel. The result shows that the average throughput of a terminal principally depends on its average contention backoff parameter as well as the channel condition. In other words, it is affected by the size of the contention window, because the contention window range decides the average backoff parameter. Therefore, when the contention window range gets smaller, the average throughput should get larger until the system resource is exhausted.

$$E[BF] \approx \frac{1}{\frac{b}{1-a} * ReqTP} \tag{10}$$

A. Basic Theory

Our goal is to achieve QoS fairness over Ad Hoc Networks, which means each terminal will get the corresponding resources that it requests. If we set the average throughput to the QoS requested throughput, we can easily predict the average backoff parameter based on the corresponding QoS requested throughput as equation (10). In equation (10), *ReqTP* denotes the QoS requested throughput. The suitable average backoff parameter is dependant on the channel condition, *a* and *b*, and the requested throughput of the application. From the previous equations, we can see that the average throughput is basically decided by the backoff parameter, but also affected by the channel condition. We cannot determine the average throughput of one terminal without considering the channel condition. Therefore, we need to analyze the external relation among terminals. Since we assume that all terminals work in a saturated channel, we can assume that the channel condition of all terminals is the same. According to this assumption, we acquire the ratio of the contention backoff parameter of all terminals within the system in equation (11).

$$E[BF]_1 : E[BF]_2 : ... : E[BF]_n \approx \frac{1}{ReqTP_1} : \frac{1}{ReqTP_2} : ... : \frac{1}{ReqTP_n} \tag{11}$$

This ratio reveals the external relation among the terminals within the system, and how the QoS requested throughput affects the average contention backoff parameter. Based on the definition of a contention window mechanism, backoff time is the random number of time slots in the range from 0 to the size of the contention window. The average backoff parameter should be half of the size of the contention window, therefore we can get another ratio between the contention window range and the QoS requested throughput in equation (12).

$$CW_1 : CW_2 : ... : CW_n \approx \frac{1}{ReqTP_1} : \frac{1}{ReqTP_2} : ... : \frac{1}{ReqTP_n} \tag{12}$$

Equation (12) is the primary theory behind our algorithm. Like HCCA, there is a QoS coordinator in our algorithm, which collects channel information and sets terminals to an appropriate contention window size corresponding to the ratio of requested throughput.

B. Overload Control

The definition of overload in our algorithm is the requested throughput exceeds the system's capable throughput. One drawback to the main theory, proposed in the previous section, is that the actual throughput of each terminal will not reach the requested throughput when the overall requested throughput is greater than the channel capacity. The reason is that channel capacity has been divided by the ratio of requested throughput, not the real number of the requested throughput. In this chapter, we will discuss this problem, and proceed to present an overload control in the system.

While the requested resource is more than the actual resource, the coordinator has to decide whose QoS request should be guaranteed. Therefore, few terminals, called guaranteed terminals, will get the QoS requested throughput, while others, called Best Effort terminals, will get the shared bandwidth in an overload condition. As shown in equation (13), $ReqTP_{grt}$ denotes the requested throughput of guaranteed terminals, and TP_{BE} denotes the assigned throughput of Best Effort terminals. TP_{actual} denotes the channel capacity. The overall throughput of the guaranteed terminals $ReqTP_{overall}$ has to be smaller than the channel capacity, so it has extra throughput for Best Effort requests. The overall throughput of Best Effort terminals should be equal to the rest of the channel capacity, therefore, they will not seize resource from the guaranteed terminals.

$$\begin{cases} 1.ReqTP_{overall} = \\ ReqTP_{grt_1} + ReqTP_{grt_2} + ... + ReqTP_{grt_n} \\ 2.TP_{BE_1} + TP_{BE_2} + ... + TP_{BE_n} = \\ TP_{actual} - ReqTP_{reserved} \\ 3.ReqTP_{overall} < TP_{actual} \end{cases} \tag{13}$$

In this phase, we have two methods by which to separate the rest of the actual throughput for the Best Effort terminals. 1) the rest of the throughput will be separated equally to all the Best Effort terminals; 2) or the rest of the throughput will be separated based on the ratio of requested throughput of the Best Effort terminals.

$$\begin{aligned} CW_{grt_1} : ... : CW_{grt_n} : CW_{BE_1} : ... : CW_{BE_n} = \\ \frac{1}{ReqTP_{grt_1}} : ... : \frac{1}{ReqTP_{grt_n}} : \\ \frac{K}{ReqTP_{BE_1}} : ... : \frac{K}{ReqTP_{BE_n}} \end{aligned} \tag{14}$$

$$K = \frac{ReqTP_{BE_{overall}}}{TP_{actual} - ReqTP_{overall}} \tag{15}$$

Equation (14)and (15) illustrates how to conduct overload control with the ratio division on Best Effort terminals. All Best Effort terminals have to multiply an extra factor, *K*, which converts the normal ratio value to Best Effort ratio value. Therefore, the throughput will be redivided based on the new ratio. To implement this mechanism, the coordinator, which retrieves data from the terminals, should measure the channel capacity, in order to control the overall throughput.

IV. SIMULATIONS

In this chapter, we will discuss the simulation results that verify our algorithm. The simulations were performed on the NS-2 simulator [16] with a built in IEEE802.11 protocol. We chose IEEE802.11, because it is one of the most widely used MAC protocols over wireless networks. The environment we built was a saturate network, with a coordinator receiving message from 30 terminals. One coordinator accept connections from terminals, and each terminal sent packets by UDP protocol with a size of 160 bytes, and sent a packet each 20ms in VOIP G.711 CODEC. The requested QoS of each terminal was chosen based on the continuous uniform distribution.

The criteria of our simulation were the success rate, which measured whether or not the assigned throughput of each terminal met its requested throughput. Success rate measured the percentage that the requested requirement was successfully satisfied. In the simulation, each terminal requests a specific throughput to transmit data. There are two types of terminals, guaranteed terminals and Best Effort terminals. Our algorithm attempts to satisfy the requests of guaranteed terminals, while best effort terminals do not receive any guaranteed performance. We only measured the success rate of the guaranteed terminal in our simulation. If there is available bandwidth, a coordinator will accept the connection and reserve the specified throughput of the terminal. This terminal becomes a guaranteed terminal. Therefore, in a situation where the network is under light workload conditions, every terminal becomes a guaranteed terminal. Figure 2 presents the simulation results when performing under a light workload network. We compare our algorithm to the DCF in the simulations. DCF is unable to satisfy all the requests from the terminals, because it does not consider the different QoS requests from each terminal. In contrast, under a light workload network, our Contention Window algorithm and Overload Control algorithm is able to provide almost complete satisfaction for the terminal requests.

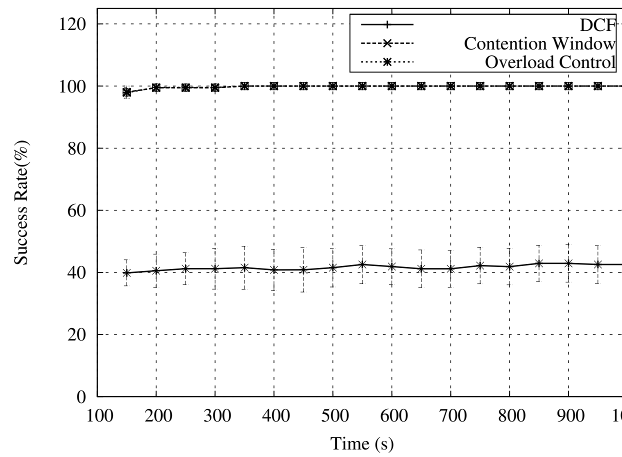

Fig. 2. simulation Result under Light Load Network Transmission

If there is a lack of network resources while a terminal requests transmission, the coordinator will still accept the connection but the terminal will not guarantee network usage. Only a portion of the terminals will receive the reserved resources. Our simulation used the "First Come First Served" approach to decide which terminal would be a guaranteed terminal for all tested the algorithms. Figure 3 illustrates the simulation results, after performing under a heavy workload network. In this simulation, we compare our algorithm with the DCF and the MILD based algorithm. The overall requested resources are 30%-50% and exceed the channel capacity. The MILD based algorithm has three priority classes, high priority, low priority and Best Effort. The total requested throughput is 50% higher than the channel capacity. Our Contention Window algorithm still performs better than the DCF algorithm, but it is unable to satisfy most of the terminals. The reason for this is that our basic Contention Window algorithm makes QoS decisions based solely on the requests. In order to consider the actual network utilization, we introduce an Overload Control Algorithm, which optimizes the system performance over an overloaded network. Our Overload Control algorithm is superior to the MILD based algorithm, even though the MILD based algorithm has a QoS classification mechanism. The MILD based algorithm sets the contention window range

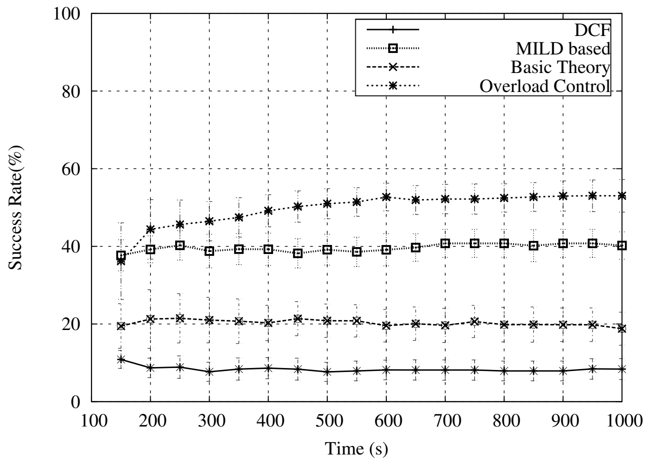

Fig. 3. simulation Result under Light Load Network Transmission

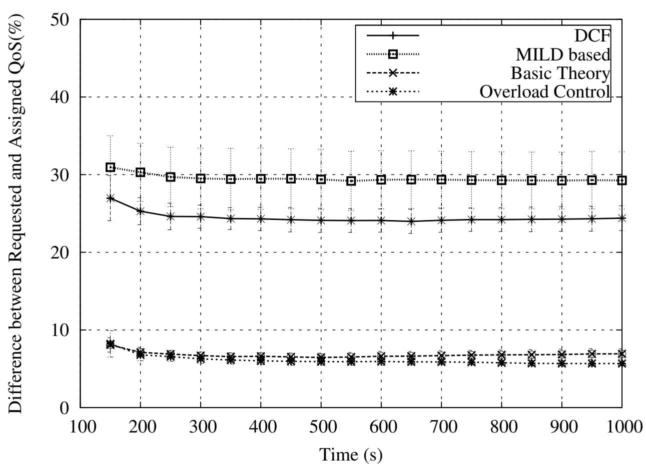

Fig. 4. Difference between Requested and Assigned QoS on Heavy load Network Transmission

only according to the priority. It doesn't consider the specific requested QoS and the relationship between these requested QoS values.

Figure 4 shows the difference between requested QoS and assigned QoS during the heavy load network transmission simulation. We can see that our Overload Control algorithm manages the difference between these two parameters around 5%, while the difference between the requested and assigned QoS of DCF and MILD based algorithms is over 20%. Even though the MILD based algorithm can be satisfactory, it wastes several resources on the unrequested portions. Though our algorithm fails some QoS requirements, it is still able to assign the terminals a resource that is approximate to the requirement. Therefore, despite some terminals not being satisfied, they have nevertheless been partially satisfied.

V. Conclusion

In this paper, we have modeled a saturated network through the Markov Chain model and estimated how the backoff parameter affects network throughput. Previous QoS classification algorithms with fixed groups cannot achieve QoS fairness, because they do not consider the differentiation and external relation of client requests. In order to control the throughput of each terminal over the network, we have designed a Contention Window algorithm, which manages the contention window range of the terminals. Considering the overall network utilization, an Overload Control algorithm has been proposed based on the Contention Window algorithm in order to optimize the system performance over an overloaded network. Our simulation experiments demonstrate that our algorithm performs better than DCF and MILD, and is able to improve the performance and limit the difference between requested resources and assigned resources when the network is overloaded.

References

[1] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE Std. 802.11, 1997.

[2] A. Ksentini, A. Nafaa, A. Gueroui, and M. Naimi, "Determinist contention window algorithm for IEEE 802.11," in *Proc. IEEE International conference on Personal, Indoor and Mobile Radio Communications (PIMRC'05)*, 2005, pp. 2712 – 2716.

[3] W. C., L. B., and L. L., "A new collision resolution mechanism to enhance the performance of IEEE 802.11 DCF," *IEEE Transactions on Vehicular Technology*, vol. 53, p. 1235, Jul. 2004.

[4] M. Mishra and A. Sahoo, "A contention window based differentiation mechanism for providing QoS in wireless lans," in *Proc. IEEE International conference on Information Technology (ICIT'06)*, 2006, pp. 72 – 76.

[5] L. Zhang, G. Wang, and W. Dong, "Adaptive contention window adjustment for 802.11-based mesh networks," in *Proc. IEEE International conference on Wireless Communication, Networking and Mobile Computing (WiCOM'08)*, 2008, pp. 1 – 4.

[6] M. Bharatiya and S. Prasad, "MILD based sliding contention window mechanism for QoS in wireless lans," in *Proc. IEEE International Workshops on Wireless Communication and Sensor Networks(WCSN'08)*, 2008, pp. 105 – 110.

[7] H. Wu, M. Yang, and K. Ke, "The design of QoS provisioning mechanisms for wireless networks," in *Proc. IEEE International Workshops on Pervasive Computing and Communications (PERCOM'10)*, 2010, pp. 756 – 759.

[8] Y. Chetoui and N. Bouabdallah, "Adjustment mechanism for the IEEE 802.11 contention window: An efficient bandwidth sharing scheme," *Computer Communications*, vol. 30, pp. 2686–2695, Sep. 2007.

[9] C. Wang, T. Lin, and J. Chen, "A cross-layer adaptive algorithm for multimedia QoS fairness in WLAN environments using neural networks," *Communications*, vol. 1, pp. 858 – 865, Nov. 2007.

[10] M. Krishnan and A. Zakhor, "Throughput improvement in 802.11 WLANs using collision probability estimates in link adaptation," in *Proc. IEEE International Workshops on Wireless Communications and Networking (WCNC'10)*, 2010, pp. 1 – 6.

[11] R. A. Howardd, *Dynamic Probabilistic Systems,Volume I: Markov Models*. John Wiley and Sons, 1971.

[12] M. Krishnan and A. Zakhor, "Local estimation of probabilities of direct and staggered collisions in 802.11 WLANs," in *Proc. IEEE International Workshops on Global Telecommunications (GLOBECOM'09)*, 2009, pp. 1 – 8.

[13] H. Vu and T. Sakurai, "Collision probability in saturated IEEE 802.11 networks," in *Australian Telecommunication Networks & Applications Conference (ATNAC'06)*, 2006, pp. 1 – 5.

[14] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *Selected Areas in Communications*, vol. 18, pp. 535 – 547, Mar. 2000.

[15] D. Guo and X. Wang, "Bayesian inference of network loss and delay characteristics with applications to TCP performance prediction," *IEEE Transactions on Signal Processing*, vol. 51, pp. 2205 – 2218, Aug. 2003.

[16] NS2. (2007) The network simulator ns-2. [Online]. Available: http://www.isi.edu/nsnam/ns/