國立交通大學

資訊科學與工程研究所

碩士論文

802.11e 無線區域網路下之跨階層 VoIP 應用程式控制系統

A Cross-Layer Control Scheme for VoIP Application over 802.11e Wireless Local Area Networks

研究生：詹雯甄

指導教授：簡榮宏　教授

中華民國　九十五　年六月

802.11e 無線區域網路下之跨階層 VoIP 應用程式控制系統

A Cross-Layer Control Scheme for VoIP Application over 802.11e Wireless Local Area Networks

研 究 生：詹雯甄　　Student：Wen-Chen Chan

指導教授：簡榮宏　　Advisor：Rong-Hong Jan

國 立 交 通 大 學

資 訊 科 學 與 工 程 研 究 所

碩 士 論 文

A Thesis

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

中華民國九十五年六月

802.11e 無線區域網路下之跨階層 VoIP 應用程式控制系統

研究生：詹雯甄　　指導教授：簡榮宏 博士

國立交通大學資訊科學與工程研究所

摘　要

近幾年來 VoIP 服務深受到人們的喜愛，成為重要的應用服務。而由於無線網路的行動特性，人們在無線網路上使用 VoIP 服務更為方便。在目前常見的無線區域網路中，IEEE 802.11e 無線區域網路可以提供 VoIP 較為良好的服務品質，但未考慮到無線頻道狀況改變帶來的影響，它需要一個調變機制來補足這一點。因此，在本篇論文中，我們實作出一個可隨著無線網路頻道訊號的狀況調變的 VoIP 應用程式。除此之外，我們也實作出一個具管理策略的 802.11e 無線區域網路，以保證有申請服務之使用者權益。最後的實驗結果顯示使用可調變的 VoIP 應用程式會比傳統的固定式 VoIP 應用程式擁有更好的效能。

A Cross-Layer Control Scheme for VoIP Application over 802.11e Wireless Local Area Networks

Student：Wen-Chen Chan　　Advisor：Dr. Rong-Hong Jan

Institute of Computer Science and Engineering

National Chiao Tung University

Abstract

In the recent years, Voice over Internet Protocol (VoIP) is the most popular technology and will be a kind of killer application. Since Wireless Local Area network (WLAN) has a characteristic of mobility, people who use VoIP over WLAN can be more convenient. In the WLAN, IEEE 802.11e can support VoIP with better Quality of Service (QoS). However, 802.11e doesn't consider the time-varying condition of wireless channel. It needs an adaptive mechanism to complement it. Thus, in this thesis, we implement a cross-layer adaptive VoIP application which can use a suitable codec according to current channel condition. Besides, we also implement a policy-based 802.11e WLAN to protect the right of user who has applied real-time services. The experiment results show the proposed cross-layer adaptive VoIP application can archive better performance than the fixed one.

Contents

List of Figures

Chapter 1

Introduction

Voice over Internet Protocol (VoIP) has become the most popular technology. People are amazed at its benefit which is low cost; therefore we believe that VoIP will be a kind of "Killer Application". Note that VoIP works under a network coverage. Wireless Local Area Network (WLAN) extends the network coverage and provides network access for WLAN users. Applying VoIP upon WLAN is called as "Voice over WLAN (VoWLAN)". In the future, the coverage of WLAN can be extend to all the living areas and VoWLAN will be similar to the cellular phone that we use it to talk everywhere.

VoIP is a real-time application, which has some strict Quality of Service (QoS) requirements, such as delay, packet loss and jitter limitation. For example, VoIP cannot afford the one-way delay of application larger than 150 millisecond (ms), in this situation VoIP users may consider the voice quality too bad. However, WLAN has two crucial limitations, which are low bandwidth and channel variation. Since the bandwidth of WLAN is lower than wired network, if there is a large amount of traffic transmitted in the WLAN at the same time, it will affect the quality of VoIP application. WLAN channels can be affect by interference,

propagation loss, and multi-path fading, therefore the channel condition varies with time. The varying channel makes the quality of VoIP unstable and may have a large delay or packet loss.

Many cities establish 802.11 WLAN in their subway stations and campuses have 802.11 WLAN in their buildings. However, 802.11 cannot support QoS by itself, it is just a general MAC protocol. To solve this problem, IEEE Working Group E enhances IEEE 802.11 to support QoS and they call this new protocol "IEEE 802.11e" [1]. 802.11e gives VoIP traffic higher priority than best effort traffic, so VoIP traffic has more chance to transmit packets and VoIP application can performs better in 802.11e than 802.11. Even though 802.11e can support QoS, it cannot guarantee QoS with its time varying channel. The parameter set and scheduling methods of the original 802.11e protocol is fixed and cannot adapt to varying channel condition, therefore it needs some adaptive mechanism.

There are two kinds of mechanisms which can do some adaptation according to channel condition. First, one adapts 802.11e MAC parameters to current channel condition. Adaptive EDCF (AEDCF) [2] increases goodput by adapting CW parameter to reduce the number of collisions when network load is high. Second method is the cross-layer mechanism. High layer protocol uses lower layer information to change their scheduling routines or protocol parameters. In [3], they proposes a cross-layer communication scheme for video stream over 802.11 WLAN. In this scheme, video streaming application can obtain the link layer information to adapt to various kinds of scenarios.

In order to give IEEE 802.11e protocol the ability to adapt to a varying environment, thus VoIP over WLAN can perform well enough. We propose and implement a cross-layer adaptive VoIP application over 802.11e WLAN. Our VoIP

application dynamically changes the voice codec according to current channel condition which is reflected by Received Signal Strength Indication (RSSI)which is the signal strength between mobile station and Access Point (AP). Besides, our WLAN is a policy-based 802.11e WLAN, it can guarantee that only traffic generating from user who has applied special service like VoIP will be served as higher priority. Using policy-based network can realize the concept of "paying user fee" to protect user's right.

The remainder chapters of this thesis are organized as follows: chapter 2 gives related works and some background knowledge. Chapter 3 states our policy based 802.11e WLAN architecture and cross-layer adaptive VoIP application mechanism. The experiment environment and performance evaluation are given in chapter 4. The conclusions and future work are given in chapter 5.

Chapter 2

Related Works and Background Knowledge

In this chapter, first, we will give related works about QoS of VoIP which includes 802.11e, adaptive VoIP codec and cross-layer mechanism. Then, we will introduce the background knowledge used to implement our proposed scheme.

2.1 IEEE 802.11e Wireless LAN

2.1.1 Overview

In this section, we briefly introduce the IEEE 802.11e protocol. 802.11e is a Medium Access Control (MAC) Protocol proposed by IEEE Working Group E. 802.11e enhances the MAC procedures of 802.11 to support some specific LAN applications (e.g. voice, audio and video application) to meet their Quality of Service (QoS) requirements over IEEE 802.11 WLAN. 802.11e has become the IEEE standard. Corresponding to the DCF and PCF in 802.11, the hybrid coordination function (HCF) of 802.11e combines functions from DCF and PCF with some enhanced QoS-specific mechanisms. HCF consists of two channel access method, Enhanced Distributed Channel Access(EDCA) and HCF Controlled Channel Ac-

cess (HCCA). In this thesis, we only focuses on EDCA.

2.1.2 EDCA

EDCA enhances DCF by adding a new concept ,called as Access Category (AC). With this concept, EDCA classifies all network traffics into four kinds of ACs and takes different action on different AC, thus traffics can be prioritized.

In a station having EDCA, each AC owns its transmission queue and transmission parameter set. These parameters include:

- Arbitrary InterFrame Space Number(AIFSN)

  AIFS is correlated with the DIFS in 802.11. Stations must wait the network medium being idle for a AIFS time before starting backoff.We can use AIFSN to derive the value of AIFS, which is explained by eq 2.1.

$$AIFS\,[AC] = AIFSN\,[AC] \times aSlotTime + aSIFSTime \tag{2.1}$$

- Maximum Contention Window (CW_{max})

  CW_{max} is the same as it in 802.11, which defines the value of maximum contention window.

- Minimum Contention Window(CW_{min})

  CW_{min} is the same as it in 802.11, which defines the value of minimum contention window.

- Transmission Opportunity (TXOP)

  The maximum interval of time in which the station can transmit data since it has obtained the access right.

Since each AC has its own parameter set, the value of AIFS and Contention Window (CW) size depend on AC.

Figure 2.1 displays the contention procedure which is illustrated by two ACs. AC_0 can has more chances to obtain the medium access right by two reasons. First, the AIFS of AC_1 is longer than AC_0 which implies that the frame of AC_1 must wait loner than AC_0 before it can start backoff procedure. Second, the random backoff value is chosen among (0, CW⟨AC⟩), but CW⟨1⟩ is larger than CW⟨0⟩, thus the random backoff value of AC_1 is bigger than AC_0. From these two reason, we can figure out why AC_0 has more access chance than AC_1.

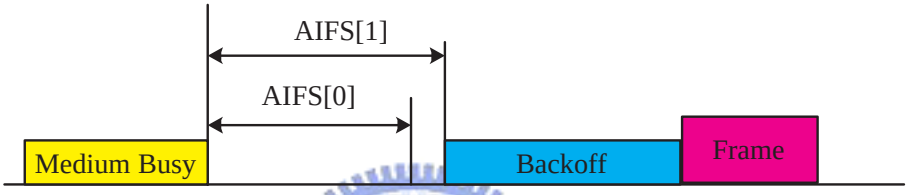

Figure 2.1: 802.11e contention procedure.

Traffics in one station are classified and then put in their traffic queue where they are waiting to send. Figure 2.2 displays the four traffic queues in one station. In 802.11e, traffics contend medium access right by AC not by station.

2.1.3 Bad link problem

In [4], they show that EDCA can support better QoS than DCF and PCF during low and medium load conditions. However, the EDCF-based ad-hoc network saturates and throughput decreases when the load increases. In [5], they mention that if the high priority users increase, high priority frames must wait long time in queue. WLAN users vary over the time scale of hours to days, thus it is hard to guarantee QoS. Especially, some high priority users who are in the

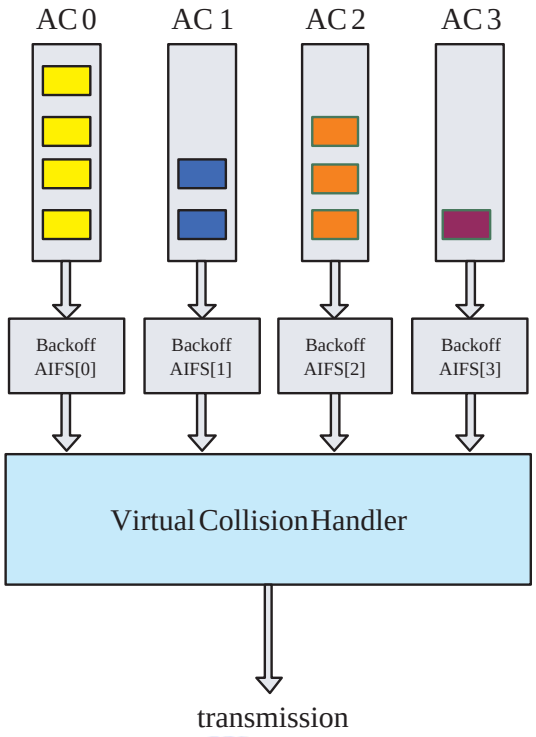

Figure 2.2: Four AC queues.

poor channel condition may waste channel resource. Figure 2.3 illustrates this case. When the signal strength between mobile station and AP turns in to weaker, the Network Interface Card (NIC) driver in the mobile station will automatically select a modulation scheme with lower data rate to encode data. Comparing with the strong signal condition in which NIC driver use the modulation scheme with high data rate, when we transmit the same size packet, the former needs longer transmission time which wastes much more network resource. The evaluation result of [6] displays how the "bad link" condition declines the performance of IEEE 802.11e WLAN. This situation will affect the quality of VoIP application, therefore we must extend 802.11e with some mechanisms to consider the time varying network condition. Cross-layer mechanism is this kind of mechanism, and we will

introduce it in the next section.

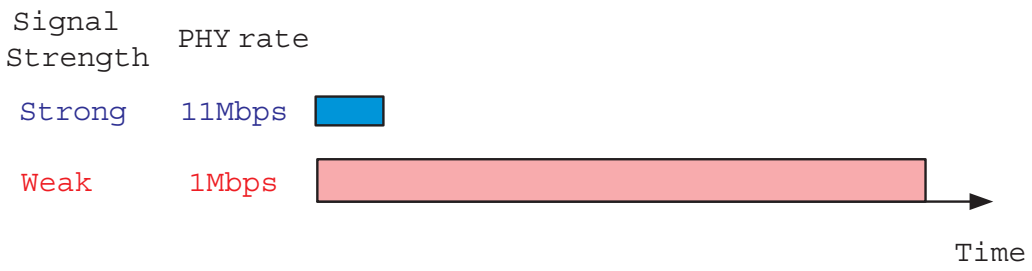

Figure 2.3: Time comparison of two modulation mode.

2.2 Cross-Layer Mechanism

Layering network model is the most common network model. Its concept is that layer independent makes the design of network protocol simpler. The network designers of one of these layers only need to work within their own layer, however it makes protocol implementation inefficient.

Cross-Layer Design is a new concept to resolve this situation. It seems like to merge all network layers into a flat layer, thus information of one layer can be shared with all other layers.

In [7], it shows that VoIP with Adaptive Multi-Rate (AMR) codec can improve its voice quality better than a traditional codec with fixed data rate. Therefore we can use the cross-layer concept to design a cross-layer adaptive VoIP application which makes VoIP application change codec with the time varying channel of WLANs. We will explain our method in the later chapter.

2.3 Policy-Based Network Management (PBNM)

2.3.1 Overview

In 802.11e WLAN, although stations can obtain QoS by service differentiation, it needs an admission control mechanisms to let AP know which stations are allowed and what kind of actions they should take on these stations. Although 802.11e has defined admission control schemes for EDCA and HCCA, they are implement-dependent. May be some APs have implemented this but others don't. This makes network administrators confused and can be troublesome, thus we need a centralized admission control scheme. It means the admission scheme is not implemented in AP but in AP's upper layer. One of these mechanisms is Policy-Based Network Management (PBNM) strategy. In this thesis, our system architecture is based on Policy-Based Network Management. Policy is defined as a set of rules. Each policy rule consists of a condition clause and an action clause. If the condition met, the actions in the action clause are allowed to execute. AP uses these rules to allocate network resource to many stations and schedules them. Policy-based management is defined as the usage of rules to manage the configuration and behavior of one or more entities. PBNM systems have one important benefit, which the majority of network configuration tasks are simple in nature and do not require a specialist. The administrator utilizes this benefit to build a central-controlled network architecture which makes the management task easier.

2.3.2 PBNM Architecture

Figure 2.4 shows the IETF defined PBNM architecture which basically consists of following components [8]:

- Policy repository

  Policy repository is a container which is used by administrator to store policy information. We usually seem policy repository as a network-connected database. Someone who needs policies would acquire policy information from Policy repository through network.

- Policy Console

  Policy Console provides administrator an interface to create policy rules, where policies are defined in a high-level declarative language. After validation and conflict test, policies are translated into Object-Oriented representations and then stored into policy repository.

- Policy Decision Point (PDP)

  RFC3198 [9] defines policy server as "A marketing term whose definition is imprecise". Now this term became more precise and known as PDP. RFC3198 defines PDP as "A logical entity that makes policy decisions for itself or for other network elements that request such decisions". PDP has a policy consumer in it and use policy consumer to retrieve policy information from policy repository.

- Policy Enforcement Point (PEP)

  RFC3198 [9] defines PEP as "A logical entity that enforces policy decision". PEP would request policy information from the PDP within its domain and continually monitor if any condition meets the condition clause of some policies. If there are some policies met, the policy actions will be executed on the target stations immediately.

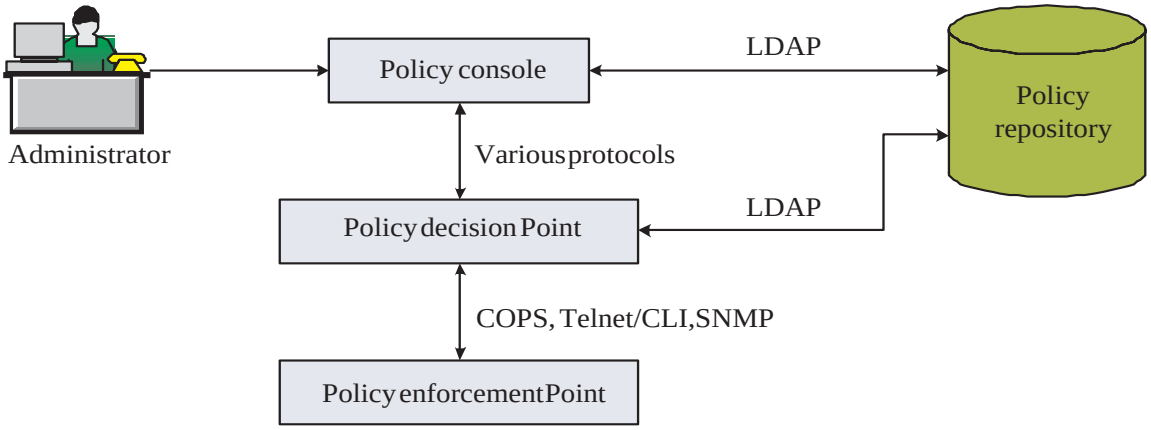

Figure 2.4: Policy-based Network Management system.

In [10], they propose a policy-based 802.11e network. It adds a server - Wireless QoS Enhancer(WQE) to 802.11e WLAN. WQE acts as the PEP and 802.11e AP as the PDP of the PBNM system. WQE has a database which holds the information about what kind of service user has been applied. Every time while a new flow wants to go through AP, the AP will query WQE about information of this flow to decide what kind of actions (set priority) should take on this flow. If WQE has no information about this flow, AP will treat this flow just as "best effort" traffic. Although WQE provides a good central-controlled admission control scheme for 802.11e, there is still a disadvantage. The judgement method of WQE is by flow. Every time a new flow comes, AP queries WQE once and gets a flow information. AP must store the information of many flows, thus it is not suitable for some APs which have strict memory limitation and the network cannot be scalable. Later we will propose a new policy-based 802.11e network judging by client which saves memory size and reduce query overhead.

In our architecture, we would like to let RADIUS server act as the PDP and AP as the PEP. For this goal, we utilize the flexibility of RADIUS protocol. In

the next section, we will introduce the concept and RADIUS protocol.

2.4 RADIUS Protocol

2.4.1 Overview

RADIUS is a client/server protocol and software, in which remote client can communicate with a central server to authenticate and authorize user's access to the requested service. In 802.11 WLAN, AP operates as a RADIUS client which passes the information of user authentication to RADIUS server and receives the response from RADIUS server to verify if this user is legal. The response packet may contain not only authentication response but also other useful information. The reason why RADIUS packet contains much information is that RADIUS packet is composed of many different type of attributes.

2.4.2 Operations

Figure 2.5 illustrates the entire authentication procedure of IEEE 802.1x, but our approach only focuses on the right side of this procedure, the RADIUS authentication procedure.

First, seeing the step 5 in Figure 2.5, the authenticator (Access Point) starts a RADIUS-Access-Request. Then, RADIUS server sends a RADIUS-Access-Challenge and waits the challenge text which comes from authenticator. Authenticator puts the challenge text in the Radius-Access-Request and sends to RADIUS Server. After receiving, RADIUS server validates this challenge text. If it is correct, then RADIUS server will send a RADIUS-Access-Accept and the supplicant can start access the network. But if not, RADIUS server will send a RADIUS-Access-Reject and the supplicant can not access network.

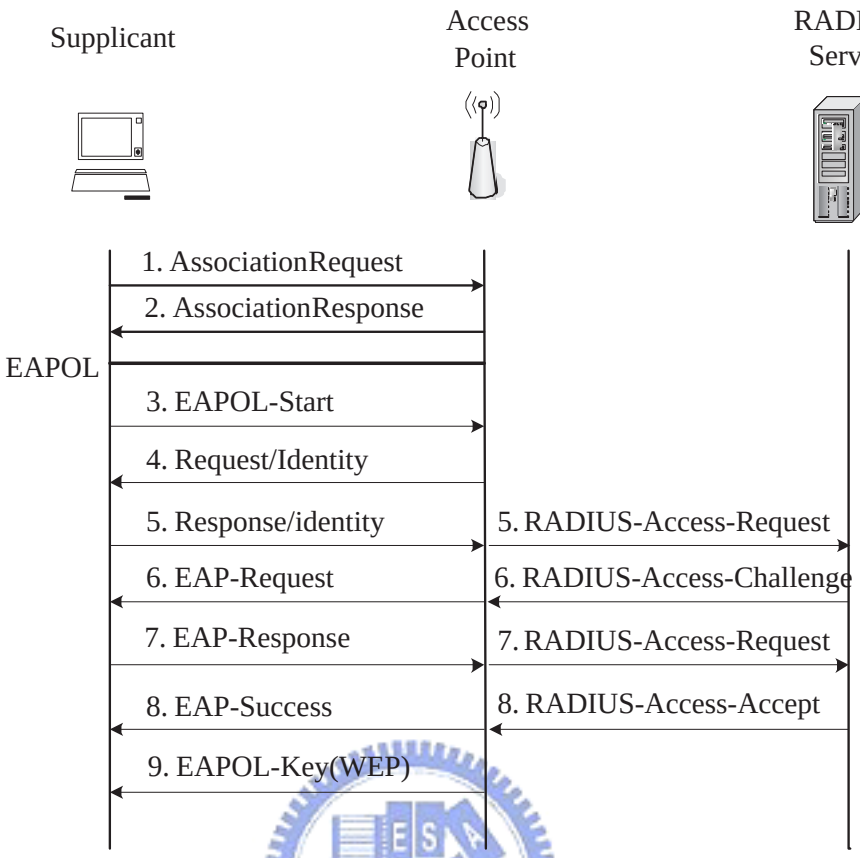

Figure 2.5: IEEE 802.1x authentication procedure.

2.4.3 Attribute

RADIUS packet is consist of many attributes. Figure 2.6 shows its attribute format [11]. Each attribute is composed of variable length Type-Length-Value 3-tuples. Besides, RADIUS has a special attribute - "Vendor-Specific". Vendors of network equipment can use this attribute to carry some useful information. Figure 2.7 shows the format of Vendor-Specific Attribute (VSA).

First, the Vendor-Id field is used to distinguish each vendor. The vendor type field is a sub-type of this attribute, therefore the same vendor can carry many kinds of information. Last, we put the information or value into Attribute-Specific

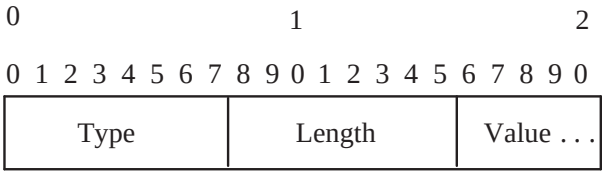

Figure 2.6: The format of RADIUS attribute.

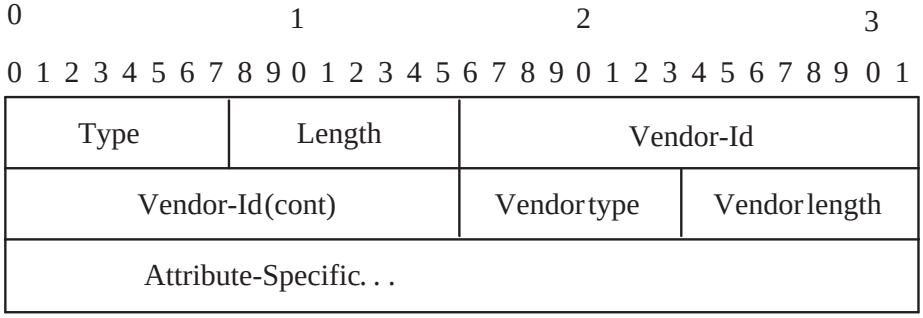

Figure 2.7: The format of RADIUS Vendor-Specific Attribute.

field. Vendor-Specific is such a useful attribute that we can use it to achieve our goals.

Chapter 3

System Architecture

In this chapter, we describe the proposed system architecture which includes policy-based 802.11e WLAN and cross-layer adaptive mechanisms. First, we will give you an overview of our mechanism is given, then explain these two main mechanisms.

3.1 The Proposed Mechanism

In chapter 2, we have mentioned about the problem of 802.11e. 802.11e protocol doesn't consider the time-varying channel condition of WLAN. However, the cross-layer mechanism can do adaptation according to current channel environment. Therefore we propose a scheme to combine 802.11e with a cross-layer VoIP application. With this scheme, we think VoIP application can perform well in the changeable wireless network.

In figure 3.1, we use a protocol stack graph to illustrate our mechanism. In the data link layer, we use 802.11e MAC protocol to handle the media access contention event and application layer is a cross-layer VoIP application. The arrow upon protocol stack graph explains that upper layer VoIP application can refer lower layer information to adapt to the current environment.

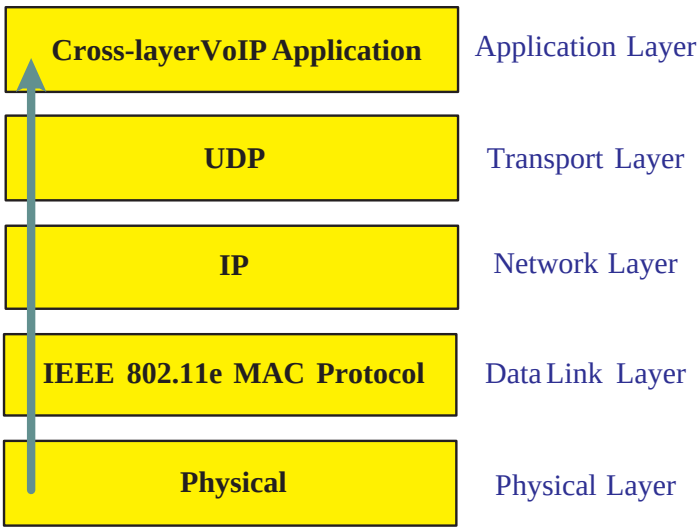

Figure 3.1: Protocol stack of proposed mechanism.

3.2 Policy-based 802.11e WLAN

Figure 3.2 shows our system architecture of the policy-based 802.11e WLAN. In this PBNM system, we use RADIUS server as the PEP, in which we store the Service Level Agreement (SLA) of each client. When a mobile station associates to AP, the hostap daemon (hostapd) in AP will help station to finish authentication procedure. First, hostapd send a Access-Request to RADIUS server to start the authentication process. Then, just as the regular RADIUS authentication procedure we have described in figure 2.5. Finally, if it is proved that the user is an authenticated one, RADIUS server responds an Access-Accept message to the hostapd with a SLA content. After finishing authentication procedure, every time while packets of this station go through the AP, it uses the SLA of this station to determine setting these packets into what kind of AC.

Figure 3.3 displays the functional block of RADIUS server, hostapd and AP. Following, we will explain the detail processes of them. (See section 3.2.1, 3.2.2

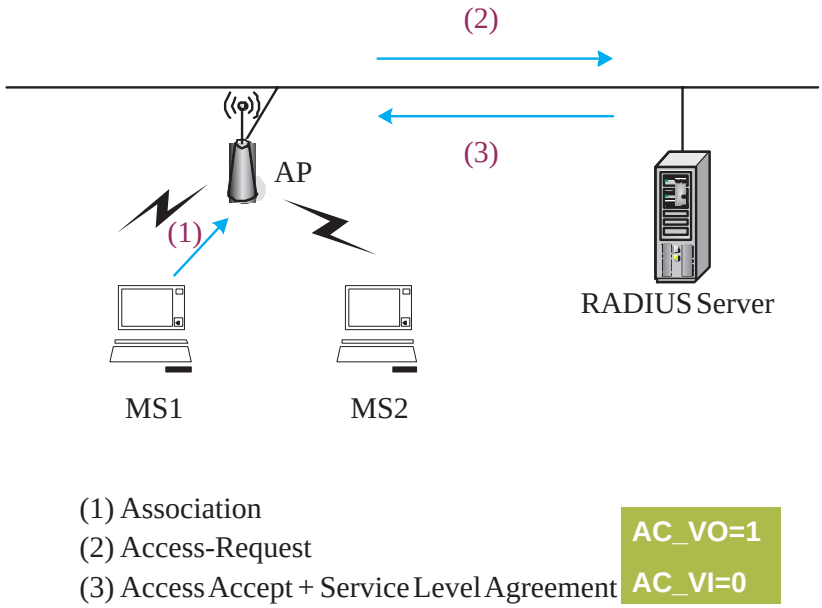

Figure 3.2: Policy-based 802.11e WLAN.

and 3.2.3)

3.2.1 RADIUS Server

We use an open-source software - FreeRADIUS [12] as our RADIUS server. Since FreeRADIUS is an open-source software, we can get the source code and modify it as we want.

In chapter 2, we have mentioned the Vendor-Specific Attribute of RADIUS message. The vendors of RADIUS server can use it to add some extra functions they need. As a vendor, before using this attribute, we must create our own vendor id and type to distinguish the contents which is added by us. We define the vendor name as netlab, vendor id as 2810 and create a new subtype - "11E" for VSA, thus we can use this sub-type to carry the SLA information to 802.11e AP.

Figure 3.4 shows the file which stores vendor name and sub-type values.

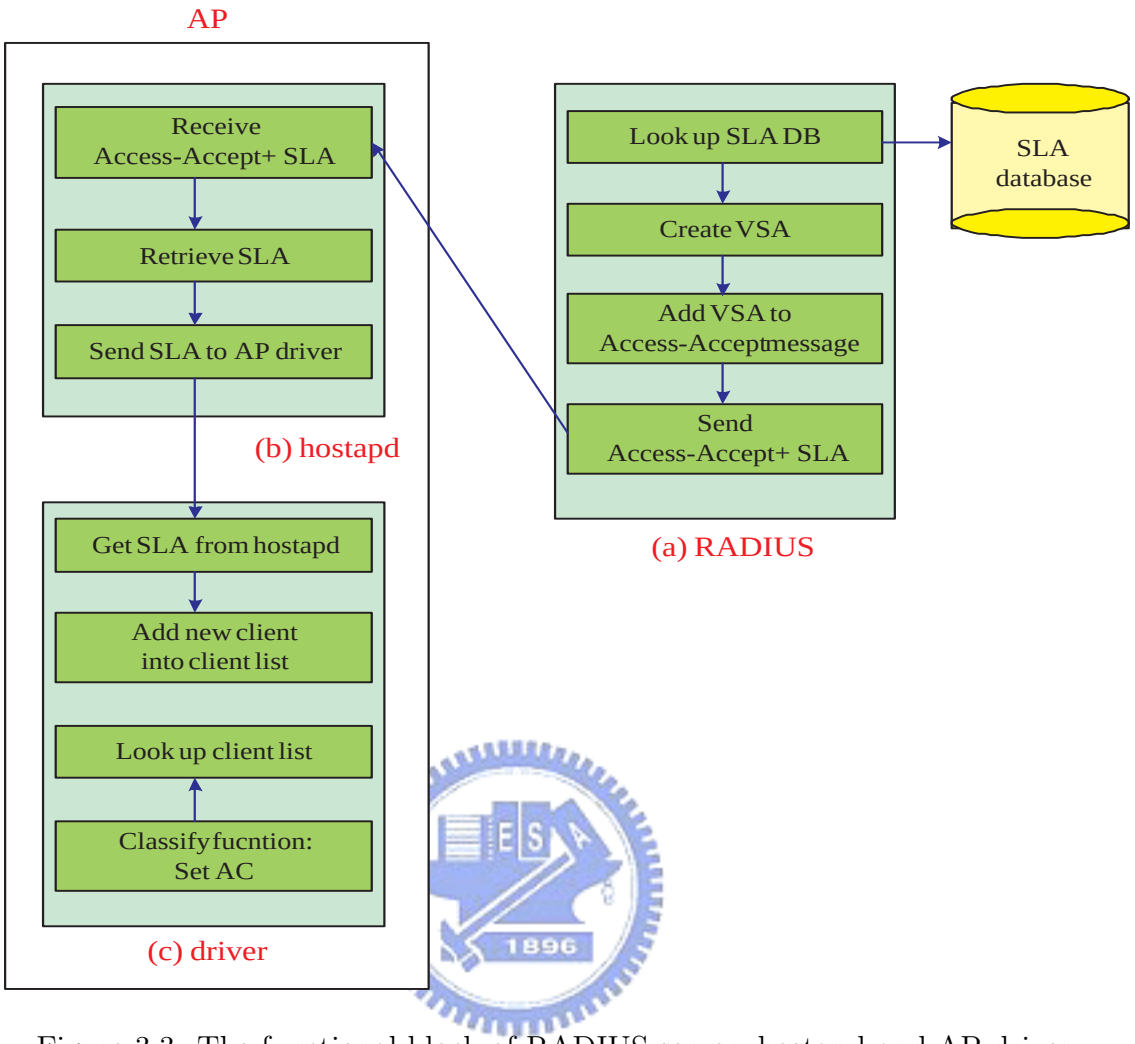

Figure 3.3: The functional block of RADIUS server, hostapd and AP driver.

FreeRADIUS call this file "dictionary". Each vendor has his own dictionary file.

Figure 3.5 displays an example of our VSA. The type value of RADIUS VSA is 26 and 2810 is the number of our self-defined vendor id. The value of sub-type 1 presents a "11E" sub-attribute which is used to carry the SLA information of some user. The value of sub-attribute is a SLA information structure.

Figure 3.6 displays the content of SLA file which acts as a small database which stores the information about someone who has applied what kind of services

```
#
#     Netlab's VSA's dictionary
#
#           $Id: dictionary.netlab,v 1.0 2006/2/22
16:09:20
#

VENDOR      netlab      2810

BEGIN-VENDOR netlab
ATTRIBUTE 11E 1 string


END-VENDOR netlab
```

Figure 3.4: The dictionary file of FreeRADIUS.

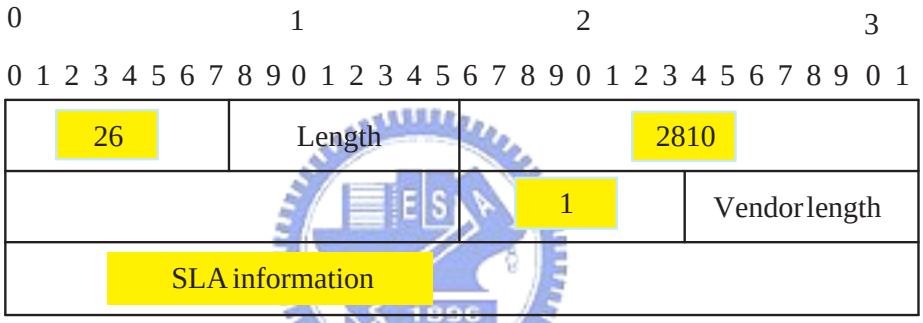

Figure 3.5: The example of Vendor-Specific Attribute.

and the expiration date of these services. In this example, user1 has applied voice and video service and the expiration dates are 2006/7/1 and 2006/8/1. User2 only has applied a video service expired at 2006/9/1 and the traffic from other clients who haven't applied any service would only severed as best effort.

After creating a new vendor and setting down his dictionary and SLA file, we modify the file - radiusd.c to append SLA information to Access-Accept message. The SLA information is formatted as a data structure. Figure 3.7 displays this structure. In this structure, the MAC address of client is stored in "addr". AC_VO

```
client user1
{
      AC_VO       EXPIRED_DATE = 2006/7/1
      AC_VI       EXPIRED_DATE = 2006/8/1
}

client user2
{
      AC_VI       EXPIRED_DATE = 2006/9/1
}
```

Figure 3.6: Service Level Agreement.

and AC_VI are two flags used to describe whether client has applied this service or not.

```
struct client_qos{
        uint8_t addr[6];
        uint8_t AC_VO;
        uint8_t AC_VI;
};
```

Figure 3.7: ClientQoS structure.

Figure 3.3(a) displays the functions we have added into RADIUS server and their flow procedure:

- Look up SLA Database

  Use the the string name of current authenticating client to search SLA database, if the client name is found which means that this client has applied some service, then return a formatted data structure - Client_QoS to main program to present the SLA information.

- Create VSA

  Create a new VSA pair which belongs to vendor netlab. This VSA only contains a "11E" sub-attribute and its value is the SLA information of this client.

- Add VSA to Access-Accept message

  Add the VSA that we made in the last step into the Access-Accept message.

- Send ⟨ Access-Accept + SLA ⟩

  Through the network socket between AP and RADIUS server, RADIUS server sends the ⟨ Access-Accept + SLA ⟩ message to AP.

3.2.2 Hostap Daemon (Hostapd)

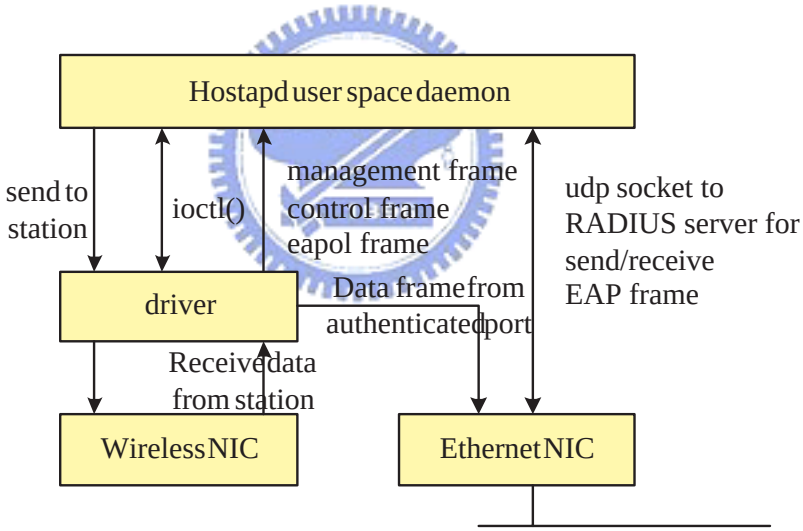

Figure 3.8: The architecture of Hostap daemon.

Just as figure 3.8 shows that hostapd [13] is a user space daemon for AP and authentication servers. It implements IEEE 802.11 AP management, IEEE

802.1X/WPA/WPA2/EAP authenticators, RADIUS client, EAP server, and RADIUS authentication server. The current version supports Linux (Host AP, Madwifi, Prism54 drivers) and FreeBSD (net80211). Hostapd is designed to be a daemon program that runs in the background and acts as the backend component controlling authentication.

Figure 3.3(b) displays the functions we add into hostapd and their flow procedure:

- Receive ⟨ Access-Accept + SLA ⟩

  Hostapd receives the ⟨ Access-Accept + SLA ⟩ from RADIUS server and starts handling it.

- Retrieve SLA

  Hostapd retrieves the SLA information of current authenticating client from ⟨ Access-Accept + SLA ⟩ message.

- Send SLA to AP driver

  Hostapd uses the ioctl function which is provided by C library to pass the SLA information over the AP driver which is at the lower layer.

3.2.3 Access Point Driver

We use a Corega wireless NIC with Atheros Chipset on it and set it at Host AP mode which allows a wireless NIC to perform all the functions of an AP. The driver that AP uses is an open-source Linux driver - Madwifi [14], thus we can modify it to add some functions that we need.

Figure 3.3 (c) shows the functions we add into Madwifi driver:

- Get SLA from hostapd

  The Madwifi driver (the driver of AP) gets the SLA information from hostapd which is at upper layer.

- Add new client into the client list

  Before this station deassociating, we need this SLA information to classify packets from it, therefore we use the memory of AP to implement a client list to store these information. Each time a new station finishing authentication, we will add its SLA information into the client list.

- Look up client list

  Each time while a packet arrives at AP, the Madwifi driver will use MAC address obtaining from MAC header to search the client from the client list.

- Classification function: Set AC

  After getting the SLA information from last step, the classification function of Madwifi driver will use the DSCP field of IP header to classify type of this packet, then according to SLA content to determine whether set this packet to some AC or not. Figure 3.9 is the code of the classification function which describes about Madwifi driver how to classify packet and set packet's AC. The code is a switch-case statement which executes a two-stage examination. First, the classification function use IP→DSCP (representing the DSCP field of IP header) to distinguish this packet as voice, video, or just best effort. After knowing about traffic type, the classification function checks if this client has applied this kind of service. If it is true, the classification function will set this packet to corresponding AC. For example, RADIUS server have

a SLA file like figure 3.6. If there is a packet from client - "user1" arriving AP and the value of IP→DSCP filed is 0x0c, this packet will be classified as a voice packet. Then, checking the SLA information of user1, the value of AC_VO is 1 which means user1 has applied voice service. Thus we set this packet to WME_AC_VO, and then AP will use voice priority to schedule this packet.

```
switch ( IP->DSCP ){
   case   0x0c:          /*Voice*/
     if (client.AC_VO == true)
          d_wme_ac = WME_AC_VO;
          break;
   case   0x0a:          /*Video*/
     if (client.AC_VI == true)
          d_wme_ac = WME_AC_VI;
          break;
   default      :        /*Best Effort*/
          d_wme_ac = WME_AC_BE;
          break;
}
```

Figure 3.9: The clarification function of AP.

3.3 Cross-layer adaptive VoIP application

Since wireless channel is likely influenced by many factors, the channel condition varies with time. However, 802.11e doesn't consider this kind of varying environment, it makes the quality of VoIP application unstable. Although 802.11e is not adaptive, the application can be. Therefore we design a cross-layer adaptive VoIP application adapting to wireless channel. In this following section, we will introduce the codec selection algorithm of our cross-layer adaptive VoIP application and the implementation details.

3.3.1 Codec Selection Algorithm

We use figure 3.10 to explain our codec selection method.

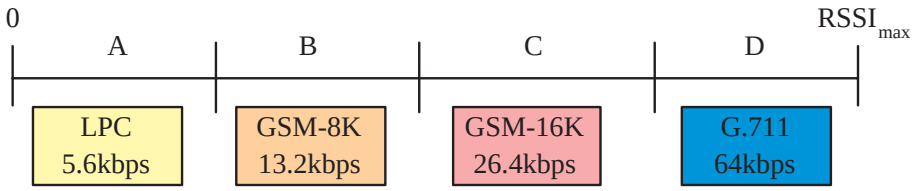

Figure 3.10: Codec selection method.

RSSI is a value representing the received signal strength of both the mobile station and the AP. This value is used to initiate a power change or handoff, thus the current condition of a wireless channel can be presented by the value of RSSI. In our cross-layer VoIP application, we request the value of RSSI from NIC driver every 0.3 second and after we get 10 RSSI values, we count the average of them. The codec selection function refers to the value of average RSSI to determine which codec will be used. We separate the value of 0 to $RSSI_{max}$ into four areas, and each maps to a codec. If the average RSSI locates on area-A, we choose LPC as the codec to be used by the VoIP application. Similarly, if the average RSSI locates on area-B, we choose GSM-8K as the codec to be used by the VoIP application and so on.

In the following two sections, we will explain how to implement this cross-layer adaptive VoIP application.

3.3.2 WRAPI

In last section, we have remarked that our codec selection algorithm must use RSSI as reference. However, a normal NIC driver cannot supply the value of

RSSI to upper layer application. Fortunately, there is a excellent project - WRAPI which provides us a convenient way to get the value of RSSI.

WRAPI [15] is a software library that allows applications running in user space on mobile end stations to query information about the IEEE 802.11 network they are attached to. Before starting to code with WRAPI, we should prepare these things which are listed on the web site of WRAPI:

- Hardware requirements
  - A wireless LAN based on the IEEE 802.11b standard, configured with one or more access points in infrastructure mode
  - A high-performance laptop or workstation with an X86 processor.
  - Wireless Network adapters (NICs) from any hardware vendor for an IEEE 802.11b-based wireless LAN.
- Software requirements
  - Windows XP operating system
  - Windows XP miniport drivers for the NIC
  - Windows XP DDK (driver development kit)

Figure 3.11 explain that by WRAPI, application can get the value of RSSI from NICs.

3.3.3 Implementation of VoIP application

The Robust Audio Tool (RAT) [16] is an open-source audio streaming application which also supports many codecs with different bit rate. Thus we modify

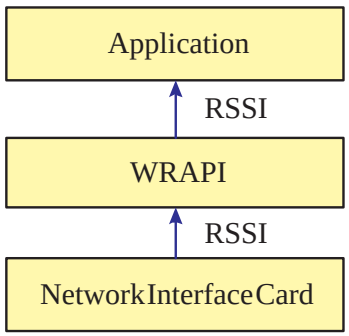

Figure 3.11: WRAPI.

RAT to refer to the value of RSSI and change the codec it used to adapt to the time varying channel condition.

We use Microsoft Visual C++ 6.0 as developer tool. In order to monitor channel condition by WRAPI, we add a program - rssi.cpp which includes several functions listed in figure 3.12. At beginning of the main program of RAT, it will create two threads. One calls the mon_rssi function to get current RSSI every 0.3 second and the other calls the count_avg_rssi function to count the average value of last ten RSSI. After getting the average RSSI, the count_avg_rssi function will determine which codec the VoIP application should use according to the value of average RSSI and then call the Set_Codec function to change codec.

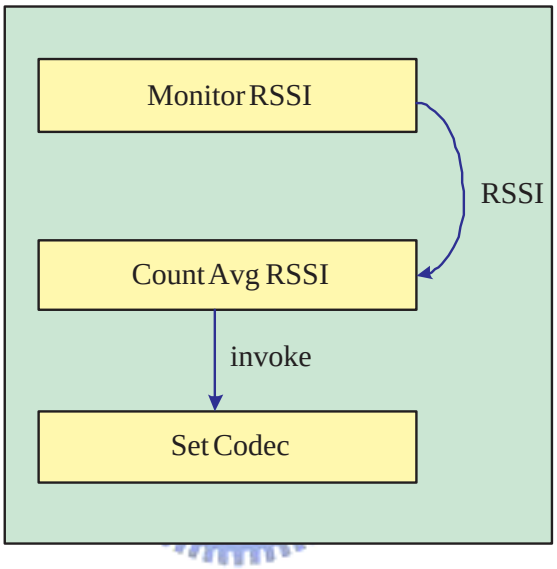

Figure 3.12: The RSSI-relatived functions.

Chapter 4

Performance Evaluation

In this chapter, we describe our experiment environment and analysis.

4.1 Experiment Environment

In our experiment environment, in order to be close to the real Internet, we use NIST Net [17] to emulate the Internet latency and packet loss probability between endpoints. Figure 4.1 shows the experiment framework which we use NIST Net.

Between subnet 140.113.167 and subnet 192.168.1 we use NIST Net to emulate a Wide Area Network (WAN). We set the latency of this emulated WAN in all of our experiment. The traffic must flow from subnet 140.113.167 to subnet 192.168.1 or from subnet 192.168.1 to subnet 140.113.167, thus it will go through the emulated WAN. Besides, we use network experimental software - NetIQ Chariot which can order endpoints to generate the original, non-adaptive VoIP traffic which we use to increase high priority traffic. With the traffic which is generated by NetIQ Chariot, our experiment will be more close to the scenario of real-world. After that, based on these traffics we use cross-layer adaptive VoIP and the original, non-adaptive VoIP flow to observe the difference between these two methods.

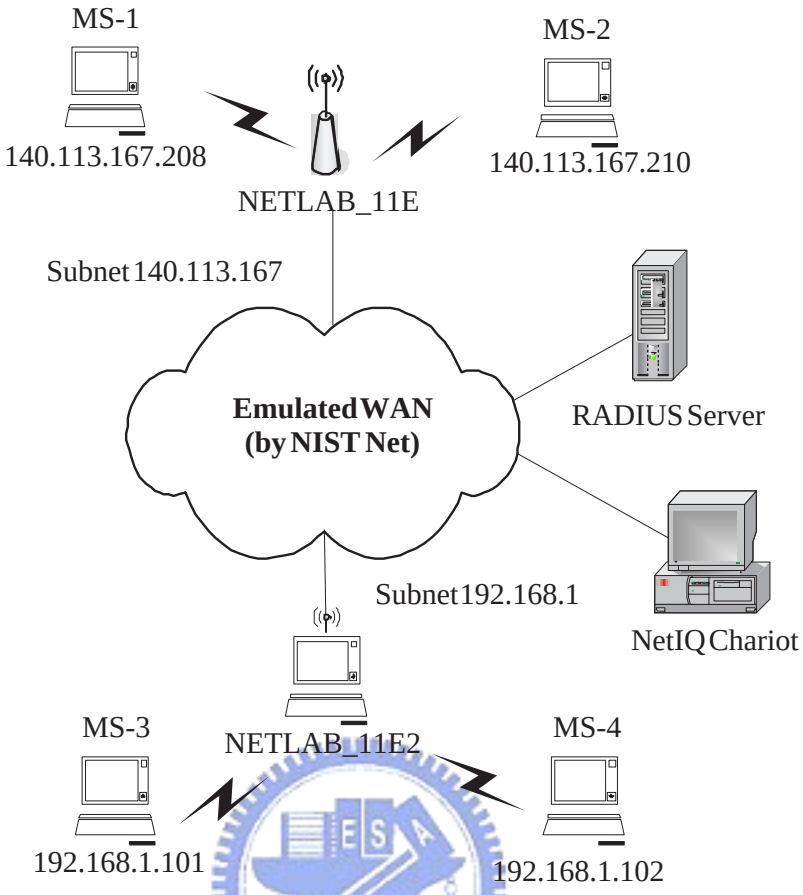

Figure 4.1: The framework of emulated WAN.

Later, we will explain the experiments that we design to observe the performance of our cross-layer method.

4.2 Experiment: adaptive v.s. non-adaptive VoIP

Figure 4.2 displays the traffic flow that we build in this experiment. Flow (1) is a g.711 (64kbps) VoIP flow which is generated by NetIQ Chariot endpoints from MS-1 to MS-4 to increase high priority traffic. Flow (2) is created by our modified-RAT VoIP application. We change RAT from non-adaptive mode to adaptive mode to observe their difference.

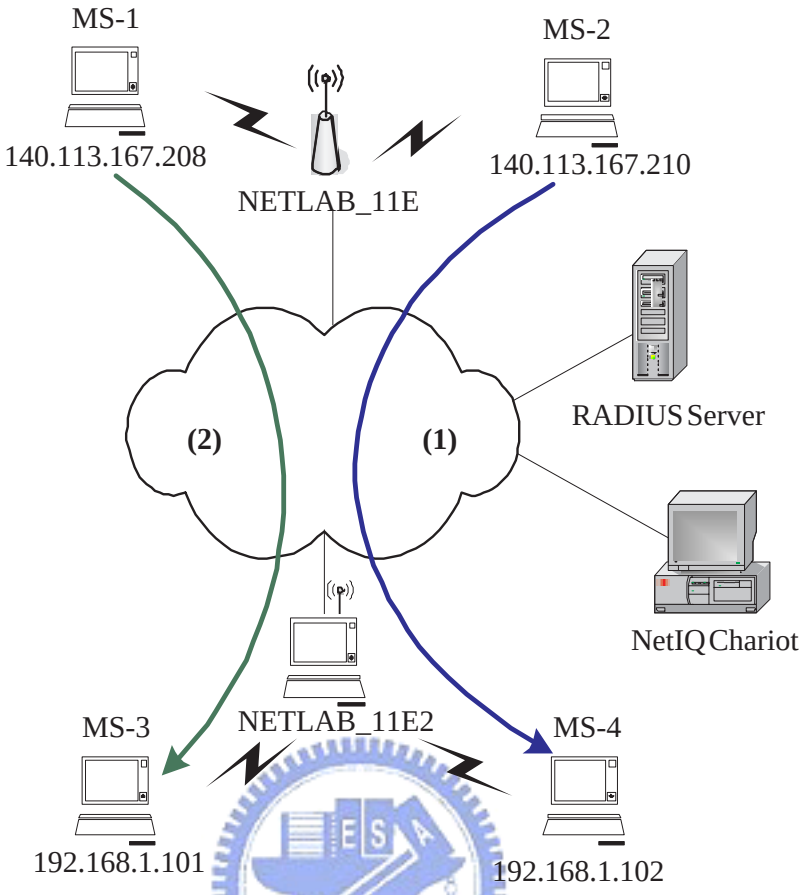

Figure 4.2: The VoIP flows of experiment.

We made three tests in different scenarios. In scenario one, RAT used fixed 64kbps data rate and let MS-1 walk back and forth to change the value of RSSI. The same as scenario one, in scenario two, RAT used fixed 5.6kbps and scenario three changes is adaptive RAT that changed data rate according to current RSSI value.

Table 4.1 shows the one-way delay of VoIP flow (2) of these three scenarios. We did five tests in each scenario. The one-way delay of fixed 64kbps VoIP flow is approximately 88 $\sim$ 134 ms, fixed 5.6kbps VoIP flow is 27 $\sim$ 39 ms and adaptive VoIP is 33 $\sim$ 46 ms.

Table 4.1: The one-way delay of three scenarios.

TestNo	Fixed64kbps (ms)	Adaptive (ms)	Fixed5.6kbps (ms)
1	104	33	39
2	134	37	27
3	111	32	33
4	109	46	31
5	88	35	30

Figure 4.3 shows the average delay of these three scenarios. Fixed 64kbps has the highest delay, and fixed 5.6kbps has the lowest delay. However, the delay of the adaptive codec lies between them.

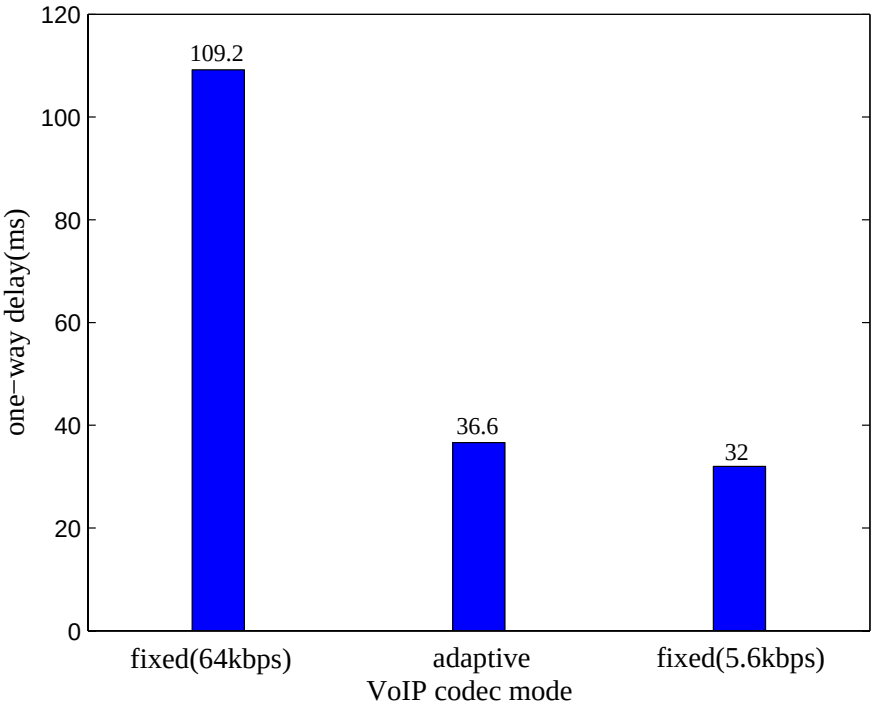

Figure 4.3: The average delay of three scenarios.

Figure 4.4 displays the average packet loss rate (PLR) of these three scenarios. The PLR of fixed 64kbps is highest, and that of fixed 5.6kbps is lowest. However, the PLR of the adaptive codec lies between them.

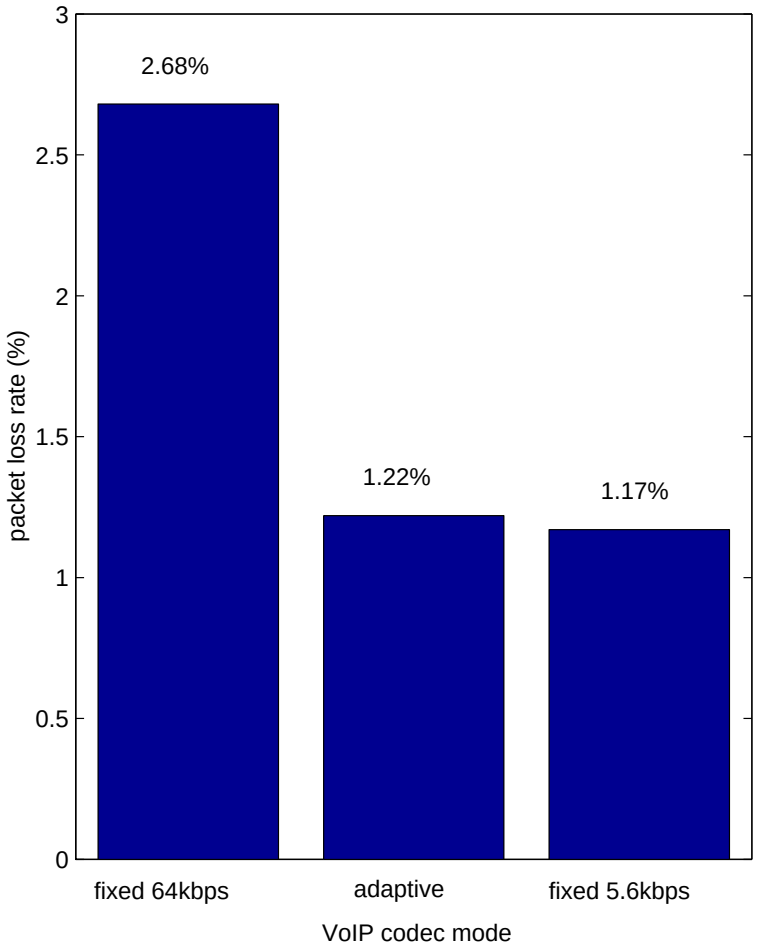

Figure 4.4: The average packet loss rate of three scenarios.

Although fixed 5.6kbps has lowest delay, its voice quality is the lowest one. Voice quality was traditionally reported as a Mean Opinion Score (MOS) on a scale from 1 to5 where 1 is the lowest and 5 the highest. Table 4.2 displays the MOS of these four codecs we used.

Table 4.2: The MOS of codecs that we use.

Codec	G.711(64kbps)	GSM-16K(26.4kbps)	GSM-8K(13.2kbps)	LPC(5.6kbps)
max MOS	4.1	3.92	3.6	2.4 $\sim$ 3

During the test of adaptive VoIP application, we also keep track of the number of packets transmitted with these four codecs. Figure 4.5 shows the percentage of packets which is transmitted with each codec.

We count the average MOS and bit rate of this adaptive codec according to the percentage values:

- average MOS=4.1×17.39%+3.92×34.38%+3.6×24.03%+3×24.2%=3.65
- average bit rate= 64×17.39%+26.4×34.38%+13.2×24.03%+5.6×24.2%=24.73kbps

In table 4.3, we summarize the experiment results. From these results, we can realize that adapting codec with time varying condition can has lower delay than a fixed codec with large data rate and better sound quality than a fixed codec with small data rate. Thus cross-layer adaptive VoIP application performs well than a VoIP application with fixed bit rate.

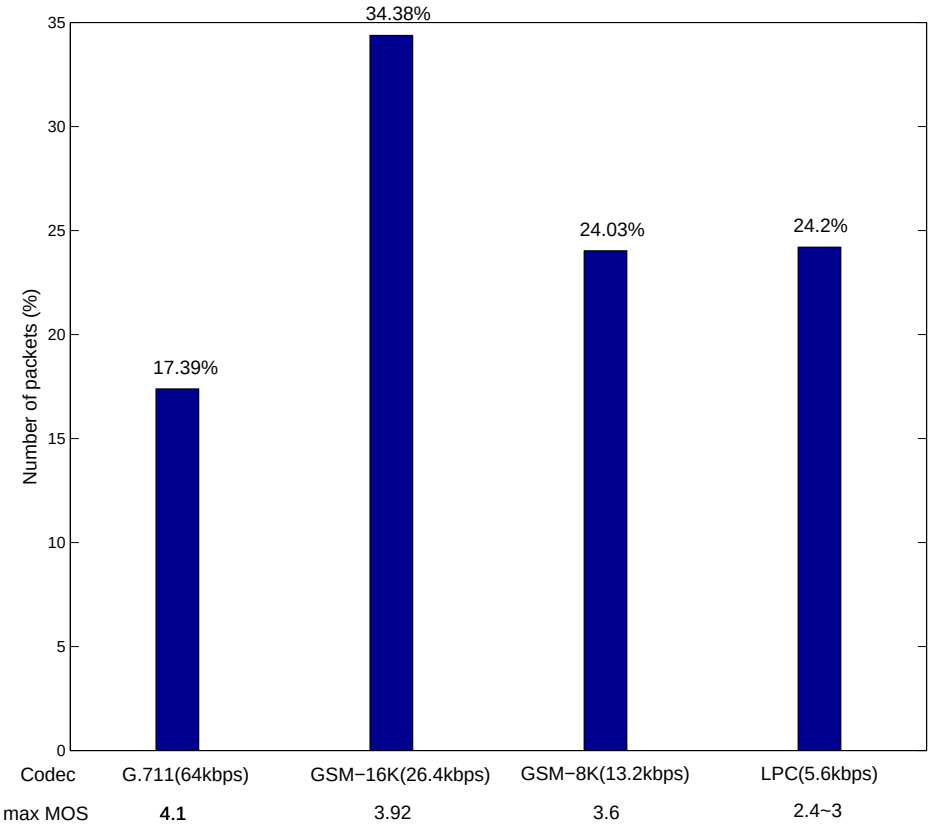

Figure 4.5: Number of packets transmitted with each codec.

Table 4.3: The summary of experiment results.

	G.711(64kbps)	Adaptive Codec	LPC(5.6kbps)
Avg. delay	109.2	36.6	32
Avg. PLR	2.68%	1.22%	1.17%
MOS	4.1	3.65	$2.4 \sim 3$

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In the past few years, VoIP becomes the most popular technology and will be a kind of "Killer Application". Wireless Local Area Network (WLAN) let people can use VoIP everywhere as long as that area is covered by WLAN radio. In order to support QoS on WLANs, 802.11e enhances the MAC procedures of 802.11 to give VoIP traffic higher access priority than data traffic.

In this thesis we implement a policy-based 802.11e WLAN which is an admission control mechanisms to let AP know which stations are allowed and what kind of actions they should take on these stations.

Although 802.11e can support QoS, it doesn't consider the varying channel condition. Therefore we use a cross-layer adaptive VoIP application which can change its codec according to current RSSI value. With this cross-layer adaptive VoIP application, one-way delay can be lower than traditional codec with fixed large data rate and gets better voice quality than a codec with fixed small data rate.

5.2 Future Work

Our policy-based 802.11e WLAN does the admission control job on traffics under this WLAN, however we can not control the behavior of user's own machine. It means we can not control the contention procedure of up-link traffic, but only down-link traffic. The thing how to schedule up-link traffic depends on the Wireless NICs on user's machine, thus AP can only schedule down-link traffic. This situation may violate the benefit of the payer. Thus in the future we will add a up-link traffic scheduling method into the transmission queue of AP to schedule up-link traffic according to policy rules as we describing in chapter 3.

Bibliography

[1] IEEE Work Group: "IEEE Standard for Information technology- Telecommunications and information exchange between systems- Local and metropolitan area networks- Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements", IEEE, Nov 2005

[2] L.Romdhani, Q.Ni, and T.Turletti: "Adaptive EDCF: Enhanced Service Differentiation for IEEE 802.11 Wireless Ad-Hoc Networks", *IEEE Wireless Communications and Networking Conference*, Volume 2, Page(s):1373 - 1378 vol.2, Mar. 2003.

[3] I.Haratcherev, J.Taal, K.Langendoen, R.Lagendijk, and H.Sips: "Optimized Video Streaming over 802.11 by Cross-Layer Signaling",*IEEE Communications Magazine*, Volume 44, Issue 1, Page(s):115 - 121, Jan. 2006.

[4] Q.Ni, L.Romdhani, T.Turletti, and I.Aad: " QoS Issues and Enhancements for IEEE 802.11 Wireless LAN ",*INRIA*, Nov. 2002.

[5] M.Matsumoto and T.Itoh:" QoS-guarantee Method for Public Wireless LAN Access Environments",*IEEE International Conference on Wireless Networks,*

Communications and Mobile Computing, Volume 1, Page(s):101 - 106, Jun.2005.

[6] J.d.P.Pavon and S.S.N: " Impact of frame size, number of stations and mobility on the throughput performance of IEEE 802.11e",*IEEE Wireless Communications and Networking Conference*, Volume 2, Page(s):789 - 795, Mar.2004.

[7] Seo J W, Woo S J and Bae K S: "A Study of the Application of an AMR Speech Codec to VoIP",*In Proceedings of IEEE Acoustics, Speech, and Signal Processing International Conference*, Volume: 3, Page(s): 1373-1376, May 2001.

[8] J. Strassner, "Policy-Based Network Management : Solution for the Next Generation", *O'REILLY*, Apr. 2002

[9] A. Westerinen, J. Schnizlein, Cisco Systems, J. Strassner , Intelliden Corporation, M. Scherling, xCert, B. Quinn, Celox Networks, S. Herzog, Policy-Consulting, A. Huynh, Lucent Technologies, M. Carlson, Sun Microsystems, J. Perry, Network Appliance, S. Waldbusser, "Terminology for Policy-Based Management", *RFC-3198*, Nov. 2001

[10] G.Pauy, D.Maniezzo, S.Das, Y.Lim, J.Pyon, H.Yu, M. Gerla:" A Cross-Layer Framework for Wireless LAN QoS Support", In Proceedings of IEEE Information Technology: Research and Education International Conference, Page(s):331 - 334, Aug. 2003.

[11] C. Rigney, S. Willens, A. Rubens, and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", *RFC-2865*, Jun. 2000

[12] FreeRADIUS. [Online]. Available: http://www.freeradius.org/

[13] hostapd. [Online]. Avaliable: http://hostap.epitest.fi/hostapd/

[14] MadWifi. [Online]. Avaliable: http://madwifi.org/

[15] WRAPI. [Online]. Avaliable: http://sysnet.ucsd.edu/pawn/wrapi/

[16] RAT, Robust Audio Tool.*University College London* [Online]. Available: http://www-mice.cs.ucl.ac.uk/multimedia/software/rat/

[17] NIST Net. [Online]. Available: http://snad.ncsl.nist.gov/nistnet/