國立交通大學

資訊工程研究所

碩士論文

IEEE 802.11 無線區域網路品質保證之柔性允入控制機制

Soft-Guarantee-based Call Admission Control for IEEE 802.11 Wireless LANs

研 究 生：林士勛

指導教授：楊啟瑞　教授

中華民國九十三年八月

國立交通大學

資訊工程研究所

碩士論文

IEEE 802.11 無線區域網路品質保證之柔性允入控制機制

Soft-Guarantee-based Call Admission Control for IEEE 802.11 Wireless LANs

研 究 生：林士勛

指導教授：楊啟瑞　教授

中華民國九十三年八月

IEEE 802.11 無線區域網路品質保證之柔性允入控制機制

Soft-Guarantee-based Call Admission Control for IEEE 802.11 Wireless LANs

研究生：林士勛　　Student：Shih-Hsun Lin

指導教授：楊啟瑞　　Advisor：Maria C. Yuang

國立交通大學
資訊工程研究所
碩士論文

A Thesis
Submitted to Department of Computer Science and Information Engineering
College of Electrical Engineering and Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Computer and Information Science

Aug. 2004

Hsinchu, Taiwan, Republic of China

中華民國九十三年八月

IEEE 802.11 無線區域網路品質保證之柔性允入控制機制

研究生：林士勛　　　　指導教授：楊啟瑞博士

國立交通大學資訊工程學系

摘要

隨著科技的演進，無線網路傳輸速率亦逐漸成長。因此，無線多媒體溝通的需求也隨之增加。IEEE 802.11媒體存取控制機制是目前用於無線區域網路相當常見的方式。但是，此方法對於多媒體通訊並不提供訊務之品質保證(例如：音訊傳輸)。另一方面，允入控制機制也是另一種重要的方法。允入控制主要用於維持已連通知多媒體連結之品質，避免新進入之連結破壞原有的保證。在此，我們希望結合媒體存取控制與允入控制機制兩種概念，建立出一套整合性的演算法Soft-Guarantee-Based Call Admission Control (SCAC)，以提供柔性允入控制。此演算法主要著眼於提供音訊連結品質上的保證並提高整體無線空間的使用率。SCAC包含三個主要的元件：QoS-enabled Call Admission Control (Q-CAC)，Efficient Contention Control Algorithm (ECCA)，and Resource Management Database (RMDB)。以舊有連結的品質要求為基礎， Q-CAC搜尋事先建立好的資料庫(RMDB)，計算出對應的參數用以限制參與競爭的MT數量，並保證原有連結的品質需求。ECCA採用兩階段之演算法已提供音訊較佳之競入控制。透過模擬實驗結果，我們可以證明SCAC能夠提供系統事先定義之柔性保證程度(例如：99%品質保證)於音訊連結，同時提供較佳的競入控制機制給緊急的訊務。

Soft-Guarantee-based Call Admission Control for IEEE 802.11 Wireless LANs

Student: Shih Hsun Lin *Advisor*: Dr. Maria C. Yuang

Department of Computer Science and Information Engineering
National Chiao Tung University, Taiwan

Abstract

With the advent of new broadband technologies providing higher user data rate, the wireless communication is clearly poised for rapid growth. IEEE 802.11 MAC protocol is the most common method used in Wireless LANs (WLANs) with poor QoS guarantee for higher priority traffic such as voice signaling requests. On the other way, call admission control (CAC) method is also important for maintaining the QoS guarantees of communications in AP's polling table. In this paper, we combine the concepts of CAC and MAC to construct a complete algorithm called Soft-Guarantee-Based Call Admission Control (SCAC) method. This method focuses on supporting QoS guarantee for voice traffic. SCAC has three major components: QoS-enabled Call Admission Control (Q-CAC), Efficient Contention Control Algorithm (ECCA), and Resource Management Database (RMDB). Based on demands of MTs in AP's polling table, Q-CAC limits the contending MTs with voice signaling requests for maintaining QoS of voice connections in AP's polling table by searching pre-constructed RMDB. ECCA adopts two phases algorithm by improving Contention Control (CC) concept to provide better contention resolution mechanism for high priority traffic. Simulation results demonstrate that SCAC can support pre-defined soft QoS guarantee (ex: 99% QoS guarantee) for voice traffic and provide lower contending time for high priority traffic such as voice signaling traffic.

Keywords: Wireless networks, Quality-of-Service (QoS), Call Admission Control (CAC), Media Access Control (MAC), Contention Access, Contention Resolution Time (CAT), Feedback Control, Distributed Coordination Function (DCF), Point Coordination Function (PCF), DCF Inter Frame Space (D-IFS), PCF Inter Frame Space (P-IFS), Random Back-off Time (RBT), Contention Control (CC).

Table of Contents

List of Fugure

Symbols

D_{Qos} : Upper limit period without breaking the original QoS guarantee

$\hat{N}$: The predicted number of MTs with voice signaling requests

P_{adm} : The probability used to determine whether join ECCA operation time or not

CCI(1) : First Contention Control Interval length

$CCI(1)_{opt}$: Optimal first Contention Control Interval length

$CCI(k)_{opt}$: Optimal kth Contention Control Interval length

b_i : The feedback signal for the ith slot

N_{adm} : The admitted number of MTs with signaling requests

$\tilde{D}$: The distribution for SCCI

$\tilde{D}_{N,j}$: The distribution of SCCI under N MTs joining and j slots in $CCI(1)$

$D_{N,j}$: The expected value of $\tilde{D}_{N,j}$

$\tilde{D}_{N,j|m}$: The distribution of SCCI under N MTs joining and j slots in $CCI(1)$ with the condition that there are m MTs choose the first slot of $CCI(1)$

$D_{N,j|m}$: The expected value of $\tilde{D}_{N,j|m}$

$Q_{N,j}(m)$: The probability under the condition that N MTs join in ECCA, there are j slots in $CCI(1)$ and m MTs choose the first slot of $CCI(1)$

Acronyms

WLAN	: Wireless LANs
BSS	: Basic Service Set
AP	: Access Point
MT	: Mobile Terminal
QoS	: Quality of Service
CAC	: Call Admission Control
MAC	: Media Access Control
CAT	: Contention Resolution Time
DCF	: Distributed Coordination Function
EDCF	: Enhanced Distributed Coordination Function
PCF	: Point Coordination Function
HC	: Hybrid Coordinator
D-IFS	: DCF Inter Frame Space
P-IFS	: PCF Inter Frame Space
RBT	: Random Back-off Time
CC	: Contention Control
CCI	: Contention Control Interval
SCCI	: Super Contention Control Interval
SCAC	: Soft-Guarantee-based Call Admission Control
Q-CAC	: QoS-enabled Call Admission Control

ECCA	: Efficient Contention Control algorithm
RMDB	: Resource Management Database
PMER	: Pdf-based Multi-user Estimator

1. Introduction

With the advent of new broadband technologies providing higher user data rate, the wireless communication is clearly poised for rapid growth. In wireless LANs (WLAN), not only best effort traffics, the demands on variant real time applications become heavy. In order to support real time traffics such as CBR, VBR, and ABR, we should guarantee QoS requirements for them. IEEE 802.11 wireless LAN (WLAN) [1] is one of the most deployed wireless technologies all over the world and is likely to play a major role in next-generation wireless communication networks.

IEEE 802.11 MAC sub-layer defined two relative medium access coordination functions, the distributed coordination function (DCF) and an optional point coordination function (PCF) [1]. DCF uses CSMA/CA as the basic channel access protocol to transmit asynchronous data and signaling request of synchronous data in the contention period. It uses a pre-defined duration called DCF Inter Frame Space (D-IFS) and a Random Back-off Time (RBT) as the timer for transmitting. The contention free service for time-bounded traffic is provided by PCF which basically implements a “polling” access method with a shorter inter frame space, named as PCF Inter Frame Space (P-IFS). Although PCF has been designed to support real-time service (synchronous data), the requests of QoS traffic (signaling request) still have to contend the channel in contention period with best-effort traffics using DCF. Due to the best-effort property of DCF, the QoS guarantee for signaling requests can not be supported. Besides, when traffic load is getting high, there is no new requests can be handled and the system performance will severely decrease. On the other way, if the new request makes a success contention in DCF mode, the system still can not decide whether the request could join into AP’s polling table or not. If the system accepts the new request without any estimation, the QoS of some requests originally in the polling table will be broken in some case.

In order to solve the problems mentioned above, there are lots of new MAC mechanisms being proposed. Those methods can be taken into two strategies: Refinement transmission timer and contention control. In refinement transmission timer strategy, it can be roughly taken into three categories: RBT refinement, *x*-IFS refinement, combined RBT and *x*-IFS refinement. In RBT refinement, the basic idea behind the design is to adjust RBT size by traffic condition. That is, RBT will be enlarged when traffic load is getting high, and vice versa. [3] proposes RT-FCR algorithm to support QoS for voice traffic reservation (signaling request) by adjusting the RBT based on the channel status. But, the low priority traffics often jam the high priority traffics frequently in this case.

Therefore, the second category is proposed, *x*-IFS refinement. Its essential is to let high priority traffic get short *x*-IFS and let them get more transmission opportunity. IEEE 802.11e [4,5] use enhanced DCF (EDCF) to support various priority traffics. It takes different *x*-IFS for different traffic priority. However, the growing load of the same priority traffics will break the QoS. In order to catch both advantages of those two strategies, combined RBT and *x*-IFS refinement is addressed. Unlike pervious two, it controlled the RBT and *x*-IFS parallelly. Different priority traffic use different *x*-IFS and RBT mechanism. For example, *voice*-IFS is shorter than *data*-IFS and RBT_{voice} grows slower than RBT_{data}. In this design, it can let voice traffic get higher priority. [6] superiorly proposes DBASE protocol to provide hard QoS guarantee for voice traffic reservation. DBASE reassigns the *x*-IFS and RBT, which lets *voice*-IFS plus RBT_{voice} is smaller than *data*-IFS. Through this design can improve the performance of higher priority traffic (ex. voice signaling traffic) than only refining *x*-IFS or RBT, problem of growing load of higher priority still occurs.

In contention control strategy, separating high priority traffic and low priority traffic into different contention periods and using distinct contention resolution algorithms for them is the key point. The popular method is called Contention Control (CC) and CC operation time is called Contention Control Interval (CCI). In this design, high priority traffics can get better QoS support evidently [7] for the reason that low priority must keep waiting until Contention Control Interval (CCI) being finished. However, the same problem that we mentioned before is still unsolved.

In addition, restraining a new signaling traffic from contending into the polling table is needed if we can not supply enough bandwidth for the request or it will break original QoS guarantee for existing connection in AP's polling table. The success hinges on the efficient support of signaling traffic is so called Call Admission Control (CAC). Many CAC strategies are presented. [8] is a static CAC planning using an analysis to compute the maximum number of voice calls in the CBR or VBR mode for the system. [9] dynamically computes the remaining channel capacity by estimating the throughput that original traffic flows would achieve. According this, it makes CAC decision when a new QoS request occurs. In these CAC designs, systems operate MAC mechanisms and call admission control (CAC) methods separately. For example, IEEE 802.11 and its enhancement mentioned before only focus on the MAC layer and PHY layer, without considering any CAC methods in it [3-7] and [8,9] only consider the CAC methods which be only used when a new signaling request makes a success contention. Unfortunately, if the bandwidth is saturated, a new request make a success contention based on the MAC mechanism, it will be rejected by CAC. Under this condition, the bandwidth will be wasted.

In this paper, we consider both MAC and CAC together to design an efficient soft-guarantee-based CAC, called SCAC. The SCAC based on IEEE 802.11e and included three major parts: QoS-enabled Call Admission Control (Q-CAC), Efficient Contention Control Algorithm (ECCA), and Resource Management Database (RMDB). Q-CAC is a call admission control algorithm. According to the QoS requests of existing connections in AP's polling table, the algorithm decides the maximum contention resolution time (CRT) which can meet the existing requirements. Then, Q-CAC determines the maximum number of MTs that system can support by searching RMDB based on CRT. Finally, these MTs with the contending permission employ a new mechanism called ECCA to contend the channel. RMDB is an off-line constructed database by analysis and neural-fuzzy prediction. It records the relation between the system load and contention resolution time which is used by Q-CAC. To formally justify the performance of SCAC, we present pre-defined percentage QoS guarantee simulations. In the analyses, SCAC is proved that soft-guaranteed condition is derived. Finally, simulation results delineate that, SCAC achieves high performance with respect to max throughput, access delay and blocking probability.

The rest of this paper is organized as follows. In section 2, we describe the network architecture. The design of SCAC and pre-defined soft guarantee justification are presented in Section 3. In section 4, we demonstrate experimental results for system performance. Finally, concluding remarks are given in Section 5.

2. Network and System Architectures

In Wireless LANs (WLAN), there are two different architectures, ad-hoc and infra-structure. The basic difference between them is having Access Point (AP) or not. Unlike ad-hoc architecture, infra-structure network contains an AP as a central controller to serve a finite set of Mobile Terminals (MTs) in the Basic Service Set (BSS), which is a region that a Mobile Terminal (MT) can receive physical signals broadcasted by AP. As shown in Figure 1, BSS is the dotted circle and a MT outside the circle just can not communicate with AP. In this paper, we only focus on the wireless infra-structure network.

IEEE 802.11e standard is the most common mechanism in the wireless infra-structure network, recently. In IEEE 802.11e, the basic access method is the Enhanced Distributed Coordination Function (EDCF) which is used to support asynchronous data and signaling request of synchronous data on a best effort basis. The contention free service for QoS traffic is provided by the Hybrid Coordinator (HC), which is a type of MTs that is typically co-located with the AP. HC provides two functions. One is Point Coordination Function (PCF) which basically implements a "polling" access method. This scheme lets MTs have priority access to the wireless medium. HC periodically polls MTs giving them the opportunity to transmit frames and thus avoiding any contention for the channel. Another function of HC is that it is able to start a Controlled Access Period (CAP) whenever required, even during an active EDCF period, in order to poll traffic with specific QoS requirement. As shown in Figure 2, the MSC layer protocol exerts PCF access for CBR/VBR/ABR traffic and the EDCF scheme for UBR and signaling traffic, over dynamically allocated contention-free and contention bandwidth, respectively.

In this paper, we design a new algorithm called SCAC based on IEEE 802.11e, which combines the advantage of MAC and CAC. In order to achieve our goal, we make several improvements from IEEE 802.11e, which are described below. (i) The superframe of SCAC is relaxed to variable length, which is suitable for AP to guarantee QoS requirement. (ii) In order to provide higher priority for signaling requests, we assign a new contention period called Super Contention Control Interval (SCCI) based on the CC strategy. Even more, SCAC employ ECCA to

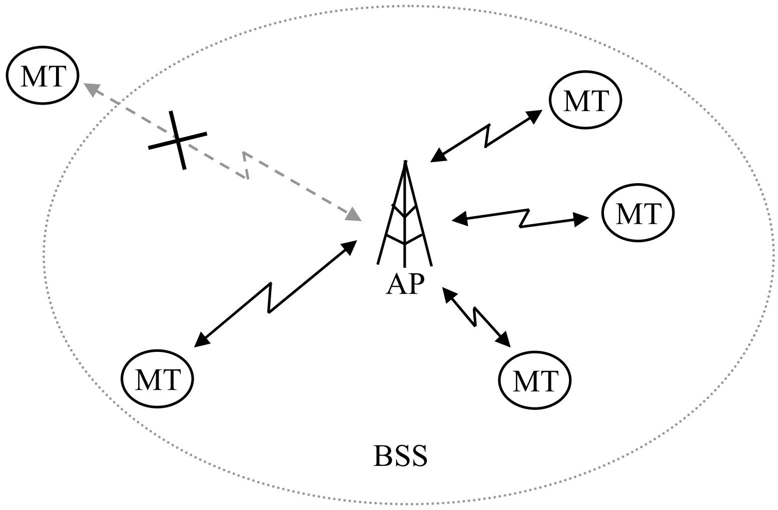
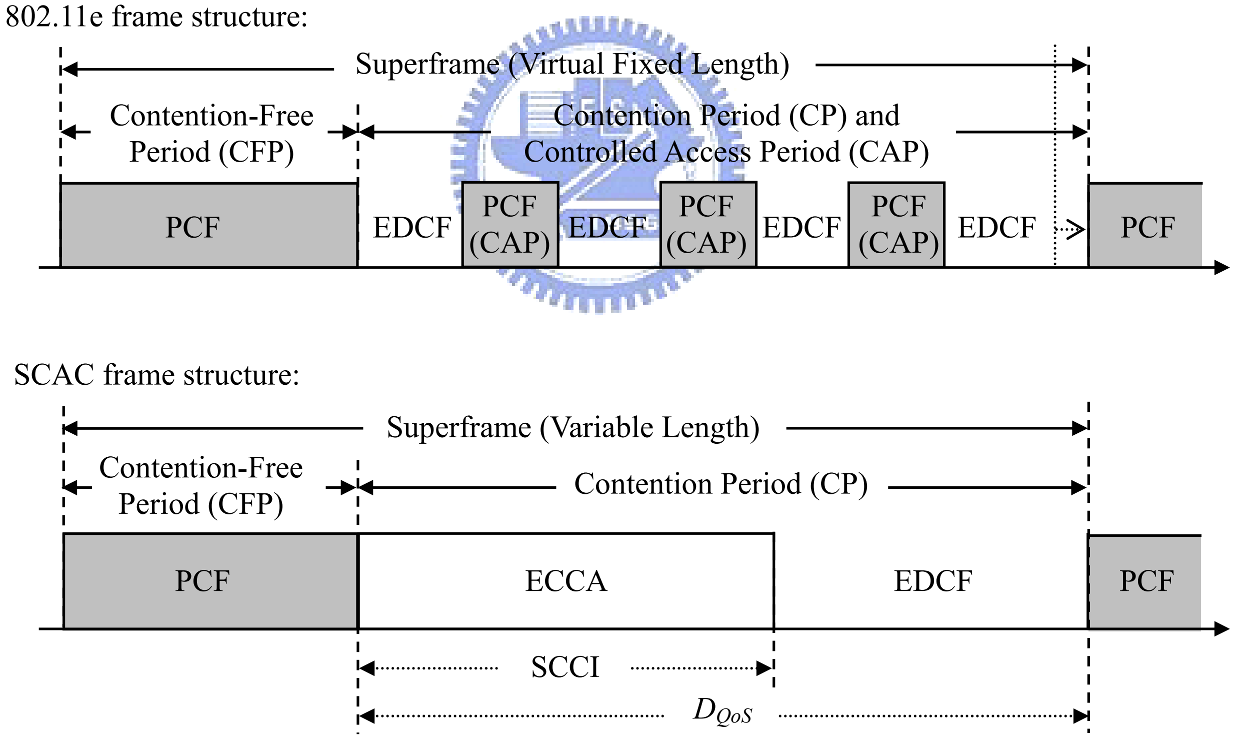


Figure 1. Wireless infra-structure network.

enhance the contending performance in SCCI. (iii) EDCF is the lowest priority period in SCAC, i.e. if ECCA running time is smaller than the upper limit period (D_{QoS}), which is computed by Q-CAC based on the original QoS guarantee, EDCF are operated during the rest time (D_{QoS} – SCCI).

3. Soft-Guarantee-based Call Admission Control (SCAC)

Soft-Guarantee-based Call Admission Control (SCAC) includes three major parts: QoS-enabled Call Admission Control algorithm (Q-CAC algorithm), Efficient Contention Control algorithm (ECCA algorithm), Resource Management Database (RMDB), and a simple 7-feedback control facilitated by a Pdf-based Multi-user Estimator (PMER) at the physical layer, which can detect the number of MTs contending in the same slot [10], as shown in Figure 3. By cooperating the three parts we can provide soft guarantee and high efficiency with low computational complexity. The detail of the three parts will be described as follow.


802.11e frame structure:
Superframe (Virtual Fixed Length)
Contention-Free Period (CFP)
Contention Period (CP) and Controlled Access Period (CAP)
PCF
EDCF
PCF (CAP)
EDCF
PCF (CAP)
EDCF
PCF (CAP)
EDCF
PCF
SCAC frame structure:
Superframe (Variable Length)
Contention-Free Period (CFP)
Contention Period (CP)
PCF
ECCA
EDCF
PCF
SCCI
D_{QoS}

Legend:

SCCI: Contention Period used for signaling request contending the channel;
ECCA: Efficient Contention Control Algorithm;
D_{QoS}: Upper limit period without breaking the original QoS guarantee;

Figure 2. 802.11e and SCAC frame structure.

3.1. QoS-enabled Call Admission Control algorithm (Q-CAC)

Q-CAC is a call admission control strategy. The major job of Q-CAC is to limit the number of contending MTs with voice signaling requests into AP's polling table. Q-CAC has three steps: First, it applies some traffic prediction methods to predict the total number of MTs with voice signaling requests ($\hat{N}$) at the beginning of SCAC. Second, using some easy constrains, such as delay, loss etc. to compute the optimal SCCI length (D_{QoS}), which must obey the original QoS guarantee that living connections need in AP's polling table. For example, if the voice traffic types are different, we can choose the minimal voice periodic transmission time around them for the optimal SCCI length. Finally, taking $\hat{N}$ and D_{QoS} as inputs, Q-CAC searches the RMDB for a admitted probability (P_{adm}) and the system parameter called optimal first Contention Control Interval length ($CCI(1)_{opt}$) which are used for ECCA algorithm and then starts ECCA operation.

3.2. Efficient Contention Control algorithm (ECCA)

ECCA is a MAC mechanism, which is used for MTs with voice signaling requests contending the shared channel. The basic idea of ECCA is that: Instead of giving all the slots like CC method, ECCA gives slots piece bye piece, which each piece can viewed as a Contention Control Interval (called $CCI(\text{k})_{opt}$, k is tagged as the kth piece) and we call the total $CCI(k)_{opt}$ as SCCI. We use a random variable $\tilde{D}$ to describe the distribution of SCCI (Figure 4). After AP runs Q-CAC and searches RMDB to get P_{adm} and $CCI(1)_{opt}$, we can decide the first piece of slots(i.e. $CCI(1)_{opt}$). Following, ECCA algorithm will decide $CCI(2)_{opt}$, $CCI(3)_{opt}$, … $CCI(k)_{opt}$ based on the contending condition by itself. The details of ECCA algorithm are described as follow.

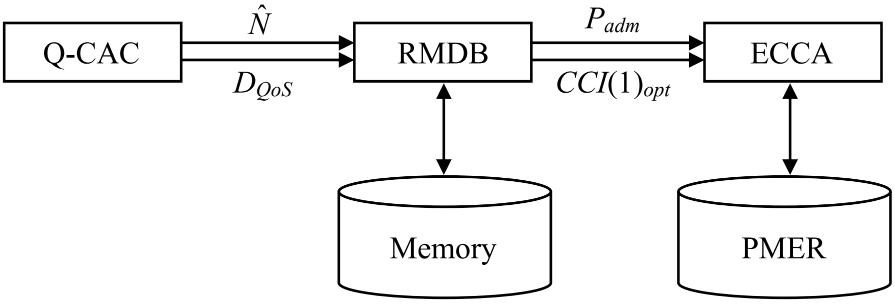


Figure 3. SCAC structure.

ECCA contains two phases (Figure 5). At the beginning of ECCA, AP awakes all MTs with signaling requests and enter ECCA two phases operation. In Phase 1, AP broadcasts P_{adm} and $CCI(1)_{opt}$ using Broadcast Packet (BP) to all MTs, which join ECCA two phases operation. These MTs desiring to contend the channel use P_{adm} as the dice probability for joining ECCA algorithm to decide whether continue or not. That is a MT wishing contending the channel run a uniform distribution from 0 to 1, and check the value. If the value smaller or equal to P_{adm}, the MT wins the diet and continue following operations. If a MT, which wins the dice continues to run ECCA operations, it will choose a slot from 0 to $CCI(1)_{opt}-1$ and jam in that slot (i.e. making reservation). And other MTs losing the dice will keep sleeping until the end of ECCA operation. Obviously, if more than one MT jams the same slot, the collision occurs. Oppositely, the MT succeeds the reservation without other MTs jamming the same slot with it. In this system, we use a hardware PMER [10] to detect the collision number of MTs in a slot. Based on the mechanism, we can detect up to five MTs colliding in the same slot and get higher channel utilization by using this information (i.e. throughput). So, in this paper, we assume that we can detect seven conditions (0 to 6, using 6 to describe up than 5). Due to the simulation observation [10], 7-Feedback mechanism is good enough comparing with Full-Feedback (Figure 6, which the throughput means success number of MTs in contending the shared channel divide the total contention time i.e. success contending rate). As shown in Figure 6, in each slot form slot number 0 to slot number $CCI(1)_{opt}-1$, AP will receive a feedback from PMER (b_i) and broadcast them ($b_0, b_1, ..., b_{CCI(1)-1}$) to MTs at the end of Phase 1 using BP. If there are any collision occurred in b_i, the system will enter Phase 2. And the colliding MTs in Phase 1 will continue the ECCA operation.

In order to achieve the soft QoS guarantee, we need a special protection at the end of Phase 1 for controlling the number of MTs that join ECCA operation besides using P_{adm} (Because running a

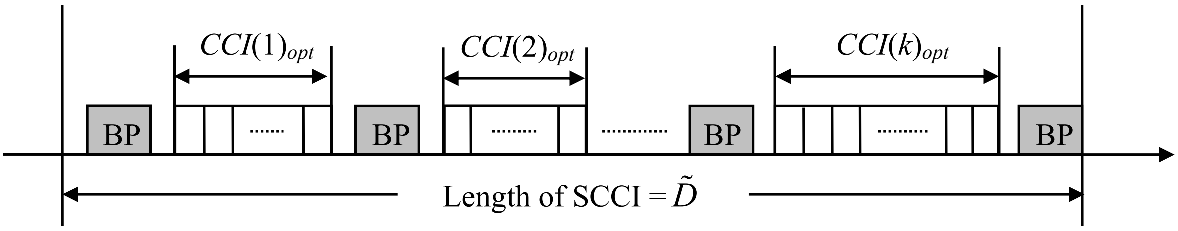


Figure 4. The SCCI diagram.

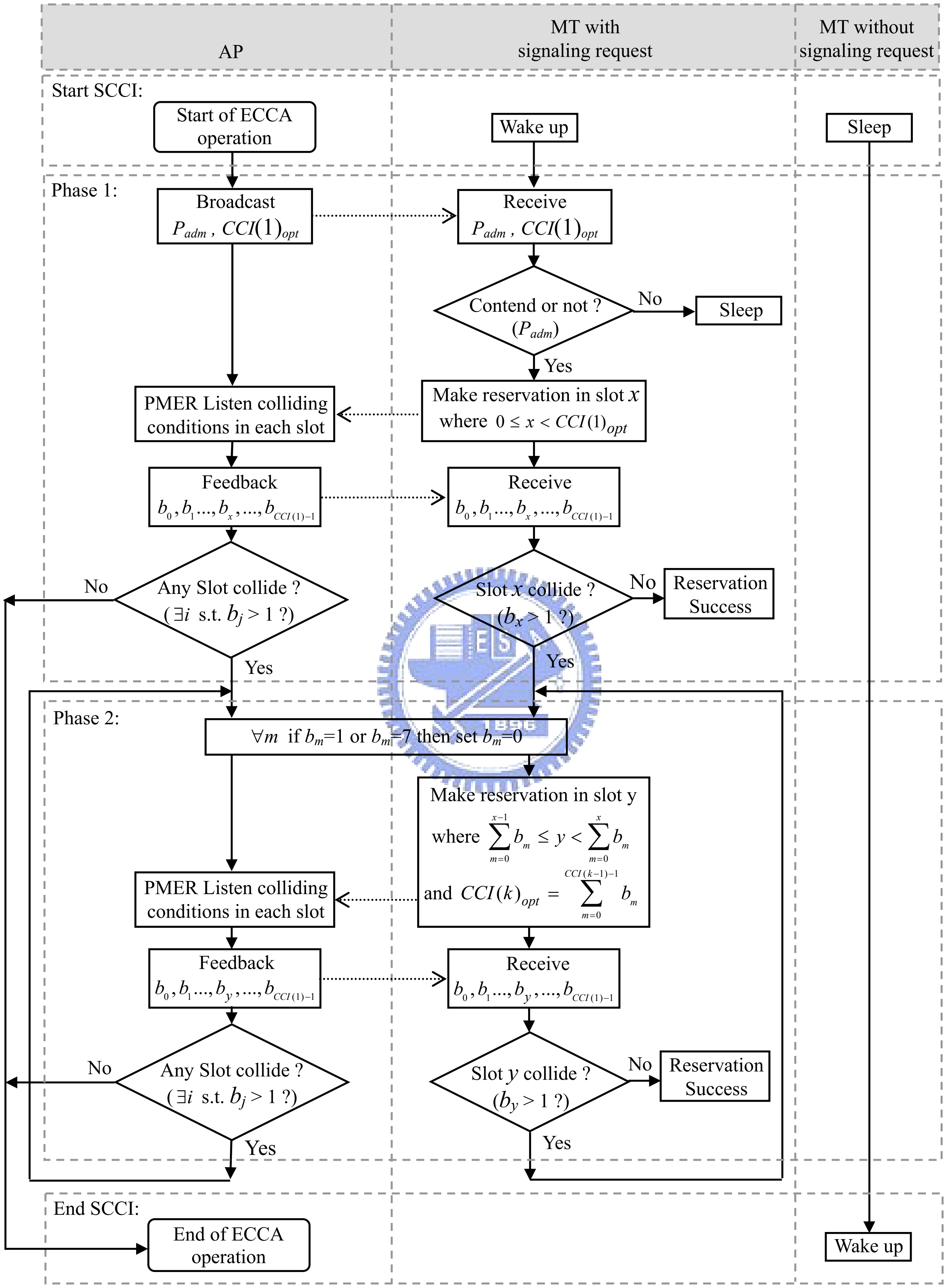

AP
MT with signaling request
MT without signaling request
Start SCCI:
Start of ECCA operation
Wake up
Sleep
Phase 1:
Broadcast P_{adm}, $CCI(1)_{opt}$
Receive P_{adm}, $CCI(1)_{opt}$
Contend or not ? (P_{adm})
No
Sleep
Yes
PMER Listen colliding conditions in each slot
Make reservation in slot x where $0 \leq x < CCI(1)_{opt}$
Feedback $b_0, b_1 ..., b_x, ..., b_{CCI(1)-1}$
Receive $b_0, b_1 ..., b_x, ..., b_{CCI(1)-1}$
No
Any Slot collide ? ($\exists i$ s.t. $b_j > 1$?)
Yes
Slot x collide ? ($b_x > 1$?)
No
Reservation Success
Yes
Phase 2:
$\forall m$ if b_m=1 or b_m=7 then set b_m=0
Make reservation in slot y
where $\sum_{m=0}^{x-1} b_m \leq y < \sum_{m=0}^{x} b_m$
and $CCI(k)_{opt} = \sum_{m=0}^{CCI(k-1)-1} b_m$
PMER Listen colliding conditions in each slot
Feedback $b_0, b_1 ..., b_y, ..., b_{CCI(1)-1}$
Receive $b_0, b_1 ..., b_y, ..., b_{CCI(1)-1}$
No
Any Slot collide ? ($\exists i$ s.t. $b_j > 1$?)
Yes
Slot y collide ? ($b_y > 1$?)
No
Reservation Success
Yes
End SCCI:
End of ECCA operation
Wake up

Figure 5. ECCA algorithm.

dice is not enough to limit the contending MTs whit signaling requests.). Because more MTs than N_{adm} that join ECCA operation will heavily break the soft QoS guarantee, which we want to support, we have to provide a second line of defense on joining number of MTs. Because AP can roughly estimate the number of MTs that join ECCA operation by summing all b_i values, AP will assign a special feedback value 7 (it can be viewed as a virtual collision) from b_m to $b_{CCI(1)-1}$ under the condition that $\sum_{j=0}^{m-1} b_j \le N_{adm}$ and $\sum_{j=0}^{m} b_j > N_{adm}$. Based on this protection, we just can limit the joining number of MTs advancedly. As a conclusion, if a MT jams the xth slot in Phase 1, it can decide whether colliding in Phase 1 or not itself by checking the value of b_x. If b_x is equal to 1, the MT succeeds the contention and needs not to join Phase 2. Otherwise, if b_x is larger than 1 and smaller than 7, the MT collides with other MTs and need to join Phase 2 for the next contention opportunity. Unfortunately, If b_x is 7, the MT will lose the opportunity on joining Phase 2 operation (virtual collision) because of the exceeding number of MTs that join Phase 1 operation.

In Phase 2, the operation is similar to Phase 1 but much simpler. First of all, AP and MTs will filter out the success slots and virtual collision slots by set them to 0 (i.e. if b_m = 1 or 7 then set b_m = 0),

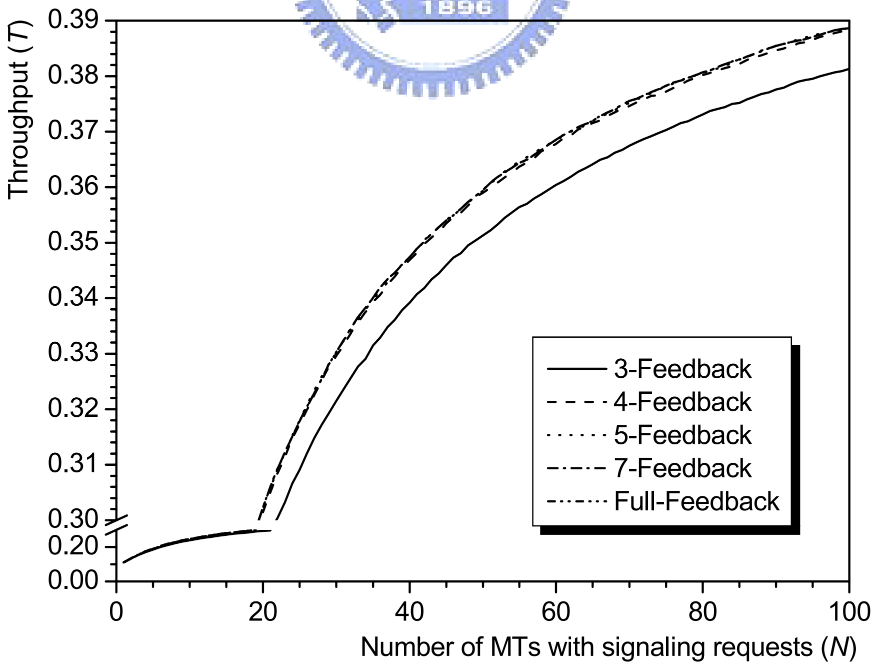


Figure 6. The performance impact from Feedback.

because the MTs jamming in these slots will not join the following operation. When AP broadcasts $b_0, b_1, ..., b_{CCI(1)-1}$ in Phase 1, it not only feedback the collision information, but also tells all MTs that it will release $\sum_{j=0,7>b_j>1}^{CCI(1)-1} b_j$ slots to MTs, which still operate in Phase 2 (Remember that if the collision number larger than 5, we use 6 to describe it). This means that we release *i* slots in Phase 2 for a group of MTs, which colliding at the same slot in Phase 1 and the collision number equal to *i*. If *i* is equal to 6, we just give 6 slots for the group of colliding MTs even through the group may be larger than 6. We let $CCI(2)_{opt} = \sum_{j=0,7>b_j>1}^{CCI(1)-1} b_j$ and the MT jams and colliding in *x*th slot of Phase 1 with $1 < b_x < 7$ will randomly choose a slot in the subset slots with total slot number b_x (i.e. b_y in Figure 5). On the other words, if we assign indexes to each slot from index 0, the MT just can choose one slot from number $\sum_{j=0,7>b_j>1}^{x-1} b_j + 1$ to number $\sum_{j=0,7>b_j>1}^{x} b_j$ randomly. If any collisions still occur in Phase 2, ECCA will repeat Phase 2 operation until no collisions occur and we let $CCI(k)_{opt} = \sum_{j=0,7>b_j>1}^{CCI(k-1)-1} b_j$.

We will see an example in Figure 7. At the beginning of ECCA, AP runs Q-CAC algorithm to get two parameters P_{adm} and $CCI(1)_{opt}$, which P_{adm} is equal to 2/3 and $CCI(1)_{opt}$ is equal to 4. Fortunately, we assume that there are exactly 6 MTs win the diet, which equal to N_{adm}. Then the operation of ECCA is depicted in Figure 4 and is the same as discussed before.

3.3. ***Resource Management Database (RMDB)***

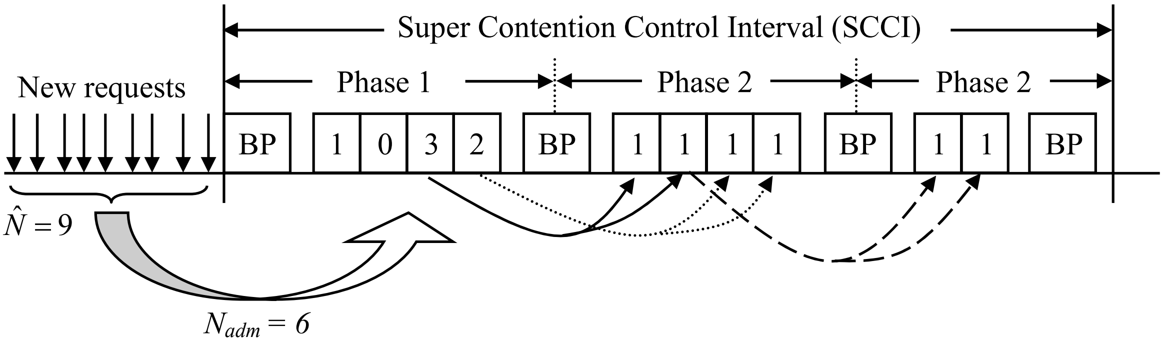


Figure 7. A SCAC running example.

Resource Management Database (RMDB) is a pre-constructed database used for Q-CAC. It can be viewed as a function, which takes $\hat{N}$ and D_{QoS} as inputs and returns P_{adm} and $CCI(1)_{opt}$ as outputs. Following, we describe how to construct the RMDB and how to compute these outputs separately.

3.3.1. Construct the RMDB

The data structure of RMDB is a table with each entry is a triple parameter. The structure of the entry is (N, $CCI(1)$, *Đ*) which N is the number of MTs with voice signaling requests, $CCI(1)$ is the system parameter, first Contention Control Interval length ($CCI(1)_{opt}$ is the optimal $CCI(1)$ used in Phase 1 of ECCA), used for Phase 1 in ECCA with respect to a specific N, *Đ* is the cumulative density function (CDF) for SCCI related to N and $CCI(1)$. In order to finish the table, we construct each $CCI(1)$ for each related N first and construct each *Đ* based on related N and $CCI(1)$. The detail of these works is described as follow:

Step 1: Construct $CCI(1)$

As mentioned before, $CCI(1)$ is the first Contention Control Interval length. Our strategy for assigning each $CCI(1)$ value into each related N is that: For a given N, we want to find the best $CCI(1)$ value related to N for achieving minimal E[*Đ*] (i.e. let the expected operation time of ECCA as short as possible, because SCCI is the running time of ECCA). When constructing all $CCI(1)$ values, we simpler ECCA operation as follow: (i) Each MT with signaling request join ECCA operation without run a diet (P_{adm}). (ii) AP will not feedback value 7 to any b_i. These two simplifications are needed, because we should not turn off the MT's contending opportunity for constructing $CCI(1)$ values. On the other word, call admission control strategy is not needed in order to find out $CCI(1)$ values without guaranteeing QoS. (iii) And also, we assume that the length of BP is zero in SCCI for simplicity, because the effect is little. For finding all $CCI(1)$ values, we define random variables and write a set of recursive equations as follow:

Definition:

Đ$_{N,j}$ = the distribution of SCCI under N MTs joining and j slots in $CCI(1)$ (i.e. ECCA operation time)

$D_{N,j}$ = $E[$*Đ*$_{N,j}]$

$\tilde{D}_{N,j|m}$ = the distribution of SCCI under N MTs joining and j slots in $CCI(1)$ with the condition that there are m MTs choose the first slot of $CCI(1)$

$D_{N,j|m}$ = $E[\tilde{D}_{N,j|m}]$

$Q_{N,j}(m)$ = the probability under the condition that N MTs join in ECCA, there are j slots in $CCI(1)$ and m MTs choose the first slot of $CCI(1)$

Equations:

$$Q_{N,j}(m) = \binom{N}{m}(\frac{1}{j})^m(\frac{j-1}{j})^{N-m} \tag{1}$$

$$D_{N,j|m} = D_{m,1} + D_{N-m,j-1} \tag{2}$$

$$D_{N,j} = \sum_{m=0}^{N} Q_{N,j}(m) * D_{N,j|m} = \sum_{m=0}^{N} Q_{N,j}(m) * [D_{m,1} + D_{N-m,j-1}] \text{ for } N \geq 2,\ j \geq 2 \tag{3}$$

$$D_{N,1} = 1 + D_{N,N} \text{ for } 6 \geq N \geq 2 \tag{4}$$

$$D_{N,1} = 1 + D_{N,6} \text{ for } N > 6 \tag{5}$$

$$D_{0,j} = D_{1,j} = j \tag{6}$$

Equation (1) is a simple permutation-constitution problem, which choosing m MTs from tatal N MTs into the first slot of total j slots. And the rest N-m MTs join any slot of the following j-1 slots. In equation (2), $D_{N,,j|m}$ can be viewed as a parallel operating process for $D_{m,1}$ and $D_{N\text{-}m,j\text{-}1}$. $D_{N,,j|m}$ proceeds each step just like summing each step of $D_{m,1}$ and $D_{N\text{-}m,j\text{-}1}$. As shown in Figure 8, $D_{10,4|3}$ can be viewed as the summation of $D_{3,1}$ and $D_{7,3}$. Equation (3) is the main body of the recursive set of equations. First, we calculate $D_{N,j}$ by condition on its colliding number of MTs of the first slot in $CCI(1)$ i.e. condition on the number of MTs, which choose the first slot in $CCI(1)$ randomly. Then, we substitute $D_{N,,j|m}$ into $D_{m,1} + D_{N-m,j-1}$ using equation (2). Equation (4) and (5) are obvious, which

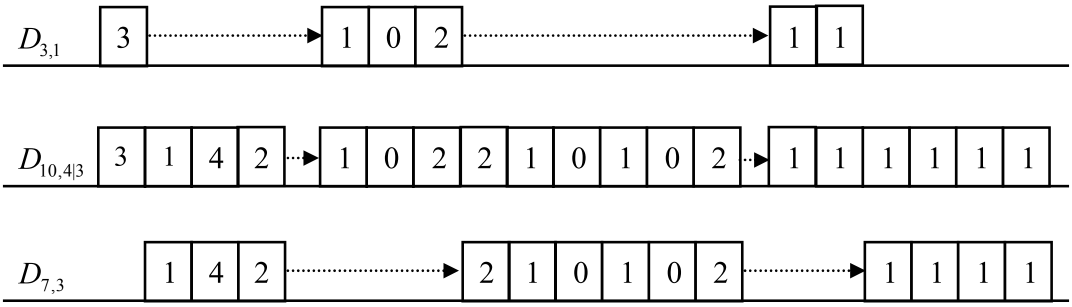


Figure 8. A $D_{N,j|m}$ computing example.

based on MAC design described before. In equation (4), 1 is $CCI(1)$ of $D_{N,j}$ in Phase 1 and the following operating time of $D_{N,j}$ is $D_{N,N}$ because PMER will return N under $2 \le N \le 6$. And equation (5) is similar to equation (4), but PMER just can return 6 for the colliding number of MTs larger than 6. Equation (6) obviously is the basis of the set of recursive equations.

Based on the set of equations, we compute each $D_{N,j}$ by using all previous $D_{N,j}$ values which have been computed recursively. For example, $D_{2,2}$ can be constructed using $D_{0,1}$, $D_{1,1}$ and $D_{2,1}$ based on equations (3) to (6). Until finishing all $D_{N,j}$ values, we define the channel contending throughput (T) as $\frac{N}{D_{N,j}}$, that is for a given N, if we use j as $CCI(1)$ value in ECCA, we will get the throughput T and this will become a index for determining $CCI(1)$ with respect to a specific N. Finally, for a choosen N, we take the value i as its related $CCI(1)$ value, which $D_{N,i}$ is the minimal value around all $D_{N,j}$ for all j (i.e. $\frac{N}{D_{N,i}}$ is maximal). For example, as shown in Figure 9, $D_{25,22}$ is the minimal value around all $D_{25,j}$ with respect to $N = 25$ (In other word, $\frac{25}{D_{25,22}}$ is maximal). And then, we take 22 as the $CCI(1)$ value

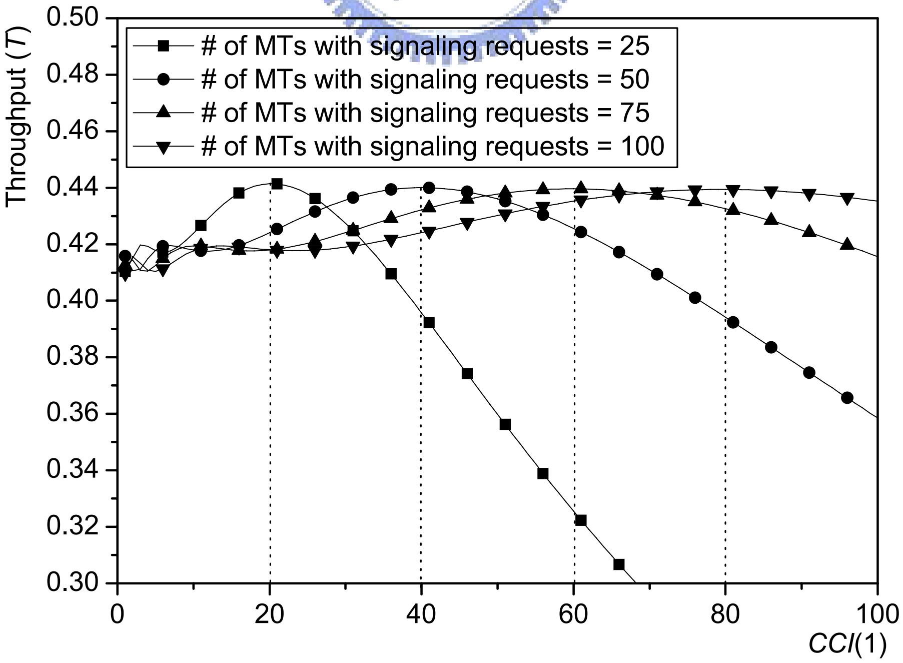


Figure 9. The relationship between $CCI(1)$ and Throughput.

related to $N = 25$. Up to now, we find out all $CCI(1)$ values for their related N.

The computation complexity for finishing the construction is O(N^3), which N is the number of MTs with voice signaling requests. Obviously, when N is getting larger, the computation time will become much longer. In order to shorten the computation time, we simplify our method for constructing all $CCI(1)$ values. First, we compute the $CCI(1)$ value for its related N from $N = 1$ to 100 using the construction mentioned before (see Figure 10). Then, we use a linear curve to fit the relation between N and $CCI(1)$. We detect that $CCI(1)$ is almost $N * 0.8$. So, for N larger than 100, instead of using algorithm to find the value, we use the value $N * 0.8$ for the related $CCI(1)$.

Step 2: Construct $\tilde{D}$

We use a neural fuzzy method [11] for constructing each $\tilde{D}$ for its related N and $CCI(1)$. Because the neural fuzzy method needs a little datum for behavior learning, we first simulate those $\tilde{D}$ related to N from $N = 1$ to 100 using computer procedure for CDF curves (simulation time is 10^7 for each $\tilde{D}$ to construct its distribution). Then, feeding these datum into the neural fuzzy function, we can construct each $\tilde{D}$, which its related N is larger than 100. Up to now, we finish RMDB

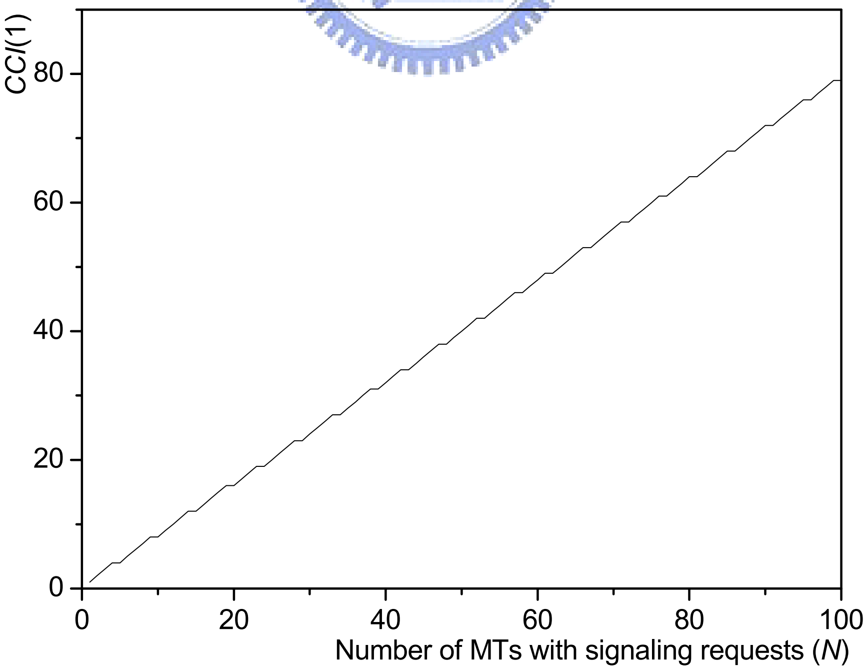


Figure 10. The analytic result of $CCI(1)_{opt}$ for different number of MTs.

construction.

3.3.2. Compute the outputs

In this subsection, we describe how to compute P_{adm} and $CCI(1)_{opt}$. The computation takes $\hat{N}$ and D_{QoS} as inputs and returns P_{adm} and $CCI(1)_{opt}$ as outputs. We separate the computation into three steps. First, based on the pre-defined degree of the system QoS guarantee (For example, 90% QoS guarantee mean that ECCA operation time has 90% chance not cross D_{QoS}), we find out all the 90% values d (i.e. $P[\tilde{D} \le d] \ge 90\%$) from searching each $\tilde{D}$ related to N which obeys the QoS guarantee. As shown in Figure 11 with 90% QoS guarantee, we can get $d = 30$ with $N = 10$, $d = 56$ with $N = 20$, $d = 80$ with $N = 30$, $d = 105$ with $N = 40$, and $d = 130$ with $N = 50$. Second, compare D_{QoS} with all d to find the maximal admitted N (N_{adm}), which its d is shorter than D_{QoS}. Based on the same example (Figure 11), we can fine that $N_{adm} = 30$ based on $D_{QoS} = 100$. The meaning is that the system, which invites 30 MTs in ECCA, will have 90% chance not cross D_{QoS} i.e. we can provide 90% QoS guarantee. Finally, compare $\hat{N}$ with N_{adm} to compute P_{adm} and $CCI(1)_{opt}$, which has two

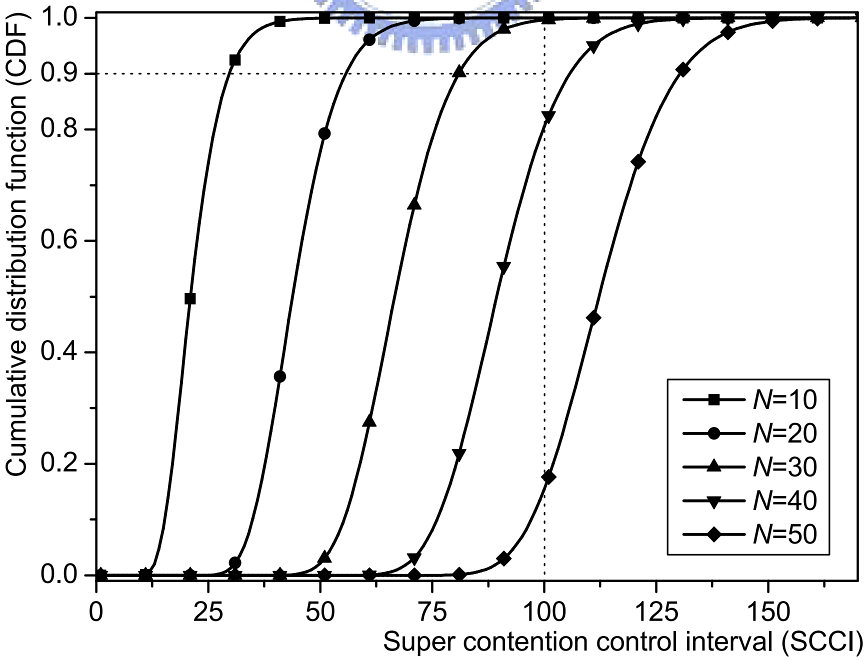
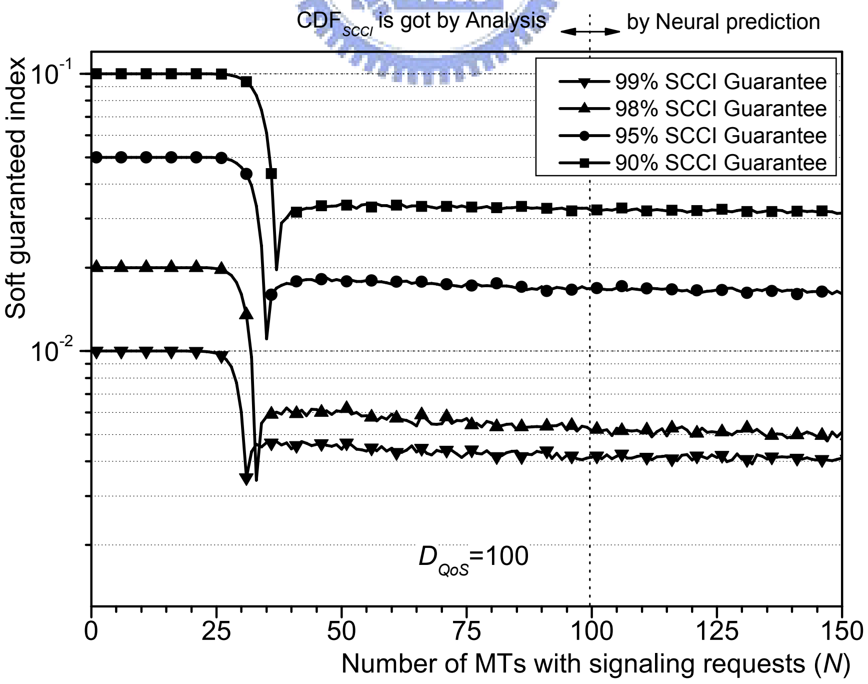


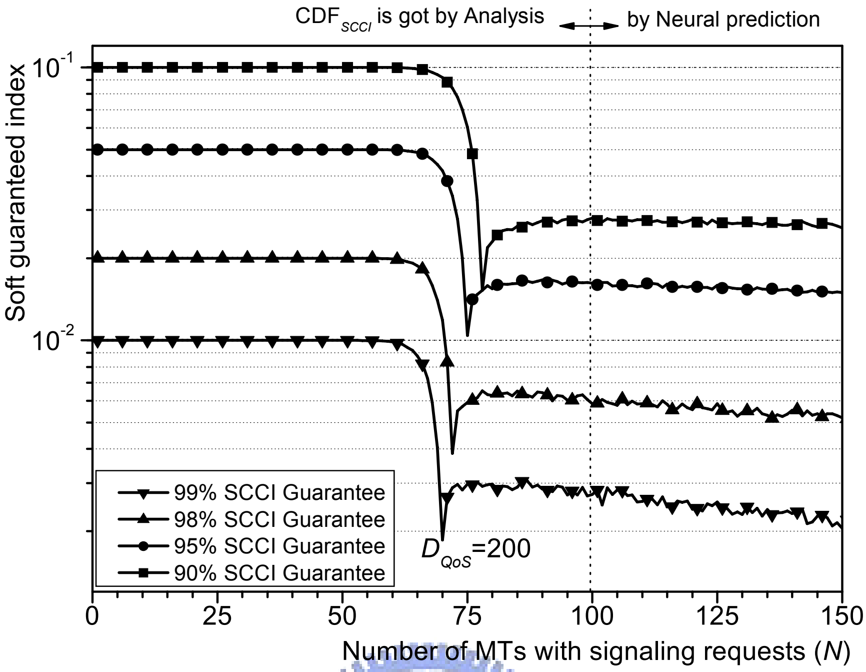
Figure 11. An example of finding P_{adm}.

conditions: (i) If $\hat{N}$ is larger than N_{adm}, we take P_{adm} into $\frac{N_{adm}}{\hat{N}}$, and takes $CCI(1)_{opt}$ as the $CCI(1)$ related to N_{adm}. (ii) If $\hat{N}$ is smaller than (or equal to) N_{adm}, we take P_{adm} into 1, and takes $CCI(1)_{opt}$ as the $CCI(1)$ related to $\hat{N}$.

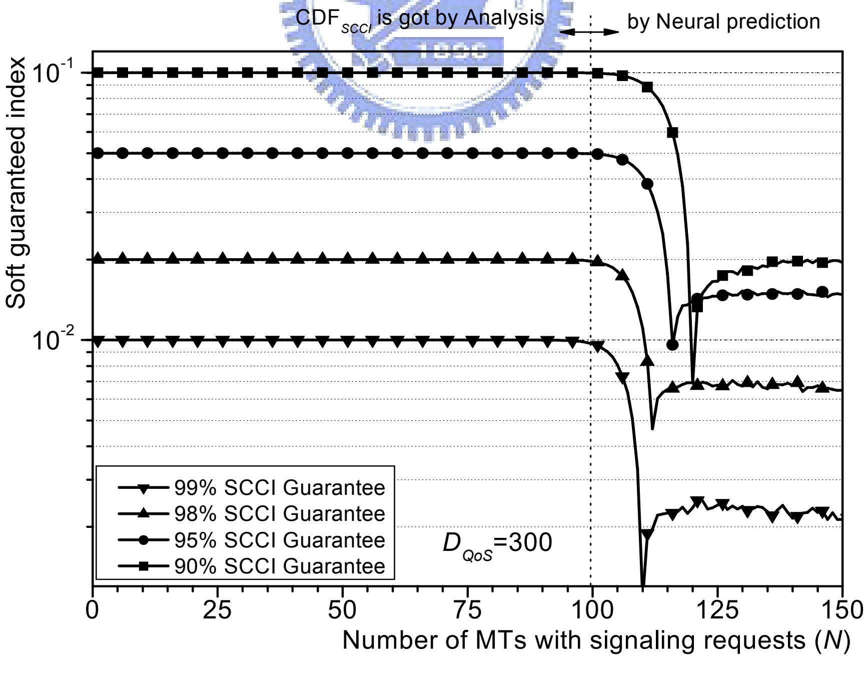
In sum, combining Q-CAC, ECCA, and RMDB, SCAC can support pre-defined soft QoS guarantee and provide better contention resolution for high priority traffic (ex: voice signaling requests). Through simulation (see Figure 12) using different QoS (90%, 95%, 98%, and 99%) degree and D_{QoS} (100, 200, and 300), we can prove that our new method can guarantee the pre-defined QoS degree. In Figure 12, the x-axis is the number of MTs with signaling requests and the y-axis is the QoS guarantee percentage minus the pre-defined QoS guarantee degree. So, these positive curves prove that SCAC achieve our goal.



(a) D_{QoS} = 100



(b) $D_{QoS} = 200$



(c) $D_{QoS} = 300$

Figure 12. SCCI guaranteed.

4. Experimental Results

We measure the SCAC mechanism on real-time traffic through simulation. We consider a simulated environment with an independent BSS. Two traffic models are considered and the traffic parameters are summarized in Table I.

CBR Voice Traffic: The voice traffic is modeled as a two-state Markov process with talk-spurt and silence states. The duration of these two states is assumed to be exponentially distributed with parameters 1s and 1.35s, respectively.

ABR Data Traffic: There are 20 MTs to generate asynchronous data traffic at a mean aggregate rate of 5 Mbps.

The performance measurements considered in our simulations are defined as follows:

Table I. Simulation Parameters

Parameter		Symbol	Value	
System	Transmission Rate	R	100	kbps
	The SIFS interval	S-IFS	80	µs
	The PISF interval	D-IFS	160	µs
	The DIFS interval	P-IFS	240	µs
	Maximum duration of the superframe	T_{SF}	48	ms
CBR Voice Traffic	Mean ON Period		1	s
	Mean OFF Period		1.35	s
	Mean Total Connection Time		120	s
	Mean Voice Request Rate		0.001	request/sec
	Voice Coding Rate		8.5	kb/s
	Voice Traffic Priority		High	
	Voice-IFS	A-IFS$_{Voice}$	240	µs
ABR Data Traffic	Mean Data Packet Arrival Rate		0.01	Packet/sec
	Mean Packet Size		2	kb/s
	Data Traffic Priority		Low	
	Data-IFS	A-IFS$_{Data}$	300	µs

Voice request delay: The slot time duration for a voice request from entering the local queue to the beginning of successful transmission.

Voice request blocking probability: The fraction of discarded requests caused by violating the delay bound.

We draw comparisons of performance with respect to voice request access delay, and blocking probability, between ECCA, EDCF and DCF. Simulation was terminated after reaching 95% confidence interval. Simulation results are depicted in Figures 13-16.

To show the fact that the usage of ECCA can increase the system performance, we perform the experiment with 20 CRB voice flows and increasing number of ABR data flows. Figure 13 shows that more ABR data flows can be accommodated in the system, given an acceptable packet drop delay bound (300 slot-times), when EDCF and ECCA are used separately. We further make comparisons of voice request delay among DEF, EDCF and ECCA. In Figure 14, we clearly observe that, DCF scheme has higher request delay than EDCF and ECCA. The reason is that the number high priority traffic (CBR voice traffic) is fixed and EDCF and ECCA can support the real-time request.

In Figure 15, we show the CBR voice request delay with the increasing number of real-time flows. The experiment is performed with 20 ABR data flows and increasing number of CRB voice flows. If the flows have been served and there is residual time in the CFP. In the simulation, when voice flow increasing, DCF and EDCF have the same high blocking probability. And, ECCA got better performance. Moreover, the voice request delay comparison showed in Figure 16 is the same result. From the simulation, we clearly observe that, ECCA significantly outperforms under the same priority traffic load increasing.

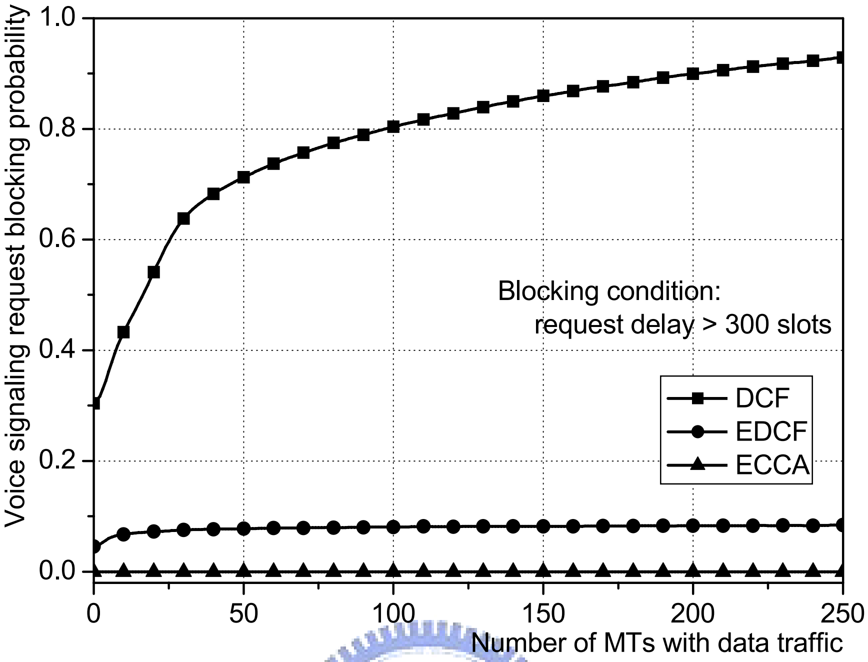


Figure 13. Comparison of voice request blocking probability by different methods.

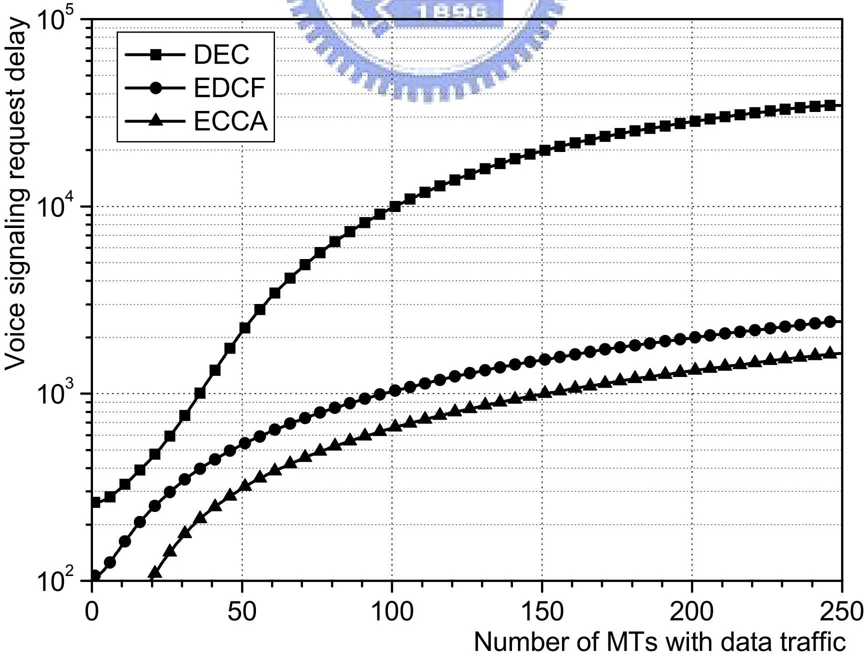


Figure 14. Comparison of voice request delay by different methods.

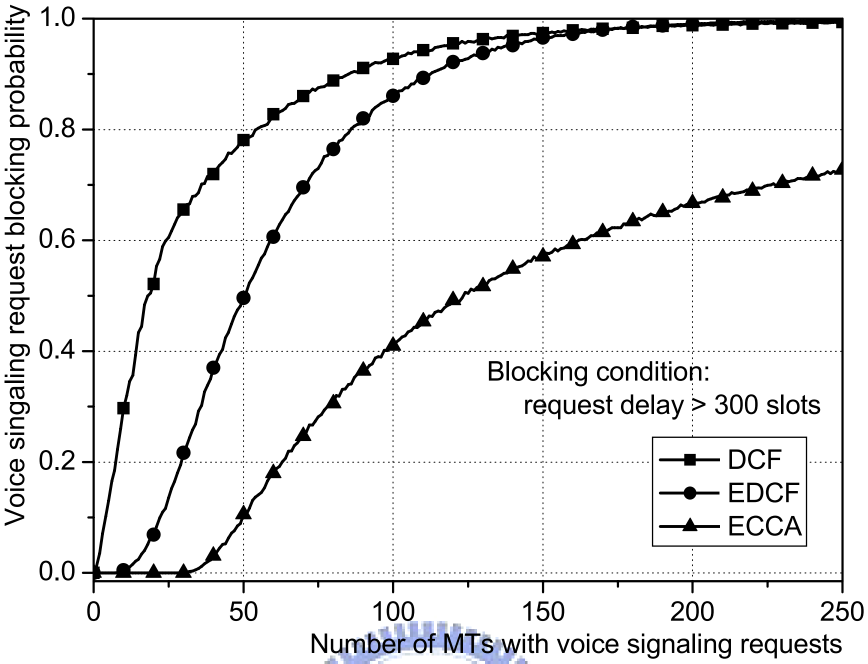


Figure 15. Comparison of voice request blocking probability by different methods.

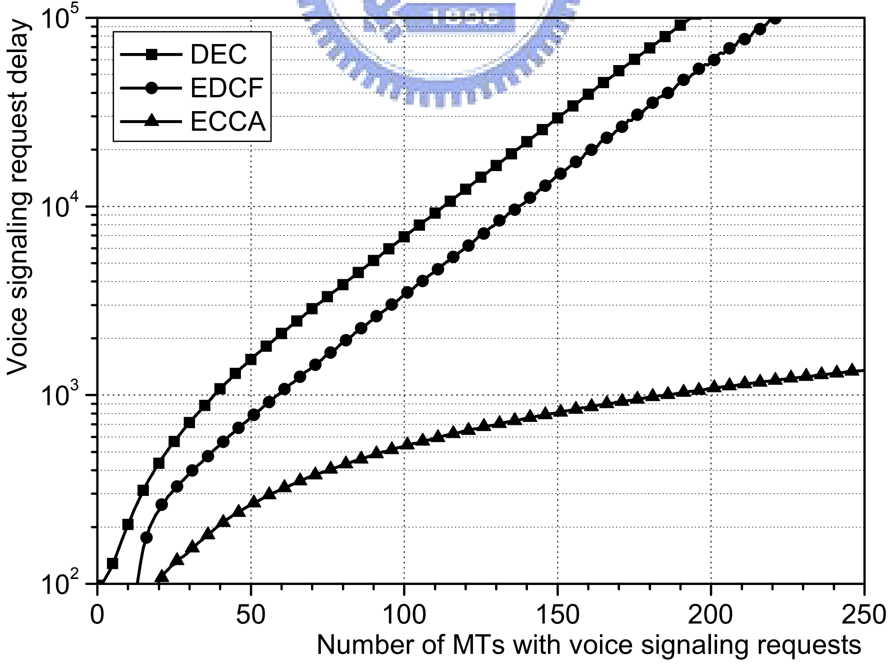


Figure 16. Comparison of voice request delay without blocked control.

5. Conclusions

In this paper, we have proposed an efficient soft-guarantee-based CAC (SCAC), which combined Q-CAC as a limiter for restricting the number of contending signaling requests, a new MAC algorithm (ECCA) for supporting higher guarantee and better success contending performance to voice signaling traffic, and a pre-constructed database (RMDB) used search system information for Q-CAC. SCAC runs Q-CAC to estimate the MTs with signaling requests ($\hat{N}$) find out the optimal QoS guaranteed duration (D_{QoS}) based on current system environments. Through searching RMDB, we can determine the admitted number of MTs (N_{adm}) to compute the admitted parameter ($P_{adm} = \frac{N_{adm}}{\hat{N}}$) and $CCI(1)_{opt}$, which is a important parameter used for ECCA. And then, SCAC enter ECCA operation. In ECCA, each MT wishing to contend the channel plays a dice (P_{adm}) to decide join ECCA two phases operation or not. The MTs, which win the dice, follow the two phase rule of ECCA to join AP's polling table. Through simulation result, we find out that, SCAC can achieve pre-defined soft QoS guarantee for living MTs in AP's polling table. And even more, when comparing with IEEE 802.11e, we also can observe that ECCA, which is a MAC mechanism in SCAC, provide better QoS guarantee and higher channel utilization for voice signaling requests.

References

[1] IEEE 802.11 WG, “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” *IEEE 802.11 standard*,1997.

[2] Lindgren A, Almquist A, and Schelen O. Evaluation, “Evaluation of quality of service schemes for IEEE 802.11 wireless LANs,” *Procedings of LCN 2001*, Tampa, Florida, USA, Nobember 15-16, 2001.

[3] Y. Kwon, Y. Fan, and H. Latchman, “Design of MAC Protocols With Fast Collision Resolution for Wireless Local Area Networks,” *IEEE Tran. Wireless Commu.*, vol. 3, no. 3, pp. 793-807, May 2004.

[4] G. Daqing, and Z. Jinyun, “QoS Enhancement in IEEE 802.11 Wireless Local Area Networks,” *IEEE Commun. Mag.,* vol. 41, no. 6, pp. 120-124, June 2003.

[5] IEEE Draft Std 802.11e, Medium Access Control (MAC) Enhancements for Quality of Service (QoS), D5.2, November 2003.

[6] S. S. Tsong, and S. T. Fang, “A Bandwidth Allocation/Sharing/Extension Protocol for Multimedia Wver IEEE 802.11 Ad Hoc Wireless LANs,” *IEEE J. Select. Areas Commun.*, vol. 19, no. 10, pp. 2065-2080, Oct. 2001.

[7] IEEE Draft Std 802.11e, Medium Access Control (MAC) Enhancements for Quality of Service (QoS), D2.0a, November 2001.

[8] M. Veeraraghavan, N. Cocker, and T. Moors, “Support of voice services in IEEE 802.11 wireless LANs,” *IEEE INFOCOM*, vol. 1, pp. 488-497, April 2001.

[9] A. Veres, A. T. Campbell, and M. Barry, “Supporting Service Differentiation in Wireless Packet Networks Using Distributed Control,” *IEEE J. Select. Areas Commun.*, vol. 19, no. 10, pp. 2081-2093, Oct. 2001.

[10] M. C. Yuang, B. C. lo, and J. Y. Chen, “Hexanary Feedback Contention Access with Pdf-based Multi-User Estimation for Wireless Access Networks,” *IEEE Tran. Wireless Commu.*, vol. 3, pp. 278-289, Jan. 2004.

[11] M. Yuang, and P. Tien, “Multiple Access Control with Intelligent Bandwidth Allocation for Wireless ATM Networks,” *IEEE JSAC*, vol. 18, no. 9, Sept. 2000, pp. 1658-1669.

[12] S. Valaee, and B. Li, “Distributed Call Admission Control for Ad Hoc Networks,” *Procedings of*

VTC 2002, vol. 2, pp. 1244-1248, Sept. 2002.

[13] D. Pong, and T. Moors, "Call Admission Control for IEEE 802.11 Contention Access Mechanism,"*Procedings of GTC 2003*, vol. 1, pp. 174-178, Dec. 2003.

[14] L. Lenzini, M. Luise, and R. Reggiannini, "CRDA: A Collision Resolution and Dynamic Allocation MAC Protocol to Integrate Date and Voice in Wireless Networks," *IEEE JSAC*, vol. 19, no. 6, June. 2001, pp. 1153-1163.

[15] N. Passas, S. Paskalis, D. Vali, and L. Merakos, "Quality-of-Service-Oriented Medium Access Control for Wireless ATM Networks," *IEEE Comm. Mag.*, vol. 35, no. 11, Nov. 1997, pp. 42-50.

[16] L. Alonso, R. Agusti, and O. Sallent, "A Near-Optimum MAC Protocol Based on the Distributed Queueing Random Access Protocol (DQRAP) for a CDMA Mobile Communication System," *IEEE JSAC*, vol. 18, no. 9, Sep. 2000, pp. 1701-1718.

[17] L. Georgiadis, and P. Papantoni-Kazakos, "A Collision Resolution Protocol for Random Access Channels with Energy Detectors," *IEEE Trans. Comm.*, vol. 30, no. 11, Nov. 1982, pp. 2413-2420.