ABSTRACT OF REPORT

Anshul Jain

The Graduate School
University of Kentucky
2003

Abstract

IEEE 802.11-based wireless local area networks have been growing immensely in the last few years. Until then, LANs were limited to the physical, hard-wired infrastructure of the building. The major motivation and benefit from wireless LANs is increased mobility. When wireless station moves out of the range of one *access point*, it has to connect to another access point in order to remain connected. The process of association from one access point to another access point is called *handoff*. In this report, we present the details of the handoff process along with its various components and total time for handoff. In our study we found significant variations in handoff latency depending upon what channels the access points are on and whether the access points have the same SSID or not.

Handoff Delay for 802.11b Wireless LANs

PROJECT REPORT

A report submitted in partial fulfillment of the
requirements for the degree of Master of Sciences
at the University of Kentucky

By

Anshul Jain

Advised By

Dr. Henning Schulzrinne, Columbia University
Dr. Zongming Fei, University of Kentucky

2003

Table of Contents

Chapter 1

Introduction

The major motivation and benefit from wireless LANs is increased mobility. Untethered from conventional network connections, network users can move about almost without restriction and access LANs from nearly anywhere.

1.1 The IEEE 802.11 Wireless LAN architecture

The 802.11 architecture is comprised of several components and services that interact to provide station mobility transparent to the higher layers of the network stack [9].

Wireless LAN Station: The station (STA) is the most basic component of the wireless network. A station is any device that contains the functionality of the 802.11 protocol, that being medium access control (MAC), physical layer (PHY), and a connection to the wireless media. Typically, the 802.11 functions are implemented in the hardware and software of a network interface card (NIC). A station could be a laptop PC, handheld device, or an Access Point. Stations may be mobile, portable, or stationary and all stations support the 802.11 station services of authentication, de-authentication, privacy, and data delivery.

Basic Service Set (BSS): 802.11 defines the Basic Service Set (BSS) as the basic building block of an 802.11 wireless LAN. The BSS consists of a group of any number of stations.

Service Set Identifier (SSID): A service set identifier (SSID) is a unique label that distinguishes one WLAN from another. So all APs and all STAs attempting to connect to a specific WLAN must use the same SSID. Wireless STAs use the SSID to establish and maintain connectivity with APs. An SSID is also referred to as a network name because essentially it is a name that identifies a wireless network. An SSID contains up to 32 alphanumeric characters, which are case sensitive.

In IEEE's proposed standard for wireless LANs (IEEE 802.11), there are two different ways to configure a network: ad-hoc and infrastructure. In the *ad-hoc* network, computers are brought together to form a network on the fly. There is no structure to the network; there are no fixed points; and usually every node is able to communicate with every other node. The *infrastructure* architecture uses fixed network access points with which mobile nodes can communicate. These network access points are sometimes connected to backbone called distributed system, typically Ethernet, to expand the LAN's capability by bridging wireless nodes to other wired nodes.

Distribution System (DS): The Distribution System is the means by which an access point communicates with another access point to exchange frames for stations in their respective BSSs, forward frames to mobile stations as they move from one BSS to another, and exchange frames with a wired network.

Extended Service Set (ESS): An extended service set is a set of infrastructure BSSs, where the access points communicate amongst themselves to forward traffic from one BSS to another to facilitate movement of stations between BSSs.

$802.11b$ follows a three non-overlapping channel based model that involves overlap and interference as per the table below.

Channel	Lower Freq (MHz)	Mid Freq (MHz)	Upper Freq (MHz)
1	2401	2412	2423
6	2426	2437	2448
11	2451	2462	2473

Channel 1 ends at 2423 MHz and channel 4 starts at 2416 MHz, that is a significant overlap. The entire 22 MHz follows a parabola pattern with power on vertical axis and frequency on horizontal as shown in Fig. 1.1 below.

In order to achieve mobility, STA should be able to move from one BSS to other BSS without loosing connnectivity. This process is called a *handoff*. Handoff and other distributed services like synchronization, scanning, association and disassociation are done by exchanging 802.11 manage-

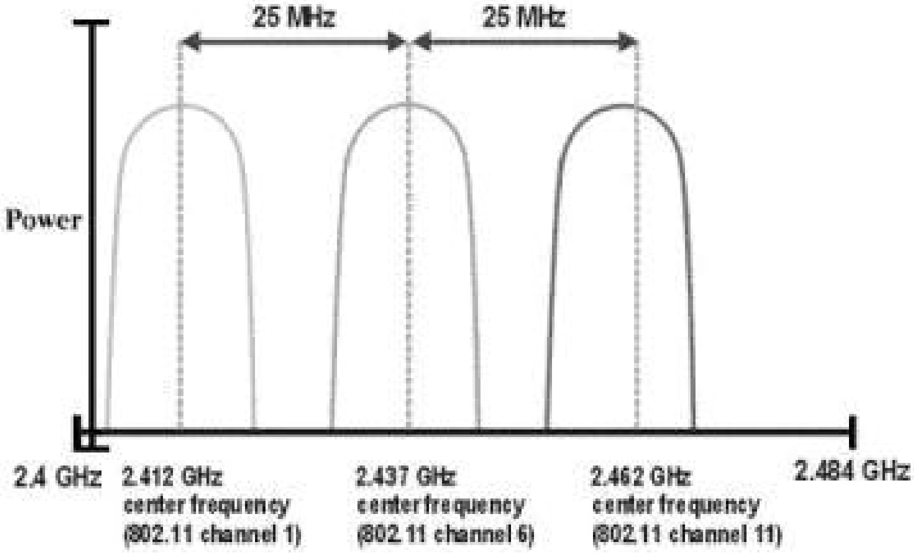


Figure 1.1: Channel Allocation scheme for 802.11b [5]

ment frames between particpating STAs and APs. Before we study the handoff procedure in detail in the next chapter, we describe the 802.11 management frames.

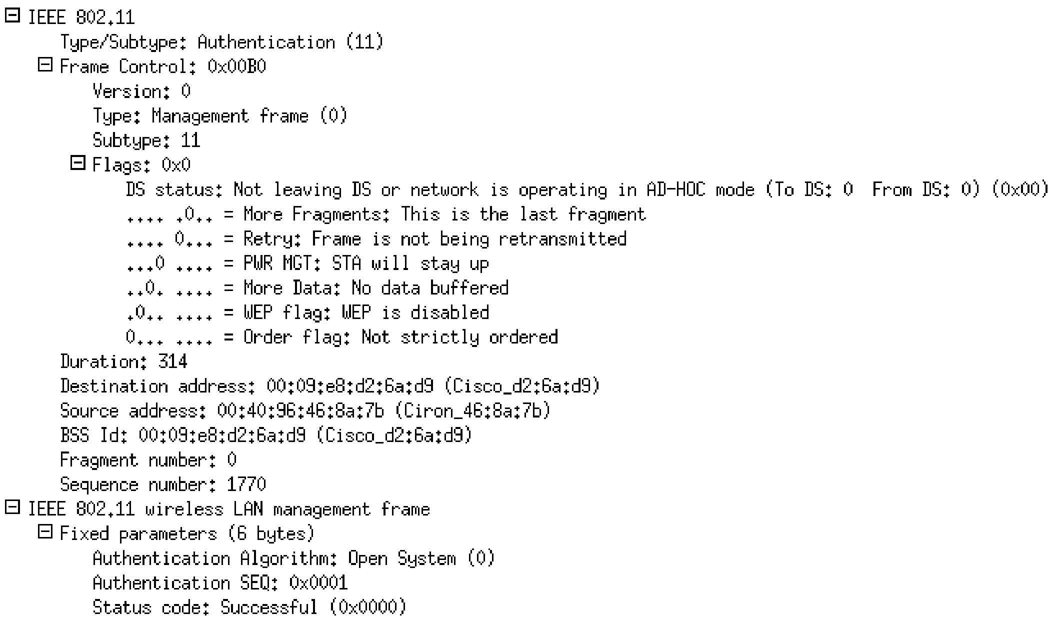
1.2 802.11 Management Frame Format

802.11 management frames enable stations to establish and maintain communications. The following are common 802.11 management frame subtypes, with the description taken from [6]:

Authentication frame: “802.11 authentication is a process whereby the access point either accepts or rejects the identity of a STA. The STA begins the process by sending an authentication frame containing its identity to the access point. With open system authentication (the default), the STA sends only one authentication frame, and the access point responds with an authentication frame as a response indicating acceptance (or rejection)”. ”See Fig. 1.2.”

Deauthentication frame: “A station sends a deauthentication frame to another station if it wishes to terminate secure communications.”

Association request frame: “802.11 association enables the access point to allocate resources for and synchronize with a STA. A STA begins the association process by sending an association



```
⊟ IEEE 802.11
     Type/Subtype: Authentication (11)
   ⊟ Frame Control: 0x00B0
        Version: 0
        Type: Management frame (0)
        Subtype: 11
      ⊟ Flags: 0x0
           DS status: Not leaving DS or network is operating in AD-HOC mode (To DS: 0  From DS: 0) (0x00)
           .... .0.. = More Fragments: This is the last fragment
           .... 0... = Retry: Frame is not being retransmitted
           ...0 .... = PWR MGT: STA will stay up
           ..0. .... = More Data: No data buffered
           .0.. .... = WEP flag: WEP is disabled
           0... .... = Order flag: Not strictly ordered
     Duration: 314
     Destination address: 00:09:e8:d2:6a:d9 (Cisco_d2:6a:d9)
     Source address: 00:40:96:46:8a:7b (Ciron_46:8a:7b)
     BSS Id: 00:09:e8:d2:6a:d9 (Cisco_d2:6a:d9)
     Fragment number: 0
     Sequence number: 1770
⊟ IEEE 802.11 wireless LAN management frame
   ⊟ Fixed parameters (6 bytes)
        Authentication Algorithm: Open System (0)
        Authentication SEQ: 0x0001
        Status code: Successful (0x0000)
```

Figure 1.2: Snapshot of 802.11 Authentication Frame as seen in Ethereal

request to an access point. This frame carries information about the STA (e.g., supported data rates) and the SSID of the network it wishes to associate with. After receiving the association request, the access point considers associating with the STA, and (if accepted) reserves memory space and establishes an association ID for the STA."

Association response frame: "An access point sends an association response frame containing an acceptance or rejection notice to the STA requesting association. If the access point accepts the STA, the frame includes information regarding the association, such as association ID and supported data rates. If the outcome of the association is positive, the STA can utilize the access point to communicate with other STAs on the network and systems on the distribution (i.e., Ethernet) side of the access point."

Reassociation request frame: "If a STA roams away from the currently associated access point and finds another access point having a stronger beacon signal, the STA will send a reassociation frame to the new access point. The new access point then coordinates the forwarding of data frames that may still be in the buffer of the previous access point waiting for transmission to the STA."

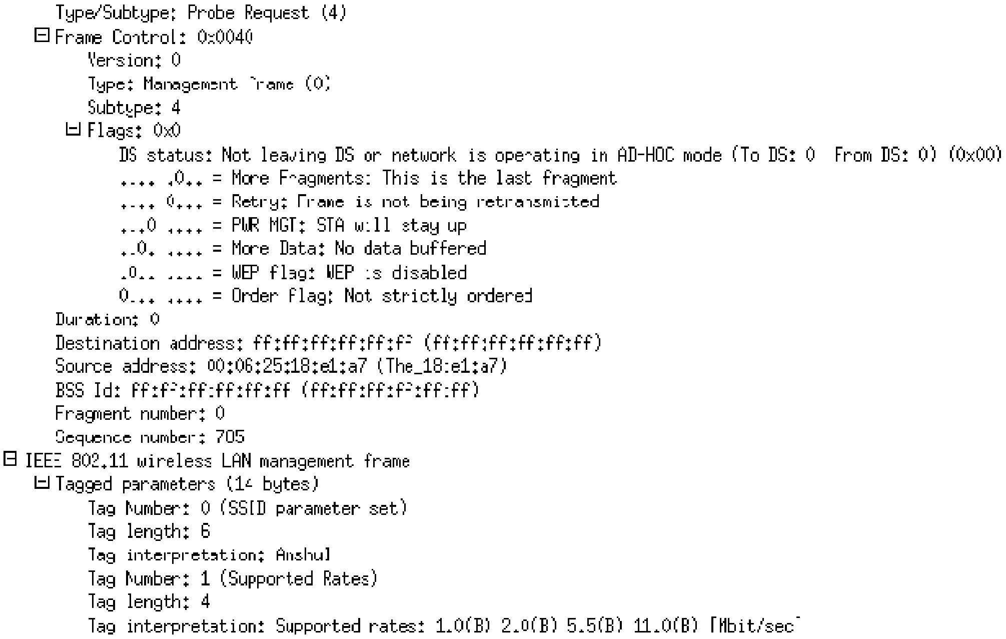
Reassociation response frame: “An access point sends a reassociation response frame containing an acceptance or rejection notice to the STA requesting reassociation. Similar to the association process, the frame includes information regarding the association, such as association ID and supported data rates.”

Disassociation frame: “A station sends a disassociation frame to another station if it wishes to terminate the association. For example, a STA that is shut down gracefully can send a disassociation frame to alert the access point that the STA is powering off. The access point can then relinquish memory allocations and remove the STA from the association table.”

Beacon frame: “The access point periodically sends a beacon frame to announce its presence and relay information, such as timestamp, SSID, and other parameters regarding the access point to STAs that are within range. STAs continually scan all 802.11 radio channels and listen to beacons as the basis for choosing which access point is best to associate with.”

Probe request frame: “A station sends a probe request frame when it needs to obtain information from another station. For example, a STA would send a probe request to determine which access points are within range”. ”See Fig. 1.3.”

Probe response frame: “A station will respond with a probe response frame, containing capability information, supported data rates, etc., after it receives a probe request frame.”



```
    Type/Subtype: Probe Request (4)
  ⊟ Frame Control: 0x0040
        Version: 0
        Type: Management frame (0)
        Subtype: 4
      ⊟ Flags: 0x0
            DS status: Not leaving DS or network is operating in AD-HOC mode (To DS: 0  From DS: 0) (0x00)
            .... .0.. = More Fragments: This is the last fragment
            .... 0... = Retry: Frame is not being retransmitted
            ...0 .... = PWR MGT: STA will stay up
            ..0. .... = More Data: No data buffered
            .0.. .... = WEP flag: WEP is disabled
            0... .... = Order flag: Not strictly ordered
    Duration: 0
    Destination address: ff:ff:ff:ff:ff:ff (ff:ff:ff:ff:ff:ff)
    Source address: 00:06:25:18:e1:a7 (The_18:e1:a7)
    BSS Id: ff:ff:ff:ff:ff:ff (ff:ff:ff:ff:ff:ff)
    Fragment number: 0
    Sequence number: 705
⊟ IEEE 802.11 wireless LAN management frame
  ⊟ Tagged parameters (14 bytes)
        Tag Number: 0 (SSID parameter set)
        Tag length: 6
        Tag interpretation: Anshul
        Tag Number: 1 (Supported Rates)
        Tag length: 4
        Tag interpretation: Supported rates: 1.0(B) 2.0(B) 5.5(B) 11.0(B) [Mbit/sec]
```

Figure 1.3: Snapshot of 802.11 Probe Request Frame as seen in Ethereal

Chapter 2

Handoff Procedure

Handoff is a procedure executed when a mobile node moves from coverage area of one AP to that of another AP. The handoff process involves a sequence of messages being exchanged between the mobile node and the participating APs. This sequence of messages can be divided into three phases: *probe*, *authentication* and *reassociation*, which are described in detail later. The transfer from old AP to the new AP results in some state information being transfered from the former to the latter. The state information that is transferred consists of authentication, authorization and accounting information. This can be achieved by an *Inter Access Point Protocol* (IAPP) that is currently under draft in IEEE $802.11f$, or by a protocol specific to the vendors. We used *Cisco Aironet* 1200 series APs that follow the *Cisco Discovery Protocol* (CDP).

2.1 Steps during Handoff

The handoff process can be divided into two logical steps: *discovery* and *reauthentication* [1].

Discovery: The discovery process involves the handoff initiation phase and the scanning phase. When the STA is moving away from the AP it is currently associated with, the *signal strength* and the *signal-to-noise ratio* of the signal from the AP might decrease. This may cause STA to initiate a handoff. Now, the STA needs to find other APs that it can connect to. This is done by MAC layer *scanning* function. Scanning can be accomplished using either in *passive* or *active* mode.

In passive scan mode, the STA listens to the wireless medium for *beacon* frames. Beacon frames provide a combination of timing and advertising information to the STAs. Using the information obtained from beacon frames the STA can elect to join or decline the AP. During this scanning mode the STA listens to each channel of the physical layer in trying to locate an AP. Active scanning involves transmission of *probe request* frames by the STA in the wireless medium and processing the received *probe responses* from the APs. The basic procedure of the active scan mode includes the following steps as explained in [2]:

1. Using the normal channel access procedure, *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA), gain control of wireless medium.

2. Transmit a *probe request* frame which contains the broadcast address as destination.

3. Start a *probe timer*.

4. Listen for *probe response*.

5. If no response received by *minChannelTime*, scan next channel.

6. If one or more responses are received by *minChannelTime*, stop accepting *probe responses* at *maxChannelTime* and process all received responses.

7. Move to next channel and repeat above steps.

After all channels have been scanned, all information received from probe responses is processed and passed to the management entity for selecting which AP to join.

Reauthentication: The reauthentication process involves an authentication and a re-association to the new AP. This phase involves transfer of STA's credentials from old AP to the new AP. Authentication is a process by which the AP either accepts or rejects the identity of the STA. The STA begins the process by sending the *authentication* frame telling its identity to the AP. With *the open system* authentication that we are using in our experiments, the STA sends one *authentication* frame and the AP responds with an *authentication* frame as a response indicating acceptance or rejection. After authentication is successful, the STA sends the reassociation frame to the new AP. The new AP then sends a reassociation frame back to the STA containing an acceptance or rejection notice. Fig. 2.1 taken from [1] shows the sequence of messages expected during the handoff.

As seen in the above figure the sequence of messages can be divided into three types:

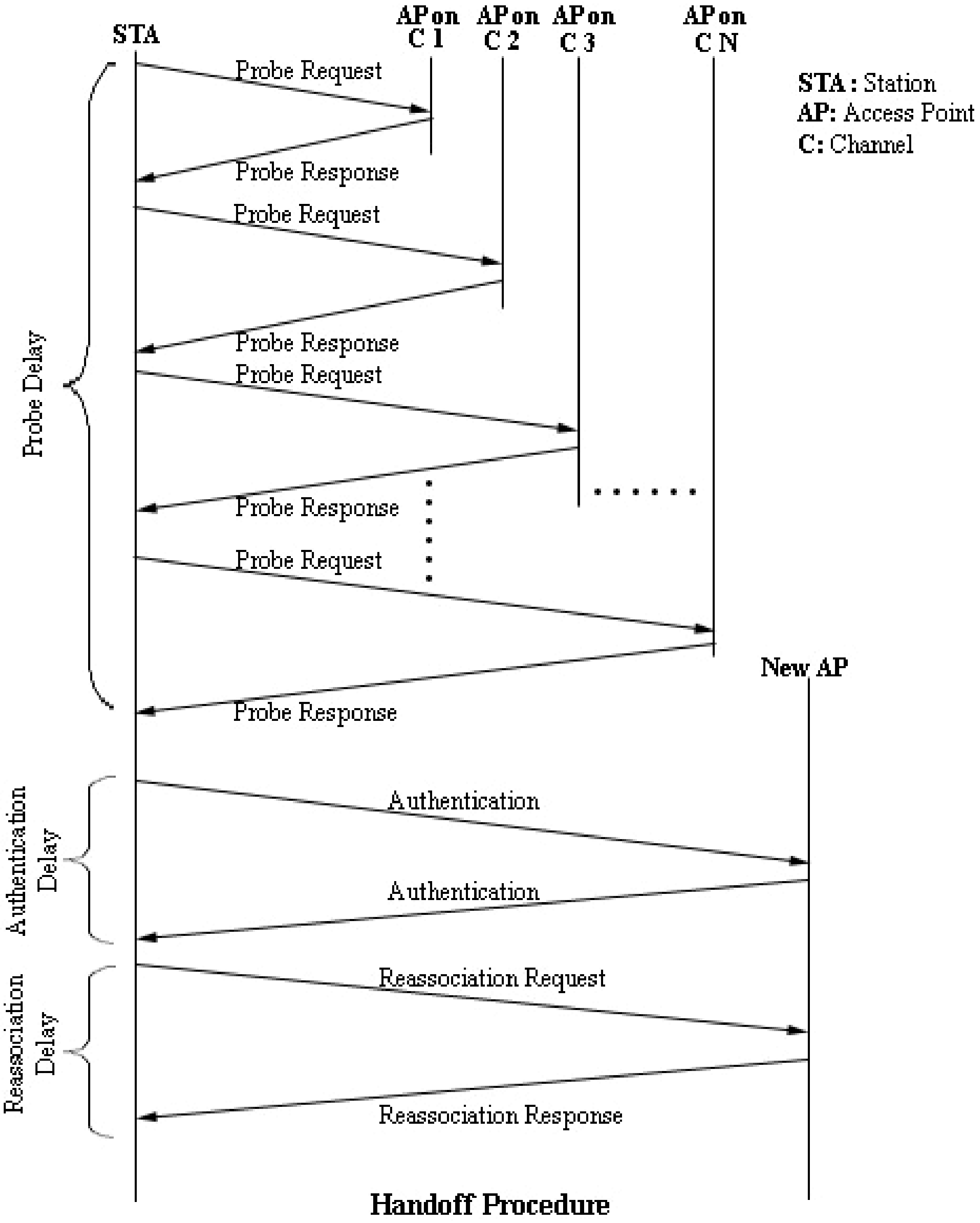


Figure 2.1: Handoff procedure for 802.11b

1. **Probe Messages:** Once the STA decides to look for other APs, the probe phase starts. The STA starts sending out *probe requests* and process received *probe responses* based on active scanning algorithm explained above. The time spent during this probing process is called as *probe delay*. The probe messages form the discovery phase of the handoff.

2. **Authentication Messages:** Once the STA decides to join an AP, *authentication* messages are exchanged between the STA and the selected AP. The time spent during this process is called *authentication delay*. The authentication messages form the reauthentication phase of the handoff.

3. **Reassociation Messages:** After successful authentication, the STA sends a *reassociation request* and expects a *reassociation response* back from the AP. The latency incurred during reassociation messages is called *reassociation delay*. Reassociation messages form the reauthentication phase of the handoff. After reassociation the new access point coordinates the forwarding of data frames that may still be in the buffer of the previous access point waiting for transmission to the STA.

Chapter 3

Experimental Setup

3.1 Hardware Specification

The experiment consists of three hardware components: a *wireless network*, a *wireless client (STA)* and *two wireless sniffer systems*.

Wireless Network: The experiments were performed in the Laboratory of Advanced Networking at University of Kentucky. A university's campus wireless network named *ukyedu* already exists in the building. Another wireless network was deployed as a part of this experiment setup, named *Anshul*. The *ukyedu* network consists of six APs, one on channel 1 and five on channel 6. The *Anshul* network consists of two APs: one on channel 1 and the other on channel 11. The two APs of network *Anshul* are hosted on channel 1, 11, respectively, to avoid interference as channels 1 and 11 are non-overlapping. However, depending upon the requirements for the experiments we have changed the channel allocation and network names. The two APs on network *Anshul* are *Cisco Aironet* 1200 series access points.

Wireless Client: The wireless client is a *Pentium III* 300 MHz, 256 MB RAM *Gateway* laptop. The wireless card used is *Cisco Aironet 350***.**

Wireless Sniffer Systems: The first sniffer system is a *Pentium IV* 1.67 GH$_z$, 256 MB RAM *Sony* laptop with *Linksys WPC11* $v3.0$ wireless PCMCIA card. The second sniffer system is a *Pentium III* 300 *MHz*, 128 MB RAM *IBM* laptop with *Linksys WPC11* $v3.0$ wireless PCMCIA card.

3.2 Software Specification

Operating System : The operating system used in experiments is *Red Hat* 8.0 with kernel version $2.4.18 - 14$.

Drivers:

1. Driver used for Cisco 350 card is airo-linux driver downloaded from http://sourceforge.net/project/airo-linux.
2. Driver used for Linksys WPC11 $v3.0$ card is linux-wlan-ng-0.1.16.pre10 driver downloaded from http://flinux-wlan.org.

Software Tools: The other software tools used in the project are as follows:

1. We used Kismet 802.11 wireless network sniffer to sniff $802.11b$ packets on the wireless network. We used the version $2.8.1$ of the software available at [13].
2. We used the Ethereal network protocol analyzer to analyze sniffed $802.11b$ packets. We used version $0.9.6 - 1$, packaged along with *Red Hat* 8.0.
3. We used Cisco Aironet Client Utility to configure the Cisco 350 cards. We used version 2.0 downloaded from Cisco's website.

3.3 Kismet Wireless Sniffer

Kismet is an 802.11 wireless network sniffer. It separates and identifies different wireless networks in the area. It sniffs $802.11b$ packets on the wireless networks and allows Ethereal/tcpdump compatible file logging. Kismet's primary user interface is divided into three primary panels [13]:

1. *Network display view*: The network display panel shows all the networks which have been discovered. This list can be sorted and manipulated.
2. *Information view*: The information panel shows the total number of packets, current packet rate, amount of time the capture has been running, etc.
3. *Status view*: The status panel scrolls information and status events. Alerts appear in this panel, as well as in the alert popup.

Chapter 4

Configuration Problems

1. The Cisco drivers downloaded from Cisco's website do not support promiscuous mode, as confirmed by calling Cisco's technical support. As a result, one can just sniff wireless packets that are destined to the sniffer station and not every packet on the wireless medium. The Aironet drivers downloaded from sourceforge work well with Cisco 350 cards and support promiscuous mode.

2. Using Kismet, one can't restrict Cisco 350 cards to sniff on one particular channel. Cisco cards have tendency to hop between channels. Therefore, Cisco cards tend to miss many frames while hopping between channels. Cisco cards have a miss rate of around 30%. The only alternative I found was to use another vendor's wireless card.

3. With Cisco cards, no current drivers on linux reports signal strength correctly. The alternative is to use *prism2*-based cards that report the signal strength per frame received correctly.

4. In order to sniff IEEE 802.11 frames, the Kismet software puts the laptop in monitor mode which precludes the ability of the wireless card to send any data to the network. In order to send packets to the network, I had to use another network wireless interface.

5. The signal strength value reported by Kismet with a prism2 based Linksys card is a *Received Signal Strength Indicator* (RSSI) value. Each vendor has their own formula to convert this RSSI value to Decibel Milliwatts (dBm). I could not find this conversion formula for Linksys even after I called the Linksys technical support.

Chapter 5

Details of Experiments

Experiments done are divided into the following categories:

1. Handoff analysis when APs having different SSIDs are on different channels (Section 5.1).
2. Handoff analysis when APs having different SSIDs are on the same channel (Section 5.2).
3. Handoff analysis when APs having the same SSID are on different channels (Section 5.3).
4. Handoff analysis when APs having the same SSID are on the same channel (Section 5.4).
5. Signal strength at the point of handoff (Section 5.5).
6. Effect of Beacon Interval on handoff Latency (Section 5.6).

In all the experiments to follow, handoff is performed such that the STA leaves association with the AP on channel 11 and associates with an AP on channel 1.

5.1 Handoff analysis when APs Having Different SSIDs are on Different Channels

5.1.1 Detailed Procedure

The two APs named *Anshul-1* and *Anshul-2* form two overlapping cells. The power of each AP is adjusted so that the coverage areas for the APs overlap at some point and they are not coincident. AP *Anshul-1* is running on channel 11 and *Anshul-2* on channel 1. To force a handoff, the wireless client (STA) that is initially associated with *Anshul-1* is moved away towards *Anshul-2*. This is considered as one run of the experiment. At some point during this run, the STA disassociates from *Anshul-1* and associates with *Anshul-2*. During this handoff, a series of messages get exchanged between STA and APs. These messages are sniffed by wireless sniffer systems running the Kismet software [13].

One sniffer system is sniffing on the channel 1 and the other is sniffing on channel 11. To overcome the inaccuracy caused by inconsistent system clocks of the two sniffer laptops, we used the Network Time Protocol (NTP). The time server used is *time.uky.edu*, the local time server at University of Kentucky. In order to synchronize the clock with the time server, the sniffer laptops need to send NTP specific packets. Sniffer laptops being in monitor mode is unable to send any packets, and hence we have to use another ethernet interface on each of the two laptops to be able to synchronize using NTP with an estimated accuracy of $5-10$ ms.

As explained in the handoff procedure in Section 2.1, a fixed set of messages are sent by both the STA and APs. It is these messages we intend to capture and study using the experiments setup described above. We collected all the IEEE 802.11 management frames and then filtered out the rest to get *Probe Request*, *Probe Response*, *Authentication*, *Reassociation Request* and *Reassociation Response*. Total handoff delay is then calculated by counting time difference between the first *probe request* (on channel 1) and the *reassociation response* (also on channel 1).

5.1.2 Results and Analysis

The Fig. 5.1 shows the handoff latency when there are two APs with different SSID, *Anshul-2* on channel 1 and *Anshul-1* on channel 11. The graph depicts the results of 15 repetitions of the experiment with the experiment number on x-axis and handoff latency measured in milliseconds on y-axis.

As can be seen from the graph, probe delay is the major contributing factor towards the total handoff latency. It is more than 99% of the overall handoff latency. Authentication delay and the reassociation delay account for a total of less than 1% of a total handoff latency.

Average handoff latency across all the repetitions of the experiment is approximately 531.6 ms. Average probe delay, authentication delay and reassociation delay is approximately 528 ms, 1.3 ms and 2.3 ms, respectively. The 95% confidence interval for handoff latency is from 516.1 ms to

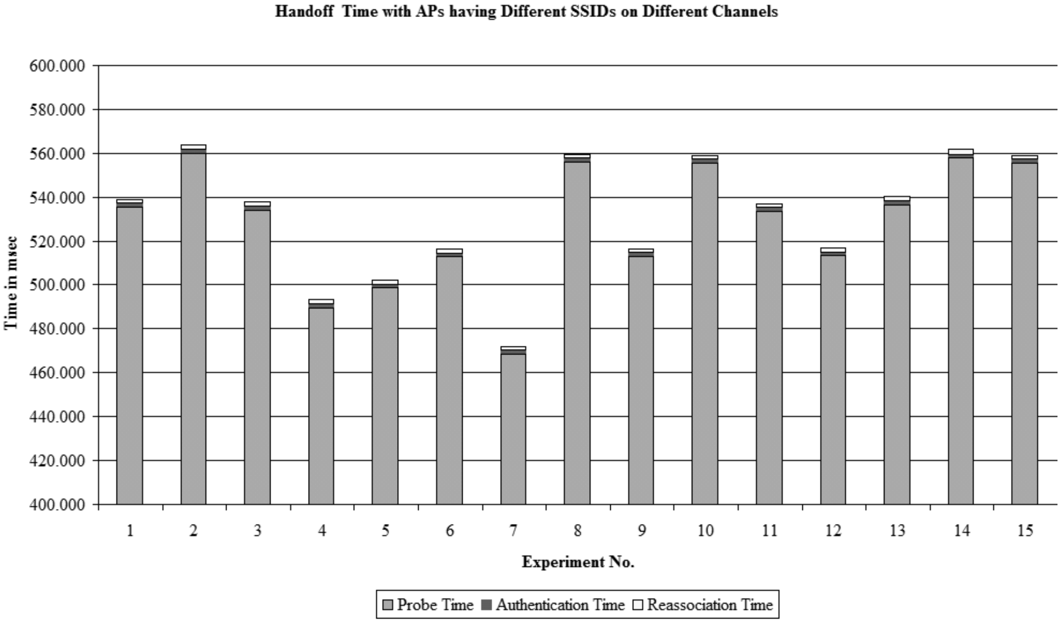


Figure 5.1: Handoff time when APs Having Different SSIDs are on Different Channels

547.106 ms calculated from [15]. This means if the same experiment is repeated 100 times, five times the handoff latency would fall outside the confidence interval.

Since probe delay is the most significant part of the handoff process, we will explain that in detail. By sniffing at two different channels simultaneously (synchronized with Network Time Protocol) in different combination a couple of times, we came to know that Cisco 350 card starts probe request from channel 1 no matter which channel the currently associated AP is on. Since we have configured the Cisco card to connect to either of *Anshul-1* or *Anshul-2*, the card sends two probe requests, one for each AP on each channel. That makes a total of 22 probe requests, which we confirmed by looking at their sequence numbers. The scanning process is expected to work according to the algorithm mentioned in Section 2.1. As can be seen in the Fig. 5.2, the probe process starts by broadcasting a probe request with SSID *Anshul-2* on channel 1. The wireless card then starts the probe timer. The AP *Anshul-2* sends a probe response back to the STA. The STA waits for the 37 ms since the transmission of first probe request and then sends another probe request with SSID *Anshul-1* on the same channel to search AP *Anshul-1*. The card does not get a response from the

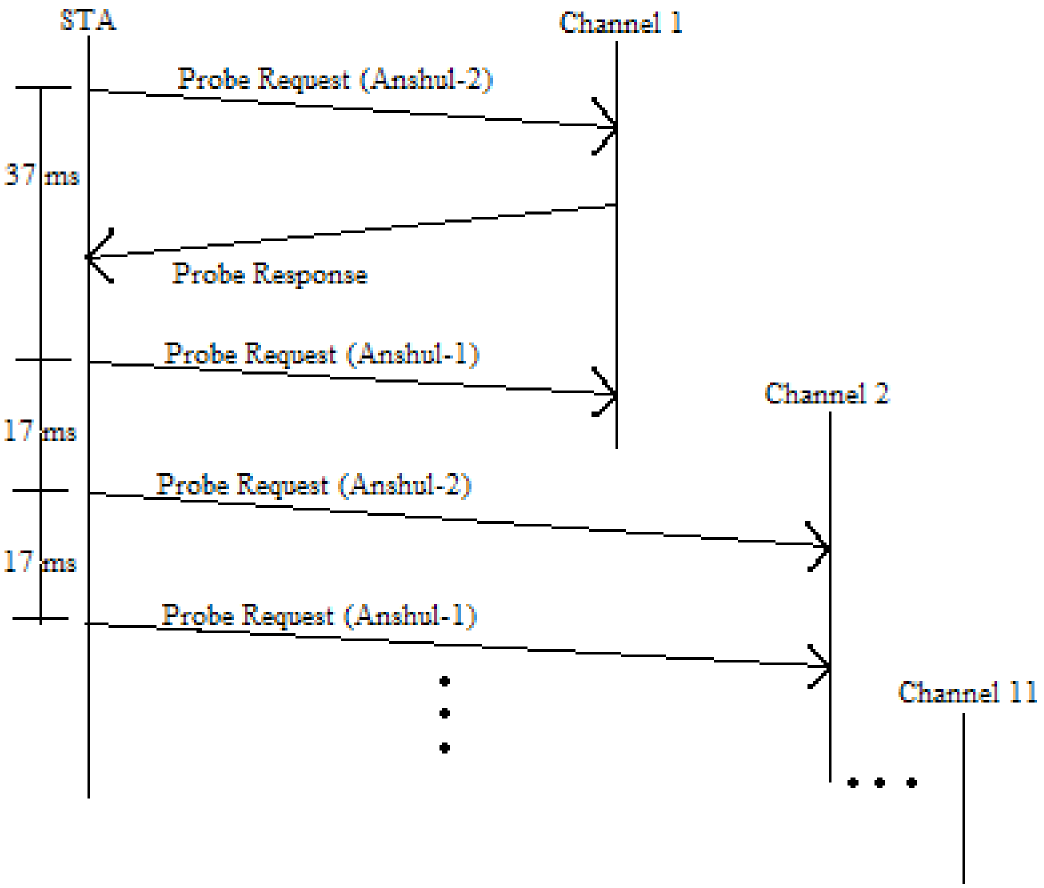


Figure 5.2: Probing Process

Anshul-1 within 17 ms and therefore goes to scan the channel 2. On channel 2, this process is repeated except that none of the two APs should send a response back to the STA. The values 17 ms and 37 ms discovered during experiments are *minChannelTime* and *maxChannelTime*, respectively, for the Cisco 350 card. The total expected probe time that is the time between first probe request on channel 1 and the reassociation response on channel 1 should be:

Channel 1: 37 ms delay for probe request to AP *Anshul-2* and 17 ms for probe request to *Anshul-1* for a total of 54 ms.

Channel 2 – 10: 17 ms delay for probe request to AP *Anshul-2* and 17 ms for probe request to *Anshul-1* for a total of 306 ms for nine channels.

Channel 11: 17 ms delay for probe request to AP *Anshul-2* and 37 ms for probe request to *Anshul-1* for a total of 54 ms.

This adds to a total of 414 ms expected probe delay. This is quite a bit less than the observed average probe delay of 528 ms. Looking in detail at the probe response messages, we observed that AP *Anshul-2* on channel 1 is sending a probe response messages in reply to the probe requests by STA on channels 2, 3 and 4 and similarly AP *Anshul-1* on channel 11 sends probe response messages in reply to the probe request by STA on channels 8, 9 and 10, when they should not. This is attributed to the channel allocation scheme for $802.11b$. As explained in Chapter 1, channel 1 and channel 4 overlap significantly and so do channel 8 and channel 11. Therefore, APs *Anshul-1* and *Anshul-2* overhear probe request messages on overlapping channels. Since the probe request messages do not contain a channel number, APs have no way to distinguish these overheard probe requests from those on the channels they are working on. And therefore, the APs respond by sending probe response messages back to the STA for these overlapping channels. The total expected probe time should be:

Channel 1: 37 ms delay for probe request to AP *Anshul-2* and 17 ms for probe request to *Anshul-1* for a total of $54\,ms$.

Channel $2-4$: 37 ms delay for probe request to AP *Anshul-2* and 17 ms for probe request to *Anshul-1* for a total of 162 ms for three channels.

Channel $5-7$: 17 ms delay for probe request to AP *Anshul-2* and 17 ms for probe request to *Anshul-1* for a total of 102 ms for three channels.

Channel $8-10$: 17 ms delay for probe request to AP *Anshul-2* and 37 ms for probe request to *Anshul-1* for a total of 162 ms for three channels.

Channel 11: 17 ms delay for probe request to AP *Anshul-2* and 37 ms for probe request to *Anshul-1* for a total of 54 ms.

This adds to a total of 534 ms expected probe latency which is approximately equal to the observed average probe latency of 528 ms.

There is a variation in handoff latency from one repetition of experiment to other. This can be accounted for lost probe requests due to weak signal strength as the STA moves away from the center frequency explained in Fig. 1.1. So, the APs may not hear the probe request messages on the overlapping channels with frequency some distance away from the center frequency of the channel they are working on. Since the STA does not get the probe response back because the AP missed the probe request, STA switches to next channel after waiting for *minChannelTime*. And, so there is a saving of *maxChannelTime - minChannelTime* on the channel.

5.2 Handoff analysis when APs Having Different SSIDs are on the Same Channel

5.2.1 Detailed Procedure

Both the APs *Anshul-1* and *Anshul-2* are on the same channel 11. To force a handoff the wireless client (STA) that is initially associated with *Anshul-1* is moved away towards *Anshul-2*. This is considered as one run of the experiment. One sniffer system is sniffing on channel 1 and the other is sniffing on channel 11. To overcome the inaccuracy caused by inconsistent system clocks of the two sniffer laptops, we used the Network Time Protocol (NTP). The time server used is *time.uky.edu*, the local time server at University of Kentucky. The total handoff delay is calculated by counting the time difference between the first *probe request* (on channel 1) and the *reassociation response* (on channel 11).

5.2.2 Results and Analysis

The Fig. 5.3 shows the handoff latency when there are two AP's with different SSID, both on channel 11. The graph depicts the results of 15 repetitions of experiment with experiment number on x-axis and handoff latency measured in milliseconds on y-axis.

As can be seen from the graph, probe delay is the major contributing factor towards the total handoff latency. It is more than 99% of overall handoff latency. Authentication delay and reassociation delay account for a total of less than 1% of total handoff latency.

Average handoff latency across all the repetetions of the experiment is approximately 532 ms. Average probe delay, authentication delay and reassociation delay is approximately 528.4 ms, 1.3 ms and 2.3 ms, respectively. The 95% confidence interval for handoff latency is from 510.306 ms to

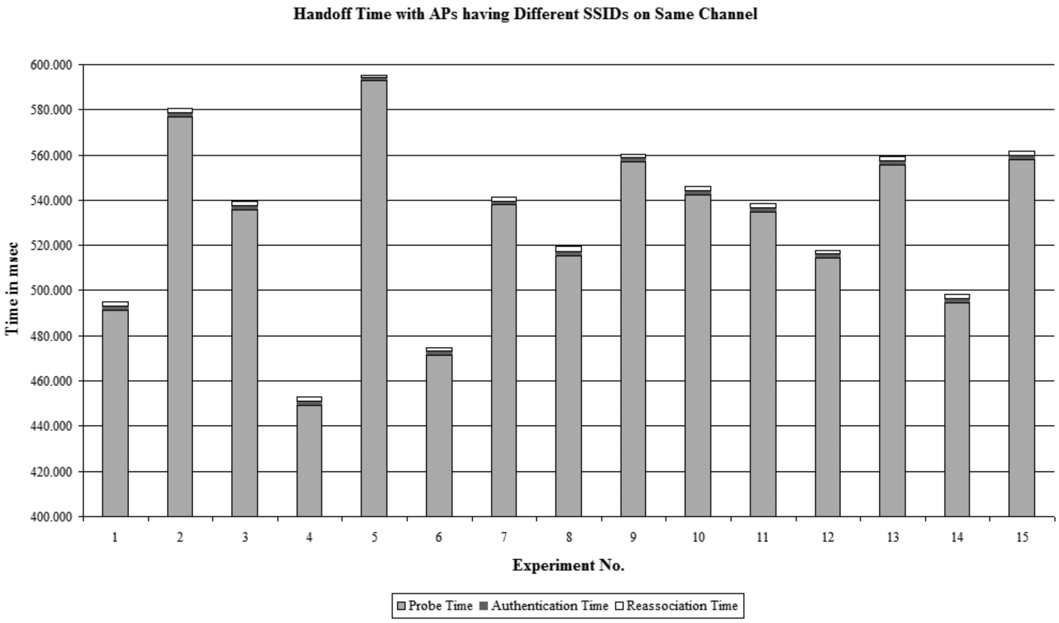


Figure 5.3: Handoff time when APs Having Different SSIDs are on Same Channel

553.694 ms calculated from [15]. This means if the same experiment is repeated 100 times, five times the handoff latency would fall outside the confidence interval.

Since both the APs *Anshul-1* and *Anshul-2* are on channel 11, therefore both the APs will respond back with a probe response for the probe request with respective SSIDs *Anshul-1* and *Anshul-2*. Also, as explained in the Section 5.1.2 above both the APs will overhear the overlapping channels and will respond back. Total expected probe time should be:

Channel 1 – 7: 17 ms delay for probe request to AP *Anshul-2* and 17 ms for probe request to *Anshul-1* for a total of 238 ms for seven channels.

Channel 8 – 10: 37 ms delay for probe request to AP *Anshul-2* and 37 ms for probe request to *Anshul-1* for a total of 222 ms for three channels.

Channel 11: 37 ms delay for probe request to AP *Anshul-2* and 37 ms for probe request to *Anshul-1* for a total of 74 ms.

This adds to a total of 534 ms expected probe latency which is approximately equal to the observed average probe latency of 528.4 ms. The variation in handoff latency from one run of experiment to other can be accounted for the same reason as stated in Section 5.1.2 above.

5.3 Handoff Analysis when APs Having the same SSID are on Different Channels

5.3.1 Detailed Procedure

For this experiment we changed the SSID of both the APs to *Anshul*. One of the two AP is running on channel 11 and other on the channel 1. One sniffer system is sniffing on channel 1 and the other is sniffing on channel 11. To overcome the inaccuracy caused by inconsistent system clocks of the two sniffer laptops, we used the Network Time Protocol (NTP). The time server used is *time.uky.edu*, the local time server at University of Kentucky. The total handoff delay is calculated by counting the time difference between the first *probe request* (on channel 1) and the *reassociation response* (also on channel 1).

5.3.2 Results and Analysis

The Fig. 5.4 shows the handoff latency when there are two AP's with same SSID, one on channel 1 and other on channel 11. The graph depicts the results of 15 repetitions of the experiment with the experiment number on x-axis and handoff latency measured in milliseconds on y-axis.

As can be seen from the graph, again probe delay again is the major contributing factor towards the total handoff latency. It is more than 99% of overall handoff latency. Authentication delay and reassociation delay account for a total of less than 1% of total handoff latency.

Average handoff latency across all the repetitions of the experiment is approximately 329.4 ms. Average probe delay, authentication delay and reassociation delay is approximately 325.8 ms, 1.3 ms and 2.3 ms, respectively. The 95% confidence interval is from 318.303 ms to 340.497 ms calculated from [15]. This means if the same experiment is repeated 100 times, five times the handoff latency would fall outside the confidence interval.

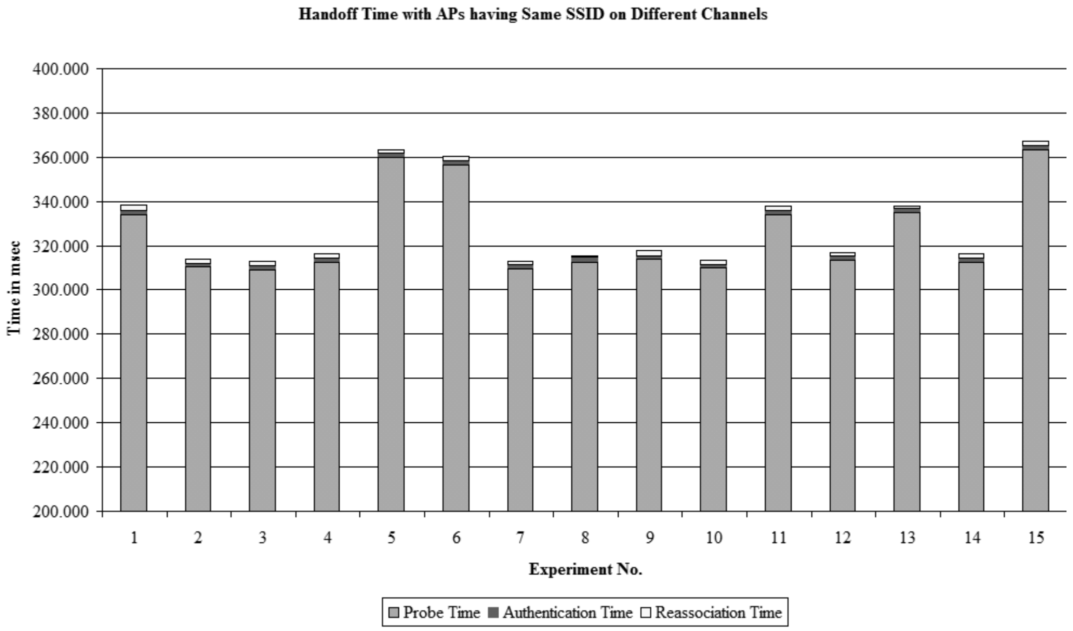


Figure 5.4: Handoff time when APs Having the same SSID are on Different Channels.

Since both the APs have same SSID, Cisco card will send only one probe request on each channel unlike two in above experiments. Total expected probe time should be:

Channel 1: 37 ms delay for probe request to AP *Anshul*.

Channel $2-4$: 37 ms delay for probe request to AP *Anshul* for a total of 111 ms for three channels.

Channel $5-7$: 17 ms delay for probe request to AP *Anshul* for a total of 51 ms for three channels.

Channel $8-10$: 37 ms delay for probe request to AP *Anshul* for a total of 111 ms for three channels.

Channel 11: 37 ms delay for probe request to AP *Anshul*.

This adds to a total of 347 ms expected probe latency which is approximately equal to the observed average handoff latency of 325.8 ms. The variation in handoff latency from one run of experiment to other can be accounted for the same reason as stated in Section 5.1.2 above.

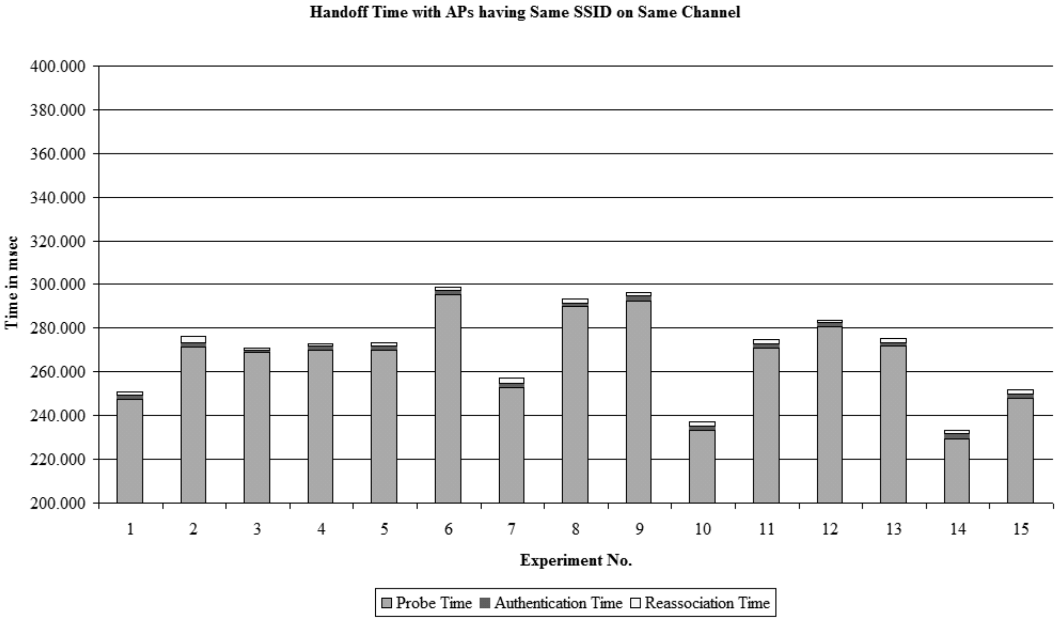


Figure 5.5: Handoff time when APs Having the same SSID are on the Same Channel.

5.4 Handoff Analysis when APs Having the same SSID are on the Same Channel

5.4.1 Detailed Procedure

For this experiment we changed the SSID of both the APs to *Anshul*. Both the APs are running on channel 11. One sniffer system is sniffing on channel 1 and the other is sniffing on channel 11. To overcome the inaccuracy caused by inconsistent system clocks of the two sniffer laptops, we used the Network Time Protocol (NTP). The time server used is *time.uky.edu*, the local time server at University of Kentucky. The total handoff delay is calculated by counting the time difference between the first *probe request* (on channel 1) and the *reassociation response* (on channel 11).

5.4.2 Results and Analysis

The Fig. 5.5 shows the handoff latency when there are two AP's with same SSID, both on channel 11. The graph depicts the results of 15 repetitions of the experiment with the experiment number on x-axis and handoff latency measured in milliseconds on y-axis.

As can be seen from the graph, once again probe delay is the major contributing factor towards the total handoff latency. It is more than 99% of overall handoff latency. Authentication delay and reassociation delay account for a total of less than 1% of total handoff latency.

Average handoff latency across all the repetitions of the experiment is approximately 269.7 ms. Average probe delay, authentication delay and reassociation delay is approximately 266.1 ms, 1.3 ms and 2.3 ms, respectively. The 95% confidence interval for handoff latency is from 258.564 ms to 280.836 ms calculated from [15]. This means if the same experiment is repeated 100 times, five times the handoff latency would fall outside the confidence interval.

Since both the APs have same SSID, Cisco card will send only one probe request on each channel. Also, for each probe request on channel 11, both the APs will respond for having the same SSID.

Total expected probe time should be:

Channel $1-7$: 17 ms delay for probe request to AP *Anshul* for a total of 119 ms for seven channels.

Channel $8-10$: 37 ms delay for probe request to AP *Anshul* for a total of 111 ms for three channels.

Channel 11: 37 ms delay for probe request to AP *Anshul*.

This adds a total of 267 ms expected probe latency which is approximately equal to the observed average probe latency of 266.1 ms.

5.5 Effect of Beacon Interval on Handoff Latency

5.5.1 Detailed Procedure

The aim of the experiment is to check if there is any effect on handoff latency if we decrease the beacon frame interval. For this experiment the beacon frame interval is reduced to 50 ms instead of the default value of 100 ms. The rest of the experimental setup is identical to the experiment in which APs having the same SSID are on different channels.

5.5.2 Results and Analysis

The Fig. 5.6 shows the handoff latency with the reduced beacon frame interval, when there are two AP's with same SSID, one on channel 1 and other on channel 11. The graph depicts the results of five repetitions of the experiment with the experiment number on x-axis and handoff latency measured in milliseconds on y-axis.

Average handoff latency across all the repetitions of the experiment is approximately 306.2 ms. Average probe delay, authentication delay and reassociation delay is approximately 302.6 ms, 1.3 ms and 2.3 ms, respectively.

This average handoff latency is approximately equal to the handoff latency when beacon frame interval was 100 ms with rest of the experiment setup being the same. This experiment shows that there is no effect of beacon interval on the handoff latency. While reducing the beacon interval to 50 ms from 100 ms, allow the STA to check the signal strength value faster and hence can make a decision to start a handoff process earlier. However, once the handoff is initiated, it will take the same time.

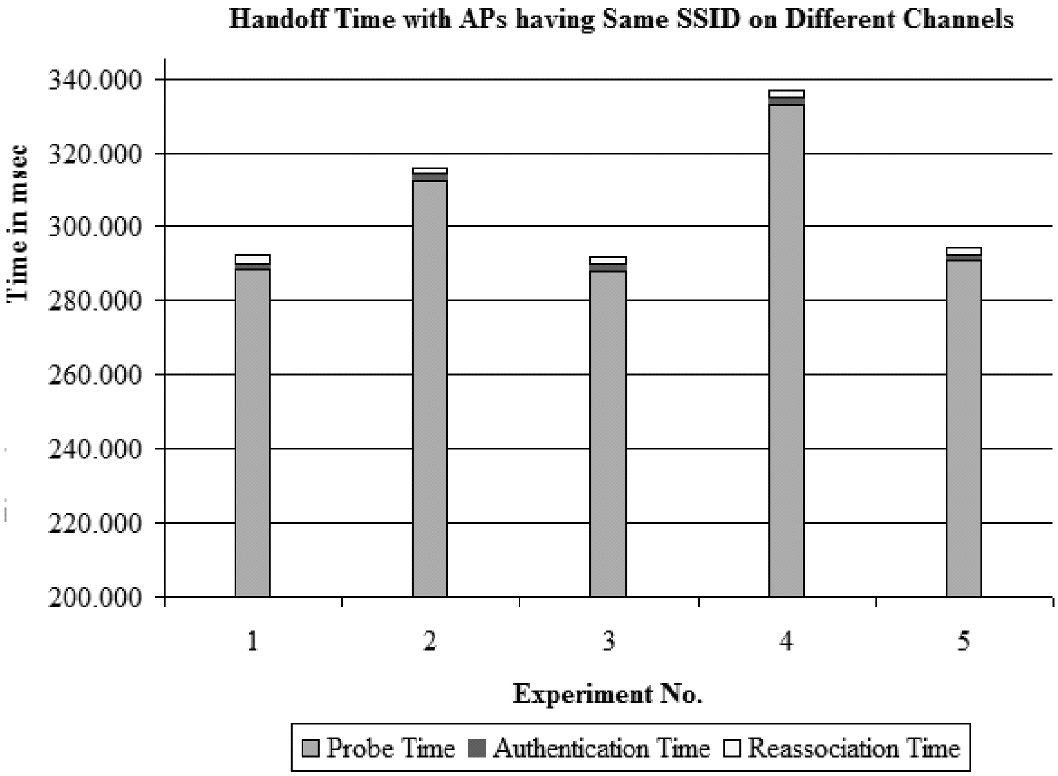


Figure 5.6: Effect of Beacon Interval on Handoff Latency

5.6 Signal strength at the Point of Handoff

5.6.1 Detailed Procedure

The signal strength value reported by Kismet with a prism2-based Linksys card is a *Received Signal Strength Indicator* (RSSI) value. Each vendor has their own formula to convert this RSSI value to Decibel Milliwatts (dBm). I could not find this conversion formula for Linksys. However, I got such a table for Cisco cards on http://www.wildpacket.com. Therefore, I used Cisco card instead of Linksys on sniffer system. Also, I did not trust the signal strength value reported by Kismet. So, I used Wildpacket's *Airopeek* software that runs on Windows XP.

For this experiment, SSID of the APs and the channel they are working on doesn't matter. We set both the APs to have the same SSID (*Anshul*). One of the APs is running on the channel 1 and the other one is running on channel 11. One of the sniffer systems is sniffing on channel 1 and other on channel 11. The sniffer system sniffing on channel 11 is running Wildpacket's *Airopeek* software on windows XP. The other sniffer system is running *kismet* software on linux Red hat 8.0 as in the other experiments. The sniffer system running *Airopeek* is moved along with the wireless STA. The assumption is that the moving sniffer system would see approximately the same signal strength value as seen by wireless STA performing handoff. To force the handoff the wireless STA that is initially associated with AP on channel 11 is moved away towards AP on channel 1. Signal strength is measured by looking at the signal strength value reported by *Airopeek* for the beacon frame just prior to the first probe request by the STA (on channel 1).

5.6.2 Results and Analysis

The Fig. 5.7 shows the signal strength at the point of handoff, when there are two AP's with same SSID, one on channel 1 and other on channel 11. The graph depicts the results of 10 repetitions of the experiment with the experiment number on x-axis and signal strength measured in decibel milliwatts on y-axis.

As can be seen in the graph, the signal strength value ranges from -75 dBm to -84 dBm with an average of 79.5 dBm. The corresponding RSSI values are 36%, 28% and 32%, respectively. The 95% confidence interval is from -81.926 dBm to -77.074 dBm calculated from [15]. This means if the same experiment is repeated 100 times, five times the handoff latency would fall outside the confidence interval.

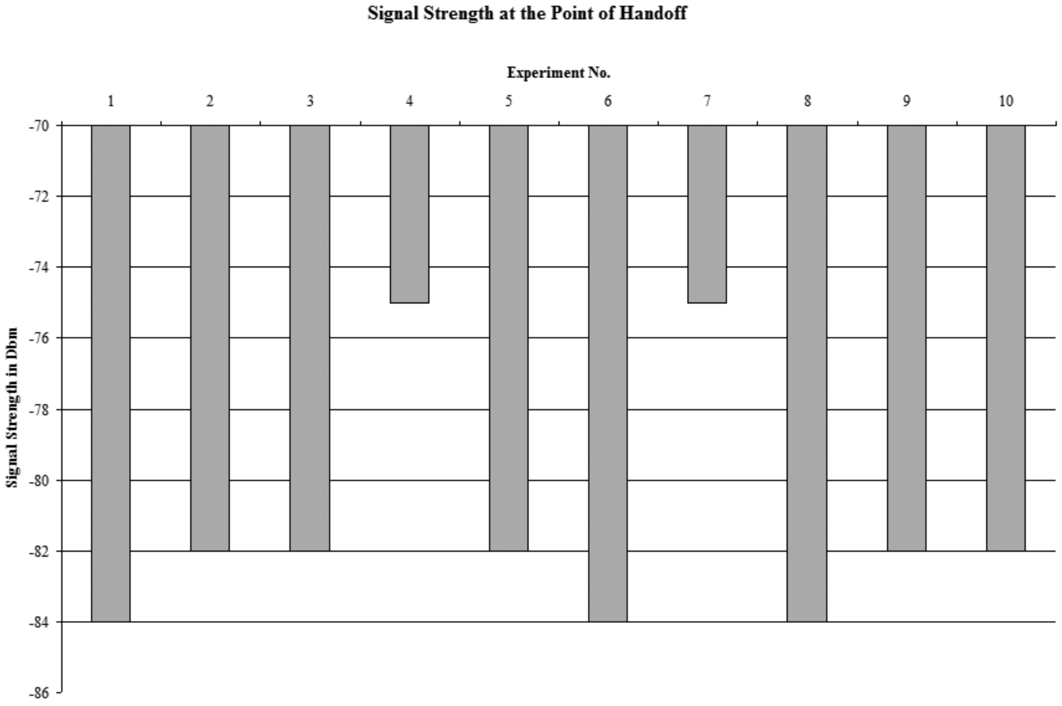


Figure 5.7: Signal strength at the point of handoff

Chapter 6

Summary of Experiments

Experiment	Expected Result	Observed Result
Handoff latency when APs with different SSIDs are on different channels	$534\,\text{ms}$	$531.6\,\text{ms}$
Handoff latency when APs with different SSIDs are on same channels	$534\,\text{ms}$	$532\,\text{ms}$
Handoff latency when APs with same SSIDs are on different channels	347 ms	$329.4\,\text{ms}$
Handoff latency when APs with same SSIDs are on same channels	267 ms	266.1 ms
Effect of beacon interval on handoff time	347 ms	$306.2\,\text{ms}$
Signal strength at the point of handoff	-	$-79.5\,\text{dBm}$

Chapter 7

Conclusion

In the project we studied handoff process in detail. We find out that there are three logical steps involved in handoff: probe, authentication and reassociation. Probe delay accounts for more than 99% of overall handoff latency. We also find out that there are significant variations in handoff latancies when the two APs are on different channels and also when they have different SSID. We saw that the handoff latency is smallest when we have APs with same SSID and on the same channel. The probe process starts when the signal strength of the beacon frames received goes below a certain threshold (specific to each vendor). There is no way to change this threshold, but, changing the threshold will not effect the handoff latency. This is because reducing or increasing the threshold will just effect the start-time of handoff process and not the overall handoff latency. We also did experiments to check if the beacon frames interval has any effect on handoff latency. Reducing the beacon interval from the default value of 100 ms to 50 ms started the probe process 50 ms earlier. But, once started the handoff took approximately the same time. So, beacon interval did not have any effect on handoff latency. Another interesting result is that the handoff latencies we measured far exceed the guidelines for jitter in voice over IP applications where the overall latency is recommended not to exceed 50 ms [14].

Bibliography

[1] A. Mishra, M. Shin, W. Arbaugh. *An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process*. Technical report, University of Maryland, 2002.

[2] *IEEE Std. 802-11b, 1999, Wireless LAN Medium Access Control (MAC) and Physical (PHY) specifications: High Speed Physical Layer Extension in the 2.4 GHz Band*. IEEE, New York, NY, 1999.

[3] W. Arbaugh, N. Shankar, J. Wan. *Your 802.11 Wireless Network has No Clothes*. Technical report, Department of Computer Science, University of Maryland, College Park, 2001.

[4] Jiang Wu. *A Mobility Support Agent Architecture for Seamless IP Handover*. Technical report, Royal Institute of Technology, Sweden, June 2000.

[5] B. Hara, Al Petrick. *The IEEE 802.11 Handbook: A Designer's Companion*. IEEE, 1999.

[6] J. Geier. *Understanding 802.11 Frame Types*. Jupitermedia Corporation, August 2002.
URL : wi-fiplanet.com/tutorials/article.php/1447501

[7] J. Tourrihes. *Wireless LAN resources for Linux*. Hewlett Packard, 1996.
URL : hpl.hp.com/personal/Jean_Tourrihes/Linux/

[8] J Bardwell. *Converting Signal Strength Percentage to dBm Values*. November 2002.
URL : wildpackets.com/elements/whitepapers/Converting_Signal_Strength.pdf

[9] P. Brenner. *A Technical Tutorial on the IEEE 802.11 Protocol*. July, 1996.
URL : sss_mag.com/pdf/802_11tut.pdf

[10] O. Meir, I. Bar. *IEEE 802.11 WLAN*. 1999.
URL : wwwcomnet.technion.ac.il/ cn3s02/presentations/IEEE_802.11_WLAN.ppt

[11] R. Cole, J. Rosenbluth. *Voive Over IP Performance Monitoring*. ACM Computer Communication Review, vol. 31, no. 2, pp. 9–24, April 2001.

[12] G. Combs. *Ethereal Network Protocol Analyser*.
URL : ethereal.com

[13] M. Kershaw. *Kismet Wireless Network Sniffer*.
URL : kismetwireless.net

[14] International Telecommunication Union. *General Characterstics of International Telephone Connections and International Circuits*. ITU-TG.114, 1998.

[15] S. Waner and S. Costenoble. *Confidence Intervals, Miscellaneous on-line topics for Finite Mathematics 2e*. September, 2000.