國 立 交 通 大 學

電信工程學系

碩 士 論 文

無線區域網路中
以多重輪詢為基礎的排程演算法

An Efficient Multipolling-based Scheduling Algorithm
For IEEE 802.11e WLANs

研 究 生：周 勁 文

指導教授：李 程 輝 教授

中 華 民 國 九 十 七 年 十 月

無線區域網路中
以多重輪詢為基礎的排程演算法

An Efficient Multipolling-based Scheduling Algorithm For IEEE 802.11e WLANs

研 究 生：周勁文 Student：Chin-Wen Chou

指導教授：李程輝 Advisor：Tsern-Huei Lee

國 立 交 通 大 學

電 信 工 程 學 系

碩 士 論 文

A Thesis

Submitted to Department of Communication Engineering

College of Electrical and Computer Engineering

National Chiao Tung University

In partial Fulfillment of the Requirements

For the Degree of

Master

In

Communication Engineering

October 2008

Hsinchu, Taiwan, Republic of China

中華民國九十七年十月

無線區域網路中以多重輪詢
為基礎的排程演算法

學生：周勁文 指導教授：李程輝

國立交通大學電信工程學系碩士班

摘要

為了在無線區域網路中提供即時性串流更佳的服務品質，IEEE 802.11e 標準制定團隊，提供了混合協調功能(HCF, Hybrid coordination Function)，其中一種非競爭的存取模式 HCCA(HCF Controlled Channel Access)中使用輪詢(Polling)的機制的，如何在這機制下設定一個好的排程演算法來決定每個工作站的服務間隔(Service Interval)和每次可以得到的傳送機會(TXOP)就是一個重要的課題。

這篇文章提出三種多重輪詢的方法，配合回報的機制，精確地分配傳送機會(TXOP)給每個工作站以符合其需要。無論是對於固定位元速率(CBR)或是變動位元速率(VBR)的串流，皆能更加有效地提升系統的效能。

An Efficient Multipolling-based Scheduling Algorithm for IEEE 802.11e WLANs

Student：Chin-Wen Chou Advisors：Dr. Tsern-Huei Lee

Department of Communication Engineering
National Chiao Tung University

Abstract

For QoS (Quality of Service) requirements of real-time traffic on WLAN, IEEE 802.11 Task Group E provides HCF (Hybrid Coordination Function),which consists an contention-free access mode named HCCA(HCF control channel access) adopting polling mechanism. How to set a good schedule algorithm to decide the service interval and the TXOP duration of STAs is an important issue.

In this paper we present three multipolling algorithms, with the response mechanism, allocate the TXOP duration of STAs accurately for the real requirement. The algorithm provides better the performance of CBR and VBR traffic.

誌謝

能夠完成這篇論文，首先要感謝我的父親周明士先生和我的母親邱玉華女士以及我的兄長周勁言先生。由於他們無怨無悔的付出，從小到大一路支持我的學業，讓我毫無後顧之憂的專注在學業上，得以進入研究所開始我真正的學術生涯。

當然更要感謝我的指導教授--李程輝教授。從他的指導中，我學習到做研究的態度以及做學問的方法，老師總是以身作則，成為我心目中學者的典範。

接者要感謝實驗室的學長景融和郁文，兩年來的熱心指導，對我的研究有非常大的助益。還要感謝實驗室的學長姐士瑋、迺倫、同窗好友耀誼、明智、世弘、凱文、明鑫、錫堯和實驗室的學弟俊德、佑信、松晏、家豪、鈞傑、晨屹、敬堯等各位陪著我度過這兩年的研究生活。

感謝大家，因為有你們，才促使了這篇論文的誕生。

謹將此論文獻給身邊所有愛我的人及我愛的人。

2008 年 10 新竹交大

Contents

List of Tables

List of Figures

Abbreviations and Acronyms

AC	access category
AMBS	aggregate maximum bursty size
AP	access point
ARROW	Adaptive Resource Reservation over WLANs
CFP	contention-free period
CP	contention period
DCF	Distributed Coordination Function
EDCA	Enhanced Distributed Coordination Access
HC	HCF coordinator
HCCA	HCF Controlled Channel Access
HCF	Hybrid Coordinator Function
MAC	media access control
MBS	maximum bursty size
MSDU	MAC Service Data Unit
mSI	minimum service interval
MSI	maximum service interval
mTD	minimum TXOP duration
MTD	maximum TXOP duration
PCF	Point Coordinate Function
PHY	physical
QAP	QoS access point
QoS	quality of service
QS	queue size
QSTA	QoS station
SI	service interval
STA	station
TC	traffic category
TD	TXOP duration

Chapter 1

Introduction

IEEE 802.11 standard [1] is the main standard on the current WLAN technique. It describes how to operate on the MAC (Media Access Control) layer and PHY (Physical) layer. But it does not provide the technique of QoS (Quality of Service) for multimedia traffic stream. IEEE 802.11 Task Group E defines a new MAC protocol to solve this problem.

There are two access modes at IEEE 802.11. One is the DCF (Distributed Coordinate Function) mode, and the other is the PCF (Point Coordinate Function) mode. The former adopts contention-based access to the medium. The latter adopts contention-free access. Both of the two modes can not provide good control mechanism for QoS requirement on WLAN. IEEE 802.11e defines the MAC protocol to support real-time applications with QoS requirement.

IEEE 802.11e [2] MAC protocol is applied on an HC (Hybrid Coordinator), which implements the HCF (Hybrid Coordinator Function). HC may be an AP (Access Point) and adopts two access modes. One is a contention-based scheme, named EDCA (Enhanced Distributed Coordination Access). The other is a contention-free-based scheme, named HCCA (HCF Controlled Channel Access). These provide QoS-enhanced access on WLAN. The EDCA mechanism provides differentiated access to QSTAs using different UPs (user priorities) or ACs (Access Categories). The HCCA mechanism uses HC to allocate TXOP to itself and other QSTAs for providing contention-free transfer of QoS data based on their TSPEC (Traffic Specifications) or

QoS requirements. To achieve the goal, the HC has to add a schedule algorithm to decide how to allocate the bandwidth to polled STAs. This algorithm is named as "traffic scheduler." It is a main topic in 802.11e. It can affect the system performance obviously.

ARROW (Adaptive Resource Reservation over WLANs) [3] is one algorithm meet IEEE 802.11e HCCA. Its basic idea is report the real data amount to HC by piggyback method, in order to get the accurate TXOP duration for transmitting real-time traffic.

Instead of estimating the TXOP timer requirement for QSTAs, ARROW reports the actual amount of data in QSTAs by piggyback and short service interval (SI). ARROW assigns the bandwidth to QSTAs dynamically. It gets better performance than IEEE 802.11e simple scheduler.

Although ARROW is a good scheduler algorithm, we can still do something for improvement. ARROW uses single polling to QSTAs. We consider adopting multipolling to QSTAS to reduce the overhead on WLAN. How to design a good schedules based on ARROW for multipolling is our target of this paper.

The remainder of this thesis is organized as follows. Chapter 2 briefly describes the related work about scheduler algorithm, which includes IEEE 802.11e simple scheduler and ARROW scheduler. Chapter 3 contains our proposed algorithms in detail, focusing on how to construct the polling list of multipolling. Chapter 4 show the performance evaluation of our proposed algorithm compared with ARROW. Chapter contains the conclusions and future work.

Chapter 2

Related Work

In this chapter, we introduce two scheduler algorithms. The first is IEEE802.11e HCCA simple scheduler, and then describe the ARROW scheduler.

2.1 IEEE 802.11e HCCA simple scheduler

There are two access modes at IEEE 802.11e, EDCA and HCCA. STAs (Stations) and APs (Access points) support these two modes are named QSTAs (Qos STAs) and QAPs (QoS APs).

The QAP announces a beacon frame periodically. The period is named a beacon interval. Beacon intervals are divided the period into two periods, contention periods (CPs) and contention-free periods (CFPs). Transmission opportunity (TXOP) is defined as a starting time or a defined maximum duration. QSTA can contention for TXOP in CPs when using EDCA. It can also be assigned by QAPs in CPs and CFPs when using HCCA.

The EDCA mechanism provides different access using eight different user priorities (UPs). The EDCA mechanism defines four access categories (ACs) to map the corresponding UPs as Table 1. In the CPs, the four ACs starts a back-off time when detecting the wireless channel is idle independently after their respective AIFS time. High priorities ACs have the shorter AIFS time and can back-off earlier than others. The contention mechanism is like the DCF at IEEE 802.11.

Table 1-UP to AC mappings

Priority	UP (Same as 802.1D user priority)	802.1D designation	AC	Designation (informative)
Lowest	1	BK	AC_BK	Background
	2	—	AC_BK	Background
	0	BE	AC_BE	Best Effort
	3	EE	AC_BE	Best Effort
	4	CL	AC_VI	Video
Highest	5	VI	AC_VI	Video
	6	VO	AC_VO	Voice
	7	NC	AC_VO	Voice

The HCCA mechanism uses an HC that has higher priority than non-AP STAs. HC transfer MSDUs and allocate TXOPs to non-AP STAs during both CFs and CFPs. The HC allocated TXOPs may be scheduled to meet the QoS requirement of a particular traffic stream (TS) or traffic category (TC). A TS is a set of MSDUs which have consistent QoS parameters. These parameters are defined through TSPEC. The typical TSPEC parameters are showed as Table 2.

For a scheduler, the most important thing is decide when to poll a QSTA and how many amount of TXOP duration for transmitting. In other words, HC should decide the service interval (SI) and the TXOP duration for a QSTA. In IEEE 802.11e, HC can get the traffic stream (TS) parameters by TSPEC field. The relative parameters are described as below:

Table 2-TSPEC parameters

TSPEC Parameter	**Symbol**	**Description**
nominal MSDU size	*L*	nominal MSDU size
maximum MSDU size	M	maximum MSDU size
minimum service interval	mSI	minimum interval between two success transfer
maximum service interval	MSI	maximum interval between two success transfer
minimum data rate	mR	minimum data rate for transport of MSDUs
mean data rate	ρ	mean data rate for transport of MSDUs
peak data rate	PR	peak data rate for transport of MSDUs
maximum burst size	MBS	maximum burst of MSDUs at peak data rate
delay bound	D	maximum delay allowed to transport an MSDU
minimum physical rate	R	the minimum physical rate to use

IEEE 802.11e provides an example scheduler, the simple scheduler. The HC schedule fixed length TXOP at constant intervals. The first step is the calculation of the SI. The second step is the calculation the TXOP duration for the SI.

The calculation of SI is down as follows:

1) Calculate the minimum of maximum SI for all admitted streams. Let the minimum is *m*.

2) Choose a number lower than *m* is a submultiple of beacon interval. This number is the SI.

The calculation the TXOP duration of the an admitted stream, the scheduler uses the following TSPEC parameters: mean data rate (ρ), nominal MSDU size (*L*), service interval (SI), minimum physical rate (R), maximum MSDU size and overhead in time units (*O*). The scheduler calculates the number of MSDUs (N_i) arrived at the mean data rate during a SI:

$$N_i = \left\lceil \frac{SI * \rho_i}{L_i} \right\rceil \quad (2.1)$$

For each $QSTA_i$ having n_j traffic streams, the calculation of TXOP duration (*TD*) of a traffic stream *ij* *(TS_{ij})* is obtained as follows:

$$TD_{ij} = \max\left(\frac{N_{ij} * (L_{ij} + O)}{R_{ij}}, \frac{M}{R_{ij}} + O\right), \quad j \in [1, n_i] \quad (2.2)$$

Then the total TXOP duration for $QSTA_i$ can be assigned as:

$$TD_i = \sum_{j=1}^{n_i} TD_{ij} \quad (2.3)$$

So the simple scheduler can establish an effective schedule on HCCA.

2.2 ARROW

ARROW (Adaptive Resource Reservation over WLANs) is one algorithm meet IEEE 802.11e HCCA. Its basic idea is report the real data amount to HC by piggyback method, in order to get the accurate TXOP duration for transmitting real-time traffic. ARROW utilizes some aggregate parameters:

Minimum TXOP Duration (mTD): The minimum TXOP duration can be assigned to a QSTA shall be at least the time to transmit one maximum MSDU size at the minimum physical rate.

$$mTD_i = \max\left(\frac{M}{R_{ij}}\right), \quad j \in [1, n_i] \quad (2.4)$$

Maximum TXOP Duration (MTD): The maximum TXOP duration can be assigned to a QSTA shall be less than or equal to the transmission of the **Aggregate Maximum Burst Size (AMBS)** of a QSTA. AMBS is the sum of burst size of all TSs of a QSTA.

$$AMBS_i = \sum_{j=1}^{n_i} MBS_{ij}$$
$$MTD_i \le \frac{AMBS_i}{R_i} \tag{2.5}$$
$$R_i = \min(R_{ij})$$

Minimum Service Interval (mSI): The minimum time interval between two successive TXOP assigned to a QSTA. The scheduler calculates the minimum of mSIs for all TSs of a QSTA:

$$mSI_i = \min(mSI_{ij}), \quad j \in [1, n_i] \tag{2.6}$$

If mSI is not specified, it is given as the nominal MSDU size / mean data rate

$$mSI_{ij} = L_{ij} / \rho_{ij} \tag{2.7}$$

Maximum Service Interval (MSI): The maximum time interval between two successive TXOP assigned to a QSTA. Before deciding the MSI, we should discus the basic concepts of ARROW first. ARROW exploits the Queue size (QS) field, a part of QoS data frame, introduce by 802.11e to report the buffer data amount at QSTAs for their TSs. The scheduler can use the information to assign TXOPs to QSTAs to meet their QoS requirement.

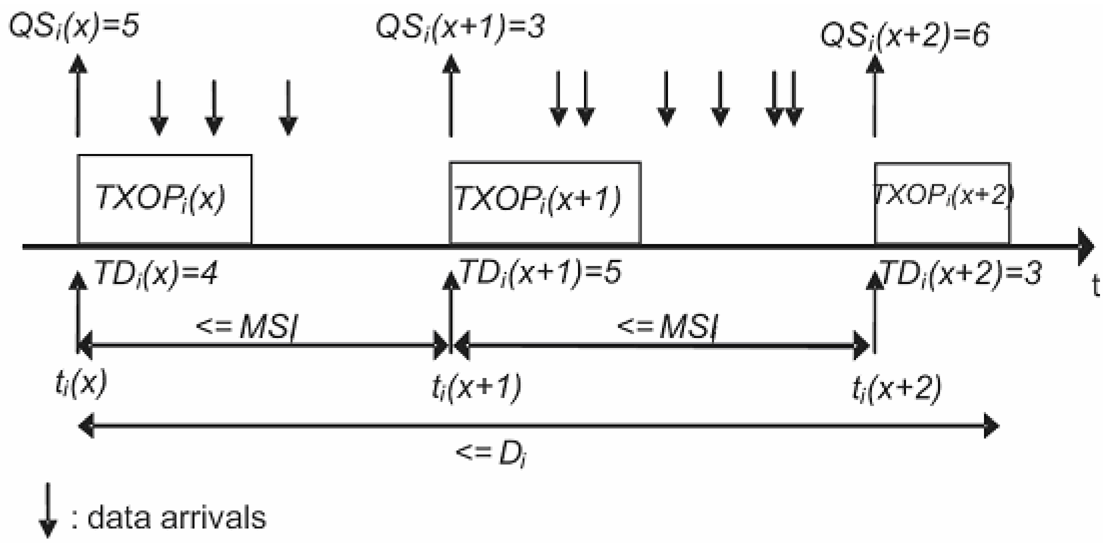

Figure 1: TXOP assignment of ARROW

An example of ARROW using QS field is depicted in Figure 1. For simplicity, we assume that only one TS per QSTA. At time $t_i(x+1)$, $TXOP_i(x+1)$ is assigned to $QSTA_i$ according to the requirements of $QS_i(x)$. The data generated of $QSTA_i$ during the interval $[t_i(x), t_i(x+1)]$ is piggybacked on the QoS data frame at $TXOP_i(x+1)$ to renew $QS_i(x+1)$. Then the scheduler assigns $TXOP_i(x+2)$ to QSTA according $QS_i(x+1)$ equaling the amount of data generated during $[t_i(x), t_i(x+1)]$. By using QS field, ARROW has sufficient information about the characteristic of each TS. The scheduler adapt the TXOP duration according these information. This is essential especially in the case of VBR or burst traffic. As we observed in figure 1, for each $QSTA_i$, the data generated during $[t_i(x), t_i(x+1)]$ can not be transmitted before $TXOP_i(x+2)$ starting at $t_i(x+2)$. In order to not exceed the delay bound of MSDUs, the worst case is service interval equaling MSI, and $TXOP_i(x+2)$ equaling MTD.

$$D_i \geq 2MSI_i + MTD_i \Leftrightarrow MSI_i \leq \frac{D_i - MTD_i}{2} \qquad (2.8)$$

If we consider the retransmission (2.8) will become:

$$MSI_i \le \frac{D_i - MTD_i}{2+m} \tag{2.9}$$

m is the maximum retransmission attempts.

(2.8) and (2.9) show that MSI should be less than the half of delay bound. ARROW has the shorter service interval and need more TXOP assignment than other algorithms. The overhead will be increased. But the disadvantage is worthy of accuracy TXOP assignment.

ARRROW incorporates a traffic policing mechanism to ensure the transmit requirement of QS field will not violate the characteristics of TSPEC. It adopts a TXOP timer. There is a timer T_i for each $QSTA_i$ with n_i TSs. The timer value is increasing with rate $r(T_i)$

$$r(T_i) = \sum_{j=1}^{n_i} \left(\left(\frac{L_{ij}}{R_{ij}} + O \right) / \frac{L_{ij}}{\rho_{ij}} \right) \tag{2.10}$$

Equation (2.10) means the increasing rate equals the ratio of data transmission time to data generated time.

The maximum timer value $\max(T_i)$ equals the time required for transmission all maximum burst size.

$$\max(T_i) = \sum_{j=1}^{n_i} \left(\frac{MBS_{ij}}{R_{ij}} + O \right) \tag{2.11}$$

The TXOP duration lager than timer value T_i can not be assigned to $QSTA_i$. The respective timer should reduced correspond to the assigned TXOP duration.

The operation of ARROW can divided with the following steps:

1) The scheduler was for the channel idled

2) If the channel is idle at moment t

$QSTA_i$ can be polled if

a) $t > t_i + mSI_i$ (2.12)

mSI requirement

t_i is the time that QSTA be polled

b) $T_i > mTD_i$ (2.13)

Timer requirement

T_i is the timer value.

3) If no QSTAs satisfy (2.12) and (2.13), the scheduler will wait until the two equations are true.

4) If there are many QSTAs which have satisfied these conditions, the scheduler polls the QSTA with the earliest deadline. The deadline is t_i+MSI_i.

5) The scheduler the TD_i for polled $QSTA_i$

a) $TD_{ij} = \max(\frac{QS_{ij}}{R_{ij}} + O, mTD_{ij}) \quad , mTD_{ij} = \frac{M_{ij}}{R_{ij}}$ (2.14)

If QS_{ij} equals zero, QSTAs still need transmit a NULL-data for updating the data amount of the previous service interval of TS_{ij}. Then TD_{ij} equals a NULL-data MSDU length.

b) $TD_i = \sum_{j=1}^{n_i} TD_{ij}$ (2.15)

The TXOP duration of $QSTA_{ij}$ is the sum of TXOP duration of all TS_{ij} of $QSTA_{ij}$

c) $TD_i = \min(TD_i, T_i)$ (2.16)

TD_i is calculated as the minimum of current timer T_i and the TXOP duration obtained from (2.15) to ensure that the traffic of $QSTA_i$ will not exceed the negotiated traffic.

6) $T_i = T_i - TD_i$ (2.17)

After scheduler assigns the TXP, the timer reduces the corresponding value according to the TXOP duration length and then returns step 1).

ARROW can be realized through the above algorithm and operated under the 802.11 architecture.

Chapter 3

Proposed Algorithms

Although ARROW is an efficient algorithm, we can still have some improvement with it. ARROW has the sufficient information of the data amount in the buffer of each traffic stream at each QSTA. At the ARROW algorithm step 2), scheduler knows that there are much data have to be assigned in many QSTAs, but the scheduler just poll the QSTAs one by one. This is wasted the bandwidth. If we can reduce the overhead of polling frame, we can reserve more bandwidth to transmit more data or support more STAs to join the WALN. On the other hand, ARROW uses the timer to constraint QSTA not exceed their negotiated traffic. But if the QSTA number is low, the mechanism may increase the packet loss probability where the bandwidth is still enough for transmit these data. In fact, under ARROW algorithm, the scheduler has knows if the bandwidth is enough, so we do not need use timer to restrict the traffic amount.

Under 802.11, for reducing the overhead due to polling frame, a concept named multipolling is wild adopted. We can use the method to improve the above problems.

We explain why multipolling does reduce overhead first.

Byte : 2	6	1	5 x Polling Count (N)			4
Frame Control	BSSID	Polling Count (N)	Polling Control			FCS
			AID (2bytes)	Rate (1byte)	TXOP (2bytes)	

Figure 2: multipolling frame format

In [4], it proposed a multipolling frame, we adopt the same frame. The length of each multipolling frame is 13+5N[bytes]. The length of a single polling frame is 18 bytes. The length of multipolling frame of 5 QSTAs is 13+5*5=38 [bytes]. The ratio of the length of single polling frame to the length of an extra QSTA in a multipolling frame is 3.6.

If we follow the 802.11a standard [7], use the basic physical rate 6Mbps transmitting the multipolling frame, the time required is depicted as follows:

Table 3-multipolling frame time require under 6Mbps

N	1	2	3	4	5	6	7	8	…
$T_{poll}(N)$ [μs]	64	72	80	84	92	100	104	112	…

As Table 3, the ratio of the time required of adding a single polling frame to the time required of an extra QSTA in multipolling frame is about eight to sixteen. This is much larger than the length ratio observed on MAC layer. Multipoll more QSTAs can really reduce the overhead of polling frame efficiency.

The simplest multipolling method is multi-poll all QSTAs each time. The polling period is calculated of the minimum of the mSI of all TSs, No matter what‘s the inter arrival rate or delay bound. If the difference of each mSI of QSTA is large, the NULL-data frame will be generated more and the bandwidth is wasted. So we do not consider adopting the method.

3.1 Proposed algorithm 1

ARROW is a good algorithm which it can be used for the case that difference packet inter arrival rate and mSI of QSTAs. Under ARROW architecture, we can consider another simple multipolling method. It is modify the ARROW algorithm in the chapter 2. We can multipoll all QSTAs that have satisfied the mSI requirement and timer requirement in the step 2). The other steps are the same as ARROW. Or we can say the step 4) is modified as follows:

4) If there are many QSTAs which have satisfied those conditions, the scheduler multipolls the QSTAs. And the transmit order is according deadline, earliest deadline first. The deadline is t_i+MSI_i.

This method may be a good algorithm. In fact, when the number of QSTAs is low, after one multi-polling, every the time gap between the end of transmit time of a QSTA and its mSI constraint time is a little distance. The channel will be idle. The scheduler waits for a QSTA which have satisfied the mSI requirement and assign TXOP duration to it. But this polling usually is just a single polling because every QSTA have different start time. When the number of QSTAs is low, the method is not effective. We must find a more efficient method.

3.2 Proposed algorithm 2

We want to design a multipolling algorithm that can multipoll more QSTAs as possible for reducing the polling overhead and not increasing the NULL data overhead to improve the system performance.

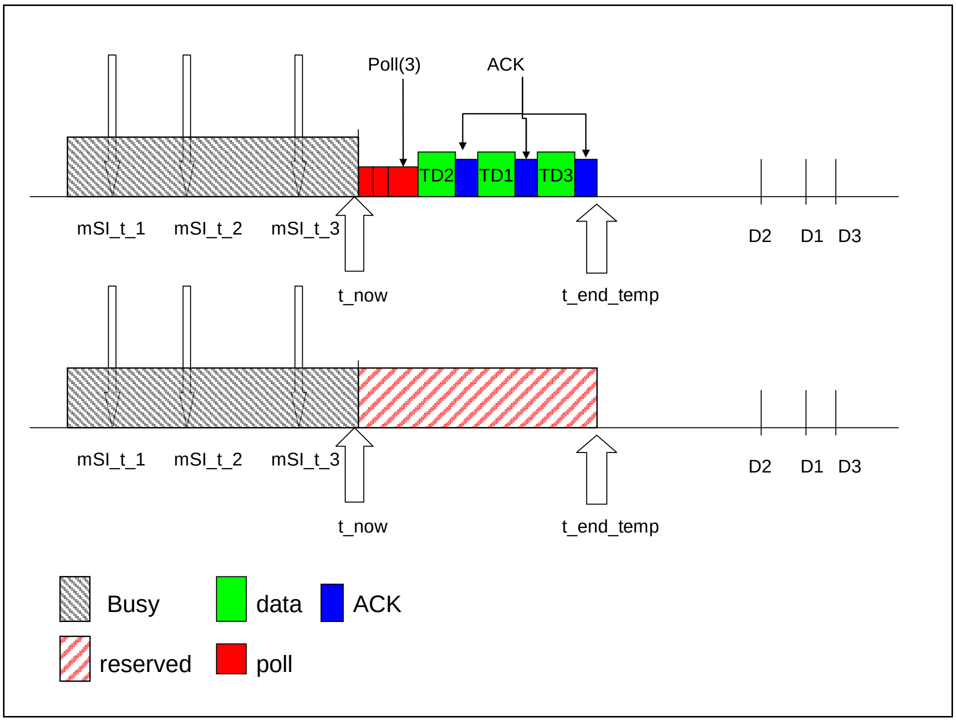

Figure 3: constructing the polling list

As Figure 3, assume the channel becomes idle at time t_now, the scheduler determines which QSTAs will be added in the polling list. mSI_t_i means the time that $QSTA_i$ will have satisfied the mSI requirement. In the case 3, when channel is idle, there are three QSTAs have satisfied the mSI requirement. So the scheduler adds the three in the polling list. For avoiding the packet loss, the transmission order is according the delay bounds, earliest deadline first. The method is the as method 1. But the method 2, considering adding more QSTAs in the polling list, we record the time that QSTAs in the polling list will be occupied and the temp end time (t_end_temp).

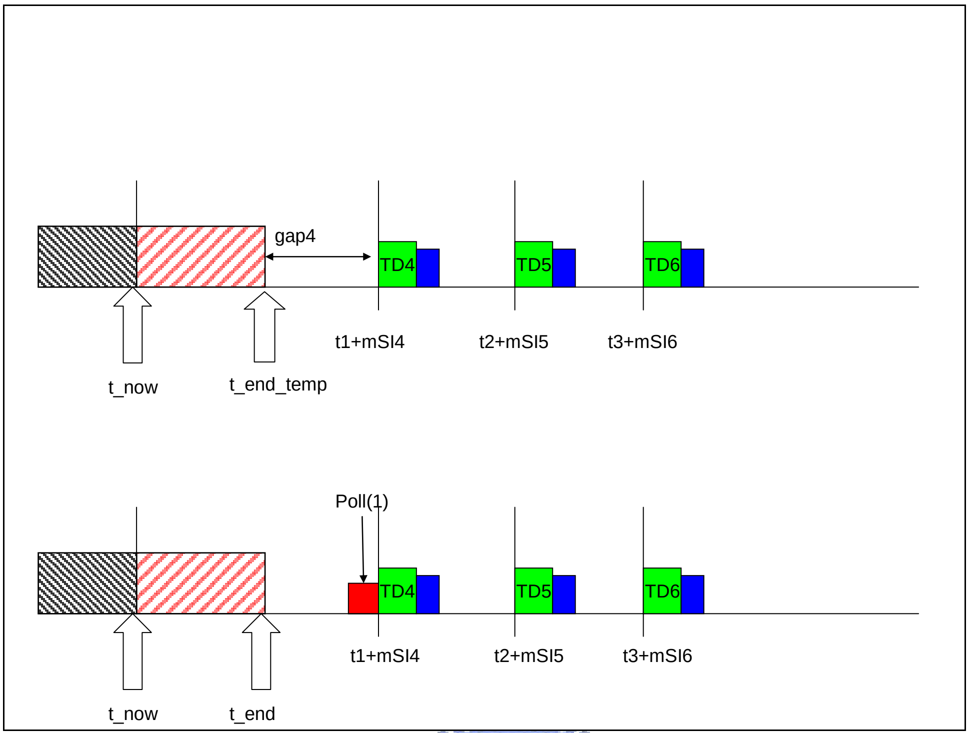

Figure 4: method 2, case 1

If the time gap of temp end time t_end_temp and the next closest mSI_t_i is larger than the time for transmitting a single polling frame ($T_{poll}(1)$), then using the t_end_temp is the real end time.

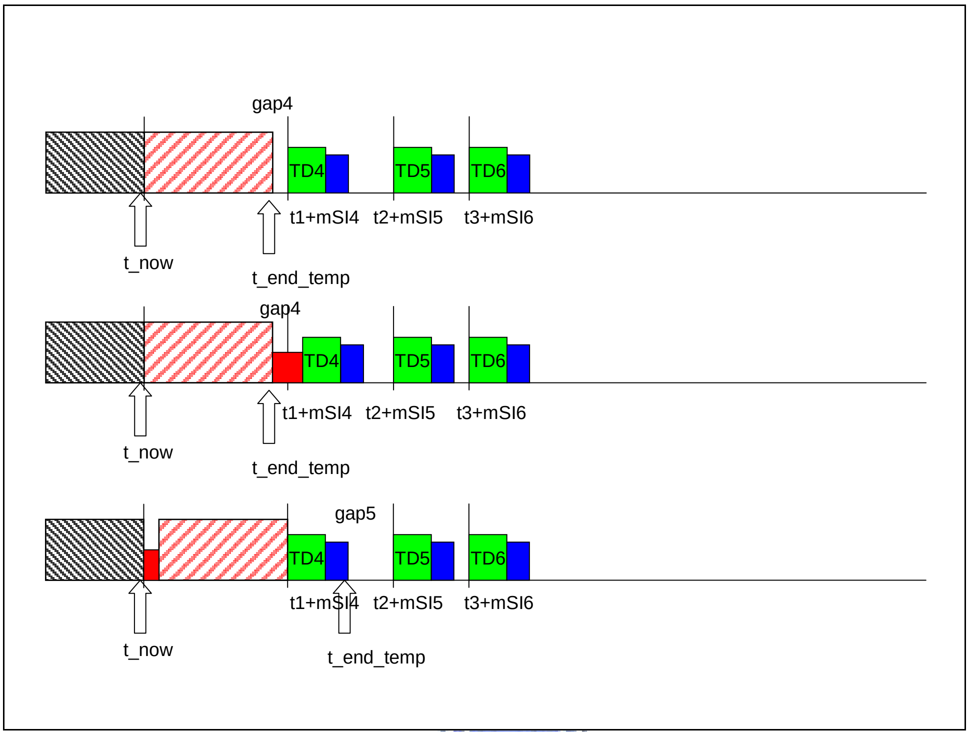

Figure 5: method 2, case 2-1

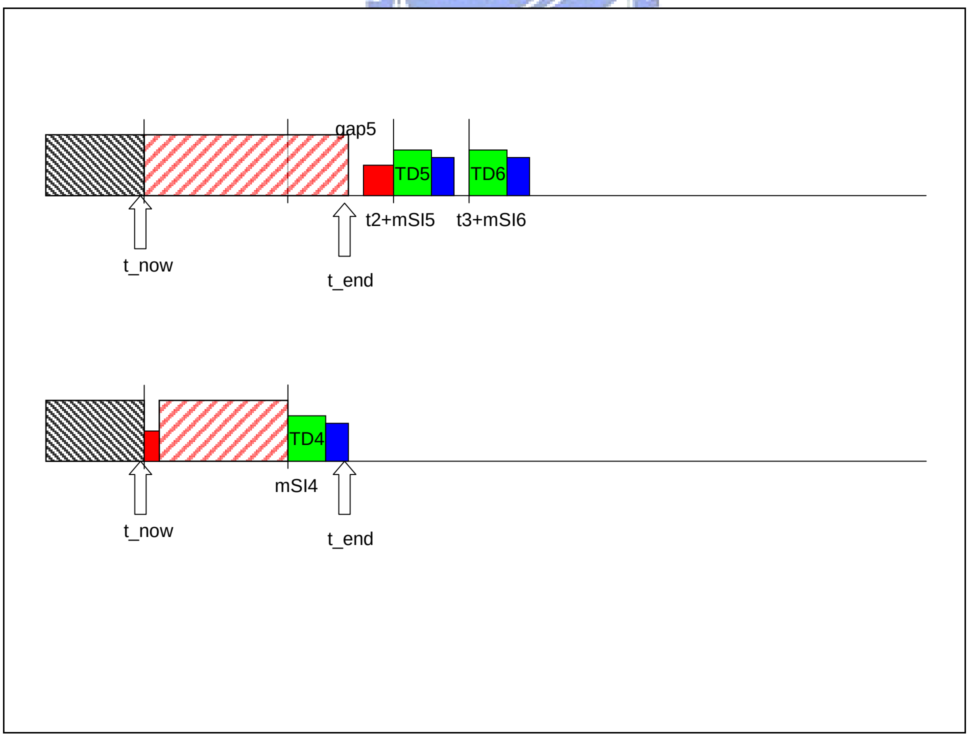

Figure 6: method 2, case 2-2

If the time gap of temp end time t_end_temp and the next closest mSI_t_i is samller than the time for transmitting a single polling frame, then add the next QSTA ($QSTA_4$) in the polling list, and then calculate the new t_end_temp and overserve the gap between next mSI_t_i($QSTA_5$). The reason of designing is hope multipolling more QSTAs. Choose a single poll frame be the criterion is hope transmit starting time will not be too earlier or too late than the desired starting time mSI.

3.3 Proposed algorithm 3

The scheduler using method 2 need calculate the end time every time after new QSTA added in polling list for being criterion that used for adding new QSTA. The step may cost much time on calculating. We wish to designing a new Method that do not use recursive method, directing using the information of mSI and needed TXOP duration , delay bound to decide the polling list.

Considering the method 2, the QSTA which have satisfied the mSI requirement are added in the polling list and the scheduler has already calculated t_end_temp. If the timer difference between t_end_temp and the closest mSI_t_i less than $T_{poll}(1)$, then the QSTA will be added in the polling list. But we consider the next STA may be added in the polling list, the decision will accord the other QSTAs.

We defined the gap as follows:

$$gap_i = mSI_i - (mSI_{i-1} + TD_{i-1}) \tag{2.18}$$

Under Method 2, if the gap_i is larger than Tpoll(1), added $QSTA_i$ in the polling list. By method 3, assume there are N QSTAs on the WLAN. We consider the gap of last QSTA ($QSTA_N$) first. The scheduler adds QSTAs by the following algorithm:

p=0;

for q=N:-1:s+1

if poll(p) $\geqq gap_q$

p=p+1;

else p=0;

p'=p+s

p: the number of multipolled STAs should be increased

P': the final numbers of multipolled STAs

s: the number of STAS that satisfied the condition (ti+mSIi<t)

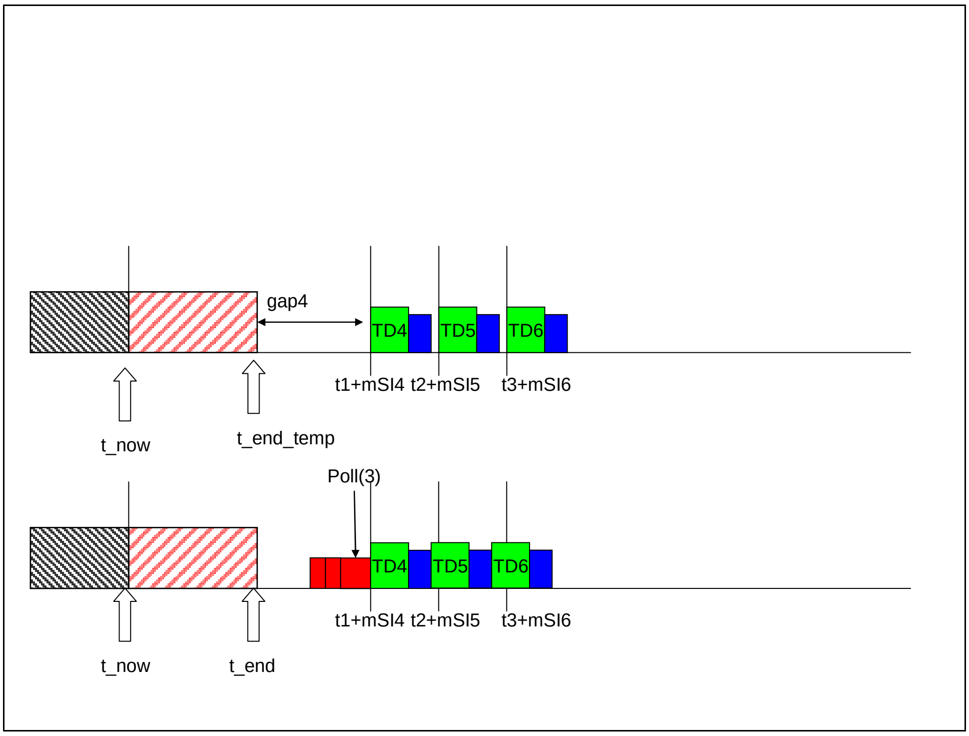

Figure 7: method 3, case 1

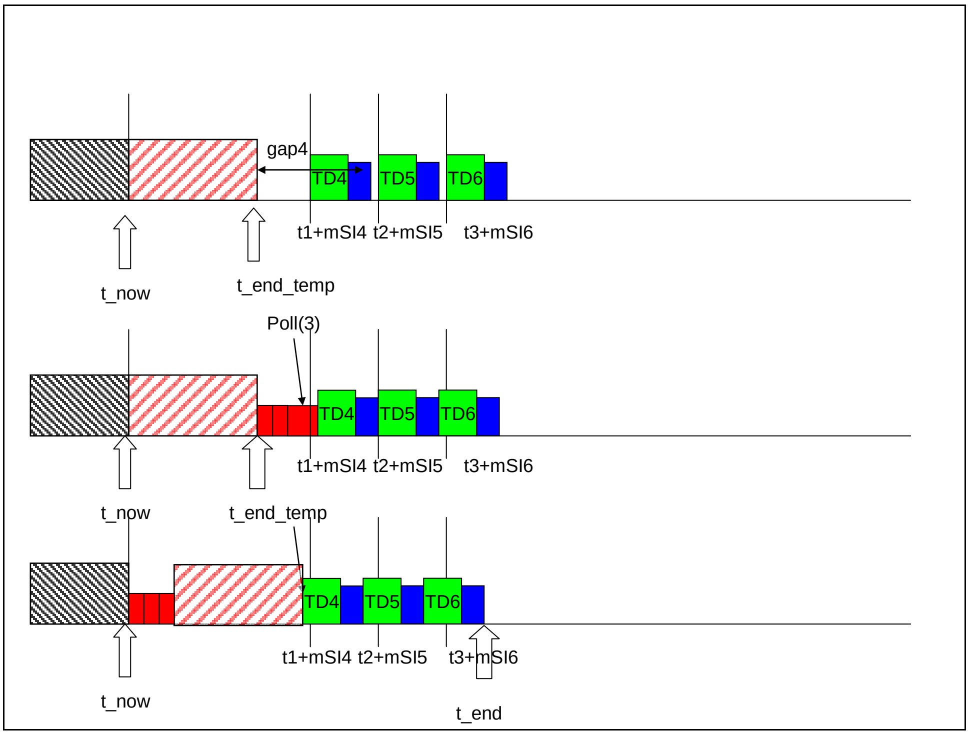

Figure 8: method 3, case 2

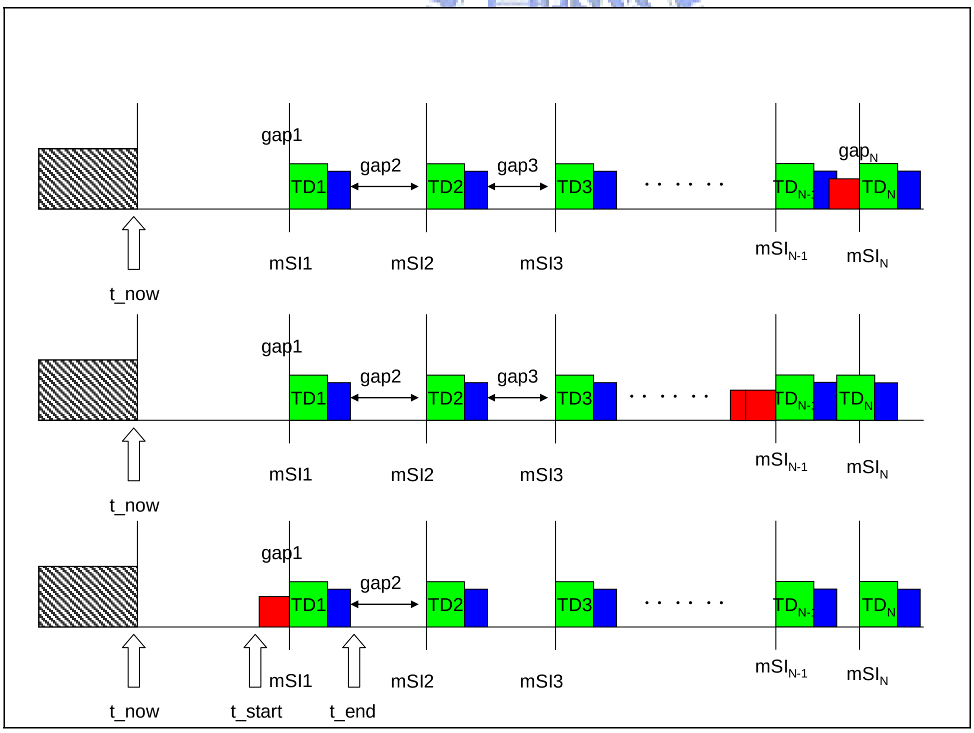

Figure 9: method 3

Chapter 4

Performance Evaluation

IEEE 802.11e defined the standard on MAC layer, but different protocol of physical layer still effect the system performance. We need choose a suitable protocol to simulate our algorithms. We choose 802.11a architecture. Packet like polling or ACK transmitted under 6Mbps except data and NULL data under 54Mbps. No considering retransmission. The simulation is achieved by MATLAB.

Each packet is fixed size. Follows are the metric we compared:

Throughput of Non-Delayed MSDUs: The throughput is calculated total success transmitted MSDUs by each algorithm.

Mean Delay of Non-Delayed MSDUs: the mean delay by MSDUs transmitted successfully

Overflow Percentage: the packet loss due to the queue overflow divided total packet generated.

HCCA System Occupancy Percentage: the total transmitting time divided the total time system operated.

Overhead Percentage: the overhead of MAC layer and PHY layer, including MSDU header, NULL data, ACK, polling frame, PHY header and IFS, divided the total transmitting time

We simulate two different traffics. The first is constant bit rate (CBR) traffic. Uploading and downloading have traffics respectively for simulating transmitting voice. The second is variable bit rate (VBR), just uploading, simulated QSTA transmitting video. The packet inter arrival time is generated by exponential distribution with mean rate ρ and peak rate PR. On the other hand, we simulate two different environments on VBR. Setting mSI=0 reduces the mean delay but bandwidth wasted. Setting mSI=peak arrival time can reduces the channel occupancy but increase the mean delay. The simulation TSPEC parameters are showed on Table 4:

Table 4-TSPEC parameters

TSPEC parameters	CBR	VBR-1	VBR-2
ρ[kbps]	83	256	256
D[mS]	60	40	40
L[bytes]	208	1279	1279
M[bytes]	208	3893	3893
MBS[bytes]	576	3893	3893
PR[kbps]	83	760	760
R[Mbps]	54	54	54
mSI[mS]	20	0	L/PR

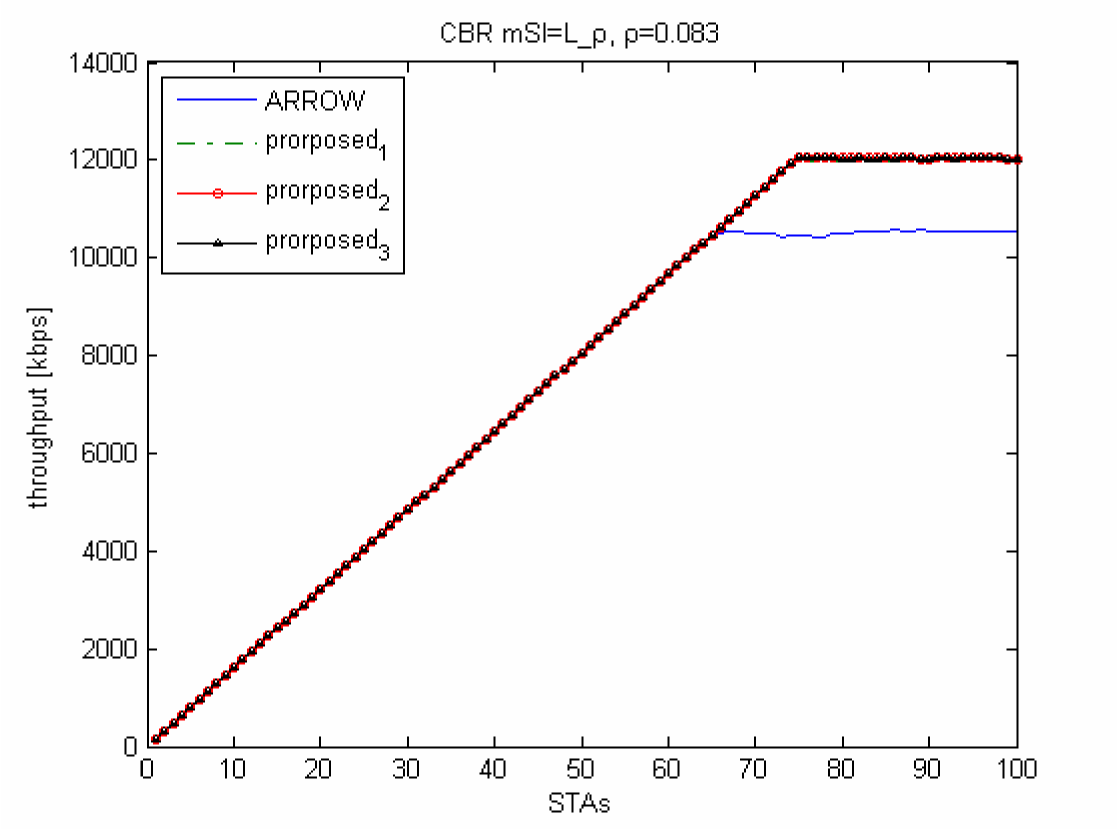

Figure 10: CBR throughput

Figure 10 depicted the throughput under CBR traffic using each algorithm. We can observe that the maximum capacity of the system is about 65 STAs under ARROW. If adopting multipolling, three algorithms can make the capacity reach 74 STAs. And the capacity can be calculated as follows:

For ARROW:

$$MSI \geq N * [T_{poll}(1) + n_i(T_{data} + T_{ACK}) * \frac{MSI}{L_{data}/\rho}] \tag{2.19}$$

It means that every STA must wait for a single poll frame and then transmit data frame and wait for ACK frame. The mean data generated can be calculated by packet arrival rate. In which T_{poll}, T_{data}, T_{ACK} has considered the MAC header and PHY header and every kind of overhead. Given relative parameters can get the capacity equaled 65 QSTAs

For multipoll:

$$MSI \geq T_{poll}(N) + N * n_i * (T_{data} + T_{ACK}) * \frac{MSI}{L_{data}/\rho} \quad (2.20)$$

At the best situation, all QSTAs are in the polling list, so we modified (2.19) to (2.20) and then get the capacity is 74 QSTAS. The result matches our simulation.

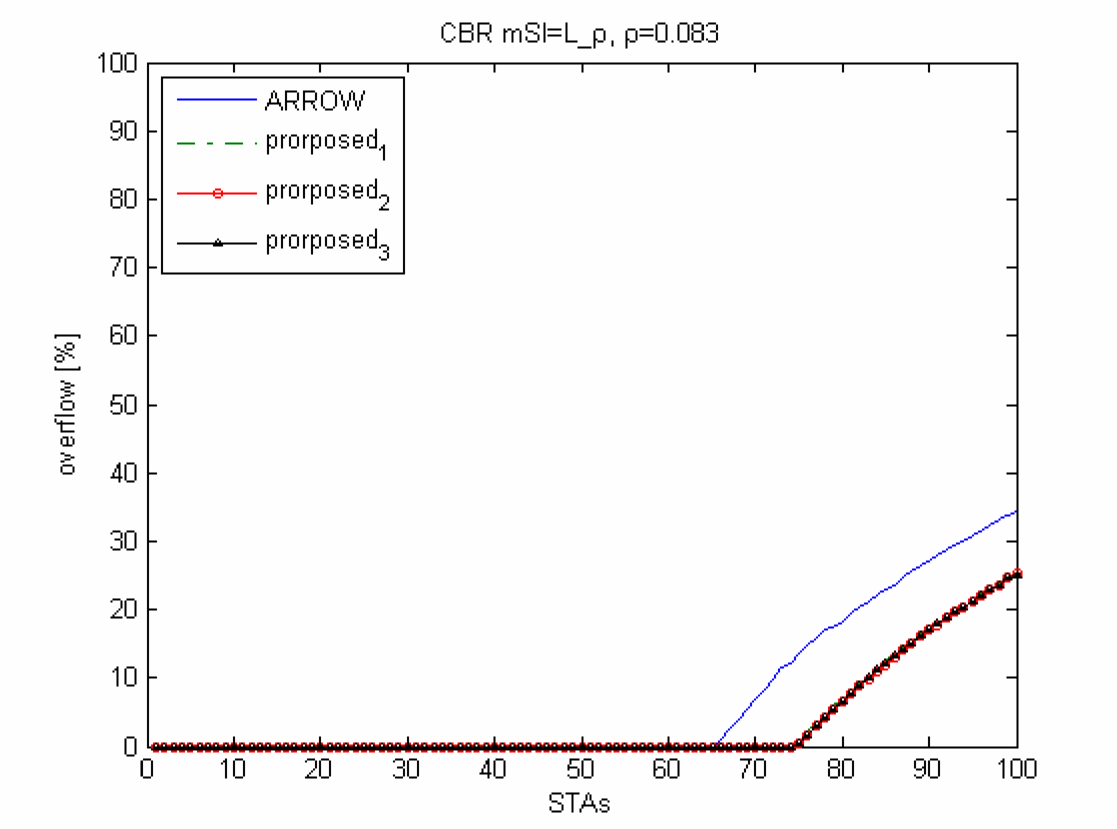

Figure 11: CBR overflow

Figure 11 shows that the capacity is 74 with multipolling algorithms, and 65 QSTAs without multipolling. As we mentioned before, there are no packet loss under 65 QSTAs on WLAN with ARROW algorithm, and no packet loss under 74 QSTAs on WLAN with multipoll algorithms.

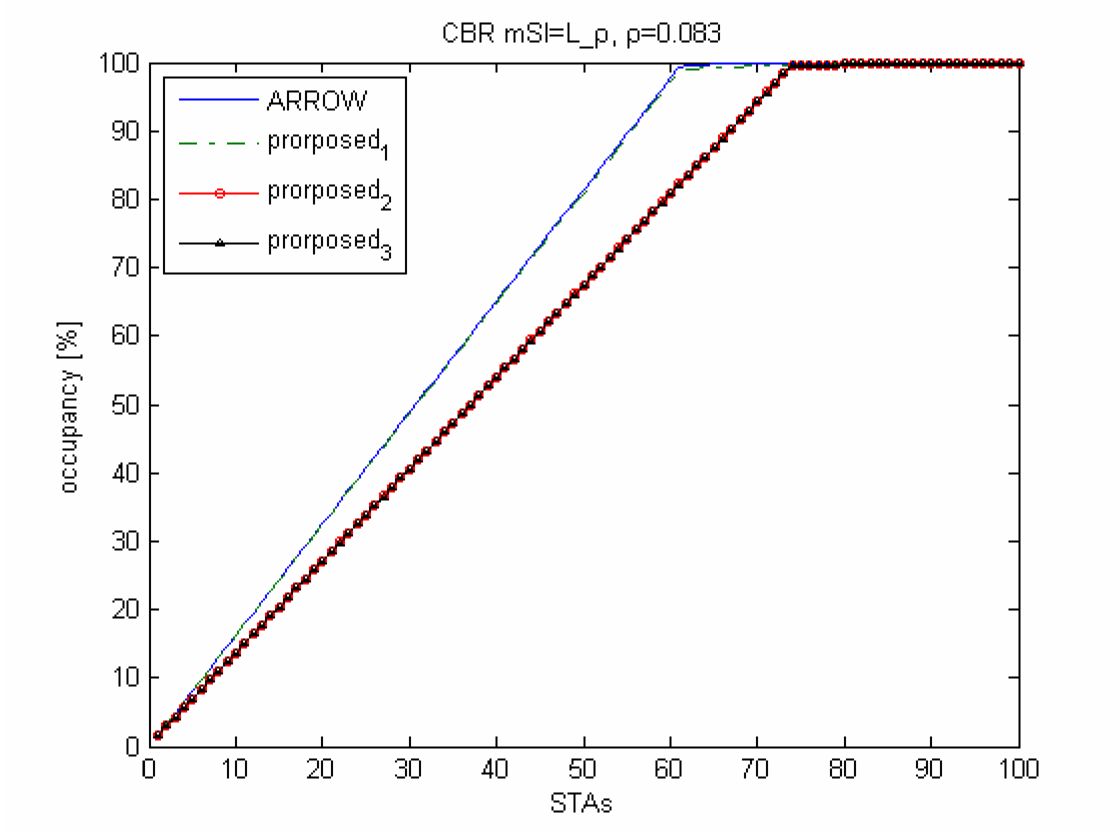

Figure 12: CBR occupancy

Figure 12 shows that the occupancy percentage with ARROW or our proposed algorithm 1 is about 100% when the number of QSTAs is 61. When the number of QSTAs is low, every STA almost be served between an mSI. For adding more QSTAs the new added QSTAs is just be served after the current QSTAs. If there is no enough space to add new QSTA, the system will increase the SI for each QSTA, so the occupancy will be full loaded before the packet loss appeared. With multipolling, due to the lower overhead, the occupancy is lower than the occupancy with ARROW. But the algorithm1, when the number of QSTAs is low, there is lower opportunity multipoll many QSTAs. The scheduler multipolls all QSTAs while the number of STAs is large. This is also the reason why its occupancy is almost 100 percent.

The algorithm 2 and 3, although the channel is idle and there are no QSTAs have satisfied the mSI requirement, the scheduler can determine the end time and decide if other QSTAs can be added in polling list. It may multipoll with little number of STAs so the overhead can be reduced but also occupancy.

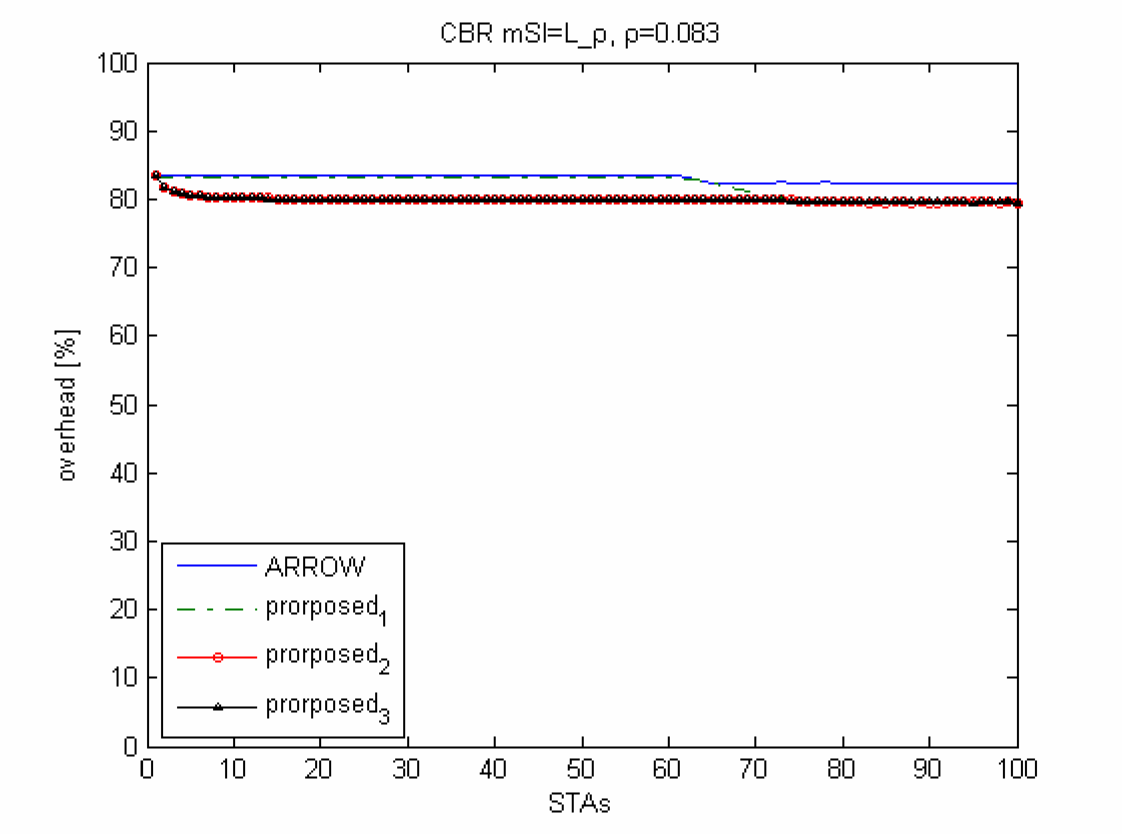

Figure 13: CBR overhead

Figure 13 compares the overhead. As mentioned before. Algorithm is almost the same as ARROW with little numbers of QSTAs.

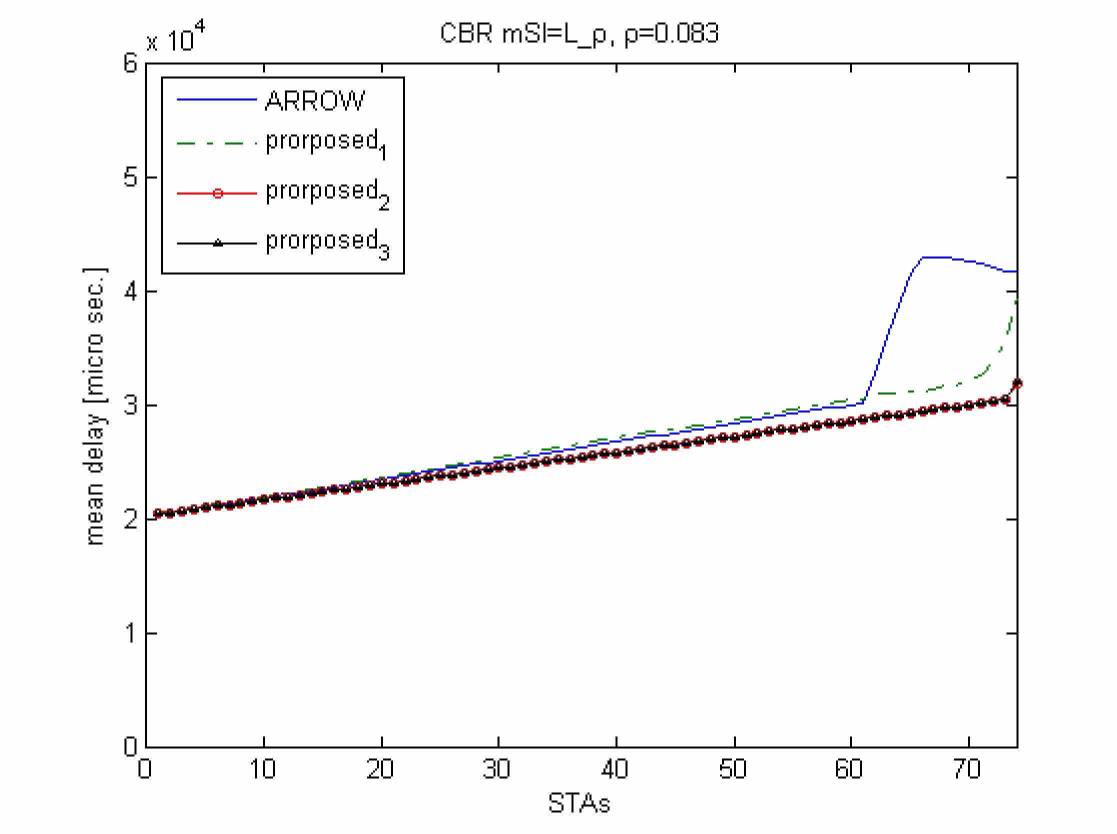

Figure 14: CBR mean delay

Figure 14 shows the mean delay. With only one QSTA, it can be served when just satisfied the mSI requirement. With more QSTAs, new added QSTA should wait the old QSTAs serving, the delay will be increased. And the sharp increasing is due to the SI is increased. The decay is due to the packet loss.

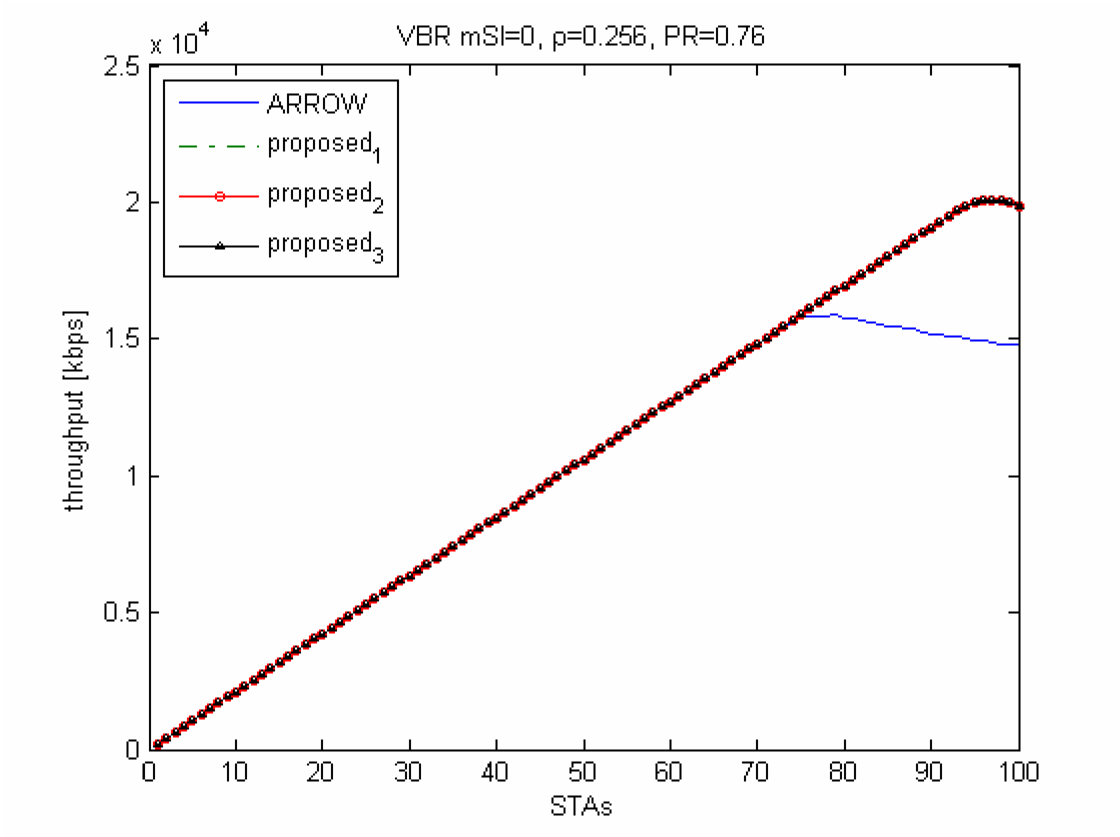

Figure 15: VBR1 throughput

Figure 15 shows the throughput with VBR traffic. We consider setting the mSI equals zero to meet the variable bit rate. On the other words, the scheduler polls QSTAs continuously. The performance of three algorithms is no different because they always multipolls all QSTAs when the channel is idle. The decay is due to the packet loss. The throughput of CBR did not decrease due to the QSTA may transmit a new packet instead of the loss packet. In fact, if the delay bound of VBR traffic is longer, the throughput will be consistent.

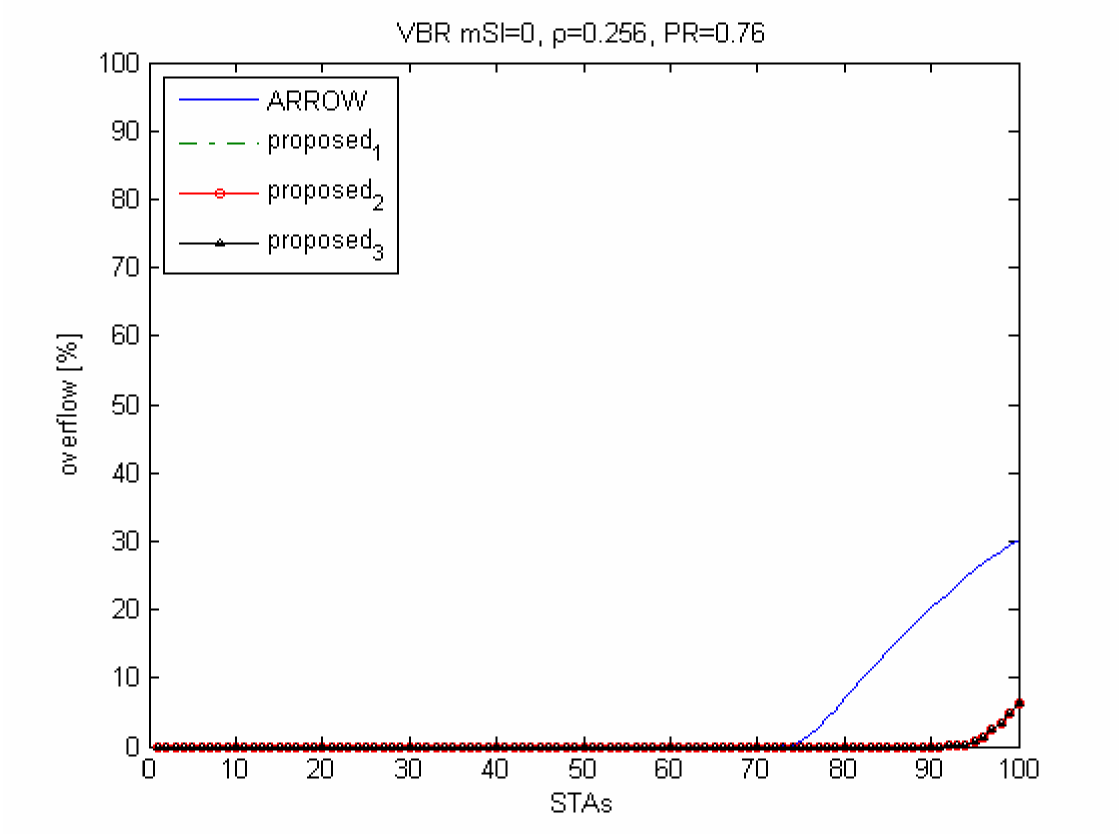

Figure 16: VBR1 overflow

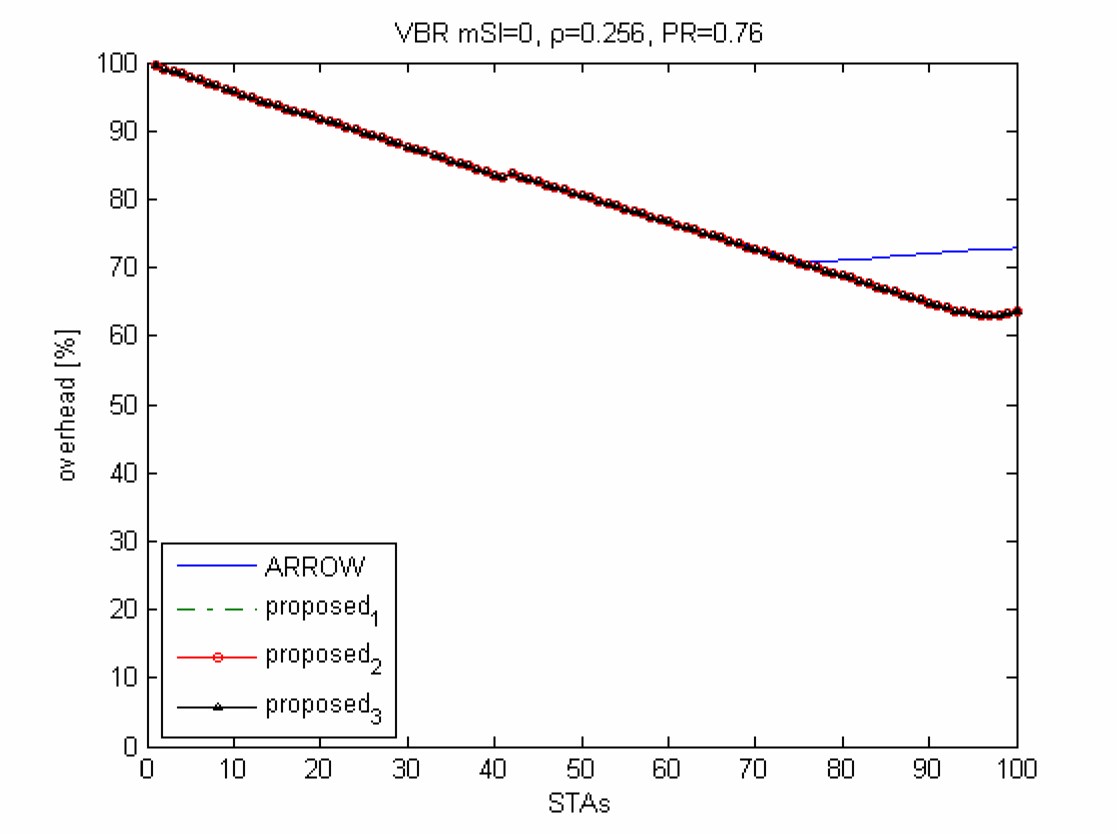

Figure 17: VBR1 overhead

Figure 17 shows the overhead. We may think multipoll algorithms are not better. But we need know the definition of the overhead metric. The definition is: the total transmission time reduce the data transmission time divided the total transmission time. Because mSI is setting equals zero, the channel is always occupied. If there is no packet loss, the overhead will decrease due to much time used for transmitting data. We may think that the overhead should be decreased. In fact, the overhead of polling is really decreased, but the NULL-data overhead is increased. The total result is that the overhead is the same as single polled with lower numbers of QSTAs.

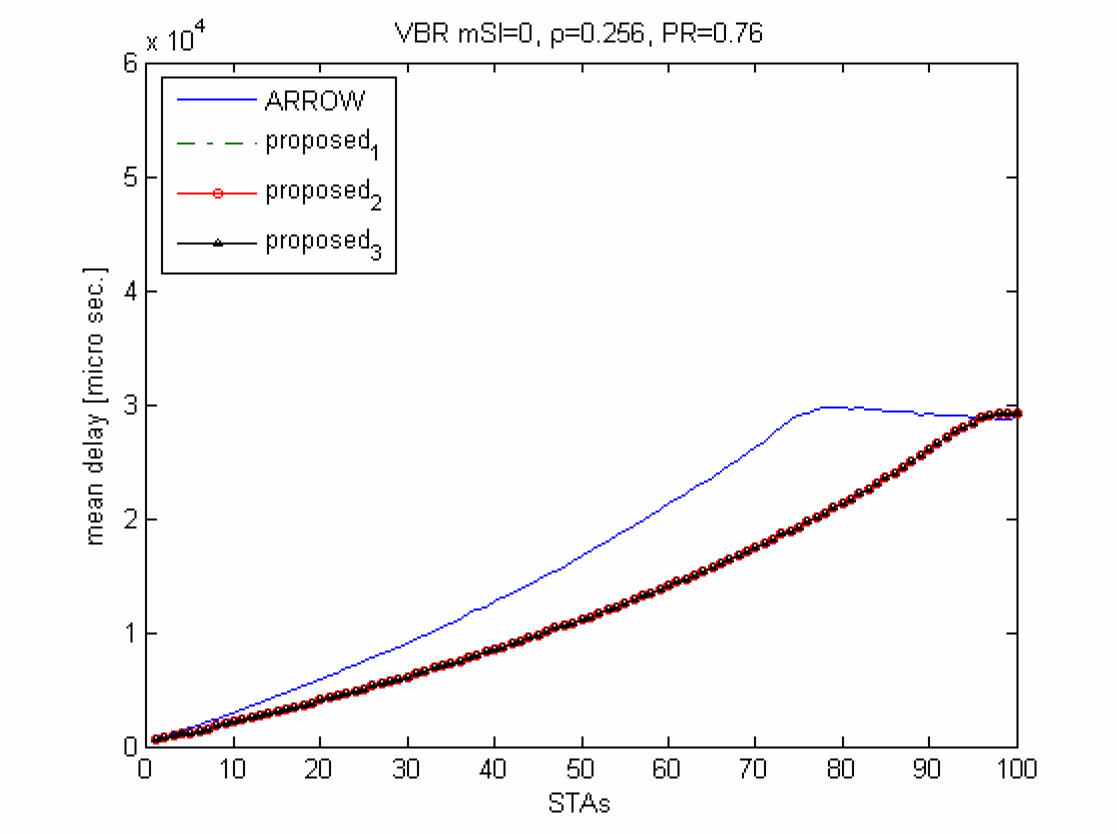

Figure 18: VBR1 mean delay

The delay with ARROW algorithm is higher than multipoll algorithms depicted in Figure 18. STAs can be served after the earlier deadline QSTAs have already served. Adopting multipolling decreases the waiting time so the mean delay decreases.

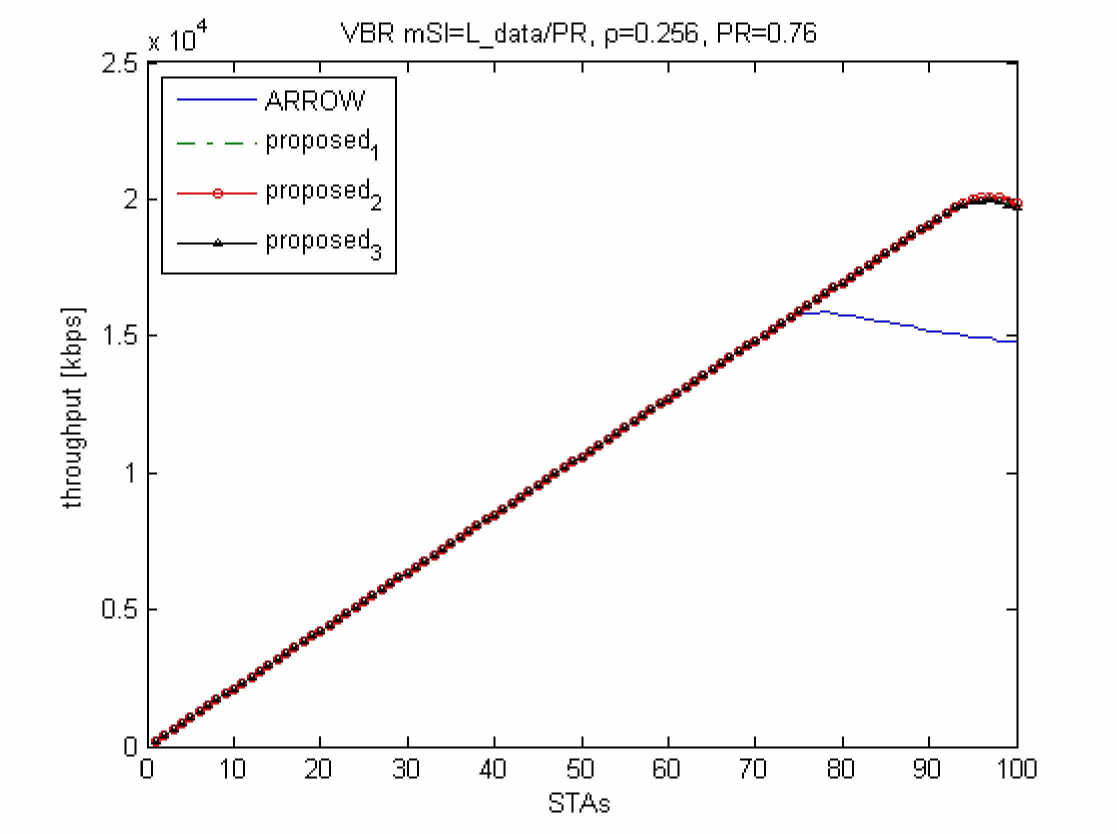

Figure 19: VBR2 throughput

As mentioned before. The overhead is too much if mSI equals zero. For VBR traffic, we can use the parameter peak rate (PR) in TSPEC to decide the peak inter arrival time. We Use the peak arrival to be mSI. The packets may be generated after an mSI, but it surely not be generated before an mSI. Observing the figure, we find the small initial SI does not reduce the system throughput.

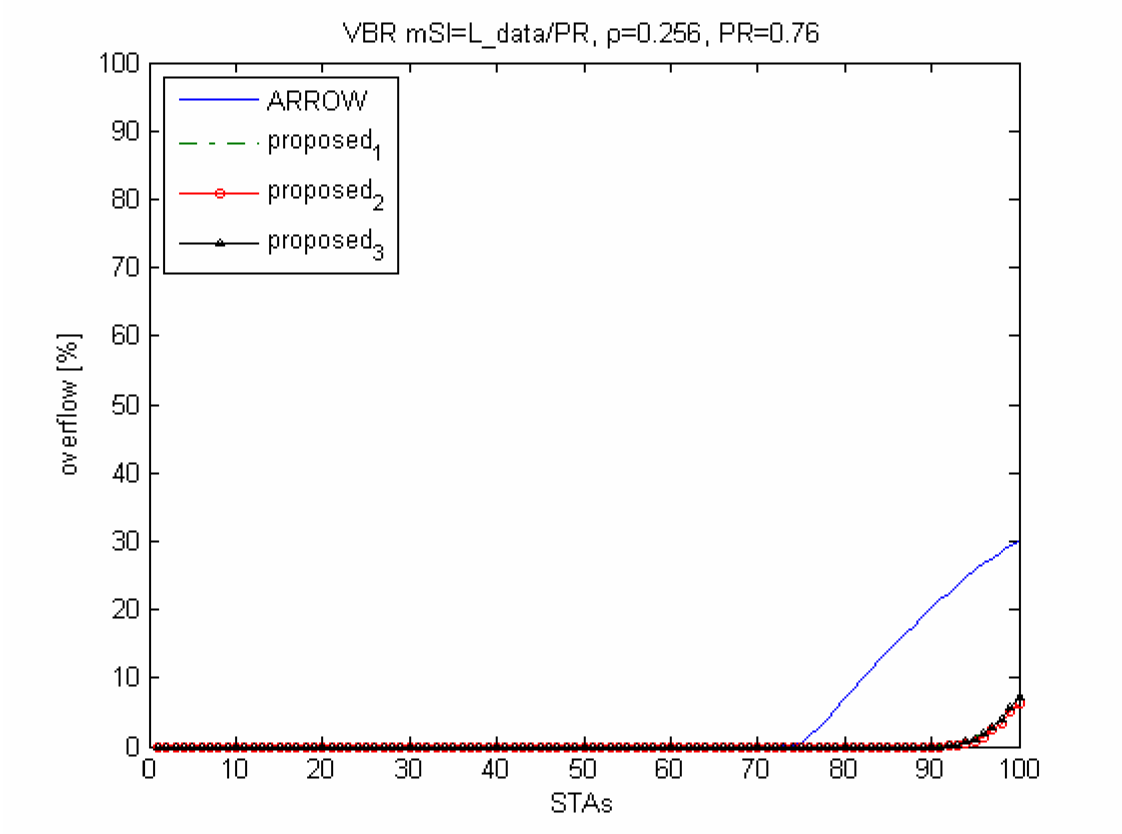

Figure 20: VBR2 overflow

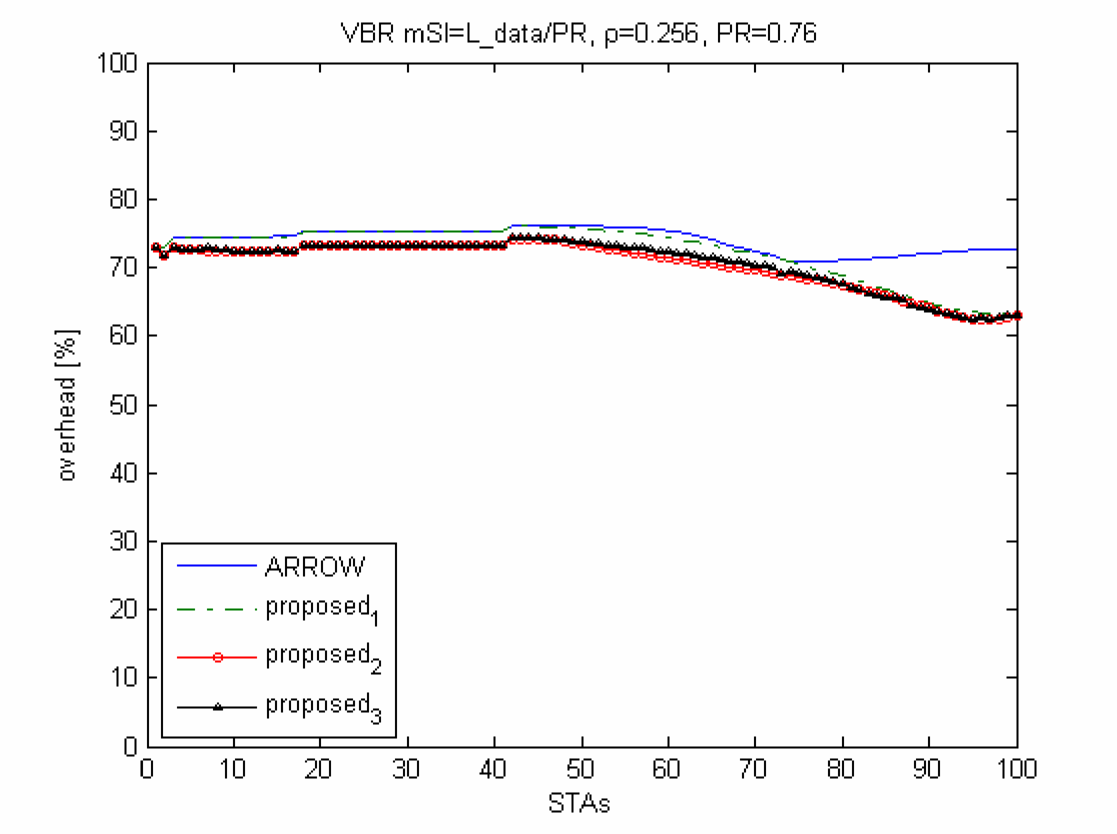

Figure 21: VBR2 overhead

The overhead of algorithm 3 is little higher than algorithm 2. Algorithm 3 may add more QSTAS in polling list easily. Every QSTA may be polled before it satisfied mSI requirement. This made more NULL-data. On the other words, we want to reduce the overhead of polling but the too earlier polling may cause much overhead of NULL data. The total effect is worse than algorithm 2

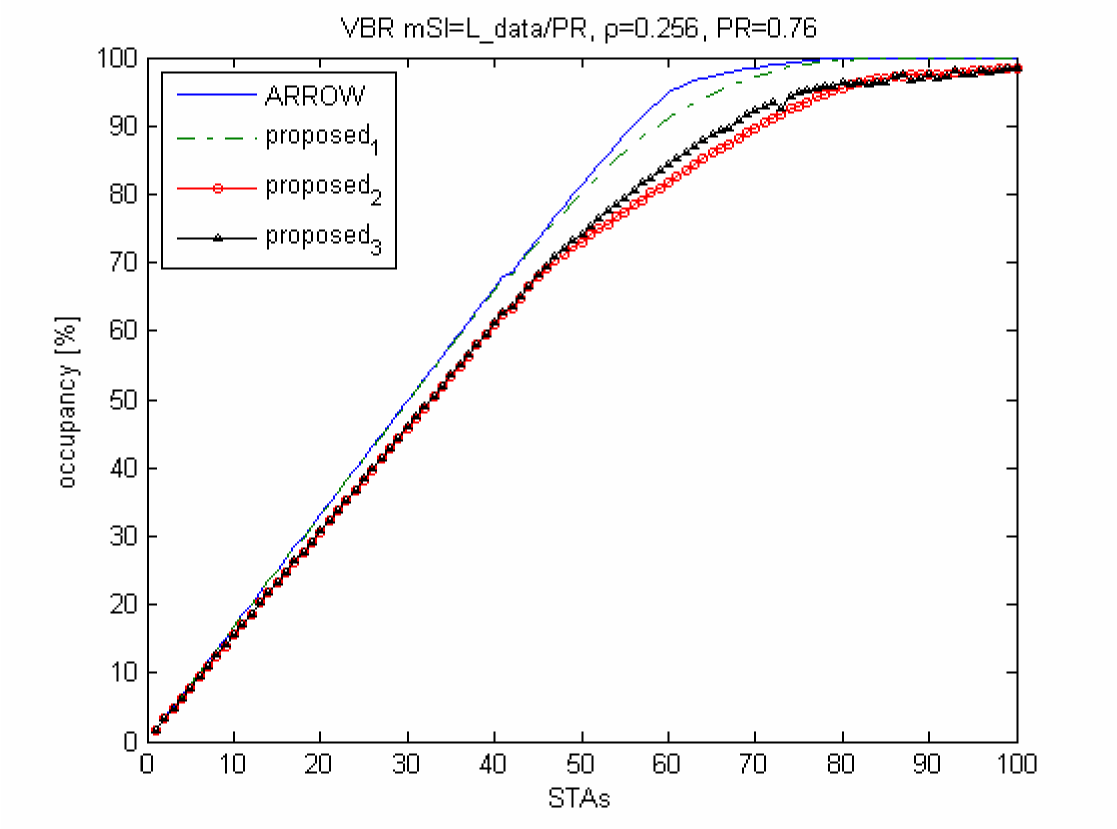

Figure 22: VBR2 occupancy

The occupancy does not be 100 % of algorithm 2 and 3 because the scheduler determined the polling list by the gap. If some QSTAs were not added in polling list, they must have bigger gap. The gap will create a little space let the channel be idle. So the two algorithms occupied less bandwidth. The occupancy will still be 100 percent when the number of QSTAs is very large.

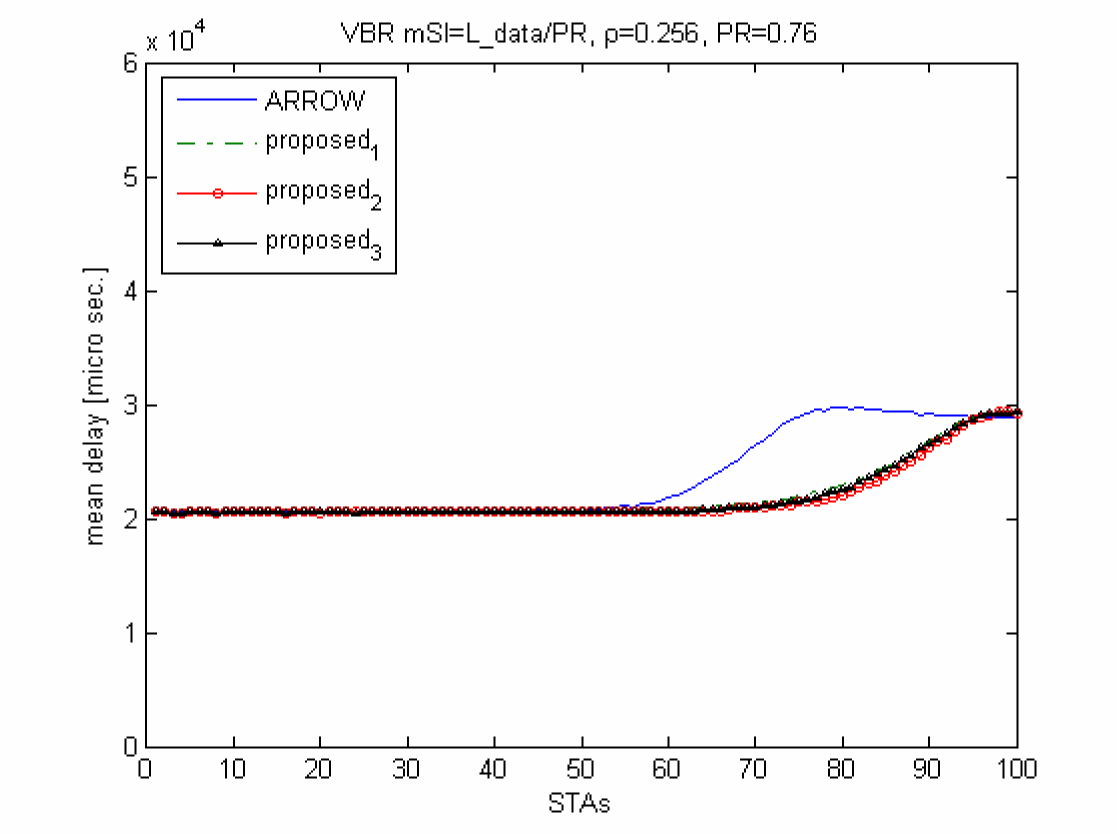

Figure 23: VBR2 mean delay

For VBR traffic under ARROW, every delay of a packet must between mSI and its delay bound. The mean delay is very consistent. When the number of STA is large and the SI is increased, the mead delay is also increased.

Chapter 5

Conclusions

We should put our goal on lower occupancy when the number of QSTAs is low so that the channel can be reuse by EDCA. The proposed algorithm 2 is the best of choice. The scheduler always multipolls all QSTAs when the number of QSTAs is very large. The performance is not different. We should focus on the number of QSTAs is about the number that the SI is just bigger than mSI but not exceed MSI. The algorithm is the best.

If we can design a good admission control system, there are less overhead of NULL data or polling, the performance will be maximize.

Bibliography

[1] IEEE Standard 802.11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, IEEE 802.11 Std., 2007.

[2] IEEE Std 802.11e/D13.0, "Draft supplement to standard for telecommunications and information exchange between systems-LAN/MAN specific requirements. Part 11: Wireless medium access control (MAC) and physical layer (PHY) specifications: Medium access control (MAC) enhancements for quality of service (QoS)," April. 2005.

[3] Dimitris Skyrianoglou, Nikos Passas and Apostolis K.Salkintzis, "ARROW: An Efficient Traffic Scheduling Algorithm for IEEE 802.11e HCCA", IEEE Transactions On Wireless Communications, VOL. 5, NO. 12, December 2006

[4] Byung-Seo Kim,Sung Won Kim, Yuguang Fang and Tan F. Wong, "Two-Step Multipolling MAC Protocol for Wireless LANs," IEEE Journal On Selected Areas In Communications, VOL. 23, NO. 6, June 2005

[5] Shou-Chih Lo, Guanling Lee and Wen-Tsuen Chen, "An Efficient Multipolling Mechanism for IEEE 802.11 Wireless LANs," IEEE Transactions On Computers, VOL. 52, NO. 6, June 2003

[6] Yang Xiao and Jon Rosdahl, "Throughput and Delay Limits of IEEE 802.11," IEEE Communications Letters, VOL. 6, NO. 8, August 2002

[7] IEEE 802.11a, Part 11: Wireless LAN, Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-Speed Physical Layer in the 5GHz Band, supplement to IEEE 802.11 Standard, Sept. 1999.

[8] Antonio Grilo, Mario Macedo and Mario Nunes, "A Scheduling Algorithm for

Qos Support in IEEE 802.11e Networks," IEEE Wireless Communications, June 2003