

行政院國家科學委員會專題研究計畫 成果報告

多媒體共享之網路服務平台--子計畫一：ADSL-WLAN 雙模接 取網路之頻寬提升技術(3/3) 研究成果報告(完整版)

計畫類別：整合型
計畫編號：NSC 95-2221-E-002-065-
執行期間：95年08月01日至96年10月31日
執行單位：國立臺灣大學電信工程學研究所

計畫主持人：蔡志宏

計畫參與人員：碩士班研究生-兼任助理：吳敏哲、徐章恆、楊志謙

處理方式：本計畫可公開查詢

中華民國 96 年 11 月 09 日

行政院國家科學委員會專題研究計畫成果報告

多媒體共享之網路服務平台-子計畫一：

ADSL-WLAN 雙模接取網路之頻寬提昇技術(3/3)

Hybrid Broadband Access Networks with ADSL and WLAN (3/3)

計畫編號：NSC 95-2221-E-002-065

執行期限：95年8月1日至96年7月31日

主持人：蔡志宏 國立臺灣大學電信工程研究所教授

一、中英文摘要

本計畫針對平行使用 ADSL-WLAN 雙模接取網路之關鍵頻寬提升技術與網路架構規劃課題進行研究。在此雙模接取網路中，個別終端使用者之路由器或電腦將可同時使用 ADSL 或是室內／公共 WLAN，以使其頻寬大幅提升。而其服務及應用也可針對不同網路介面而機動調整其路由設定。各應用享有之服務品質將可根據其交通特性調整而加以維持，而 WLAN 及 ADSL 兩種鏈路間之負載也將以智慧型式加以平衡，為提升即時之接取頻寬。前兩年我們已經完成以下技術議題之初步研究：

- a. 雙模接取網路之智慧型應用層路由器負載平衡機制及個人電腦中介軟體
- b. 高效率網路狀態監測機制及聰明交通分類機制
- c. 雙模接取網路架構之網路規劃模型及其最佳化方法
- d. 考慮不同交通混合之負載平衡演算法

- e. 在 NAT 的限制之下無縫切換同一個連線的傳輸介面

第三年我們研究並實作了封包串接和標頭壓縮這兩種機制，用來提升小封包交通流在此環境中之頻寬使用效率，以期達到倍增網路電話容量之目的。同時由於多網系統之建置成本日漸降低，我們亦設計了一套能夠提升網路電話語音品質之多網平行傳輸演算法。

關鍵詞：服務品質，網路服務排程，網路頻寬分配，負載平衡，交通混合，無縫交接，封包串接，多網接取。

Abstract

This project aims to study the key enabling technologies and the architecture planning issues for the *Dual-Mode Hybrid Broadband Access*, using ADSL and WLAN in parallel as two complementary access technologies. Under this Dual-Mode Hybrid access scenario, each end user's router or host computer will be provided with both ADSL access as well as Outdoor/Public

WLAN access so that the total access bandwidth can be boosted, while the service and applications over this network can be provisioned dynamically over different network interfaces. The quality of service for each specific application stream shall be maintained to meet its own traffic characteristics and the traffic load of between the WLAN link and ADSL link can be balanced intelligently. In order to boost the instantaneous access bandwidth, in the first year, we have completed the preliminary studies and proposed designs for the following key technologies and issues:

- a. A smart application level load balancing routing mechanism for the router and the middleware for the host computer directly on the Dual-Mode Hybrid Broadband Access
- b. An efficient network status monitoring mechanism and a smart classification mechanism
- c. The Network Planning model and optimization issues for the Dual-Mode Hybrid Broadband Access architecture
- d. The load balancing routing algorithm that takes the effects of traffic mixing into consideration
- e. Seamless handoff among different network interfaces with the constraint of NAT

In the third year, we have investigated the packet concatenation and header compression mechanisms,

which can lift the bandwidth utilization of the traffic flows with large amount of small packets in such environments. By deliberately combining these two mechanisms, we have designed and implemented a Linux module that can dramatically increase the capacity of VoIP calls of the system. In addition, we have also completed a design of parallel transmission algorithm between two multi-radio gateways that can improve the voice quality of the VoIP calls.

Keywords: Quality of Service, Call Admission, Scheduling, Load Balancing, Traffic Mixing, Seamless Handoff, Packet Concatenation, Multi-homing Access.

二、研究背景及目的

隨著Internet的快速普及和發展以及網路應用的多樣化，使用者對於頻寬的需求也愈來愈高。例如FTP及P2P (Peer-to-Peer) 等需要大頻寬的網路應用，以及VoIP、Netmeeting、VoD等語音、多媒體即時影音服務交通量愈來愈大，使得接取頻寬的需求日漸增加，以致於目前的ADSL [1]及Cable modem 等網路接取服務所提供的頻寬已漸感不足。對一個已經大量鋪設ADSL 線路及設備的固網業者而言，解決上述問題的方法是重新開始一個網路規劃及部署之程序，其中包含了部署新的光纖及新的遠端DSLAM 設備。或者是將舊的DSL 網路轉換到其它較高階之DSL 技術如ADSL2 [2]、VDSL [3]等。但此類DSL技術之高頻寬會因使用者與機房之距離增加而快速

下降，而且部署時間可能會因新產品的成熟程度不足或者新的網路設備的部署困難度而有會所延遲。

因此我們提出了一個雙網接取系統架構，利用現有之ADSL鏈路與802.11g WLAN的連線來做雙網寬頻網路的接取，以達到快速增加接取頻寬之目的。我們的目標是在使用者端之多網(multi-homing)系統內完成一個智慧型的負載平衡機制，依據目前網路的狀況以及網路應用的種類來把交通流分配到不同的網路介面上，以期能夠有效的運用所有的頻寬。

在第三年的研究中，我們致力於提高網路的傳輸效率和網路電話之語音品質，藉由標頭壓縮(header compression)、封包串接(packet concatenation)等方法減低在WLAN或使用VPN之ADSL等有線網路傳輸中overhead所佔之比例，以改善系統之效能並維持及時語音傳輸之品質。同時，由於無線網路模組的價格日趨低廉，所以各種配有多重無線模組的設備也越來越多，我們也提出利用多重無線閘道器平行傳輸的方法來改善網路電話語音品質。

三、去年進度回顧

我們初步探討了在不同網路上交通混流的現象，並且重新改良了我們第一年研究中提出的負載平衡演算法，將該現象列入路由決定時的考量，並依此預測接受某一個應用對網路資源及其他的應用所造成的影響，做為選擇傳輸用網路的基準。我們已經在Linux平台上完成了系統雛型，同時使用ADSL及IEEE 802.11g WLAN

這兩種傳輸網路，並且支援小封包之多媒體影音串流(如VoIP)及長時間之Best Effort資料傳輸(如FTP)這兩類應用。

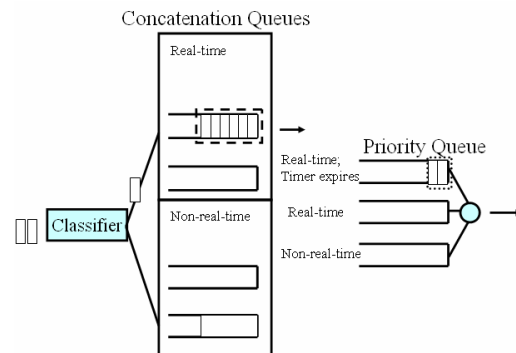
初步的實驗結果證明我們所提出的負載平衡演算法相較於傳統的trunking系統更能有效率的使用頻寬。我們也提出了IP虛擬通道的方法，配合一台獨立運作的NAT server，便可以解決前一年所提出的「如何在NAT的限制之下，讓同一連線可以動態切換傳輸介面而不使其斷線」的問題。實驗結果也證明了我們提出的方法的確可以達成此目的。

另一方面，我們也初步探討了如何以封包串接技術來提升WLAN傳輸能力，進而改善整體系統效能的可能性。

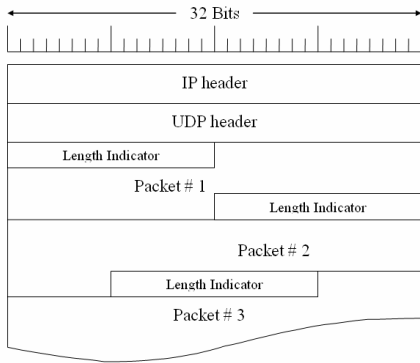
四、研究方法與結果

4.1 無線網路語音封包串接機制之設計分析與實作

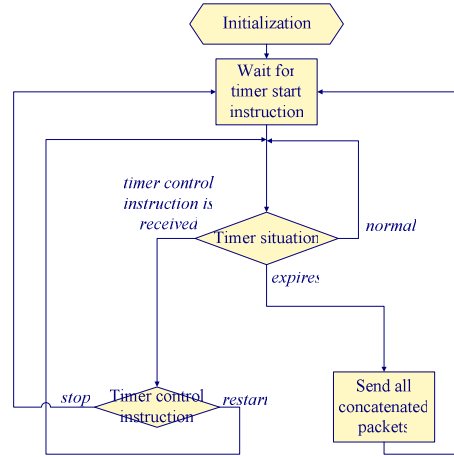
因為傳送小封包所帶來的大量虛耗(Overhead)，直接傳送大量小封包嚴重地降低了網路效率。為了改善此類型網路的通道效率與有效容量，串接封包的方法在產學界皆吸引了高度關注。然而，若要將數個封包串接為一



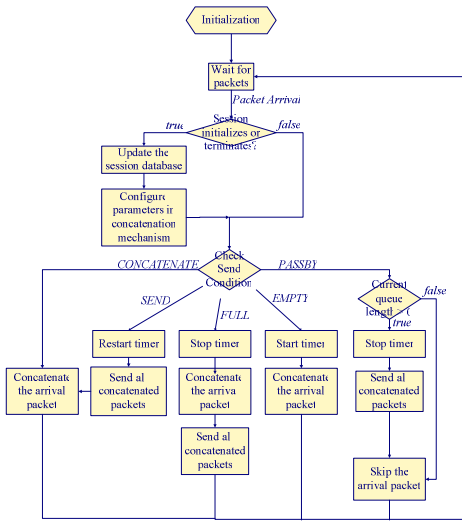
圖一 封包串接系統架構



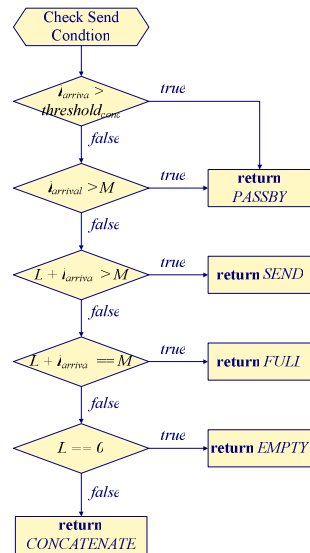
圖二 串接與還原之間所用的封包格式



圖四 Timer 運作流程圖



圖三 封包串接運作流程圖



圖五 串接決策流程圖

個訊框，封包會滯留於緩衝區中因此其延遲會增加。不幸的是，VoIP 的語音品質對於通話雙方間的延遲非常敏感，因此封包串接可能與服務品質相互衝突。

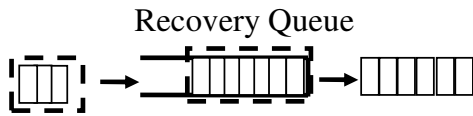
雖然已有許多文獻[4-12]探討封包串接的議題，但其往往以網路 throughput 為最佳化的目標而忽略了 VoIP 的服務品質。因此我們設計一語音封包串接機制並控制串接所造成的延遲。此外，我們實作此封包串接機制於 Linux 平台，並且實際評估其表現。

4.1.1 封包串接機制系統架構與演算法介紹

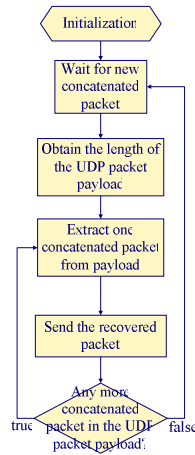
封包串接系統架構如圖一圖。左方 classifier 原則上以封包的長短與即時性分為四類。而串接佇列(concatenation queues)負責串接封包。我們將串接後的封包封裝於 UDP 封包內，在將其送至 priority queue 送出。封包格式如圖二。

封包串接機制中有兩項嚴重影響延遲的參數：

- M : maximum concatenation length, 最大串接封包數量。
- T : time constraint, 當封包抵達串接佇列時，若串接佇列沒有任何封



圖六 封包還原機制架構圖



圖七 封包還原流程圖

包，則啟動 timer 並設成 T 。

詳細的封包串接機制與 timer 運作於圖三、四、五中。

封包還原機制的架構與流程圖分別為圖六與圖七。藉由 UDP 封包中的 length indicator，封包將可一一還原。

我們以[13]的分析方法為基礎，分析與模擬以獲得串接所造成的 VoIP 封包平均串接延遲，並利用其結果以設定封包串接機制中的參數 T 和 M ，進而達成要求的平均串接延遲。

假設 VoIP 電話通數為 N_{VoIP} 且平均延遲限制為 \bar{d}_{max} ，我們定義一數值 $T_{max}(N_{VoIP}, M)$ ，此數值為參數 T 的上限。

$$T_{max}(N_{VoIP}, M) = \left\lceil \frac{M-1}{N_{VoIP}} \right\rceil T_p$$

我們提出的演算法針對每一種可能的 M 數值，搜尋可符合平均延遲限制且同時最大化通道效率的參數 T 。因為當 time constraint 增加時，通道效率為非

遞減，所以演算法從 $T = T_{max}(N_{VoIP}, M)$ 開始搜尋並檢查其結果是否符合所要求的平均延遲限制。如果相對應的平均延遲超過 \bar{d}_{max} ，演算法將嘗試 $T = T_{max}(N_{VoIP}, M) - timer_res$ 直到 $\left\lceil \frac{T}{T_p} \right\rceil \cdot N_{VoIP} < M$ 。 $timer_res$ 表示計時器的解析度。

我們使用一變數 $max_efficiency$ 紀錄目前取得的最大通道效率。在檢查 time constraint 是否符合平均延遲限制之前，會先檢查相對應的通道效率是否大於 $max_efficiency$ 。如果並非大於 $max_efficiency$ ，演算法將嘗試下一個 M 數值，否則演算法將繼續搜尋相同的 M 數值。

在所有可能的 M 都已被搜尋過之後，演算法將回覆所找出的 M 與 T 。如果沒有找到答案，則回覆 NOT_FOUND 。演算法之虛擬碼描述於圖八中。

4.1.2 封包串接系統實作與效能驗證

我們將此串接機制實作於 Linux 平台上，其實驗環境如圖九。我們使用 switch 的 port mirror 的功能，讓每個封包在串接前與還原後各重製一份送往協定分析儀，如此可藉由比對封包 ID 與 timestamp 而計算出每個封包的 one-way delay。

圖十為平均封包延遲之結果。在沒有封包串接的情況下，802.11b[14]只能容納最多 9 通 VoIP 電話，但是如果加入封包串接機制，即使 25 通 VoIP 電話依然有著極低的平均封包延遲，網路的容量增加了 2.7 倍。

我們接著使用 SmartBits 產生大量

```

PARAMETER-CONFIGURATION ( $N_{VoIP}$ ,  $\bar{d}_{max}$ ,
channel_efficiency, average_delay)
max_efficiency = 0;
T_chosen = NULL;
M_chosen = NULL;
for(m = 2; m <= maximum protocol data unit; m++)
{
    T = T_max(N_VoIP, m);
    while(  $\left\lceil \frac{T}{T_P} \right\rceil \cdot N_{VoIP} \geq m$  &&
channel_efficiency[N_VoIP][T][m] > max_efficiency)
    {
        if(average_delay[N_VoIP][T][m] <  $\bar{d}_{max}$ )
        {
            T_chosen = T;
            M_chosen = m;
            max_efficiency =
channel_efficiency[N_VoIP][T][m];
        }
        T = T - timer_res;
    }
}
if(T_chosen == NULL)
{
    return NOT_FOUND;
}
else
{
    return (T_chosen, M_chosen);
}

```

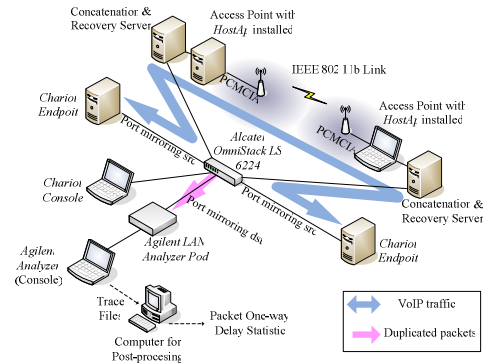
圖八 參數設定演算法之虛擬碼

的 VoIP 封包灌入啟動封包串接機制之伺服器中，以觀察此機制對 CPU 負載之影響，其結果如圖十一。在傳送多達 40,416 fps(80%)的流量時，CPU 0 的負載才接近滿載。利用以下的算式，計算出每個封包的平均 CPU 處理時間約為 31 μ s。

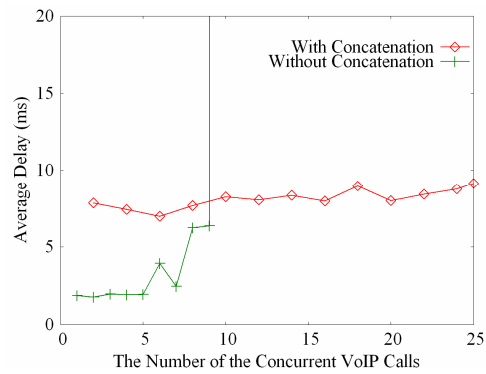
CPU time per packet

$$= \frac{\text{CPU 0 load} + \text{CPU 2 load}}{\text{Arrival packet per second}}$$

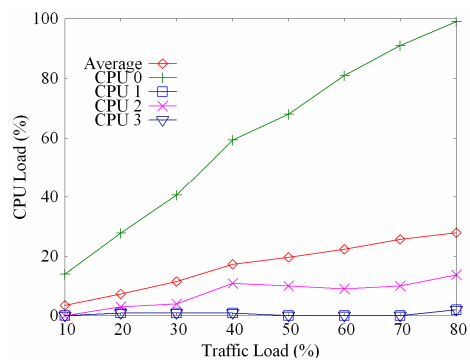
為了驗證先前推導與模擬所得的平均延遲時間是否正確，我們使用 Ethernet 代替圖九中之 WLAN，以單純



圖九 實驗環境



圖十 不同 VoIP 電話數量下的平均延遲

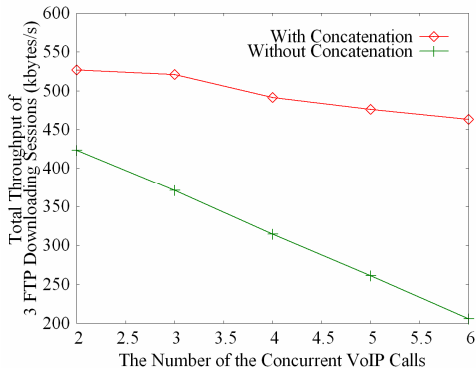


圖十一 封包串接機制的 CPU 負載

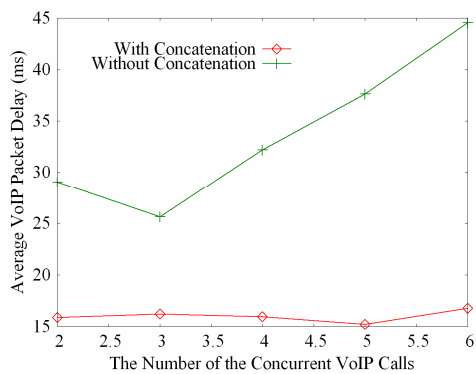
N_{VoIP}	$\bar{d}_{max} = 5$ ms		$\bar{d}_{max} = 10$ ms		$\bar{d}_{max} = 15$ ms	
	simulation	experiment	simulation	experiment	simulation	experiment
2	4.955	5.576	4.955	5.576	14.996	14.286
3	4.720	4.567	6.702	7.415	14.856	14.678
4	4.970	5.269	10.000	10.449	14.997	15.528
5	4.931	5.709	8.013	8.330	12.080	12.616
6	4.989	5.334	8.332	8.449	10.033	10.575

表一 模擬與實驗之封包串接延遲比較圖

測量串接延遲。從表一可知，實驗所量得的 VoIP 串接延遲與模擬結果非常接近。



圖十二 3個FTP sessions的throughput總和



圖十三 VoIP 封包平均延遲

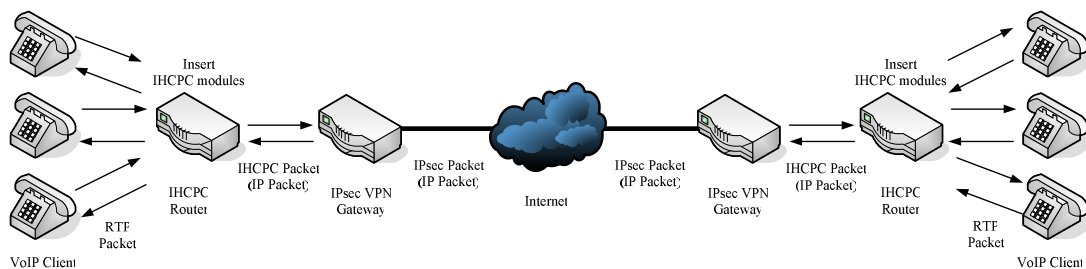
我們同樣使用圖九的實驗環境，並加入 3 個 FTP sessions，觀察 VoIP 封包串接機制在混合 FTP 交通流情形下的效能表現。圖十二與圖十三分別為 FTP 交通的 throughput 與 VoIP 封包的平均延遲結果。沒有封包串接機制時，由於小封包的頻寬使用效率不佳，每加入一通 VoIP call 均會佔掉相當大的頻寬，因而造成 FTP 交通 throughput 明顯地下降；而在封包串接機制下，即使加入數通 VoIP 電話，FTP

依然維持很高的 throughput。另一方面，VoIP 的平均封包延遲在有封包串接機制時，也不會因為電話通數的增加而升高，都維持差不多的水準。

4.2 設計與實作即時通訊服務之標頭壓縮與封包串接機制

近年來隨著網路價格下降及頻寬提昇，越來越多企業選擇改用網路電話取代傳統電話服務。在此同時，為了確保通話及企業資料的機密性，分公司間之網路則必須選擇建置安全的虛擬私有網路(VPN: Virtual Private Network)[15]。因此 VoIP 及 VPN 兩種技術在企業網路上，經常必須同時運作。然而 VoIP 過小的封包除了導致頻寬使用效率不佳的問題之外，也會對 VPN 匝道器造成極大的負擔，因此我們提出同時使用標頭壓縮與封包串接兩種機制來解決此一問題。

我們所提出的標頭壓縮與封包串接機制 IHPC (IP-Based Header Compression with Packet Concatenation) 是以 IP 協定為基礎，其架構圖十四。我們已將此機制實作成 Linux 系統之模組，可以被安裝在 IPsec VPN 路由器上，用來提升即時通訊服務在 IPsec VPN 的頻寬使用效率，並且可以有效降低 IPsec VPN 路由器的負擔。同樣的模組也可用於提升 VoIP 交通流在 WLAN 上的頻寬使用效率。



圖十四 IP-Based 標頭壓縮與封包串接系統架構示意圖

4.2.1 標頭壓縮標準簡介

目前已知的標頭壓縮標準有 CRTP[16]、ECRTP[17]、ROHC[18]與 IHC[19]四種。ROHC 支援多種標頭格式壓縮，缺點是壓縮機制過於複雜。IHC 只能用來壓縮 IP 標頭，UDP/RTP 標頭將被視為 IP 承載無法進行壓縮。CRTP 與 ECRTP 是特別針對 RTP 所設計，最適合用來壓縮 VoIP 標頭。

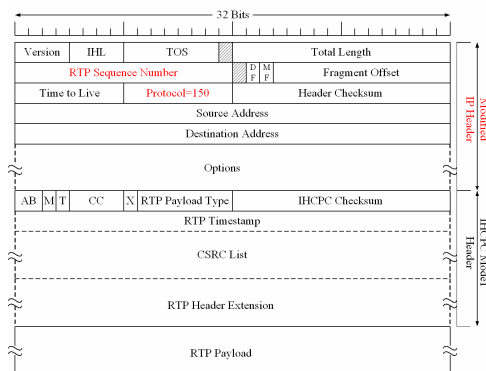
然而 CRTP 與 ECRTP 均使用差值編碼(delta coding)，在更新欄位差值時，通訊雙方所在的標頭壓縮路由器必須要同步，才能正確建立標頭還原資料庫，不然 RTP 封包將無法還原。過去因為數據機撥接等低速網路的封包遺失率很低，通常不會發生同步問題，標頭可以正確被還原。

此外，傳統的標頭壓縮技巧因為沒有保留 IP 標頭，必須藉助資料連結層，將壓縮過的封包以 MAC frame 的形式在網路裝置之間傳遞。因為每個網路裝置間的標頭壓縮動作是獨立進行，封包流經的所有網路裝置都要有進行標頭壓縮的能力。

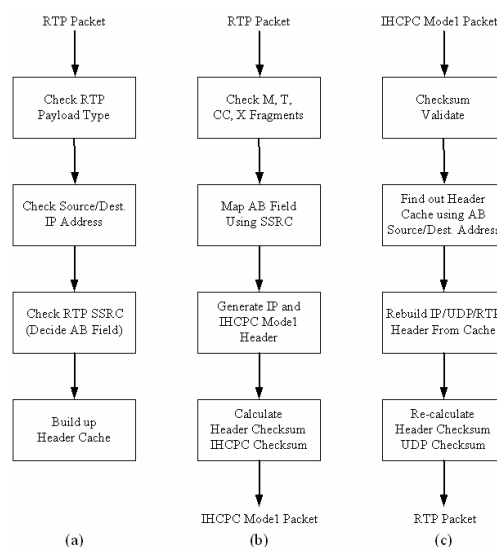
4.2.2 IHCP 機制介紹與系統架構

為了改進傳統標頭壓縮技巧的缺點，我們以 IP 協定為基礎來實現標頭壓縮。保留 IP 標頭好處是可以直接用在任何以 IP 協定為基礎的網路上，自然也包含 IPsec VPN。同時，為了改進 CRTP 與 ECRTP 用差值編碼所可能產生的問題，IHCP 不使用差值編碼，而是選擇保留完整的 Sequence Number 欄位。

IHCP 機制包含兩種模式，IHCP Mode1 及 IHCP Mode2，前者針對 RTP 封包單獨進行標頭壓縮，後者特



圖十五 IHCP Mode1 封包示意圖



圖十六 IHCP Mode1 協定運作示意圖

(a)建立標頭資料庫流程，(b)標頭壓縮流程，(c)封包還原流程。

別針對小 RTP 封包(VoIP)同時進行標頭壓縮與封包串接動作。

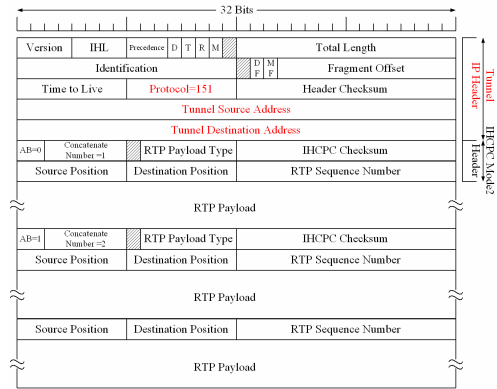
A. IHCP Mode1

IHCP Mode1 是一種新的 IP 協定，針對一般 RTP 封包，將 UDP/RTP 標頭簡化成 IHCP Mode1 標頭，並對原始 IP 標頭做小幅修改(見圖十五)，使用 IHCP Mode1 可以將 40bytes 的 IP/UDP/RTP 標頭壓縮到 24bytes。IHCP 之協定運作流程如圖十六所示，其中(a)說明如何建立標頭資料庫，(b)說明如何進行標頭壓縮動作，

(c)為封包還原動作之流程。

B. IHCP Mode2

IHCP Mode2 也是一種新的 IP 協定，針對 VoIP 封包，先進行標頭壓縮動作，再將 IP/IHCP Mode1 標頭進一步簡化成 IHCP Mode2 標頭。另外，多個 RTP 承載與 IHCP Mode2 標頭可以共用相同的 IP 標頭，以完成封包串接動作。圖十七即為 IHCP Mode2 串接後之封包示意圖，圖十八為封包串接與還原之流程。

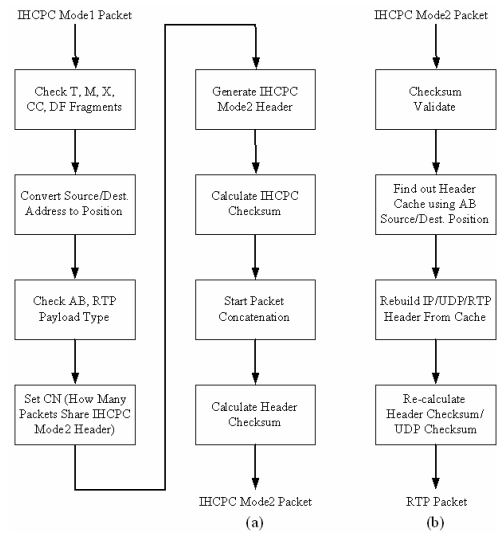


圖十七 IHCP Mode2 封包示意圖

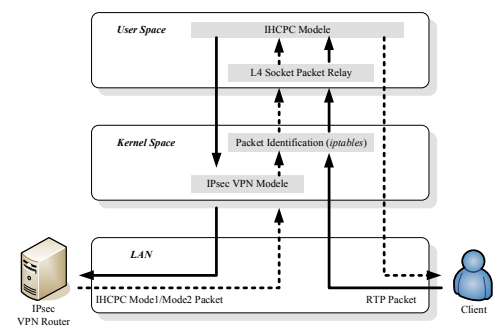
4.2.3 IHCP 機制實作方法

IHCP 標頭壓縮與封包串接模組已經成功開發在 Linux 平台上，實作環境如圖十四所示，完成之 IHCP 模組直接安裝在 IPsec VPN 路由器上，由於 IHCP 模組與 IPsec VPN 模組分屬 user-space 與 kernel-space，兩種模組互相獨立運作，並不會產生衝突。

圖十九為完成之 IPsec VPN + IHCP 路由器內部網路架構，共分三層，在 LAN 層，網路卡負責接收與傳送封包。第二層為 kernel-space 層，當 RTP 封包流入 kernel-space 後，先使用 *iptables* 將封包導入 user-space，等待 IHCP 模組進行標頭壓縮與封包串接；如果是 IPsec VPN 封包流入 kernel-space，IPsec VPN 模組會先解開 IHCP Mode1 封包或 IHCP Mode2 封包，再使用 *iptables* 將封包導入 user-space，等待 IHCP 模組解開串接封包與還原標頭。最上層是 user-space 層，也是 IHCP 真正實作的地方，利用 *libipq* 開啟 L4 socket，如果是 RTP 封包，經由 IHCP 模組壓縮標頭與封包串接後，利用 *ipq_set_verdict* 將封包



圖十八 IHCP Mode2 協定運作示意圖。
(a)封包串接流程，(b)封包還原流程。



圖十九 IPsec VPN + IHCP 路由器內部網路架構

導回 kernel-space；如果是 IHCP 封包，使用 IHCP 模組還原 RTP 封包，

再利用 *ipq_set_verdict* 將封包導回 kernel-space。

圖二十為完成之 IHCP 之軟體流程圖。

4.2.3 IHCP 效能之理論分析與實驗結果

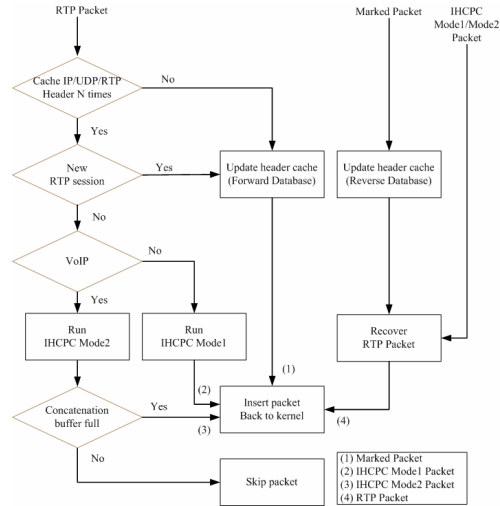
仔細計算各程標頭壓縮與串接協定之封包欄位長度，再經由下列公式

$$\text{Bandwidth Utilization (\%)} = 1 - \frac{\text{Header Length (Bytes)}}{\text{IP Packet Length (Bytes)}}$$

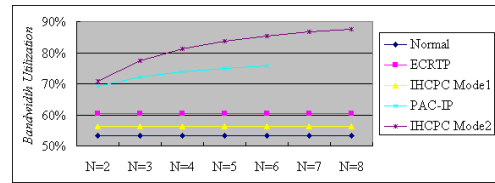
即可計算其頻寬使用效率。圖二十一為各種方法之頻寬使用效率，由此比較圖不難發現，IHCP 在結合了標頭壓縮與封包串接兩種機制之下，其頻寬使用效率均高出單單使用某一種方法之結果，相較於完全不使用標頭壓縮與封包串接機制，其改進更高達 34.3%。

接著我們使用 Chariot 產生真實的 VoIP 封包流，來測試實作完成的 IHCP 模組搭配 IPsec VPN 路由器之實際效能。由圖二十二之 MOS 比較可發現，儘管標頭壓縮與封包串接均會增加封包之延遲，但實測上對 VoIP 聲音品質之影響不大，平均 MOS 都能維持在 4 左右，可滿足業界標準之通話品質。

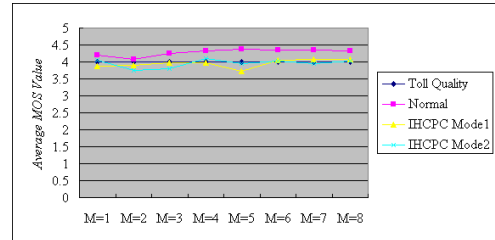
我們監測不同機制下 IPsec VPN 路由器於 kernel-space 之 CPU 平均負載結果，由圖二十三可發現不論使用何種機制其結果大致上都隨著 VoIP 電話通數呈線性成長。使用 IHCP Mode1(只壓縮標頭)時 kernel-space 之 CPU 負載稍微高於完全不用任何機制的情形，這是因為 IHCP 必須動用 kernel-space 之 *iptables* 模組來將封包



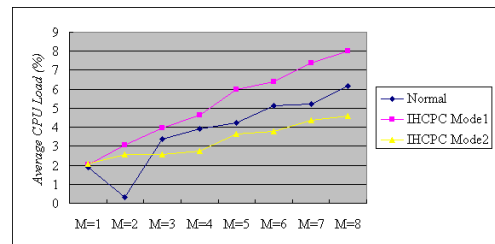
圖二十 IHCP 軟體流程圖



圖二十一 VoIP 頻寬使用效率比較圖



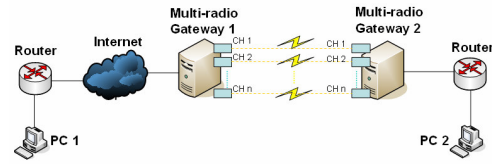
圖二十二 平均 MOS 比較圖



圖二十三 IPsec VPN 路由器 CPU 平均負載比較圖(kernel-space)

導入 user-space，因此會增加 kernel-space 之 CPU 負載。然而使用 IHCP Mode2(標頭壓縮+封包串接)時

卻能得到較低的 kernel-space CPU 負載，這樣的結果是因為封包串接機制大幅減少了 kernel-space 必須處理的封包數量，進而降低了 CPU 的負載。

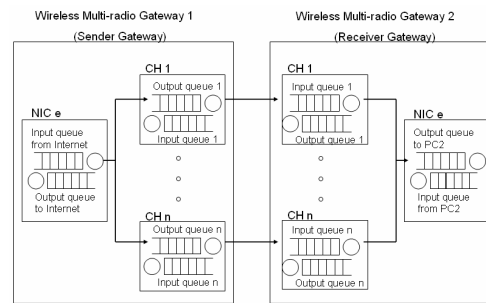


圖二十四 多重無線開道傳輸系統架構

4.3 藉多重無線設備平行傳輸增加網路電話語音品質

近年來，網路電話(VoIP)越來越受到人們的歡迎。網路電話在剛發展的階段會受到歡迎，主要原因是它是供免費的服務。但網路電話能持續受到歡迎的關鍵絕不在於它的廉價，而是它能提供持續進步的通話品質。近年來已有大量的研究著眼於提升網路電話的語音品質[20-26]。

同時由於無線網路模組的價格越來越便宜，所以各種配有多重無線模組的設備也就越來越多。我們提出改善網路電話語音品質的方法即是利用多重無線開道器平行傳輸，在沒有 ADSL 可用的時候，也可以只運用無線通道即提升語音品質。

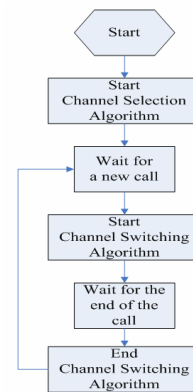


圖二十五 系統佇列模型

4.3.1 系統架構與演算法

我們提出的系統架構如圖二十四，PC1 和 PC2 透過網際網路使用網路電話通訊，並通過 Multi-radio Gateway 1 和 Multi-radio Gateway 2。現在考慮單方向由 PC1 到 PC2 的語音封包，Multi-radio Gateway 1 當做 Sender Gateway，Multi-radio Gateway 2 當做 Receiver Gateway。Sender Gateway 和 Receiver Gateway 的佇列模型如圖二十五。

Sender Gateway 必須執行通道選擇演算法，以決定把收到的語音封包轉送到那些通道。Receiver Gateway 必須執行通道切換演算法，以決定如何



圖二十六 系統操作流程圖

處理從 Sender Gateway 傳送來的語音封包。系統操作流程如圖二十六。

在說明演算法之前，我們先定義幾個簡稱和變數：

簡稱：

- connected_ch: 如果 Sender Gateway 決定把語音封包轉送到這個通道，這個通道即為 connected_ch。
- disconnected_ch: 如果 Sender Gateway 決定不把語音封包轉送到這個通道，這個通道

即為 disconnected_ch。

- forwarding_ch: 若 Receiver Gateway 決定把從此通道收到的語音封包傳送給另一端使用者，此通道即為 forwarding_ch。

變數：

- num_of_con: connected_ch 的數量。
- max_con: num_of_con 的上限，即 num_of_con 不能超過此值。
- min_con: num_of_con 的下限，即 num_of_con 不能低於此值。
- ch_add: 若 disconnected_ch 的 MOS 大於此值，即改為 connected_ch。
- ch_del: 若 connected_ch 的 MOS 小於此值，即改為 disconnected_ch。

通道選擇演算法和通道切換演法如圖二十七與二十八所示。

4.3.2 系統效能驗證

為了驗證我們演算的效能，我們用 C 語言撰寫了模擬程式，並假設 802.11a WLAN 為我們的通道類型，並設定使用的通道數目為 5。我們用兩狀態馬可夫鏈來模擬通道，其預設參數值如圖二十九。

在以下的實驗中，我們調整較差的通道狀態位元錯誤率和好的通道狀態的平均時間，並調整 max_con 和 min_con，得到的結果如圖三十、三十一和三十二。

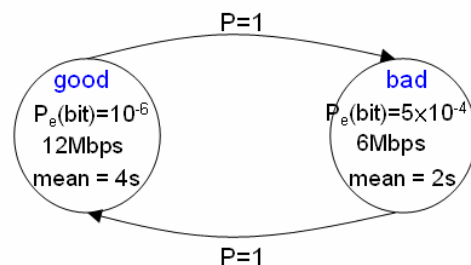
我們定義 ave. MOS > 3 為可接受的網路電話語音品質，並在表二列出

```
Design max_con, min_con, ch_add, ch_del
Schedule Send(0)
while (1) {
    switch (Event) {
        case Send(i):
            send ten consecutive probe packets to channel i within 200 ms
            schedule Send ((i+1) mod n)
        case Rec(i):
            update R-factor of Channel i
        case Call_Arrival:
            break
    }
    set the top min_con which has the highest R-factor to connected_ch
    compute ch_add and ch_del
    while (1) {
        switch (Event) {
            case Send(i):
                send ten consecutive probe packets to channel i within 200 ms
                schedule Send ((i+1) mod n)
            case Rec(i):
                update R-factor of Channel i
                if ( it is connected_ch and its R-factor < ch_del and num_of_con > min_con)
                    set it to disconnected_ch
                if ( it is disconnected_ch and its R-factor > ch_add and num_of_con < max_con)
                    set it to connected_ch
            case Call_End:
                break
        }
    }
}
```

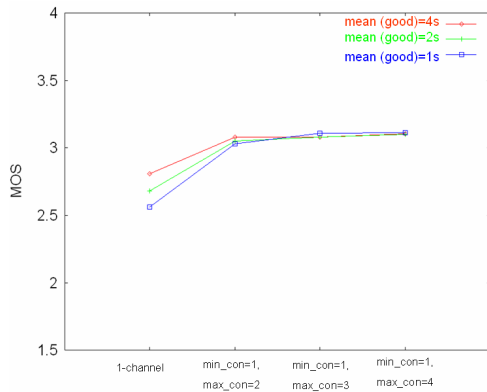
圖二十七 通道選擇演算法

```
Schedule Timeout
Schedule Scan_Channel(0)
while (1) {
    switch (Event) {
        case Scan_Channel(i):
            if (i is the for_ch)
                forward the oldest packet
                reschedule Timeout
            else if (the oldest packet of channel i has been forwarded)
                discard the oldest packet of channel i
            else if (the oldest packet of channel i has not been forwarded)
                store the oldest packet in Buffer i
                schedule Scan_Channel((i+1) mod n)
        case Timeout:
            reset_for_ch to i if Buffer i has maximal number of packets
            forward all packets in Buffer i
            drop all packets in Buffer 1 ~ Buffer n
            schedule Timeout
        case Call_Terminated:
            break
    }
}
```

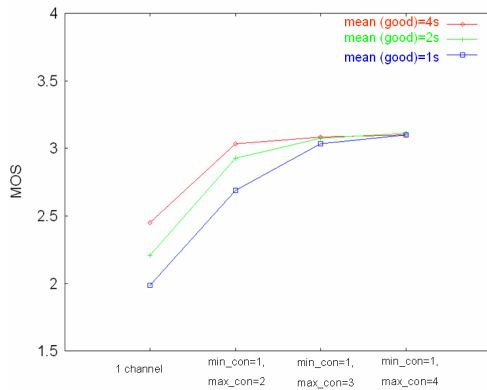
圖二十八 通道切換演算法



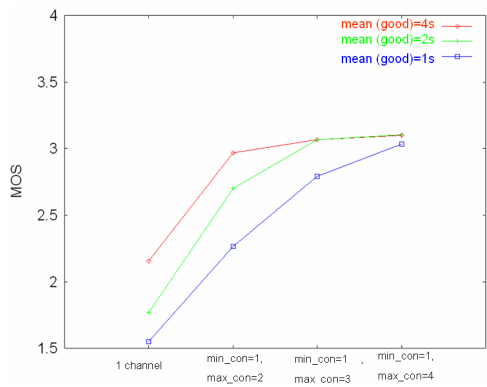
圖二十九 通道模型所使用之兩狀態馬可夫鏈與預設參數值



圖三十 較差通道狀態之位元錯誤率為 5×10^{-4} 之實驗結果



圖三十一 較差通道狀態之位元錯誤率為 7×10^{-4} 之實驗結果



圖三十二 較差通道狀態之位元錯誤率為 9×10^{-4} 之實驗結果

在各種通道特性下的 max_con 和 min_con 的建議數目。

綜合以上的實驗結果我們可發現，只要通道的位元錯誤率不是非常

	$P_e(\text{bit}) = 5 \times 10^{-4}$	$P_e(\text{bit}) = 7 \times 10^{-4}$	$P_e(\text{bit}) = 9 \times 10^{-4}$
mean (good) = 4s	min_con=1, max_con=2	min_con=1, max_con=2	min_con=1, max_con=2
mean (good) = 2s	min_con=1, max_con=2	min_con=1, max_con=3	min_con=1, max_con=3
mean (good) = 1s	min_con=2, max_con=3	min_con=1, max_con=3	min_con=1, max_con=4

表二 max_con, min_con 建議表

的高，在執行我們的通道選擇演算法和通道切換演算法後，我們的系統都可提供可接受的網路電話語音品質。

五、結論與未來展望

我們提出了三種機制來提高 VoIP 於不同特性網路中之語音品質和頻寬使用效率。封包串接與標頭壓縮兩種方法不論在無線網路或虛擬私有網路中，均能有效地降低 overhead 和提高頻寬使用效率，除了理論推導和模擬之外，我們亦實作了這兩種機制之 Linux 模組，以實驗驗證此一模組所造成之封包延遲與 CPU 負載增加均不致影響 VoIP 之通話品質，更進一步證明這兩種機制之實用性。同時我們也提出利用多重無線開道器平行傳輸的方法來改善網路電話語音品質，透過定時量測各無線通道之語音傳輸品質，我們提出了一套通道選擇與通道切換演算法，並以模擬結果找出各項參數之最佳設定值，在不同通道狀態之下均能得到良好的 VoIP 語音品質。

綜合三年研究之成果，我們不僅實作了 ADSL-WLAN 雙網接取系統，同時研究不同種類交通流於不同網路上之傳輸特性，以及各種交通混合時交互影響之效應，藉以設計交通流分配和負載平衡演算法，之後又致力於提升 VoIP 之通話品質和頻寬使用效率。這些成果結合起來，將可進一步

打造更高效能、更具智慧多網接取系統，不僅提供使用者更順暢之上網環境，還有更高品質的網路電話服務。

六、參考文獻

- [1] ITU-T, *G.992.1 – Asymmetric Digital Subscriber Line (ADSL) transceivers*, ITU-T, 1999.
- [2] ITU-T, *Draft G.992.3 AAP –ADSL2, Asymmetric Digital Subscriber Line*, ITU-T, 2002.
- [3] Committee T1 – Telecommunications, *TIE1.4 VDSL – Very-high bit-rate Digital Subscriber Line (VDSL) Metallic Interface, Part 3 : Technical Specification of a Multi-Carrier Modulation Transceiver*, Committee T1 - Telecommunications, 2001.
- [4] X. Cao, J. Li, Y. Chen, and C. Qiao, “Assembling TCP/IP Packets in Optical Burst Switched Networks,” in *Proc. of IEEE GLOBECOM Taiwan*, 2002.
- [5] J. Kang, Z. Lu, and M. Reed, “Multi-time scale adaptive optical packet assembly based on Internet traffic prediction,” in *Proc. of 31st European Conference on Optical Communication*, 2005.
- [6] D. Kliazovich and F. Granelli, “On packet concatenation with QoS support for wireless local area networks,” in *Proc. of IEEE International Conference on Communications*, 2005.
- [7] S. Ganguly, V. Navda, K. Kim, A. Kashyap, D. Niculescu, R. Izmailov, S. Hong, and S. R. Das, “Performance Optimizations for Deploying VoIP Services in Mesh Networks,” *IEEE Journal on Selected Areas in Communications*, Vol. 24, No. 11, Nov. 2006.
- [8] Y. Xiao, “Concatenation and piggyback mechanisms for the IEEE 802.11 MAC,” in *Proc. IEEE WCNC*, 2004, pp. 1642-1647.
- [9] A. Jain, M. Gruteser, M. Neufeld, and D. Grunwald, “Benefits of packet aggregation in ad-hoc wireless network,” Dept. Comput. Sci., Univ. Colorado, Boulder, CO, Tech. Rep. CU-CS-960-03, 2003.
- [10] D. Niculescu, S. Ganguly, K. Kim, and R. Izmailov, “Performance of VoIP in a 802.11-based wireless mesh network,” in *Proc. IEEE INFOCOM*, Barcelona, Apr. 2006, pp. 1–11.
- [11] W. Wang, S. C. Liew, and V. O. K. Li., “Solutions to performance problems in VoIP over a 802.11 wireless LAN,” *IEEE Transactions on Vehicular Technology*, Vol. 54, No. 1, Jan. 2005.
- [12] H. Zhai and Y. Fang, “A distributed packet concatenation scheme for sensor and ad hoc networks,” in *Proc. of IEEE Military Communications Conference*, Oct. 2005.
- [13] Z. Tsai, W. D. Wang, C. H. Chiou, J. F. Chang, and L. S. Liang, “Performance analysis of two echo control designs in ATM networks,” *IEEE/ACM Trans. Networking*, vol. 2, pp. 30-39, Feb. 1994.
- [14] IEEE Std. 802.11. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. Institute of Electrical and Electronics Engineers, 1999.
- [15] Virtual Private Network Consortium, available at <http://www.vpnc.org/>
- [16] S. Casner, V. Jacobson, “Compressing IP/UDP/RTP Headers for Low-Speed Serial Links,” RFC 2508, February 1999.
- [17] T. Koren, S. Casner, J. Geevarghese, B.

- Thompson, P. Ruddy, "Enhanced Compressed RTP (CRTP) for Links with High Delay, Packet Loss and Reordering," RFC 3545, July 2003.
- [18] C. Bormann, C. Burmeister, M. Degermark, H. Fukushima, H. Hannu, L. Jonsson, R. Hakenberg, T. Koren, K. Le, Z. Liu, A. Martensson, A. Miyazaki, K. Svanbro, T. Wiebke, T. Yoshimura, H. Zheng, "RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed," RFC 3095, July 2001.
- [19] M. Degermark, B. Nordgren, S. Pink, "IP Header Compression," IETF RFC 2507. Feb. 1999.
- [20] R. Ramjee, J. Kurose, D. Towsley, and H. Schulzrinne, "Adaptive Playout Mechanisms for Packetized Audio Applications in Wide-Area Networks," in *Proc. IEEE Infocom'94*, 1994.
- [21] S. B. Moon, J. Kurose, and D. Towsley, "Packet audio playout delay adjustment: Performance bounds and algorithm," *ACM/Springer Multimedia Systems*, 6:17--28, Jan. 1998.
- [22] [3] C. Perkins, O. Hodson, and V. Hardman, "A survey of packet loss recovery techniques for streaming audio," *IEEE Network*, vol. 12, pp. 40–48, Sept-Oct 1998.
- [23] V. Hardman, A. Sasse, and A. Watson, "Reliable audio for use over the Internet," in *Proceedings of INET '95*, June 1995, pp. 171–178.
- [24] S. Tao, K. Xu, A. Estepa, T. Fei, L. Gao, R. Guerin, J. Kurose, D. Towsley, and Z.-L. Zhang, "Improving VoIP Quality Through Path Switching," in *INFOCOM 2005*, Mar. 2005.
- [25] M. Ghanassi and P. Kabal, "Optimizing Voice-over-IP Speech Quality Using Path Diversity," in *Multimedia Signaling Processing, IEEE*, 2006
- [26] D. J. Goodman, G. B. Lockhart, O. J. Wasem and W. C. Wong, "Waveform Substitution Techniques for Recovering Missing Speech Segments in Packet Voice Communications," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. 34(6), pp. 1440-1448, 1986.
- [27] L. Ding and R. Goubran, "Speech quality prediction in VoIP using the extended E-model," in *IEEE GLOBECOM*, San Francisco, USA, 2003.
- [28] Y. Amir, C. Danilov, S. Goose, D. Hedqvist, and A. Terzis, "1-800-OVERLAYS: Using Overlay Networks to Improve VoIP Quality," in *Proceedings of the ACM International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 2005)*, pages 51-56, Skamania, WA, June 2005.
- [29] H. Melvin and L. Murphy, "Time Synchronization for VoIP Quality of Service," *IEEE Internet Computing*, Vol. 6, pp. 57-63, 2002.