Improving Channel Scanning Procedures for WLAN Handoffs[1]

Shiao-Li Tsao and Ya-Lien Cheng

Department of Computer Science, National Chiao Tung University
sltsao@cs.nctu.edu.tw

Abstract. WLAN has been widely deployed over public and private areas and has become one of popular access technologies for mobile Internet services in recent several years. Handoff between WLAN access points (APs) that introduce packet loss and delay is one of the critical issues for mobile Internet applications, especially for real-time communications. Previous studies indicated that channel scanning time contributes a significant portion of handoff latency and introduces packet loss and delay. Therefore, solutions based on active scan were proposed to reduce total scanning time of channels so that the service disruption of a communication can be minimized. The other solutions based on passive scan scattering scans between packets did not optimize the total scanning time but avoid packet loss and delay. However, solutions for channel scanning procedures which combine active and passive scan strategies and take total scan latency, packet loss, and delay together into consideration have not yet been investigated. In this paper, a generic channel scanning model is proposed and solutions to improve scanning procedures for WLAN handoff are presented. Simulation results demonstrate that the proposed approaches achieve faster scan time than the existing solutions without violating packet delay and loss requirements specified by the applications during WLAN handoff. Moreover, the implementation of the proposed mechanisms on a WLAN SoC (System-on-Chip) is also discussed in this paper.

Keywords: WLAN, mobility management, channel scan, handoff

1 Introduction

The development of the IEEE 802.11 standards offers new opportunities of wireless accesses for mobile communications, services and applications [1][2][3][4]. However, the coverage of a WLAN access point (AP) is normally 50 to 300 meters and the small WLAN coverage results in frequent handoffs between APs for moving users. A handoff that may disrupt a communication for hundreds of milliseconds to several seconds introduces serious packet delay and loss. The handoff latency significantly influences the qualities of communications, especially for real-time streaming

[1] This work was supported by the NCTU-MediaTek Research Center under Grant Q583.

applications and voice communications [3][9][10]. Hence, to minimize handoff delay becomes one of the most important research issues for WLANs.

A WLAN handoff composes of three phases, i.e. scan phase, re-authentication phase and re-association phase. The scanning phase discovers the APs that an STA can hear and measures the signal strengths of these APs. It takes about several hundred milliseconds. 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 [2]. For a WLAN without IEEE 802.1x and IEEE 802.11i security, previous studies have investigated that the scanning phase contributes up to 90% of the total handoff latency [5]. To scan WLAN channels in order to obtain signal strengths from APs before handoff, the IEEE 802.11 specification defines two scan strategies, i.e. passive scan and active scan [1][7]. For passive scan, a station (STA) scans a channel by switching to the channel and listening beacons from access points (APs) in the channel. Since an STA may not know the arrival time of beacons, the STA typically has to stay on the channel for a beacon interval, say 100ms, and waits for beacons. Generally speaking, the scan latency for passive scan strategy is determined by the length of beacon interval and usually introduces long channel scanning time. On the other hand, an STA can actively broadcast probe request messages to all APs on a channel, receives response messages from APs and then obtains their signal strengths. The channel scanning time for active scan is determined by the number of channels to scan, and the time to stay on a channel and wait for the responses message. Several studies have been worked on reducing the active scan time. For example, an STA can learn from the environment by the historical data or via pre-configurations [4][5] and then the STA uses the cache or neighbor information to eliminate the unnecessary scans on these channels without APs [8]. Although the total scanning time is reduced, they do not consider the service disruption for active connections. The service disruption is especially sensitive for real-time communications such as voice over IP and video streaming. Therefore, SyncScan was proposed by Ramani and Savage [6] based on the 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 and can obtain the signal strengths from APs through beacon messages. Without staying on the channel and waiting for probe response messages, this approach greatly reduces the packet loss and delay for real-time connections. However, this mechanism requires all APs to be synchronized and broadcast their beacons in a scheduled manner. Also, the total scanning time of the passive scan approach is longer than that of the active scan strategies because scans for all channels are not scheduled together but scattered between packet transmissions. The disadvantage of this approach is that the APs need to reconfigured and synchronized, and the approach might not be very practical for the existing WLAN infrastructures which have been already deployed. Moreover, it takes more time to scan all channels and the method cannot support urgent scan and handoff requests which need a fast scan results.

In this paper, a generic channel scanning model which combines active scan and passive scan strategies is proposed. We first present the optimal solution and then propose a heuristic algorithm to obtain fast and near-optimal results. The proposed scan strategies not only consider packet delay and loss requirements specified by applications but also minimize the total channel scanning time. The rest of the paper is organized as follows. In Section 2, the problem is described. Section 3 presents the combined active and passive scan strategies, including the optimal solution and a heuristic algorithm. Performance evaluations and implementation of the proposed method on a WLAN SoC (System-on-Chip) are discussed in Section 4, and finally Section 5 concludes this work.

2 Problem Statements and Modeling

Consider that a WLAN hotspot has *N* WLAN APs which are denoted as AP_1, AP_2 …, and AP_N, they are deployed over *C* channels, i.e. CH_1, CH_2 …, and CH_C. Different from SyncScan [6] which requires extra management and configuration procedures on APs in order to synchronize APs and broadcast beacons in a scheduled manner, we assume no extra management procedures have been applied to the APs. In other words, APs are not synchronized and APs broadcast beacons based on their own schedules. Thus, if the length of beacon interval for AP_i is B_i and AP_i broadcasts its beacon at time T_i, its next beacon should be announced at $T_i + B_i$. We assume an STA has the information including the channels that each AP stays, the lengths of the beacon intervals and the beacon broadcasting time for all APs. The information can be pre-configured on STAs or maintained on a server where the STA can query. For STAs without pre-configurations, it can also learn and cache the information by listening different channels while it has no packet to send or receive.

For an STA that knows the beacon broadcasting time and applies passive scan strategies for channel scanning, it has to switch to different channels before the beacons are broadcasted. An STA has to spend T_{sw} to switch the channel and T_b for waiting the beacon. However, for two or more APs broadcast the beacon at a similar time window which is called a collision, the STA has to decide which beacon to receive first. For the other beacons which are collided, the STA has to wait for another beacon interval. Figure 1 gives an example that an STA applies a passive scan strategy for channel scanning. In Figure 1, there are five APs. There are AP#1, AP#2, AP#3, AP#4 and AP#5 which are configured on channel 1, 1, 6, 11, and 11. The STA initially stays on channel #1 so that it first receives beacon from AP#1. After receiving a beacon from AP#1, it stays in channel #1 and listens for AP#2. After completing channel scan on channel #1, it switches to channel #6 to listen beacon from AP#3. Finally, it switches to channel #11 and receives beacons from AP#4 and AP#5. On the other hand, STAs can also use active scan strategy for channel scanning procedures. For an STA which uses active scan, it only has to switch to a channel, sends probe request messages to the channel, and then waits for responses from all APs in the channel. According to the IEEE 802.11 specification, the STA has to stay on a channel with APs for a maximal channel time T_{max} which is a manageable parameter. Active scan needs not consider the beacon schedules and can be performed at any time. Figure 2 shows an example that an STA applies an active scan strategy for channel scanning. The STA first scans channel #1, and then switch to the channel #6. After completing scan on channel #6, it switches to channel #11 and scans that channel.

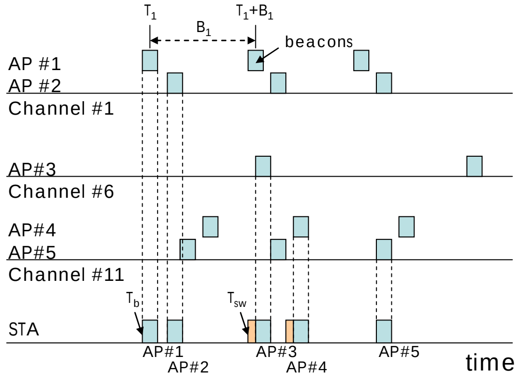

Figure 1. Channel scanning based on the passive scan approach

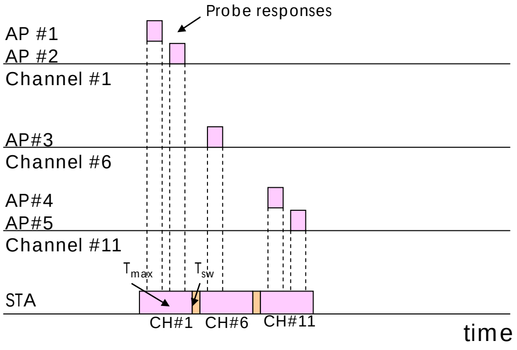

Figure 2. Channel scanning based on the active scan approach

If we further consider an STA currently has connections and has to perform the channel scanning procedures simultaneously, the packet transmission and receiving should be also considered together with the channel scanning schedule. For example, if there is a packet should be transmitted at a particular time and has been specified a maximal delay bound, the packet must be scheduled before its deadline. In other words, while an STA switch to a neighboring channel for an active or passive scan, it has to switch back to the serving channel and receives the packet before deadline. The channel scanning procedures should not influence QoS of the active connection, and this constraint is particularly important for these real-time connections such as voice over IP with QoSs.

In this paper, a generic channel scanning problem is defined. Assume that an STA knows the AP and beacon information and has connections with QoS requirements. The QoS requirement here is the deadline for each packet. The STA uses both active and passive scan strategies for scheduling scanning procedures. The objective function of the scheduler on the STA is to optimize the total channel scanning time without violating packet delay requirements specified by the applications during WLAN handoff. Figure 3 shows an example where the STA uses the combined active and passive scan strategy to minimize the scan time and packet delay. In this example, AP#1 is the serving AP for an STA, and sends voice packets to the STA periodically. The STA knows the packet arrivals and has to switch back to the channel #1 in order to receive the packets from AP#1. The first example shows that the STA applies passive scan to scan the APs. The second example reveals that the STA uses active scan to scan the channels. The third example uses passive scan to scan AP#2, AP#4 and AP#5, and use active scan to scan channel #6. This example demonstrates that combined active and passive scan strategy may reduce the total scanning time. In the next section, an optimal solution for channel scanning and a heuristic algorithm are proposed.

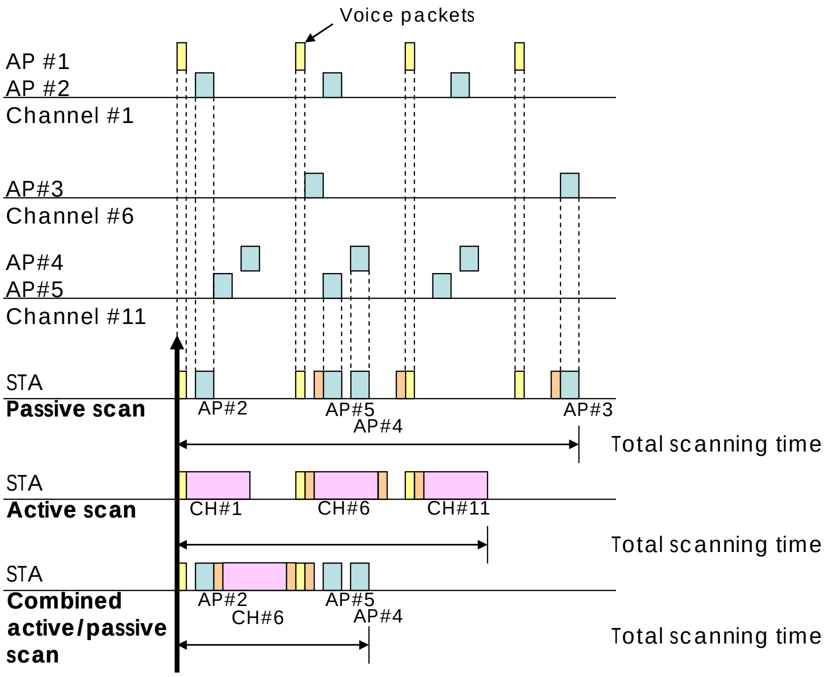

Figure 3. Concurrent packet transmissions and channel scanning based on different channel scanning approaches

3 Optimal and Heuristic Algorithms

First, a scheduled list is defined. A scheduled list contains the scanning sequence and scanning time of all APs that the STA should follow. The starting time of the scheduled list should be the time that the STA wants to perform a channel scanning. The length of period that this scheduled list lasts for is called total scanning time. The total scanning time here is defined as the period between the start of the first scan and the complete of the last scan. In other words, an STA schedules the scanning procedures of APs or channels in the scheduled list and minimizes its length. The schedule policies are as follows. First, packets with delay constraints are scheduled. Second, the passive scan is then considered. The reason why we consider passive scan first is because beacon broadcasting time from APs are determined already. An STA has to follow APs' beacon broadcasting schedules if they want to receive their beacons. Third, active scan which can be performed at any time is finally scheduled. In order to meet the packet delay constraints specified by the application, these packets with delay constraints are first scheduled, i.e. inserted into the scheduled list, before channel scanning is scheduled. That is because we assume the QoS has a higher priority rather than channel scanning. For these applications that do not have delay constraints, packets do not have to insert into the list and the proposed model can be still used for these applications without packet delay requirements. For these applications allows certain packet delays, the delay constraints will be also added into the scheduled list so that the scanning scheduler can adjust the packets to accommodate channel scheduling procedures without violating its delay requirement. After the packets are inserted, passive scan is then considered. Beacons from APs are then inserted into the scheduled list if there is no collision in the current list. The insertion function has to verify if the new inserted beacon is collided with the existing beacons or not. If the new beacon is collided, the two beacons and their next scheduled beacons will be marked in the scheduled list. That information helps the optimal algorithm to decide which beacon to scan or maybe an active scan is required.

3.1 The Optimal Algorithm

The optimal solution is to perform an exhausted search on all possible channel scanning sequences and to find the one with the minimal channel scanning time. Since active scan can be scheduled at any time, passive scan are scheduled first. After passive scan is scheduled, active scan are then insert and replace passive scan if the active scan can further reduce scanning time. Assume the scheduled list denotes a {AP#1, AP#2, AP #3 …, AP #N}. The exhausted search on all possible combinations of passive scan can be derived. Through the number of APs increases, the number of possible solutions has exponential growth. To perform the exhausted search, dynamic programming technique has been applied. The previous results can be used to derive the new possible solutions. For example, only when AP#1 and AP#3 can be scheduled without collisions for passive scan, AP#1, AP#3 and AP#4 should be considered. Otherwise, it is no need to test if {AP#1, AP#3 and AP#4} is schedulable. The optimal algorithm by applying dynamic programming technique reduces the search cases.

After passive scan sequences have been derived, active scan for a specific channel can be inserted. If active scan is applied and the total scanning time can be reduced, APs in the same channel will be replaced by a channel number, say CH#M for example. That means the active scan will be applied to that channel instead of using passive scan. The optimal algorithm is described in Figure 4. Although the dynamic programming algorithm simplifies the solution-deriving process, it has a time complexity of $O(2^N)$. When there are too many APs to scan, it may become impractical due to the long scheduling time and the performance of the scanning phase will be seriously damaged. Thus, a heuristic algorithm is considered and proposed.

```
OPTIMAL()
Input: Lists of APs, their channel identifier and beacon
broadcasting time.
Output:    The scheduled list with the minimal channel
scanning time
ScanTime: Minimal total scan time
TMPNode: ($AP_{ID}$, pTMPNode)
pTMPNode: pointer to the previous TMPNode
```

```
**TMPList**: temporary scheduled list
Create a new TMPNode, assign the channel identify as NULL,
and insert all APs to TMPList;

for $AP_i$
   for each $TMPNode_j$ in TMPList
       backup $TMPNode_j$ scheduled list
       insert $AP_i$ into $TMPNode_j$ scheduled list
       if success then
          create a new newTMPNode
          copy TMPNode scheduled list to newTMPNode
          refer newTMPNode to $TMPNode_j$, and insert
newTMPNode into TMPList
          Restore TMPNode scheduled list
        else
          Mark APi as scanned
        end if
    end for
end for
for TMPNodej in TMPList
    add active-scan for scanned AP
    calculate new ScanTime
end for
Return the best ScanTime;
```

Figure 4. Optimal channel scanning algorithm

3.2 Heuristic Algorithm

To reduce the complexity to search all possible solutions, some APs which are not suitable for passive scan are removed from the passive scan search spaces. If only a part of APs is considered during passive scan schedules, the complexity can be considerably reduced. For example, for the channels have very few APs, they may be suitable for passive scan. Otherwise, if there are many APs in the same channel, active scan which can obtain all responses at one time should be applied to these channels. Therefore, we first screen the APs and remove these APs which are not suitable for passive scan. The test is basically to examine if the total passive scan time on these APs in the same channel exceeds the time for an active scan in that channel. After the procedure, the number of APs participating passive scan search is reduced.

After deciding the scanning type of each AP, an adjusting function is performed to further improve the scan time. 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. Figure 5 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. The detail heuristic algorithm is shown in Figure 6.

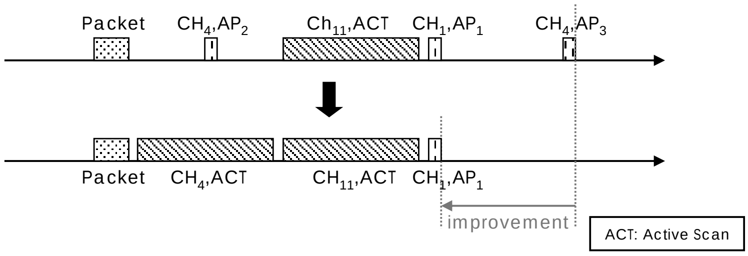

Figure 5. Adjustment channel scheduling procedures

```
CANDIDATE()
Input: Lists of APs, their channel identifier and beacon
broadcasting time.
Output: CandiList: candidate list of potential passive
scan APs

for Channel$_i$
    calculate $T_{act}$, which is the minimal active scanning
time on APi under delay constraints
    calculate the passive scanning time for all AP in the
channel
    if the passive scanning time is shorted than $T_{act}$
        insert AP in the channel i to CandiList and sort
the list by the number of APs
end for
Return CandiList

HEURISTIC ()
Input: Lists of APs, their channel identifier and beacon
broadcasting time.
Output:    The scheduled list with the minimal channel
scanning time
ScanTime: Minimal total scan time

Call CANDIDATE()identify potential passive-scan APs
for $AP_i$ in CandiList
    insert $AP_i$ into scheduled list
    if failure
        mark $AP_i$ as active scan
    end if
    remove redundant passive scan
    insert the active scan to scheduled list for these
APs marked as active scan
```

```
    call ADJUST to further improve the final schedule
    calculate ScanTime
end for
Return ScanTime;
```

Figure 6. Heuristic algorithm

First, a candidate AP list where APs by applying passive scan may have a shorter scanning time than that by applying active scan is built. The APs in the candidate list will be sorted by the channel according to the number of APs on the same channel. The channels with less total passive scanning time are scheduled first. Then the channels in the candidates are checked one by one, until all of them are scheduled. The channels did not included into the candidate list, they should be scanned by using 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. After all channels are scheduled, the adjusting function is performed. The scheduled result is enhanced by the adjusting function to get better resource utilization and time performance. The heuristic solution has a time complexity of $O(N_b^2)$, where N_b is the number of beacons which are scheduled by using passive scan. The heuristic solution considerably reduces the time complexity compared with the optimal solution.

4 Simulation and Evaluation

In this section, simulations are conducted to evaluate the performance of the proposed approaches. The simulation program is written in C language, and four scanning mechanisms, including enhanced passive scan, enhanced active scan, optimal scan and the proposed heuristic mechanisms are implemented and evaluated. We assume the STAs know the information of all APs including the channel number, and beacon broadcasting time. The enhanced version passive and active scan mechanisms are denoted as ePAS and eACT which take advantages of the environment information to reduce the channel-waiting time. In the eACT mechanism, the STA always skips the channels without APs. In the ePAS mechanism, the STA knows the beacon arrival time of each AP and can hear the beacons exactly whenever they arrive to the channel. To simulate the WLAN network, one to ten APs are randomly generated and distributed over eleven channels. The beacon intervals for all APs are 100ms but they are not synchronized, i.e. the initial beacons for APs are different. Voice packets are sent to the STA every 20ms. For other WLAN parameters used in the simulations, they are the channel switching time (T_{sw}=5ms) [6], the maximal channel time (T_{max}=11ms), time to receive a beacon (T_b=1ms) [10]. Below evaluations are all based on the average results for 1000 time simulations which imply 1000 different channel configurations of WLAN APs are tested

First, the delay constraint of voice packets is assumed 20ms and the total channel scanning time for the four different approaches is investigated. Figure 7 shows the simulation results. It can be seen from the figure while the number of AP increases, the average total scanning time also increases for all four approaches. For the eACT approach, it is because in our simulation, APs are randomly distributed to eleven WLAN channels. While the number of AP increases, the number of channels that have APs also increases. The eACT thus spends more time on average for the channel scanning. For the ePAS approach, the total scanning time increases faster that other three approaches while the number of AP increases. The reason is for a network with more APs, the STA has to spend more time to listen beacons from all APs passively. Moreover, more APs results in higher probability of beacon collisions than less APs. To compare the performance between the ePAS, eACT and the proposed approaches, it can be found that when the number of AP is ten, the optimal solution, denoted as OPT, has the best performance of the average total scanning time which is 93.5 milliseconds. The proposed heuristic, denoted as HEU mechanism, achieves the second best result of 100.4 ms which is very close to the optimal solution. The optimal and heuristic solutions reduce 50% channeling scanning time than the ePAS. Also, the optimal and heuristic solutions can reduce 26% and 25% channel scanning time than eACT, respectively.

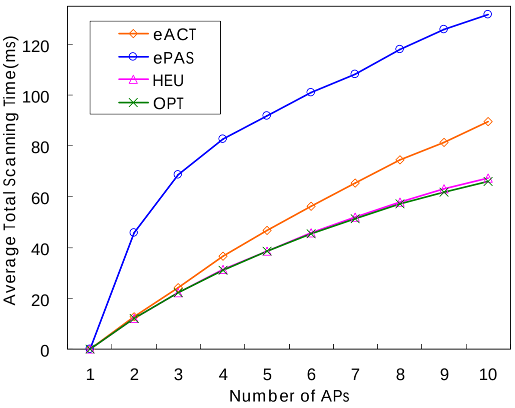

Figure 7. Total scanning time for different scan mechanisms

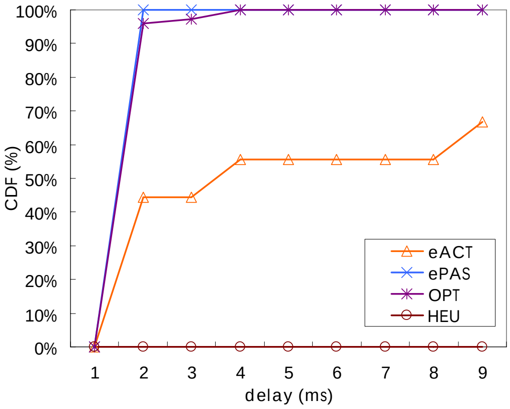

Figure 8. Delay distribution of different mechanisms while the maximal delay is 20ms

Figure 8 shows the cumulative distribution of packet delay by applying different channel scanning mechanisms. In this simulation, only the extra delay due to channel scanning procedures is counted. The maximum delay of the voice packet due to channel scanning procedures is set to 20 ms for all four mechanisms in this simulation. It is important to note that if applications can tolerate more delay due to channel scanning procedures, the scanning time can be further reduced. The STA can schedule channel scanning first and then schedules packet transmission. Thus, the channel scanning time can be reduced. The channel scanning time and the maximum packet delay specified by applications are tradeoff. Figure 8 depicts the ePAS, the proposed optimal and heuristic method can all minimize packet delays. It can be seen that more than 90% of voice packets have less than one millisecond delay during handoff. ePAS is similar to SyncSCAN performs good in reducing packet delay during handoffs. Simulation results demonstrate that the proposed mechanisms outperform the eACT solution.

Finally, the execution time of heuristic and optimal channel scanning approach is compared. The two algorithms are run on a personal computer with AMD Athlon 1.83GHz CPU. The result is as shown in Figure 9. The execution time of the optimal solution increases exponentially while the number of total APs increases. When there are ten APs, the optimal solution needs 60ms to calculate the scanning schedule while the heuristic approach merely needs 5ms. Although the optimal solution can also obtain the results fast, the computation overhead may not be acceptable if the optimal algorithm is executed in a mobile device which the computation power is much lower than a PC.

For most of WLAN chipsets or SoCs, the handoff policy is implemented in either firmware or software drivers. The chipsets and SoCs provide means such as control registers for software to control the scan procedures. For example, some WLAN chipsets or SoCs allow software to switch to a specific channel, send a specific message, and these functions can be integrated together to realize the proposed mechanisms. Similar to [6], the concept of proposed mechanisms can be implemented into the drivers. For these WLAN chipsets or SoCs which fully implement scan procedures using ASIC or does not provide control registers fro software to change its scan behaviors, the implementation of the proposed mechanisms by using software is not possible. In that case, the modification of hardware is required.

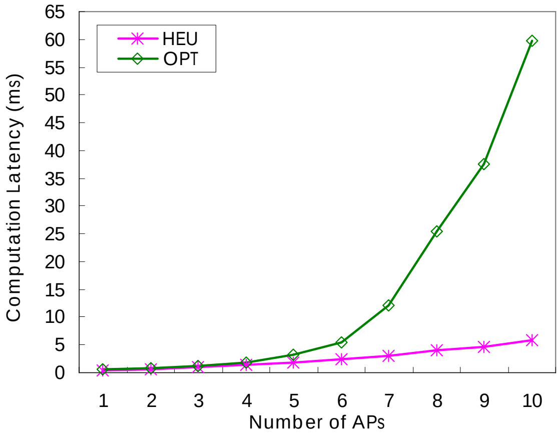

Figure 9. Execution time for different mechanisms

5 Conclusions

In this paper, a generic channel scanning strategy which combines both active and passive scan mechanisms was presented. Based on the strategy, an optimal and a heuristic algorithm were proposed to minimizes the total scanning time without violating the packet loss and delay requirements specified by the application during WLAN handoff. Simulation results demonstrate that about 30% to 50% scanning time can be reduced by applying the proposed algorithms while the QoS requirements can be also met. Comparing with optimal solution, the heuristic algorithm that significantly reduces the computation complexity can be easily implemented on mobile devices and achieves near-optimal results.

References

1. IEEE, “Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” IEEE standard 802.11, 1999.
2. 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.
3. IEEE, “Media Access Control (MAC) Enhancements for Quality of Service (QoS),” IEEE Standard 802.11e, 2006.
4. IEEE, “Radio Resource Measurement Enhancements,” IEEE 802.11k D1.0, September 2004.
5. 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.
6. Ishwar Ramani and Stefan Savage, “SyncScan: Practical Fast handoff for 802.11 Infrastructure Networks,” in Proc. IEEE INFOCOM 2005, March 2005.
7. Matthew S. Gast, “802.11Wireless Networks- The Definitive Guide,” O’REILLY, 2005.
8. Minho Shin, Arunesh Mishra, and William A. Arbaugh, “Improving the Latency of 802.11 Hand-offs using Neighbor Graphs,” ACM MOBISYS 2004, June 2004.
9. 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.
10. 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.