

國立交通大學

資訊科學與工程研究所

碩士論文

無線區域網路換手之適應性頻道掃描演算法



Adaptive Scanning Algorithms for Fast WLAN Handoffs

研究生：鄭雅聯

指導教授：曹孝櫟 教授

中華民國九十五年七月

無線區域網路換手之適應性頻道掃描演算法

Adaptive Scanning Algorithms for WLAN Handoffs

研究生：鄭雅聯

Student：Ya-Lien Cheng

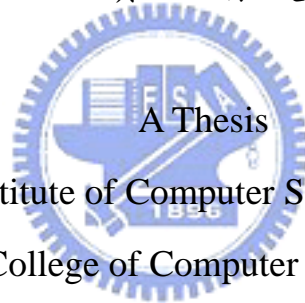
指導教授：曹孝櫟

Advisor：Shiao-Li Tsao

國立交通大學

資訊科學與工程研究所

碩士論文



Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

June 2006

Hsinchu, Taiwan, Republic of China

中華民國九十五年七月

# 無線區域網路換手之適應性頻道掃描演算法

學生：鄭雅聯

指導教授：曹孝櫟 教授

國立交通大學資訊科學與工程學系（研究所）碩士班

## 摘 要

近年來由於無線區域網路的廣泛佈建，已成為行動網路服務的一個重要傳輸媒介。然而使用者在存取點間換手動作造成的資料封包延遲及遺失將嚴重影響行動網路服務品質，這個問題對於即時音視訊服務而言更是不容忽視。已有相關研究指出通道掃描佔據大部分換手時間，造成資料封包延遲及遺失，並致力於縮短通道掃描時間。然而這些方法未能考慮即時視訊與語音封包，造成即時通訊的中斷和品質下降，為了改善這樣的問題，部分研究也提出在不影響即時通訊下的通道掃描機制，然而在現有的研究中，都未能同時考慮掃描時間或封包遺失或延遲來設計適當的通道掃描策略以適應使用者個別的需求。

有鑒於此，本論文提出了適應性的掃描機制，在滿足使用者對封包遺失及延遲要求的前提下，使整體掃描時間達到最短。由效能評估可以看出本論文提出的方法相較於現有技術而言有效縮短 30% 以上無線區域網路通道掃描時間(Channel Scan Time)，大幅降低換手動作對即時視訊與語音服務造成的干擾。

# Adaptive Scanning Algorithms for WLAN Handoffs

Student : Ya-Lien Cheng

Advisors : Dr. Shiao-Li Tsao

Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

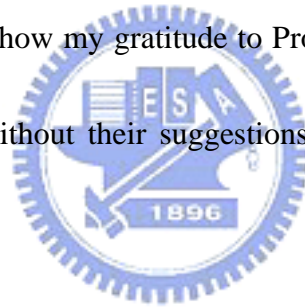
## ABSTRACT

Recently, WLAN has been widely deployed over public and private areas and has become one of popular access technologies for mobile Internet services. Handoffs between WLAN access points (APs) that introduce packet loss and delay become one of the critical issues for mobile Internet applications, especially for real-time communications. Previous studies have already indicated that WLAN AP scanning time contributes a significant portion of a handoff delay, and a number of solutions were proposed to address this issue. Unfortunately, the existing approaches either try to minimize the time period of an active scan process, or optimize packet loss and jitter by using passive scan approaches. Solutions considering packet loss, jitter, delay, and also scan latency are not yet investigated. In this thesis, novel scan algorithms that combine active and passive scan strategies for WLAN smooth handoffs are proposed to minimize the scan time without violating the QoS requirements such as packet delay, jitter, and loss for applications. Simulation results demonstrate that the proposed scan strategies achieve faster scan time without introducing packet jitter and delay than the existing solutions.

## Acknowledgement

I would like to express my sincere gratitude and thanks to my advisor, Professor Shiao-Li Tsao for giving me the opportunity to work in the research area of *Broadband Radio Access and Software*, for his professional guidance, and for his encouragement and support in all respects. I would also like to thank Professors Ming-Feng Chang, Ray-Guang Cheng, and Jen-Yi Pan for reading this thesis and offering insightful and constructive comments.

I would like to thank my dear family and friends for all their continuous support, encouragement, and assistance in my life. Without their love this work would never have been possible. I would also like to show my gratitude to Professor Ying-Ping Chen for his helpful comments and discussions. Without their suggestions, I could not do well in research and finish my thesis.



鄭雅聯 誌於

國立交通大學資訊科學與工程研究所

中華民國九十五年七月

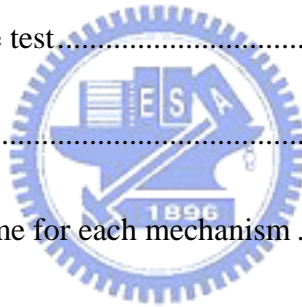
# Contents

<b>I. INTRODUCTION</b> .....	- 1 -
<b>II. BACKGROUND AND RELATED WORKS</b> .....	- 4 -
<b>III. PROBLEM STATEMENTS AND MODELS</b> .....	- 10 -
<b>Environment Information</b> .....	- 10 -
<b>Consider Voice Packets</b> .....	- 11 -
<b>Expected Result</b> .....	- 13 -
<b>IV. ASA ALGORITHMS</b> .....	- 15 -
<b>A. Insert a Passive Scan Channel to the Scheduled List</b> .....	- 15 -
<b>B. The OPTIMAL ALGORITHM</b> .....	- 17 -
<b>Find All Possible Solutions</b> .....	- 17 -
<b>Use Dynamic Programming to Shorten the Search Path</b> .....	- 18 -
<b>The Algorithm</b> .....	- 19 -
<b>C. A HEURISTIC ALGORITHM</b> .....	- 21 -
<b>Determine the Candidates for Passive Scan to Get More Benefit</b> .....	- 21 -
<b>Adjusting After Scheduling</b> .....	- 22 -
<b>The Algorithm</b> .....	- 23 -
<b>Simulation Settings</b> .....	- 26 -
<b>Performance Evaluation</b> .....	- 27 -



## Figure List

Fig. 1. Standard IEEE802.11 handoff procedures [9].....	- 5 -
Fig. 2. Handoff latencies of NICs from different vendors [9].....	- 5 -
Fig. 3. The bottleneck of a passive scan .....	- 8 -
Fig. 4 . Environment information: reserve the beacons before deadline.....	- 11 -
Fig. 5. Insert a beacon slot into the scheduled list.....	- 12 -
Fig. 6 . Expected result: a non-colliding time slot list.....	- 14 -
Fig. 7. Dynamic programming using temp result nodes .....	- 19 -
Fig. 8. Occupied time test.....	- 22 -
Fig. 9. Adjustment .....	- 23 -
Fig. 10. Total scan time for each mechanism .....	- 28 -
Fig. 11. Enhancement comparing to active scan under maxDelay=20ms.....	- 29 -
Fig. 12. Delay distribution of each mechanism under maxDelay=20ms .....	- 30 -
Fig. 13. Execution time for scheduling .....	- 31 -





## Table List

Table 1. Latencies of handoff procedures.....	- 4 -
Table 2. The comparison of the standard scanning mechanisms.....	- 7 -
Table 3. Find all possible solutions .....	- 18 -



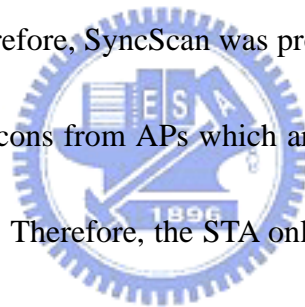
## I. INTRODUCTION

The development of 802.11 protocols and systems offers new opportunities for mobile communications, services and applications. Among all kinds of applications and services over WLANs, real-time communications have attracted a great interesting from both academia and industries. However, WLAN whose indoor and outdoor coverage areas are normally 35 to 100 meters and 100 to 300 meters introduces frequent handoffs between APs for moving users. A handoff that blocks an active connection for hundreds of milliseconds to several seconds introduces serious packet delay, jitter and loss. The handoff latency significantly influences the qualities of the real-time streaming applications and voice communications which can tolerate an end-to-end delay less than 150 milliseconds [15]. Hence, minimizing handoff delay becomes one of the important research issues for WLAN technologies.



Previous studies have investigated the delays for handoff procedures and concluded that the AP scanning phase contributes a considerable portion of the handoff delay [9]. To scan WLAN APs in order to obtain signal strengths from APs before a handoff, the specification defines two scan strategies, i.e. passive scan and active scan. By applying passive scan, a station (STA) scans the channels by listening beacons from all access points (APs). Since an STA doesn't know the arrival time of beacons, the STA typically has to stay on a channel and waits for the whole beacon interval which is normally 100 ms. Generally speaking, the passive scan strategy introduces long channel scanning time. On the other hand, to avoid the

long channel scanning time, an STA can apply active scan, sends probe requests actively to all APs on a channel and obtains the signal strengths from APs through response messages. Although the channel scanning time is reduced, an active scan may block connections during a scan procedure. Several studies have been worked on the reduction of the active scan time. For example, an STA can learn from the environment by the historical data or via pre-configurations [13] and then the STA uses the cache or neighbor information to eliminate the channel scanning time on empty channels [13]. Although the scanning time is reduced, they do not consider the packet delay for the current connections and qualities of real-time connections may degrade. Therefore, SyncScan was proposed by [10] based on a passive scan strategy. An STA can hear beacons from APs which are synchronized and broadcast beacons at the scheduled time intervals. Therefore, the STA only needs to switch to different channels at proper time intervals, and obtains the signal strengths from APs. Without staying on the channel and waiting for probe response messages, the approach greatly reduces the packet loss and delay for real-time connections. However, this mechanism requires that all APs to be synchronized and broadcast their beacons closely. This approach might not be able to apply to the existing WLAN infrastructures which are already deployed. Moreover, the beacon arrival time of APs on different channels may collide and an STA may have to wait for next beacons which are normally 100 ms away. In this thesis, a new scanning mechanism which combines active scan and passive scan concept is proposed. The combined scan strategies not only



consider the delay requirements of real-time connections but also minimize the total WLAN AP scanning time. Both the optimal solution and a heuristic algorithm are proposed in this work.

The rest of the thesis is organized as follows. In Section II, the standard 802.11 handoff procedures are introduced. Section III formalizes and models the problems. Section IV presents the combined active and passive scan strategies, including the optimal solution and a heuristic algorithm. Performance evaluations are discussed in Section V, and finally Section VI concludes this work.



## II. BACKGROUND AND RELATED WORKS

A WLAN handoff composes of three phases, as shown in Fig. 1. The scanning phase discovers the APs that an STA can hear and measures the signal strengths from the APs. The re-authentication phase verifies the access rights of an STA to a specific AP. Finally, the re-association phase negotiates with the target access point and re-establishes the connection [11]. Table 1 shows the latency of each phase. It can be seen that the latter two phases spend less than 10 ms while the scanning phase contributes the major part about 90% of the total handoff latency. The same observations can be made from different implementations of wireless network interface cards, as shown in Fig. 2.

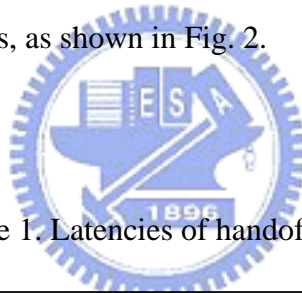


Table 1. Latencies of handoff procedures

PHASE	TIME
Scanning	350-500ms
Authentication	<10ms
Association	<10ms

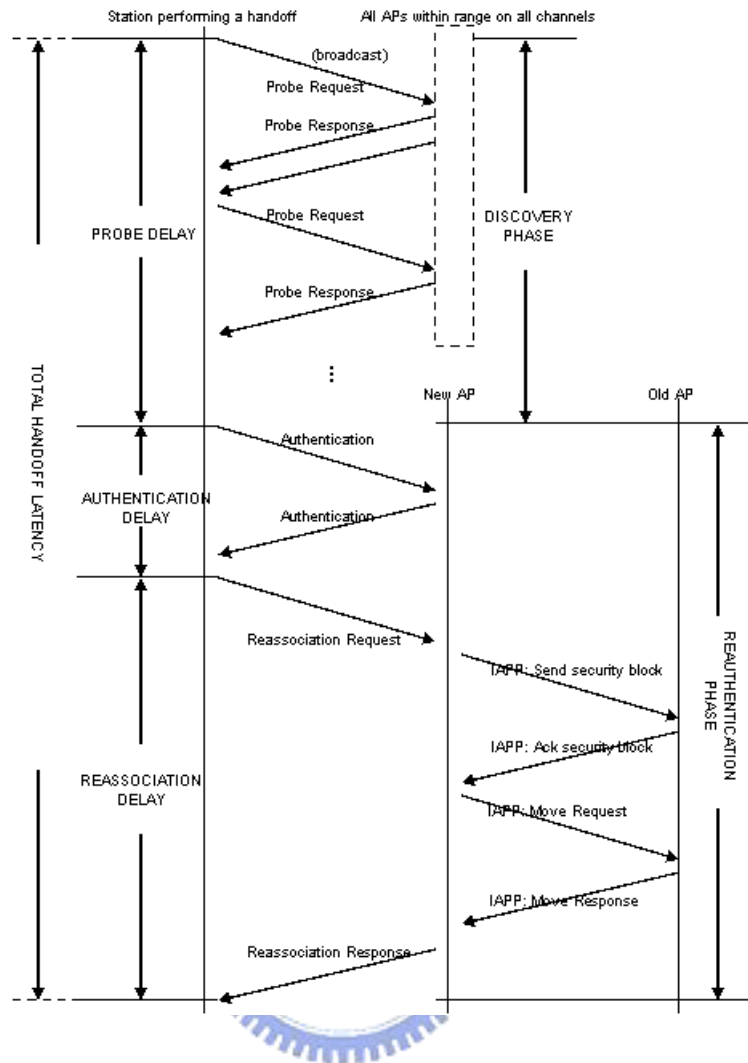


Fig. 1. Standard IEEE802.11 handoff procedures [9]

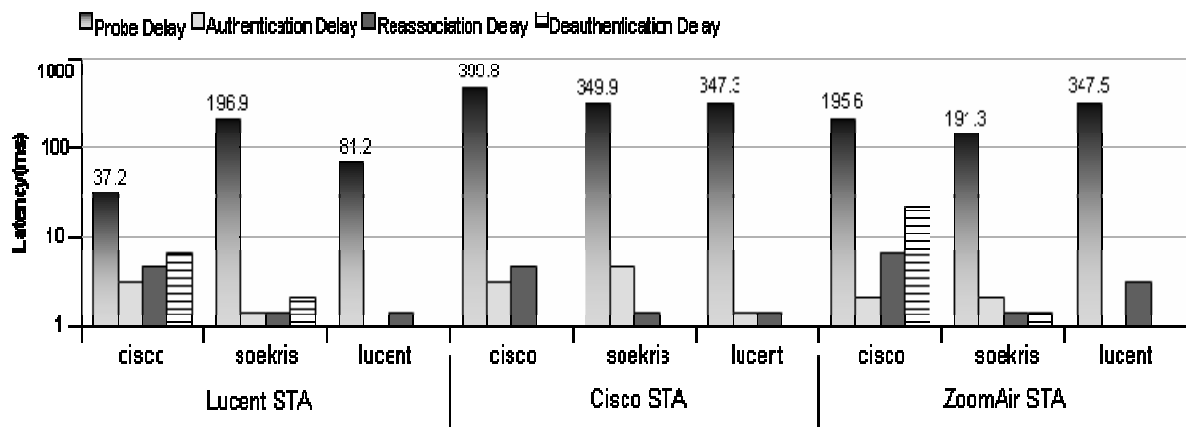


Fig. 2. Handoff latencies of NICs from different vendors [9]

The IEEE 802.11 specifies two scanning mechanisms, i.e. active and passive scan. While performing an active scan, an STA sends probe requests and waits for the responses from APs on each channel. For a passive scan, an STA has to wait for certain amount of time to receive beacons from all APs on a channel, and then the STA continues with another channel. Table 2 compares these two kinds of mechanisms. The passive scanning mechanism suffers from a major defect that it has to tie in with the exact time when the APs send the beacons. This characteristic results in a long waiting time on a single channel. The active scanning mechanism does not require staying in a channel for that long. An STA sends a request whenever it needs to update the environment information. However, since the active scanning mechanism stays on each channel with APs for the maximum channel time, it may waste too much time on channels with only few APs. To overcome this drawback, previous works learn from the environment, intelligently ignore the empty channels [13] and complete the scan right after all messages from the expected APs' responses are successfully received [13]. These enhancements based on the active scan reduces the channel waiting time, however, they do not consider the requirements such as packet delay for the connection. The QoS requirements of the connection may not be satisfied if the WLAN scan takes too long time to complete.

Based on the above observation, some enhanced solutions proposed to take advantage of

the beacon regularity in order to avoid overheads and reduce the channel waiting time [10].

The idea is to receiving AP beacons which are predictable and to eliminate the channel occupied time. Then, an STA can get the background information without the overheads caused by probe requests and responses in the active scan. The difference can be seen in an ideal scenario as follows. To investigate the information of a single AP, the standard active scan takes (R+M) milliseconds, in which R stands for the request transmission time, 1 ms for example, and M stands for the predefined maximum channel time, typically 11 ms. Therefore, the latency of the active scan for a single AP is about 12 ms. However, listening to a beacon for one AP requires only a packet receiving time which is less than 1ms.



Table 2. The comparison of the standard scanning mechanisms

	Min	Max	
Active Scan	1ms*10=10ms	11ms*10=110ms	<b>Typical:</b> minChTime=1 ms MaxChTime=11 ms
Passive Scan	in seconds	in seconds	<b>Typical:</b> 100ms*10=1000ms

Although the existing work improves the scanning latencies, they have to obey the beacons from APs, which leads to the bottleneck and limits the enhancement. The more APs which need to be scan are, the higher possibility the beacons collide. If beacons collide with each other, the performance is downgraded. As shown in Fig. 3, because of a beacon collision



for AP<sub>1</sub> and AP<sub>2</sub>, an STA cannot process AP<sub>2</sub> (the dot-line block) after finishing AP<sub>1</sub> scan. The STA has to wait for another beacon interval to get the next beacon (the solid block). The performance downgrades significantly due to a long beacon interval. The total scan time may become double as the latency in the non-collision case. SyncScan [10] solves this problem by asking the APs on the same channel to send the beacons at the same time  $t$  (or very close to it), and APs on the next channel to send the beacons at time  $(t+d)$ , and so on. The beacon arrival time is staggered to avoid collisions between channels. However, the APs on the same channel broadcast their beacons at a very close time, and STAs are not guaranteed to receive beacons from all APs successfully in the first try. Moreover, this approach requires the coordination between APs and may have deployment problem.

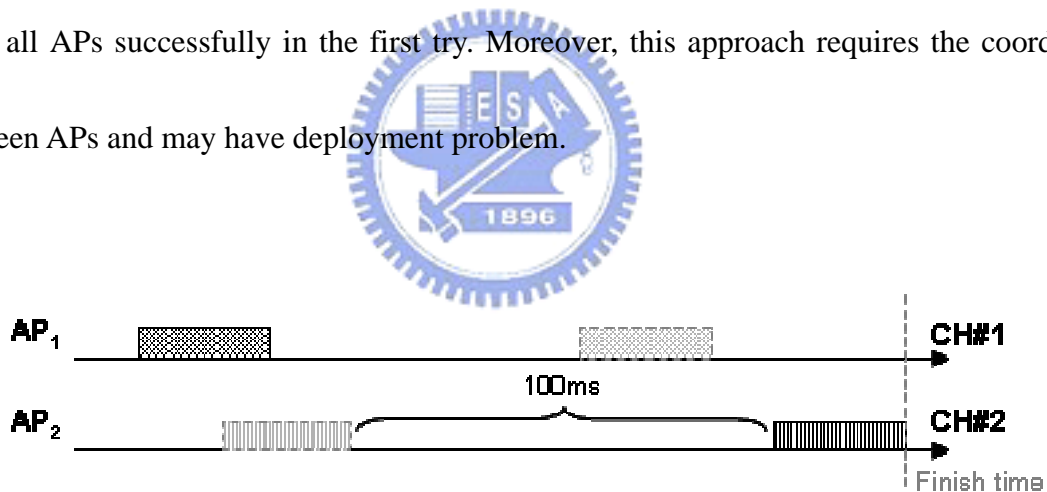


Fig. 3. The bottleneck of a passive scan

All the previous works mentioned above have good performance only in some specific cases. The active scan mechanisms may spend long channel waiting time for hotspots with low AP density. On the other hand, passive scan mechanism may suffer from serious beacon conflicts in hotspots with high AP density. Moreover, the existing methods do not consider the

various delay constraints of real-time services. User applications send the data packets (ex. voice or video) regularly and tolerate various ranges of delays. In this thesis, a novel mechanism which minimizes the channel waiting time by the passive scan approach and resolves the beacon conflicts by using the active scan strategy is proposed. Moreover, the proposed mechanisms consider both handoff delay and various QoS requirements such as delay and jitter constraints for different applications.



### III. PROBLEM STATEMENTS AND MODELS

Under pre-configured environment information and pre-defined delay constraints, a scan schedule which arranges the scan type for each channel needs to be done in order to reduce the total scan time. The environment information includes the layout and the beacon arrival time of each AP in the background. Users define the delay constraints which limit the maximum latency between the expected and the actual packet arrival time. A scheduled time slot are arranged to control the NIC driver and determine when to listen to a beacon, to perform a single-channel active scan, or to receive a voice packet. The detail models for the problem are defined as follows.



#### Environment Information

Through a cold-start procedure or pre-configured knowledge, an STA keeps the environment information, including that how many APs are in the background and when each beacon arrives.

Taking advantages of the regularity of the beacons, an STA calculates and predicts the arrival time of the next beacons. The background information can be modeled as:

$$\text{Environment} = \text{ENV} = \{\text{CH}_1, \text{CH}_2, \dots, \text{CH}_i, \dots, \text{CH}_{11} \mid \text{CH}_i: \text{the } i\text{th Channel}\} \dots(1)$$

$$\text{Channel} = \text{CH} = \{\text{B}_1, \text{B}_2, \dots, \text{B}_i, \dots, \text{B}_n \mid \text{B}_i: \text{the } i\text{th Beacon}\} \dots(2)$$

$$\text{Beacon} = \text{B} = (\text{ApId}, \text{T}_{\text{arri}}); \text{ApId: the identity of an AP; T}_{\text{arri}}: \text{beacon arrival time} \dots(3)$$

The environment information contains 11 channel elements. Each channel element lists the beacons arriving before the deadline. A deadline is set as the worst-case scan time, which is defined as the minimal turn-around time needed to perform a modified active scan without breaking the constraints of packet delay.

$T_{\text{worst}}$ : turn-around time of a delay-constrained active scan .....( 4 )

Notice that there may be more than one beacons belonging to a specified AP in the beacon list. Each beacon indicates the AP identification and the beacon arrival time. Fig. 4 illustrates an example of background information.

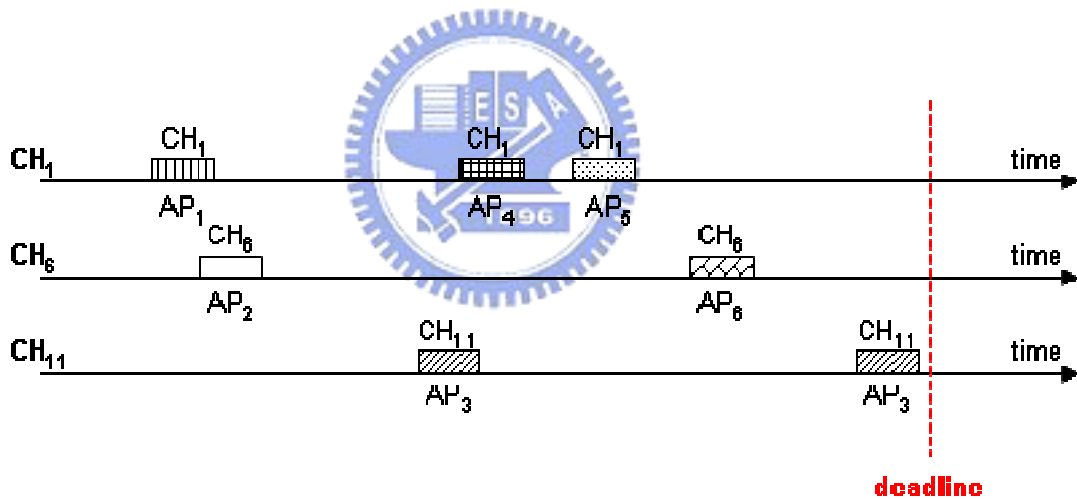


Fig. 4 . Environment information: reserve the beacons before deadline

### Consider Voice Packets

The problem model then considers the voice delay constraints. Users or applications define a maximum amount of delay for each voice packet according to the service property and the jitter buffer size. Voice packets are inserted into the scheduled list before scheduling.

After that, each channel is considered whether an active scan or a passive scan should be performed to result in the better performance which finishes the scan for all channels before  $T_{\text{worst}}$ . A scanning slot is then created according to the decided scan type and arranged to the scheduled list under the restrictions of all other scanning slots and the maximum voice delay.

$$\text{Voice} = V = (T_s, T_e, \text{Delay}); T_s: \text{start time}; T_e: \text{end time}; \text{Delay: delay amount which} \leq \text{maxDelay} \dots\dots(5)$$

Fig. 5 describes the procedure to insert a scanning slot into the scheduled list. A time slot for beacon-listening (i.e. a beacon slot) of AP<sub>3</sub> is ready to be inserted. Since the target slot has to be inserted without any collision, the scheduler detects and tries to eliminate the collisions with the existing time slots. After scheduling the new beacon slot for AP<sub>3</sub> is inserted to the scheduled list meanwhile meets user requirements for voice delay.

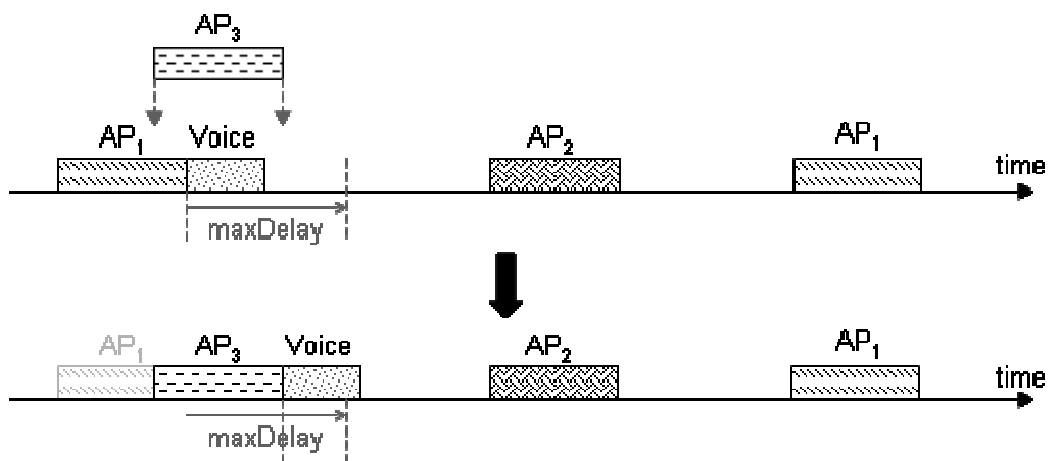


Fig. 5. Insert a beacon slot into the scheduled list

## Expected Result

After scheduling, the final result is generated as a scheduled list, which scans all APs on the channels and the delay amount of each voice packet is no more than the value pre-defined by users or applications. Meanwhile, the final result has a turn-around time less than  $T_{\text{worst}}$  and thus improves the scanning performance.

The content of the scheduled result is defined in (6)-(10).

**ScheduledList = SL = {TS<sub>1</sub>, TS<sub>2</sub>, ..., TS<sub>i</sub>, ..., TS<sub>m</sub> | TS<sub>i</sub>: the *i*th TimeSlot} .....( 6 )**

**TimeSlot = TS = (Ts, Te, CH, Type, Delay, ApId) .....( 7 )**

**Type = SCAN\_ACT | SCAN\_PAS | SCAN\_ORI | SCAN\_UNDEFINED .....( 8 )**

**Delay = -1 | 0 | positive integer .....( 9 )**

**ApId = the AP identity for a beacon slot | -1 for voice and active scan slots .....( 10 )**

The scheduled list consists of non-colliding time slots, which includes the active scan slots (typed as SCAN\_ACT), the beacon receiving slots (SCAN\_PAS), and the voice slots on the original channel (SCAN\_ORI). Each time slot indicates the start and end time, the scanning channel, the scanning type, and the amount of delay marked as -1 for beacon slots which can't be delayed, zero for the active-scan slots and non-negative integers for the voice slots. This list is then used to control the NIC driver, arrange the scanning phase and update the environment information under the constraints of voice-maximum-delay.

An example of the final result is illustrated in Fig. 6.

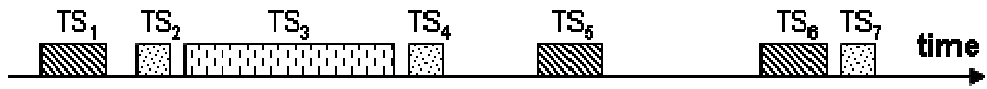


Fig. 6 . Expected result: a non-colliding time slot list



## IV. ASA ALGORITHMS

### A. Insert a Passive Scan Channel to the Scheduled List

To meet the user-defined voice delay constraints, periodical voice slots are inserted to the scheduled list in advance before scheduling. Beacon slots are then created and inserted successfully if there is no collision in the scheduled list. Only when all necessary beacons on a channel can be inserted without colliding with any voice packet and existing beacon slot on the schedule, it is said that a passive scan channel is inserted to the scheduled list successfully. By performing the insertion test, the target channel can be arranged and collisions can be avoided.

The proposed insertion test algorithm is described as follows.



REMOVE\_COLLISION\_LIST(CL, SL)

Input. CL: collision list to be delayed or deleted

SL: the existing scheduled list

Output: SL: the modified scheduled list

RCODE: result code, which can be SUCCESS or FAILURE

```
1 Backup SL;
2 for each  $TS_i$  in CL
3     do
4     if  $TS_i$  is a voice slot:
5         then
6             Try to delay  $TS_i$ ;
7             if  $TS_i$ .Delay is greater than maxDelay
8                 then
9                     Recover SL;
10                    Return FAILURE;
11        else
12            Try to delete  $TS_i$ ;
13            if  $TS_i$  can be discarded
```



```

14         then
15             Delete TSi;
16         else
17             Recover SL;
18             Return FAILURE;
19     end
20 Return SUCCESS and SL;

```

#### INSERT\_TO\_SL(CH, SL)

Input. CH: the channel to be inserted  
 SL: the existing scheduled list

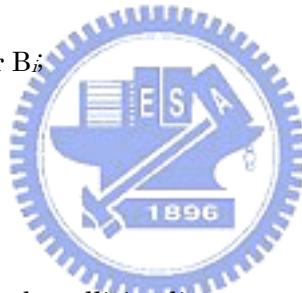
Output. SL: the modified scheduled list

RCODE: result code, which can be SUCCESS or FAILURE

```

1 Backup SL;
2 for each Bi in CH
3     do
4         Create a new TS NewTS for Bi;
5         for each TSj in SL
6             do
7                 if TSj collides with Bi
8                     then
9                         Insert TSj to the collision-list;
10            end
11            Call REMOVE_COLLISION_LIST to delete or delay the elements in the collision list;
12            if FAILURE
13                then
14                    if Bi can not be discarded:
15                        then
16                            Recover SL;
17                            Return ERROR;
18                    else
19                        Insert NewTS to SL;
20            end
21 Return SUCCESS;

```



The insertion function attempts to insert each beacon on the target channel ( $CH_{\text{target}}$ ) into

the scheduled list. First a collision list is built which collects all time slots colliding with the target beacon. Then it tries to remove all time slots in the collision list by either delaying or deleting them in order to insert the target beacon slot. If removing the collision list is failed, the insertion function checks if the target beacon can be ignored. A beacon can be ignored only when there are other beacons in  $CH_{target}$  which belong to the same AP. Otherwise an error occurs. After checking all beacons without any error, the target channel is said successful to perform passive scan. Moreover, because only the beacons before  $T_{worst}$  are reserved in each channel, this method makes sure to get a shorter scan time than  $T_{worst}$  when performing passive scan on a channel.



## B. The OPTIMAL ALGORITHM

### Find All Possible Solutions

Since active scan can be scheduled at any time, a possible solution can be seen as a set of passive-scan channels.

$$\text{PossibleSolution} = \text{PS} = \{\text{ChId}_1, \text{ChId}_2, \dots, \text{ChId}_i, \dots, \text{ChId}_k \mid \text{ChId: channel identity}\} \dots\dots(11)$$

Channels which are not in this set perform active scan. The combination of two standard scanning mechanisms improves the performance. Table 3 derives all possible solutions and thus finds the best solution to minimize the total scan time.

Table 3. Find all possible solutions

Num <sub>CH</sub>	Num <sub>PSS</sub>	Possible Solutions
1	1	{1}
2	2	{2}, {1,2}
3	4	{3}, {1,3}, {2,3}, {1,2,3}
4	8	{4}, {1,4}...{3,4}, {1,2,4}...{2,3,4}, {1,2,3,4}
5	16	{5}, {1,5}...{4,5}, {1,2,5}...{3,4,5}, ..., {1,2,3,4,5}
i	2 <sup>i-1</sup>	

Through the number of channels increases, the number of possible solutions has exponential growth as shown in Table 3.



**Use Dynamic Programming to Shorten the Search Path**

New possible solutions can be derived based on the previous ones, as shown in Table 3. For example, only when channel 1 and 3 are allowed to perform passive scan without collision, channel 1, 3, and 4 are possible to perform passive scan simultaneously. Thus, the solution {1,3,4} is possible only when the solution {1,3} exists. The dynamic programming approach takes advantage of the experience and simplifies the generating path.

The following data structure is used to implement the dynamic programming algorithm.

**TmpRes = (ChId, pTmpRes) .....( 12 )**

**pTmpRes: pointer to the previous TmpRes .....( 13 )**

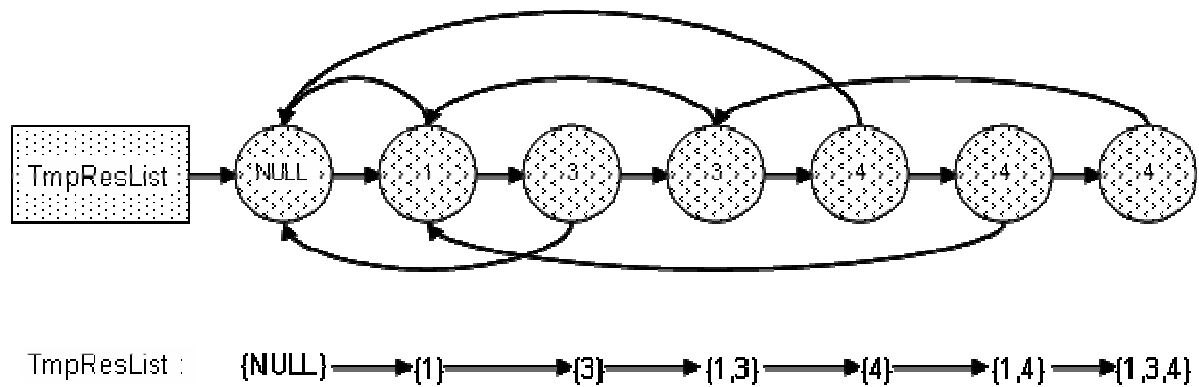


Fig. 7. Dynamic programming using temp result nodes

Fig. 7 illustrates an example list of possible solutions. In this example, seven possible solutions are derived. Then the best solution can be found from these possible solutions.



### The Algorithm

The optimal algorithm is described as follows.

ASA\_OPTIMAL(ENV)

Input. ENV: environment information which contains channel elements

Output. SL: the best scheduled list which has the shortest scan time

ScanTime: total scan time of the best SL

- 1 Create a new TmpRes, assign the channel identify as NULL, and insert it to TmpResList;
- 2 for each  $CH_i$  in ENV
- 3     do
- 4     for each TmpRes $_j$  in TmpResList
- 5         do
- 6             Backup TmpRes $_j$ .SL;
- 7             Call INSERT\_TO\_SL to insert  $CH_i$  to TmpRes $_j$ .SL;
- 8             if SUCCESS
- 9                 then
- 10                     Create a new TmpRes NewTmpRes;

```

11             Copy TmpResj.SL to NewTmpRes.SL;
12             Point NewTmpRes to TmpResj, and insert NewTmpRes to TmpResList;
13             Recover TmpResj.SL
14         else
15             Mark CHi as SCAN_ACT;
16     end
17 end
18 for each TmpResj in TmpResList
19     do
20         Remove redundant beacon slots to keep each AP scanned only once;
21         Add active-scan slots for all SCAN_ACT channels;
22         Calculate ScanTime according to TmpResj.SL;
23         Keep the best result;
24     end
25 Return the best SL and ScanTime;

```

By building the temporary result list, all possible solutions are derived through a simplified path. Some unfeasible solutions are not considered to reduce the execution time. Before calculating ScanTime for each possible solution, the redundant beacons which are from the same APs have to be eliminated. Thus the information from each AP in the background is updated just once. The other channels which are not scheduled as passive scan perform the active scanning mechanism. The active-scan slots are then created and inserted into the scheduled list to complete the scheduling. Finally the scan time of all the possible solutions is calculated and the best solution is found.

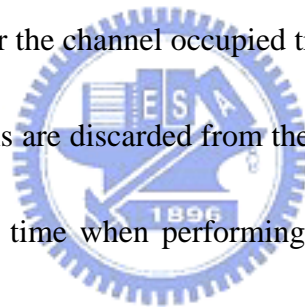
Although the dynamic programming algorithm simplified the solution-deriving process, it still has a time complexity of  $O(2^{n_{CH}})$ . When there are too many non-empty channels in the background, it may be impractical to perform the optimal algorithm due to the long

scheduling time and the performance of the scanning phase will be seriously damaged.

### **C. A HEURISTIC ALGORITHM**

#### **Determine the Candidates for Passive Scan to Get More Benefit**

To simplify the derivation of possible solutions, some channels are selected as the candidates for passive scan. That is, the channels which few APs are configured on are selected first, thus only few beacons need to be heard. It is said that a channel has a short channel occupied time when STAs only need to wait on that channel for a few time for scanning. Therefore, the shorter the channel occupied time is, the fewer influence it has on the real-time services. The channels are discarded from the candidate list when they have channel occupied time longer than the time when performing a normal active scan. An example is illustrated in Fig. 8. A channel with too many APs may have a channel occupied time longer than a single-channel active scan time and will be discarded from the candidate list. The occupied time test reduces the number of potential passive-scan channels which have more possibility to satisfy the delay requirements.



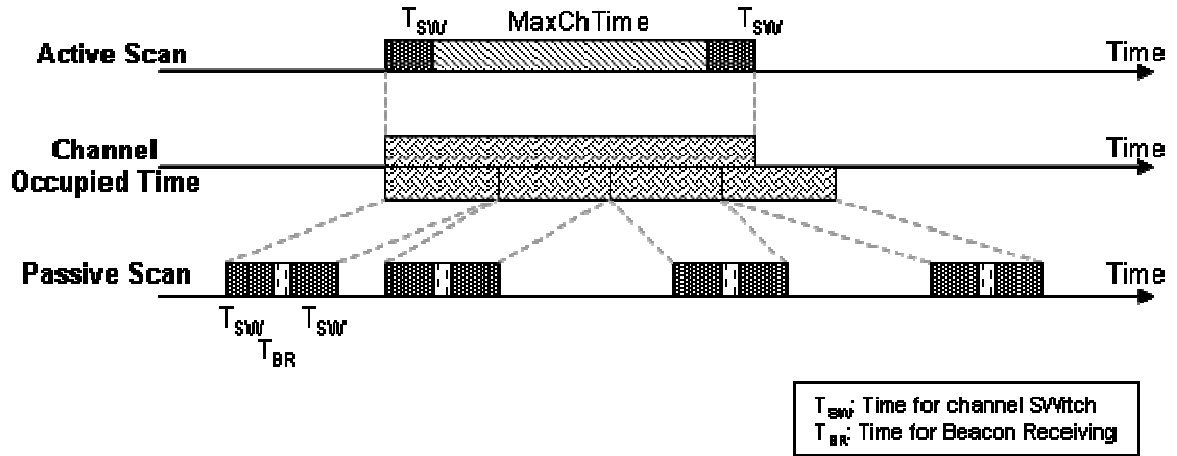


Fig. 8. Occupied time test

The selected candidates are then sorted by the number of APs. Thus the channels with fewer channel occupied time are scheduled first to reduce the influence on real-time services and also the potential collisions with other potential channels.



### Adjusting After Scheduling

After deciding the scanning type of each channel, an adjusting function is done to further improve the scan performance. The adjusting function checks the scheduled list and tries to replace the passive-scan slots of a channel with a single active scan slot to shorten the total turnaround time. Fig. 9 presents the improvement of replacing the passive-scan slots of channel 4 with a single active scan slot. The adjusting function checks the time slots in the scheduled list from tail to head until the time slot checked cannot be replaced by any other earlier time slot. Through this mechanism, the resource utilization is improved during the

scanning time while the delay constraints are satisfied. The total turnaround time is shortened as well.

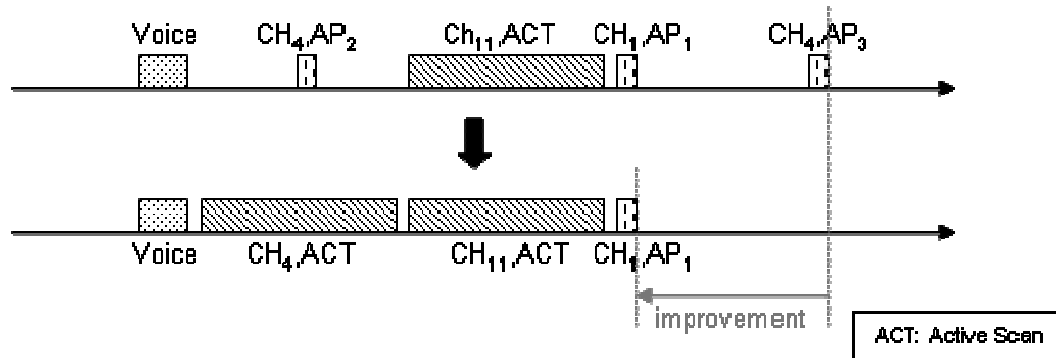


Fig. 9. Adjustment



### The Algorithm

The function INSERT\_TO\_SL mentioned above is used to insert a passive-scan channel into the scheduled list. The detail algorithm is described as follows.

#### BUILD\_CANDIDATE\_LIST(ENV)

Input. ENV: environment information which contains channel elements

Output. CandiList: candidate list of potential passive-scan channels

- 1 for each  $CH_i$  in ENV
- 2 do
- 3 Calculate  $T_{act}$ , which is the minimal turnaround time to finish a standard active scan on  $CH_i$
- 4 under delay constraints;
- 5 Calculate the channel occupied time for  $CH_i$ ;
- 6 if the channel occupied time of  $CH_i$  is shorter than  $T_{act}$
- 7 then
- 8 Insert  $CH_i$  to CandiList and sort the list by the number of APs;
- 9 end
- 10 Return CandiList;



ASA\_FAF(ENV)

Input. ENV: environment information which contains channel elements

Output. SL: the result scheduled list

ScanTime: the total turn-around time for scanning

```
1 Call BUILD_CANDIDATE_LIST to list the potential passive-scan channels which are sorted
2 according to the number of APs;
3 for each CHi in CandiList
4     do
5         Call INSERT_TO_SL to insert CHi to the scheduled list;
6         if FAILURE
7             then
8                 Mark CHi as SCAN_ACT;
9     end
10 Remove redundant beacon slots to keep each AP scanned only once;
11 Insert the active slot to SL for each channel marked as SCAN_ACT;
12 Call ADJUST to further improve the final result;
13 Calculate ScanTime as the total turnaround time according to SL;
14 Return SL and ScanTime;
```



First a candidate channel list is built. Channels in this list have shorter channel-occupied time when performing passive scan rather than active scan. The less time-occupied channels are scheduled first, thus the candidates are sorted according to their number of APs, which is from fewest to most. Then the candidates are checked one by one, until all of them are scheduled. The channels failed to join the scheduled list with passive scan are marked as active scan. Since there may be more than one beacon slots from a single AP are inserted into the scheduled list, the algorithm checks the time slots and removes the redundancies. Then active-scan slots are inserted for the other un-scheduled channels. After all channels are scheduled, the adjusting function is called. The scheduled result is modified to get better

resource utilization and time performance.

This heuristic solution has a time complexity of  $O(N_{\text{beacons}}^2)$ , which is much lower than the time complexity of the optimal solution.



## V. SIMULATION AND EVALUATION

### Simulation Settings

In this section, six scanning mechanisms, including two standard scanning mechanisms, two modified mechanisms as mentioned in the related work section, and the proposed ASA algorithms are analyzed. The modified mechanisms are named preactive active scan (preACT) and preactive passive scan (prePAS), which take advantage of the environment information to reduce the channel-waiting time. In the preACT mechanism, the NIC skips the empty channels rather than waits until minimum channel time. In the prePAS mechanism, the NIC knows the beacon arrival time of each AP and hears the beacons exactly whenever they arrive. Both of these two modified mechanisms satisfy the user-defined delay constraint. The optimal and heuristic ASA algorithms are implemented and can be used directly in the driver to arrange the scanning phase.

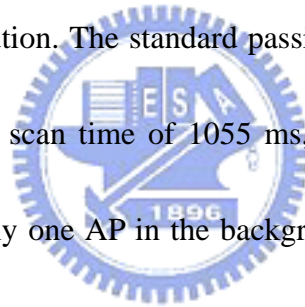
A virtual background environment is generated. The number of total APs from one to ten are randomly distributed over 11 channels. Each AP has the first beacon arriving randomly within 100 ms and periodically arrives again every beacon interval, which is set as 100 ms. Voice packets are sent every 20 ms and user-defined maximum delay constraints are set to 20 ms, 60 ms, 120 ms, and infinite for each case. An average of 1000 runs is taken for the performance evaluation.

Other parameters for simulation are as follows. Channel switching time is set as 5 ms

according to the measurement in [10]. Min/max-channel time for active scan are 1 and 11 ms. Time to receive a beacon is no more than 1 ms according to [16]. The short-time scale measurement indicates that the clock drift in APs is negligible [10].

## **Performance Evaluation**

The six mechanisms are performed under the delay constraint of 20 ms to analyze the total scan time, as shown in Fig. 10. When  $n_{AP}=10$ , the optimal solution OPT has the best performance of 93.5 milliseconds in average. The FAF mechanism shows a result of 100.4 ms which is close to the OPT solution. The standard passive scan waits for a beacon interval on each channel, thus has a total scan time of 1055 ms, no matter how many APs are in the background. When there is only one AP in the background, which is also the associated AP, all mechanisms except for the standard ones don't have to spend any time on scanning because of the awareness of background information, thus the total scan time is zero in these cases. Besides, since the performance of prePAS mechanism depends on the beacon arrival time, it has a total scan time as long as 213.3 ms when there are 10 APs in the background.



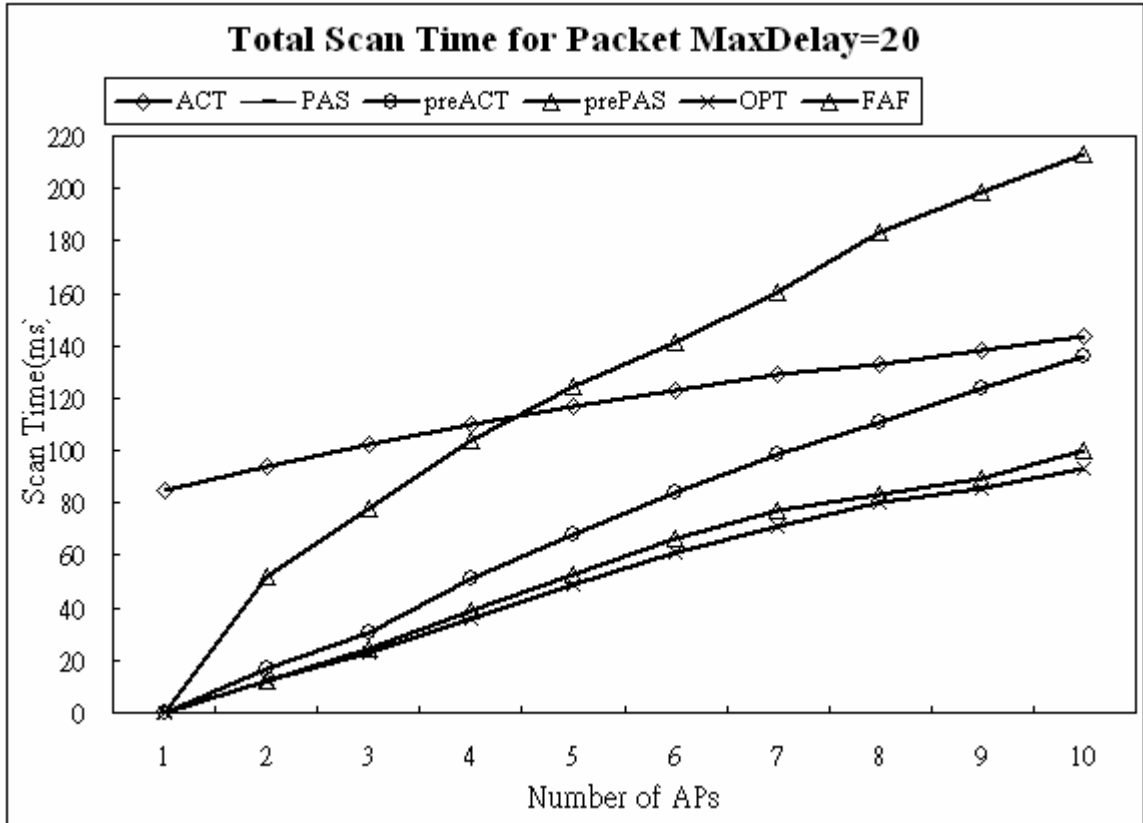


Fig. 10. Total scan time for each mechanism



The enhancement comparing to the standard active scan mechanism is shown in Fig. 11.

The FAF mechanism has an enhancement of 30.06% when  $n_{Ap}=10$ , which is much better than the preACT and prePAS mechanisms and very close to the enhancement of the OPT solution, which is 34.87% better than the standard active scan.

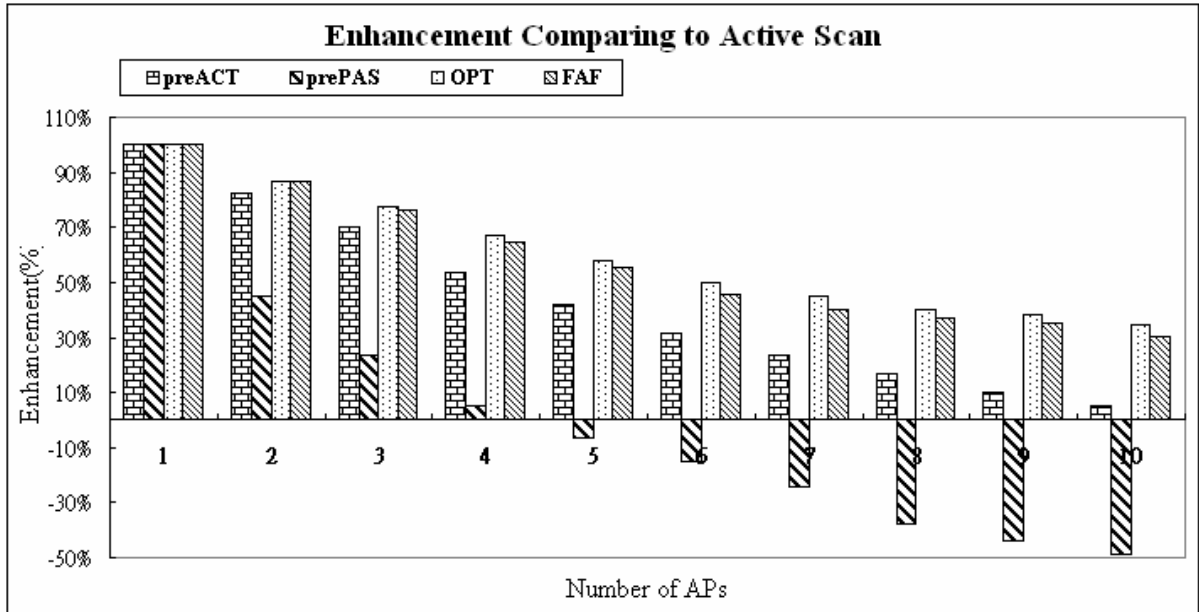


Fig. 11. Enhancement comparing to active scan under maxDelay=20ms

Fig. 12 shows the cumulative distribution of packet delay of each mechanism. The maximum delay is set to 20 ms. More than 90% of voice packets have negligible delay (<1ms) in both OPT and FAF mechanisms, which obviously outperform the standard mechanisms and the preACT solution.



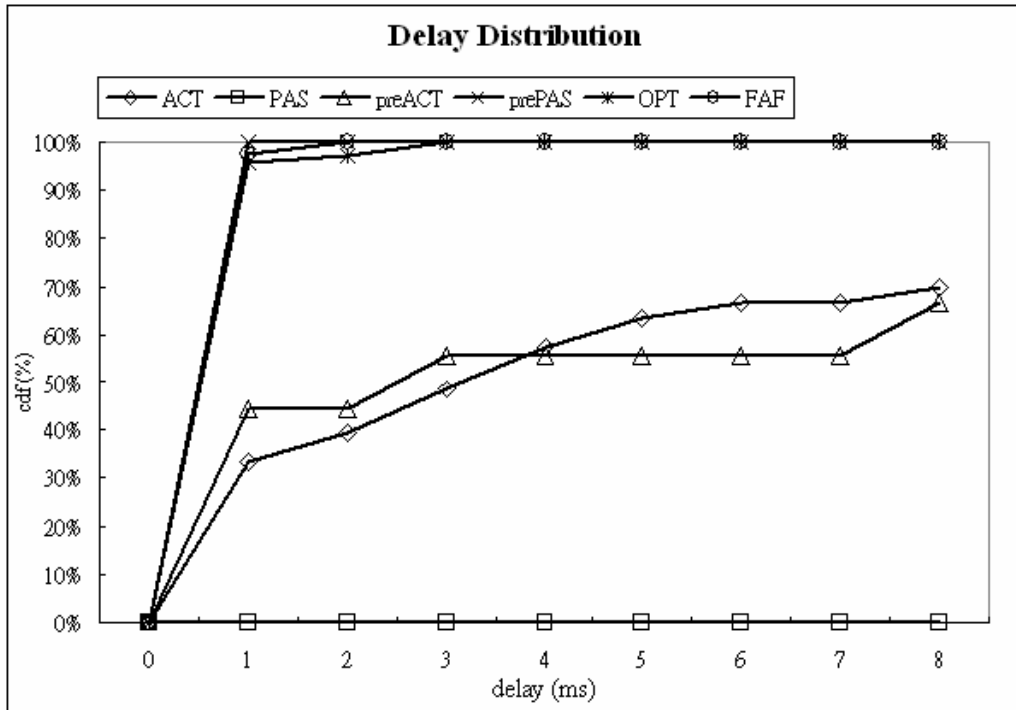


Fig. 12. Delay distribution of each mechanism under maxDelay=20ms



In the last part of this section, the execution time of FAF and OPT scheduling is compared. The two algorithms are run on a personal computer with AMD Athlon 1.83GHz cpu. The result is as shown in Fig. 13. The execution time of the optimal solution increases exponentially through the number of total APs increases. When there are 10 APs in the background, the optimal solution has a schedule latency of 59.84 ms which is much longer than the FAF algorithm.

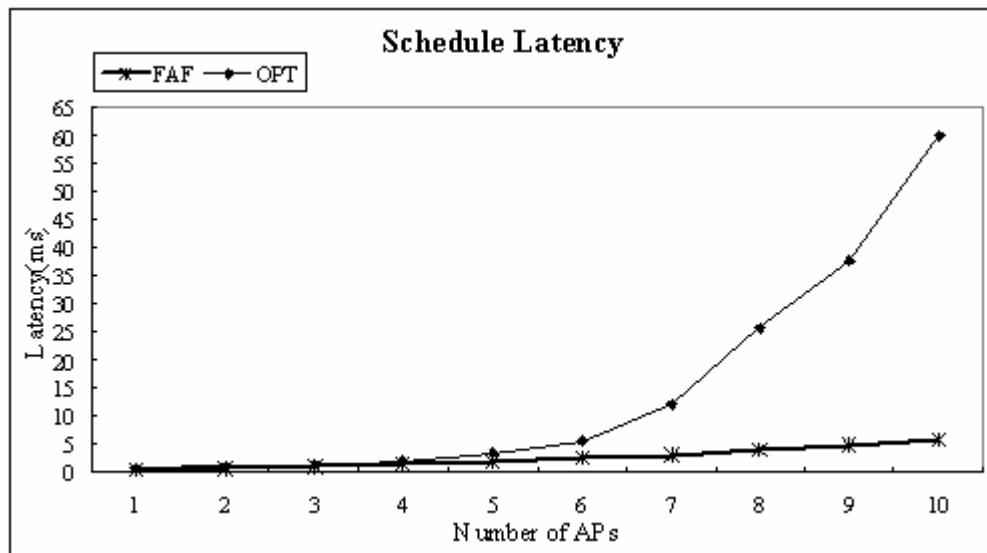


Fig. 13. Execution time for scheduling





## VI. CONCLUSION

By combining active and passive scan strategies, the proposed ASA mechanism minimizes the total scan time without introducing packet loss, delay and jitters. Simulation results show that about 30% scan time can be reduced while the QoS requirement of the connection can be guaranteed during handoffs. A heuristic algorithm was further presented to reduce the computation complexity of the proposed ASA approach. The simulation results also demonstrate that the heuristic algorithm can be easily implemented and achieves near-optimal solutions.



## References

- [1] Sean Christensen, “**Voice over IP solutions**,” Juniper Networks, June, 2001.
- [2] **ETSI**, European Telecommunications Standards Institute, <http://www.etsi.org/>
- [3] Alan Sicher and Randall Heaton, “**GPRS technology overview**,” Dell, February 2002.
- [4] IEEE, “**Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications**,” IEEE standard 802.11, 1999.
- [5] IEEE, “**Trial-Use Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation**,” IEEE standard 802.11f, July 2003.
- [6] IEEE, “**Draft Amendment to Standard for Information Technology - Telecommunications and Information Exchange Between Systems - LAN/MAN Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 2: Fast BSS Transition**,” P802.11r D1.0, November 2005.
- [7] IEEE, “**Media Access Control (MAC) Enhancements for Quality of Service (QoS)**,” IEEE Standard 802.11e/D6.0, December 2003.
- [8] IEEE, “**Radio Resource Measurement Enhancements**,” IEEE 802.11k D1.0, September 2004.
- [9] A. Mishra, M. Shin, and W. A. Arbaugh, “**An Empirical Analysis of the IEEE 802.11**

**MAC Layer Handoff Process,”** ACM SIGCOMM Comput. Commun. Rev. 33, 2, pp.93 – 102, April 2003.

[10] Ishwar Ramani, and Stefan Savage, “**SyncScan: Practical Fast handoff for 802.11 Infrastructure Networks,**” in Proc. IEEE INFOCOM 2005, March 2005.

[11] Matthew S. Gast, “**802.11 Wireless Networks- The Definitive Guide,**” O’REILLY, 2005.

[12] Marc Emmelmann. “**Influence of Velocity on the Handover Delay associated with a Radio-Signal-Measurement-based Handover Decision,**” In Proc. of IEEE Vehicular Technology Conference (VTC 2005 Fall), September 2005.

[13] Minho Shin, Arunesh Mishra, and William A. Arbaugh, “**Improving the Latency of 802.11 Hand-offs using Neighbor Graphs,**” ACM MOBISYS 2004, June 2004.

[14] S. Pack and Y. Choi, “**Fast Inter-AP Handoff Using Predictive Authentication Scheme in a Public Wireless LAN,**” in Proceedings of IEEE Networks Conference, Aug. 2002.

[15] A. Markopoulou, F. A. Tobagi, and M. J. Karam, “**Assessment of VoIP quality over internet backbones,**” IEEE INFOCOM 2002 - the Conference on Computer Communications, June 2002.

[16] A. Trad, F. Munir, and H. Afifi, “**Capacity Evaluation of VoIP in IEEE 802.11e Network Environment,**” CCNC'06 - IEEE Consumer Communications and Networking

Conference, January 2006.



94

碩士論文

無線區域網路換手之適應性頻道掃描演算法

交通大學  
資訊科學院  
工程研究所

鄭雅聯

9317560

94

碩士論文

無線區域網路換手之適應性頻道掃描演算法

交通大學  
資訊科學院  
工程研究所

鄭雅聯

9317560

94

碩士論文

無線區域網路換手之適應性頻道掃描演算法

交通大學  
資訊科學院  
工程研究所

鄭雅聯

9317560



94

碩士論文

無線區域網路換手之適應性頻道掃描演算法

交通大學  
資訊科學院  
工程研究所

鄭雅聯

9317560

94

碩士論文

無線區域網路換手之適應性頻道掃描演算法

交通大學  
資訊科學院  
工程研究所

鄭雅聯

9317560