南　　華　　大　　學

資訊管理學系

碩士論文

於 Gigabit 被動光纖網路下多媒體串流鑑識系統之設計與實作─以 SIP 網路電話為例

Design and Implementation of Multimedia Streaming Forensics System in a Gigabit Passive Optical Network －The Case Study of SIP Phone Applications

研 究 生：洪丞緯

指導教授：吳光閔、蘇暉凱

中華民國 102 年 07 月 19 日

於 Gigabit 被動光纖網路下多媒體串流鑑識系統之設計與實作
—以 SIP 網路電話為例

Design and Implementation of Multimedia Streaming Forensics System in a Gigabit Passive Optical Network
－The Case Study of SIP Phone Applications

研　　究　　生：洪丞緯　　Student: Cheng-Wei Hung
指　導　教　授：吳光閔　博士　　Advisor: Dr. Guang-Ming Wu
　　蘇暉凱　博士　　Dr. Hui-Kai Su

南　華　大　學

資　訊　管　理　學　系

碩　士　論　文

A Thesis

Submitted to Department of Information Management
College of Science and Technology
Nan-Hua University
in partial Fulfillment of the Requirements
for the Degree of
Master of Information Management
July 2012
ChiaYi Taiwan.

中華民國　102　年　07　月

於 Gigabit 被動光纖網路下多媒體串流鑑識系統之設計與實作 — 以 SIP 網路電話為例

學生：洪丞緯　　指導教授：吳光閔　博士

蘇暉凱　博士

南　華　大　學　資訊管理學系碩士班

摘　　要

隨著影音串流技術之發達，多媒體會談串流服務已成為電腦網路使用者最常使用服務之一，在啟用會談服務時，會談之控制通道 (Control Channel) 傳輸為利用固定已知埠號 (Well-Known Port)，但其資料傳輸通道 (Data Channel) 則利用動態非已知埠號 (Unknown Port)，需透過觀察 SIP 封包中所夾帶之 SDP 資訊得知，故在數位鑑識上之實現有其難度。由於網路架構之迅速發展，從傳統撥接存取網路到現今所提供之光纖網路，網路速度與品質不斷地提升，同時也導致網路攻擊難以追縱與記錄，延伸出許多網路安全問題，故須透過數位鑑識還原網路使用之記錄。數位鑑識又稱為電腦鑑識，乃透過電腦鑑識技術輔助偵查與還原環境，透過分析與比對還原案發當時之環境。傳統 GPON 網路鑑識大多利用封包在骨幹傳輸時以局端 OLT 作為監聽點，透過軟體監聽封包，但由於 GPON 骨幹網路擁有 2.5Gbps 之上下行對稱速率，如以傳統之鑑識架構在 GPON 環境下單一節點高速傳輸之流量會難以完全負荷，進而造成封

包遺漏不完整。本論文提出兩層式負載分散之架構，以自行開發之系統元件：Snooping Agent、Analyzing Server、與 Media Processing Server。透過系統建構於 GPON 環境上，先於用戶端 ONU 建構 Snooping Agent 監聽 SIP 網路電話控制通道並回傳至後端 Analyzing Server 分析找出資料傳輸通道所使用之傳輸埠號 (Port Number)，再將監聽得到之 Port Number 給予 ONU Snooping Agent 元件，將目的地 IP address 與影音 Port Number 設定至過濾條件中過濾影音封包並回傳至多媒體處理伺服器，並將結果儲存於資料庫中並透過 Web 介面查詢。透過本論文兩層分散監聽負載之作法可將 GPON 網路龐大之流量於用戶端 ONU 先行過濾與分析，並且降低集中式分析器與資料儲存負載，以提升網路鑑識效能。

關鍵字：千兆位元被動光纖網路、影音串流、會談初始協定、會談描述協定、數位鑑識

Design and Implementation of Multimedia Streaming Forensics System in a Gigabit Passive Optical Network – The Case Study of SIP Phone Applications

Student: Cheng-Wei Hung

Advisors: Dr. Guang-Ming Wu
Dr. Hui-Kai Su

Department of Information Management
The Graduated Program
Nan-Hua University

ABSTRACT

With the mature development of video and audio streaming applications, the multimedia session streaming services have become one of the popular internet services. While using the session service, the session control channel is fixed and using a well-known port, but the data channel is using select a dynamical and unknown port. The data channel would be decided in the control messages. For SIP (Session Initiation Protocol) applications, the voice data channel would be aware from the SDP (Session Description Protocol) information of SIP messages. Therefore, it's difficult to implement a digital forensics system for multimedia session streaming services. Because of the rapid development of network architecture, the speed and quality of networks is increasing continually, such as from traditional dial-up access networks to fiber optic networks. The malicious attack from internet becomes difficult to tracking and record the illegal network behavior. Many network security problems are spread. Thus, it has to be redrawn by using digital forensics system to diagnose and recover the security events. Digital forensics is also called computer forensics. The network situation and behavior of the security

events would be replayed by using computer forensics technology. The network packets are captured in OLT (Optical Line Termination) by using a traditional network forensics for GPON (Gigabit-capable Passive Optical Networks). Due to the symmetrical network speed with 2.5Gbps, the forensics task could not be handled in the high-speed situation. Some packets could be lost and the forensics is incomplete. This thesis proposed a two-tier architecture of forensics system with distributed loading. The system components were developed: Snooping Agent, Analyzing Server, and Media Processing Server. The System is design for GPON environment. Snooping Agent on the ONU (Optical Network Unit) deals with the packet capturing of SIP control channel, and the captured SIP packets are sent to the back-end component (Analyzing Server). The port numbers of the data channels will be figured out by Analyzing Server. According the port numbers, the audio and video packets will be captured and delivered to Media Processing Server. All of the session information and users data is stored in database and presented with web interface for event search. This thesis presented the two-tier structure of forensics system with distributed loading can reduce the loading of the centralized analyzer and data storage. The most packets are filtered in each ONU, and only the captured packets would be analyzed or stored.

Keyword: Gigabit Passive Optical Network, multimedia streaming, Session Initiation Protocol, Session Description Protocol, digital forensics

目錄

圖目錄

第一章、緒論

1.1 研究背景與動機

隨著資訊時代的來臨與網際網路技術的快速演進，網際網路之頻寬日漸增加，傳統之 Cable Modem 或 ADSL 上網服務已無法滿足使用者之需求。光纖接取網路中之 GPON (Gigabit Passive Optical Network) 架構可提供高速且穩定之網路服務，預計將成為未來 FTTx (Fiber To The x) 光纖通訊網路架設之重要技術。所謂 FTTx 意即各種光纖通訊網路之總稱，其中 x 代表目的地，最常見之服務為 FTTH (Fiber To The Home)，即光纖到府之服務。

由於網路知識容易取得之特性，同時也延伸出許多網路安全之問題，使用者透過網路學習駭客行為攻擊其他使用者之比例日漸提高，故資訊在傳輸時所帶有之風險也日漸增加。駭客攻擊行為可能在瞬間造成整個網路服務之癱瘓，當事件發生後其追查之時效性有限，使用者所使用之電腦環境亦容易遭到更改或破壞，加上網路流量日漸增大之緣故，同時也大幅提升數位鑑識之困難度。

隨著網路服務漸趨廣泛與傳輸速度之加快，使用者可透過網際網路工作開會、購物交易、網路交友或取得各種知識…等。其中，使用者經常透過即時通訊軟體所提供之影音會談與檔案傳輸之服務進行遠端會議，透過影音會談，使用者可於電腦前與世界各地之使用者進行面對面之會議，並可透過檔案傳輸之功能將會議所需發佈之檔案傳輸至各使用者之電腦。此時，使用者便有可能接收到帶有攻擊內容之檔案而導致電腦環境因遭受攻擊而癱瘓。

因此，對於數位鑑識如何適應當前網路架構並針對影音會談之鑑識行為，透過紀錄與保存使用者使用會談通訊所傳輸之封包並分析儲存至資料庫中以提供鑑識人員還原當前受到攻擊之環境。

1.2 系統簡介

本論文之系統架構以 Gigabit 被動光纖網路為底層架構，在 GPON 環境上研究與設計多媒體串流鑑識系統，分析 SIP 網路電話。由於傳統之被動光纖網路鑑識架構大多以局端 OLT 作為監聽點，並透過骨幹網路傳輸所監聽之封包，此做法對於 GPON 環境擁有上下行頻寬 2.5Gbps 之流量如以軟體實現監聽系統在擁有高速封包交換之局端 OLT 恐怕難以完全負荷，而導致監聽封包之流失而不完整。此為本系統鑑識時所遭遇之問題一。

目前網路使用者常使用之網路服務中多媒體影音串流為目前最常使用網路服務之一，使用影音串流會談服務時，由於會談之 Control Channel Port Number 為已知式 (Well-Known Port)，而其 Data Channel 為非已知式 (Unknown Port)，在會談將要建立時之 Control Channel 中才會告知，故在鑑識上有其難以實現之難度。此為本系統鑑識時所遭遇之問題二。

為解決上述兩個所遭遇之瓶頸，本論文提出兩層式架構分散負載之技術，將負載分散於用戶端 ONU 與後端之分析伺服器中，並將其所分析之資訊儲存至資料庫中以供後續鑑識時可以使用。本論文於用戶端 ONU 上設計 Snooping Agent 元件監聽並擷取封包，並以 UDP Tunnel 之技術傳送至後端所設計之 Analyzing Server 分析所擷取影音封包之 Control Channel 中所夾帶之 Data Channel Port Number，並將其傳輸至後端 Media Processing Server 分析還原成原始資訊並儲存於資料庫中。透過此做法可將原本於局端 OLT 監聽之單點流量，先行於用戶端 ONU

與後端分析伺服器與多媒體處理伺服器上將流量分散掉，以提升網路監聽之效能，以適應未來更複雜之網路環境。

1.3 論文架構

本論文初期將會以 SIP 網路電話之封包傳輸行為進行觀察，透過 Wireshark 軟體觀察其傳輸行為並進行程式撰寫，規劃整體系統之環境，以 GPON 環境為架構進行探討研究與設計。本論文內容之架構如下：

第一章 「緒論」。對於論文之研究背景動機與系統簡介之相關說明。

第二章 「背景知識」。對於研究與實作時之相關背景知識進行說明。

第三章 「系統設計」。對於系統所設計之元件與整體架構進行說明。

第四章 「系統實作」。對於所設計之系統進行實作之進行說明。

第五章 「系統測試與效能分析」。對於系統實作測試後之結果與效能分析進行說明。

第六章 「結論與未來展望」。對於本論文之測試結果與未來發展進行相關說明。

第二章、背景知識

本章節首先將介紹本系統架構建置之 Gigabit 被動光纖網路相關背景知識;接著介紹本系統所使用之相關會談技術,內容為 SIP 網路電話；最後將介紹數位鑑識之相關背景知識。

2.1 Gigabit 被動光纖網路

被動光纖網路 (Passive Optical Network , PON) 是由國際電信聯盟電信標準化部門 (ITU Telecommunication Standardization Sector, ITU-T) 所制定，早期由 ITU-T 所制定之 G.983 主要規範出 APON (ATM Passive Optical Network) 與 BPON (Broadband Passive Optical Network) [1]，此兩種技術標準主要佈建方式為光纖到路邊 (FTTC, Fiber To The Curb)，應用於商業應用。隨著技術演進，IEEE 於 2004 年完成 EPON (Ethernet Passive Optical Network) 標準之制定，PON 之網路架構也由 APON、BPON 演進至 EPON，EPON 為使用乙太網路數據標準光纖網路技術，並提供影像、語音與數據之服務應用。基於 BPON 技術發展，ITU-T 於 2008 年完成對於 GPON (Gigabit Passive Optical Network) 標準之定義 G.984 [2]，定義中指出 GPON 提供更高傳輸速率與支援多種服務，並支援服務品質 (Quality of Service; QoS) 之保證能力。由於光纖接入技術推陳出新，A/B/GPON 之技術對於未來 FTTx 之推廣與應用具有相當之潛力與幫助。

根據法國顧問公司 IDATE 指出，全歐洲光纖到府 (FTTH) 用戶數應可到達 1170 萬用戶 [3]，由第七屆 FTTH 亞太委員會 2012 年會提到，全球光纖到府市場有 70% 在亞洲，而 IDATE 亦指出亞洲國家中，

目前光纖滲透率最高國家為韓國高達 73% [4]。IDATE 分析師提到，推動 FTTH 快速發展之應用主要為娛樂，娛樂又分為：電視、IPTV 與網路視訊串流等應用，有數據指出在 2015 年時，對於頻寬的需求中多媒體串流將佔有 60% 之需求，以日、韓兩國發展較快之市場評估，串流視訊、遠程教育、IPTV 與遠程醫療等服務對於網路頻寬之要求正是未來網路發展之重點方向。故評估 GPON 對於未來國際各國通訊發展之建設，由於 GPON 具有高頻寬、高效能、傳輸距離遠，並支援多種服務 (包含：ATM、Ethernet、TDM)、OAM&P 能力、保護安全和可升級之能力，為目前支援較多之 PON 網路技術。並提供種類廣泛之服務，如語音 (Voice Communications)、影像 (Video)、視訊會議 (Video Conferencing)、數據流量 (Data Traffic) 與綜合數位信號傳輸 (Digital Signal Transmission) 等均可在 GPON 架構下傳輸 [5]。故 GPON 網路架構對於未來之應用與發展將存在龐大的潛力。

由於近年來智慧型手機與平板電腦之普及，行動網路之應用越來越廣泛，用戶透過行動裝置瀏覽網際網路、收看串流影音、撥打網路電話與線上遊戲…等應用，造成網路流量之負擔越來越大。由於 GPON 擁有動態調整頻寬與支援 QoS 品質保證之能力，故 GPON 亦適合被應用在行動後端網路 (Mobile Backhaul Network)，擔任基地台接入網路之角色。

GPON 之架構有別於傳統光纖網路點對點 (Point to Point, P2P) 之拓樸架構，所採用為點對多點 (Point to Multi-Point, P2MP) 之拓樸架構，並透過局端設備 OLT (Optical Line Terminal) 經由被動式分光器 (Optical Splitter) 分散給多個用戶端設備 ONU (Optical Network Unit)。GPON 傳輸模式與 EPON 類似，下行 (OLT 至 ONU) 傳輸採用使用廣播模式 (Broadcast)，上行 (ONU 至 OLT) 則使用分時多工之模式 (Time

Division Multiplex, TDM)，當用戶需要上透過上行頻道傳送資料時，OLT 會對 ONU 採用輪詢 (Polling) 之方式傳輸，即當 ONU 之動態平寬分配 (Dynamic Bandwidth Assignment, DBA) 狀態報告時向 OLT 提出將要傳送之時槽大小 (Timeslot Size)，而 OLT 會於所要求之大小，可選擇採用固定式 (Fixed) 或動態配置 (Dynamic) 之方式來決定配置方式，並於傳送上傳頻寬對應 (Upstream Bandwidth Mapping) 封包時告知 ONU 所要使用之方式。ITU-T 於 2003 年制定 GPON 標準 G.984.1、G.984.2 與 G.984.3，根據此標準 GPON 之下行速率擁有 1.2Gbps 與 2.4Gbps 兩種，上行速率有 155Mbps、622Mbps、1.2Gbps 或 2.4Gbps 共四種速率，而 GPON Service Requirement (GSR) 並無規定對稱或非對稱之傳輸方式，因此電信業者或網路服務提供商可依所要提供之規格搭配出上下行傳輸速率。依照 GPON 規範之內容，GPON 實體傳輸之最大距離為 20 公里，邏輯傳輸之最大距離為 60 公里；不同於 APON 與 EPON 僅能提供 32 台 ONU 連接，GPON 之定義可提供 64 台 ONU 之連接。

在本論文之系統設計上，根據 GPON 上行傳輸 TDM 之特性，於用戶端之 ONU 設計一 Snooping Agent 元件，透過先行擷取影音封包之 Control Channel 並傳送至後端 Analyzing Server 分析出其多媒體會談所使用之 Port Number，再回傳至 Snooping Agent 過濾並擷取出封包，並將其所擷取之影音會談封包傳送至後端 Media Processing Server 重組還原其資訊並儲存於資料庫中；透過本論文所設計之兩層式負載分散做法，系統可於 ONU 先行分散掉監聽封包時之負載，以提高多媒體串流鑑識之效能。

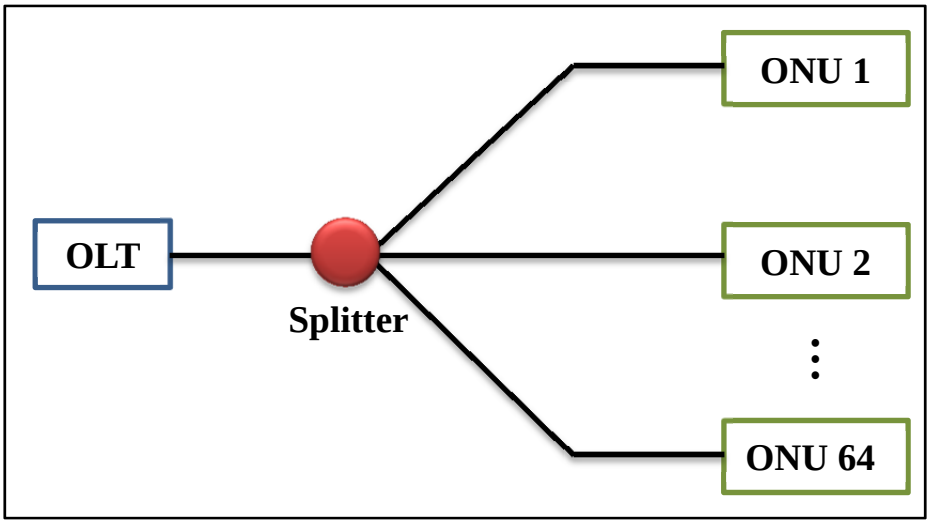

圖 2.1.1 GPON 網路架構圖

圖 2.1.1 為典型之 GPON 系統架構，主要由局端 OLT、用戶端 ONU 與被動式分光器 POS 三者所組成。

1. OLT (Optical Line Termination)：OLT 又稱為光纖網路局端，放置於中心機房，在下行方向 OLT 提供 PON 之光纖介面；在上行方向 OLT 提供 Gigabit Ethernet (GE) 介面。OLT 可用來控制用戶傳輸品質、實現網路安全控制、獲取系統狀態與用戶狀態資訊以及提供有效用戶隔離…等功能。OLT 應用以下兩種功能：1) 提供網路服務供應商之設備與 PON 所使用之光纖信號間之轉換；2) 協調網路中轉換裝置與用戶端 ONU 間之多工。
2. ONU (Optical Network Unit)：ONU 又稱為光纖網路用戶端，其功能為用來接收由 OLT 下行所發送之廣播數據，並回應由 OLT 所發送之控制命令並做出相對應之調整。ONU 對用戶間主要採用乙太網路協定，故通訊過程中無須再進行傳輸協定之轉換，以實現 ONU 對於用戶資料之透明傳送，進而達到 OLT 至 ONU 間之高速資料轉發。
3. POS (Passive Optical Splitter)：POS 又稱為被動式分光器，為連接局端 OLT 與用戶端 ONU 之無電源設備，當下行傳輸經過 Splitter 時，

Splitter 會將傳輸之資料分發，其示意圖如圖 2.1.2；當上行傳輸經過 Splitter 時，Splitter 會將傳輸之資料集中上傳。GPON 之 POS 分光率可從 2 至 64，可依照網路需求選擇所需之設備，並進行多層級連接。

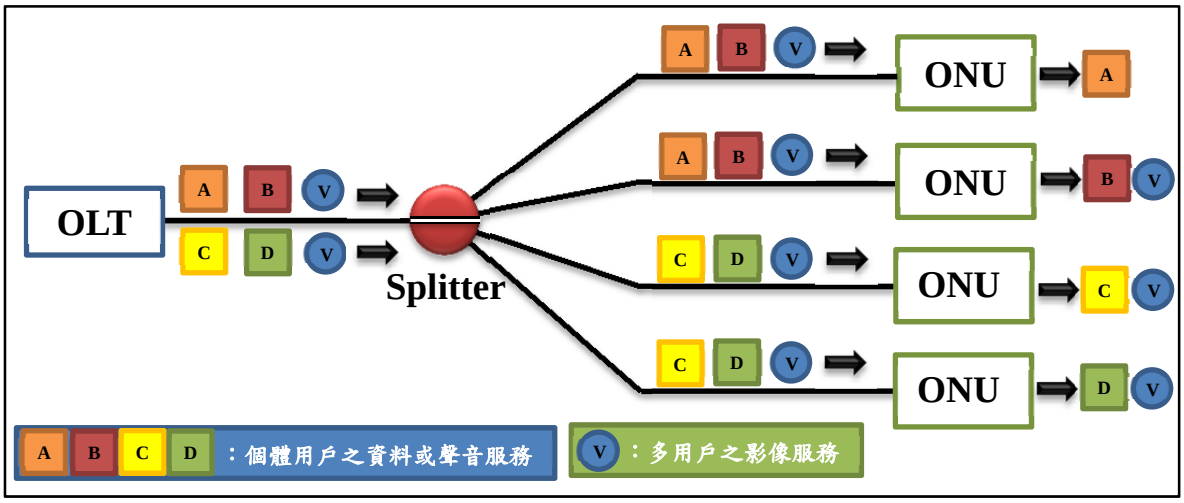

圖 2.1.2 GPON 下行傳輸廣播示意圖

而 GPON 網路架構目前大多利用 Static Time Division Multiple Access (靜態 TDMA) 之多工存取方式，由 OLT 分配於每個所連接之 ONU 固定時槽進行傳送，ONU 必須在被分配之有限時間內將緩衝區中所暫存之資料傳送至 OLT 端。GPON 服務品質控制之相關研究可參考。

2.2 會談串流應用

在本論文之會談串流應用為 SIP 網路電話，以下將介紹 SIP 網路電話軟體之發展與架構。

2.2.1 SIP (Session Initiation Protocol) 網路電話

SIP (Session Initiation Protocol) 源於美國哥倫比亞大學教授 Henning Schulzrinne 及其研究小組於 1996 年設計發展，其內容主要

為多方多媒體會談控制 (Multiparty Multimedia Session Control, MMUSIC) 標準，其中包含 SIP 之內容。1999 年，Schulzrinne 教授將其研究中有關多媒體之內容刪除後提交至 IETF (Internet Engineering Task Force) 審查，隨後 IETF 工作小組正式制定第一個 SIP 之標準，即 RFC 2543 [10]。由於 SIP 定義逐漸受到重視，故同年 IETF 並針對 SIP 成立 SIP working group 針對 SIP 2.0 標準研究制定，於 2001 年發佈 RFC 3261 標準 [11]。隨著 RFC 3261 標準之發佈確立 SIP 之基礎，隨後更發佈數個 RFC 增訂版本充實 SIP 協定之內容，例如：RFC 3262 [12] 規範 SIP 臨時回應之可靠性、RFC 3263 [13] 確立 SIP 代理伺服器之定位規則、RFC 3264 [14] 提供 SDP 應答模型 (Offer/Answer Model) 以及 RFC 3265 [15] 確立 SIP 具體事件通知。

SIP 只是單純 Call Setup 之方式，主要處理會談之建立、變更、以及結束，本身並不提供服務，只提供原始之架構，故 SIP 必須結合其他協定，例如：SDP (Session Description Protocol)，用來描述會談媒體之特性；利用 RTP (Real-Time Transport Protocol, RFC 1889) [16] 傳送串流資料 (Streaming Data) 與控制訊號 (Control signals)。SIP 之控制訊號與多媒體串流傳輸示意圖如圖 2.2.1。

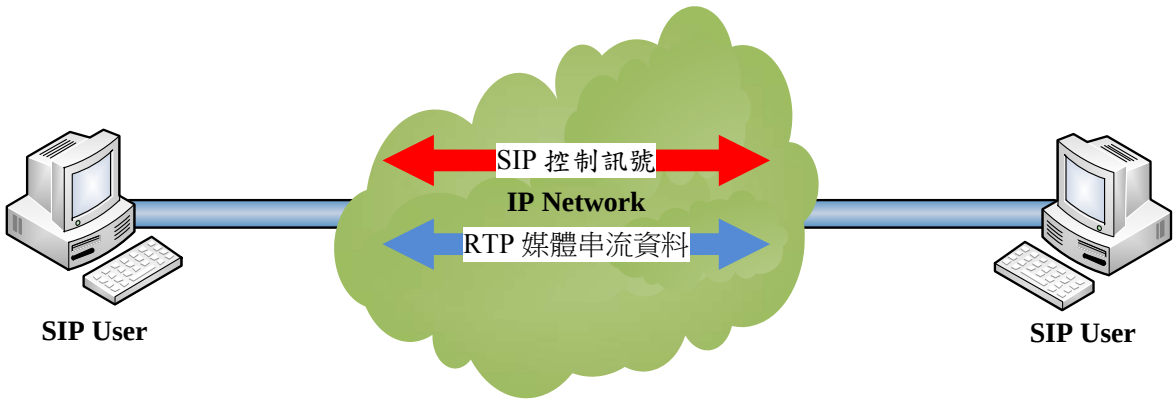

圖 2.2.1 SIP 之控制訊號與多媒體串流傳輸示意圖

SIP 之使用者定址方式利用 SMTP 定址方式，SIP address 近似於 E-mail address 之表示方式 (如： user1@host，假設用戶 user1 於南華大學之 Domain 下，其 SIP address 便以 ”sip:user1@nhu.edu.tw” 呈現)，此識別方式也更容易辨識其註冊之所在位置。SIP 擁有以下之特性：

- **利用文字 (Text-Based) 方式編碼**：其方式類似於 HTTP/1.1，由於 SIP 是以明文傳送，故可直接使用 HTTP 等格式直接溝通，其擴充性與相容性也較優越。
- **Client-Server 架構**：SIP 之網路實體可分為 Clients 端之 SIP 撥話者，稱為 Caller，負責產生呼叫；以及 Servers 端之受話者，稱為 Callee，負責回應呼叫。
- **訊號與資料獨立**：會談建立之架構中，SIP 負責多媒體會談之控制訊號，多媒體資料則透過 SDP 描述，使用 TCP、UDP、RTP 等協定傳遞資料。
- **擴充性與相容性**：SIP 可與 IETF 所制定之協定配合，如：SDP、HTTP、URL…等。

SIP 之 Client-Server 架構中，其標準系統元件包含以下三大類共五項元件。SIP 系統元件圖如圖 2.2.2，詳細描述如下：

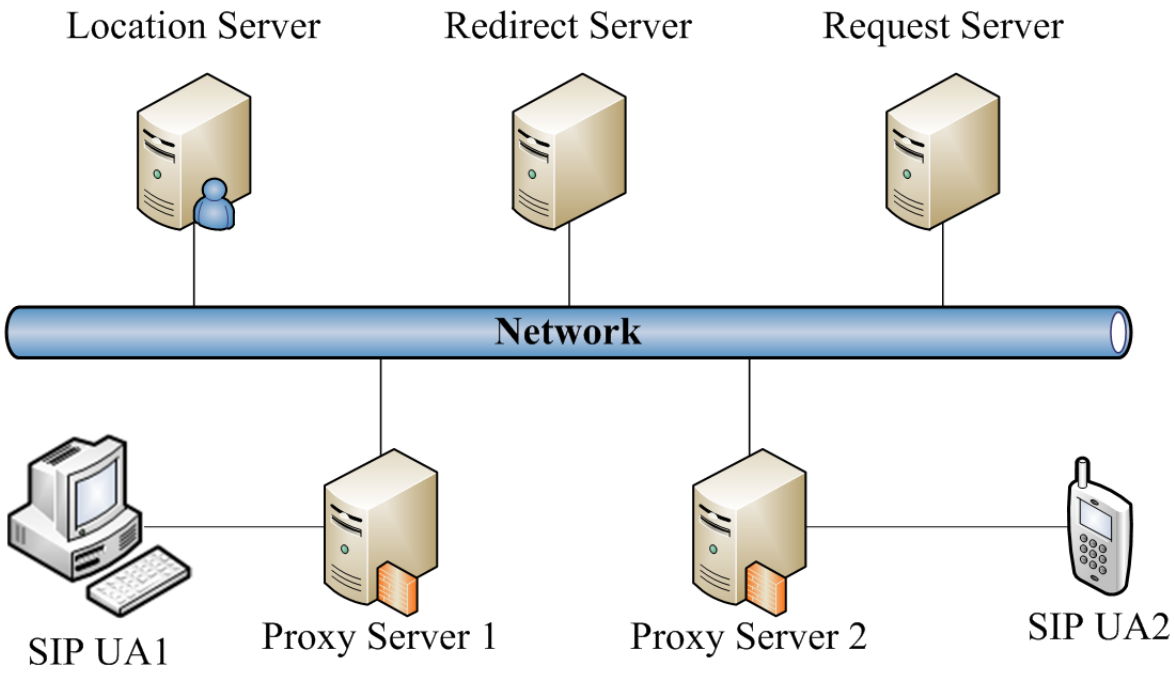

圖 2.2.2 SIP 系統元件圖

1. **用戶端代理元件 (User Agent; UA)：**

   UA 為 SIP 基礎之用戶端代理元件，負責產生 SIP 之要求會談建立訊號，用來建立多媒體會談 (media session) 與傳送及接收多媒體資料。UA 又細分為用戶端使用者 (User Agent Client; UAC) 與伺服器端使用者 (User Agent Server; UAS) 兩種，UAC 指的是起始連線之發話端，負責產生 Request 與處理回船隻 Response，UAS 則為受話端，負責接收由 UAC 發送之 Request 並產生 Response 訊息回傳。

2. **伺服器 (Servers)：**

   根據 RFC 2543 之定義，SIP 之伺服器主要分為代理伺服器 Proxy Server、重新導向伺服器 (Redirect Server)、與註冊伺服器 (Register Server) 三種：

   ✓ **代理伺服器 (Proxy Server)：**

   SIP Proxy 為 UA 與 UA 間之溝通代理人，負責接受

UA 或其他 proxy 所發送之 SIP Request，並轉送連線控制訊息至其他地方。而在網路中，要將訊息由 UAC 傳送到 UAS 則需透過多個 Proxy Server 轉送，其轉送之路徑資訊皆會記錄至訊息中，當 UAS 要回傳時便會由訊息中所紀錄之路徑繞送回 UAC。

- ✓ **重新導向伺服器 (Redirect Server)：**

  Redirect Server 用來告知 UAC 要尋找之 UAS 位置資訊，其功能只負責告知 UAC 位置資訊，並不對 UAC 所發出之訊息做任何處理動作。

- ✓ **註冊伺服器 (Registrar Server)：**

  Registrar Server 用來處理註冊及認證，當此伺服器接收到 SIP Registrar 請求時，會讓 UA 記錄當下位置資訊，並更新 UA 在 Location Server 或其他資料庫中之資訊。使用 Registrar Server 最重要之功能即能適應使用者之移動性。通常 Registrar Server 會與 Proxy Server 或 Redirect Server 結合。

3. 位置伺服器 (Location Servers)：

根據 RFC 2543 所制定，Location Servers 可將它視為用來儲存 UA 之 URLs 資訊、IP Address 資訊、身分資訊與特性…等資料之資料庫。SIP UA 不能直接存取 Location Server，需透過 Proxy Server、Redirect Server、或是 Register Server 存取其資訊。

4. SIP 登入與會談建立流程

1. SIP register 註冊

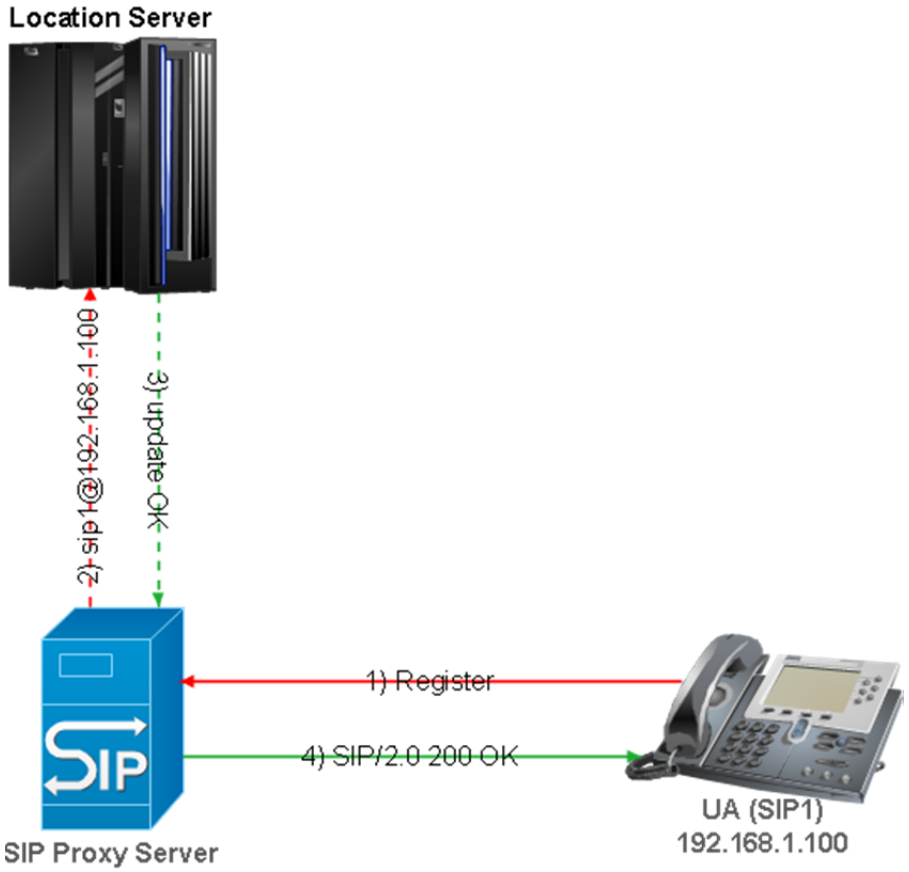

圖 2.2.3 SIP 網路電話註冊流程圖

SIP 網路電話之註冊流程如圖 2.2.3 所示，當使用者登入或變更 IP Address 時便需執行註冊之行為。當使用者登入時，UA 會透過 Register 訊息通知 Proxy Server 進行註冊之行為，當 Proxy Server 收到訊息後，會依據 UA 所提供之資訊將其 IP Address、SIP URL 與 Proxy Server 之 IP Address 傳送至 Location Server 記錄。註冊成功後，Location Server 會回傳 OK 訊息至 Porxy Server ，當 UA 接收到由 Proxy Server 所發出之 200 OK 訊息時，即表示 SIP 網路電話之註冊已完成。

2. Session call setup 會談通話建立

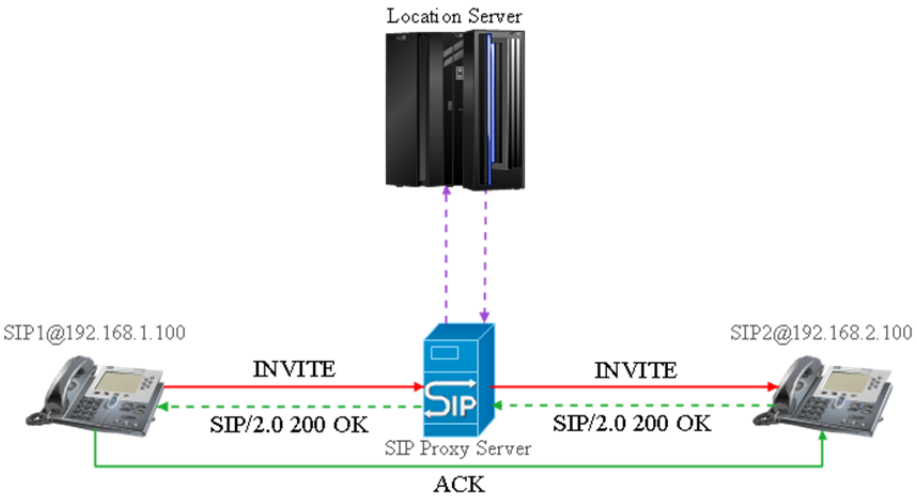

圖 2.2.4 會談通話建立 - Proxy Mode

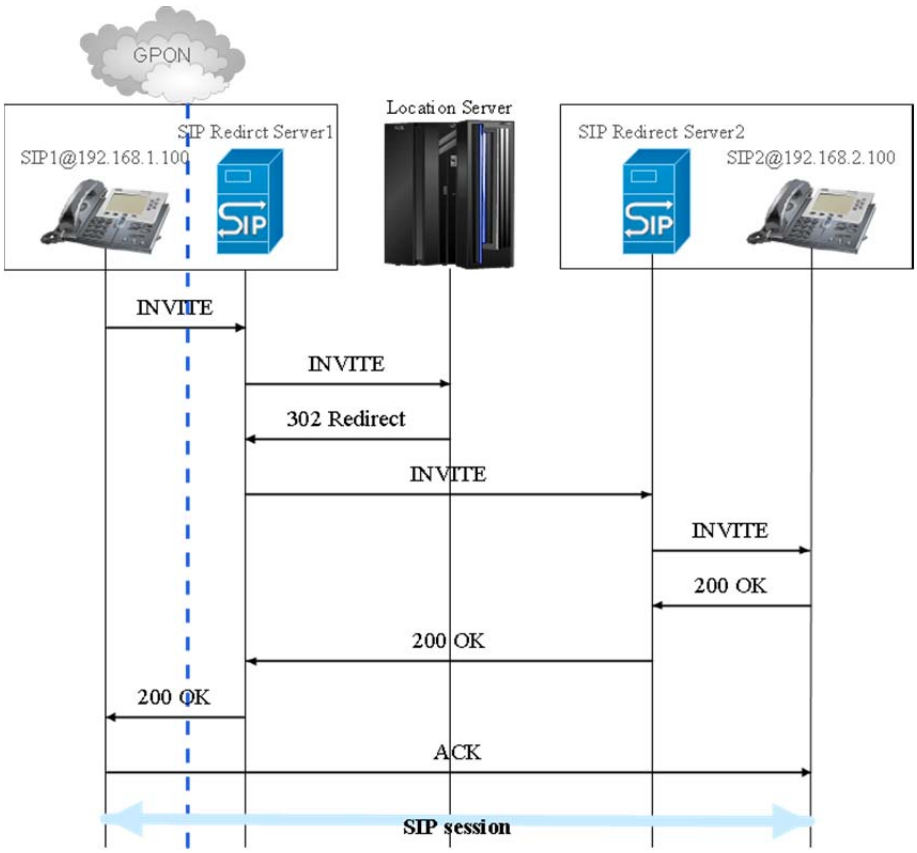

圖 2.2.5 會談通話建立 - Redirect Mode

SIP 會談建立有兩種方式，分別為 1) Proxy Mode、2) Redirect Mode，其建立流程如圖 2.2.4、圖 2.2.5，當 SIP1 需要對 SIP2

建立通話時，SIP1 會送出 INVITE 訊息，當 Proxy Server 收到後便會向 Location Server 查詢 SIP2 之實際位址並代為發送 INVITE 訊息給 SIP2；當 SIP2 收到訊息後會依據 INVITE 訊息中所夾帶之 SDP 資訊判斷是否接受會談通話，如果接受後便會依原路徑發出 SIP/2.0 200 OK 至 SIP1，當 SIP1 收到 200 OK 後會回傳 ACK 訊息給 SIP2，此時即代表 SIP 會談建立完成。

而 Proxy Mode 與 Redirect Mode 之差別在於 Proxy Mode 之 INVITE 訊息是由 Proxy Server 代為發送；而 Redirect Mode 之 INVITE 訊息是由 SIP1 自行重新發送。

3. 結束通話

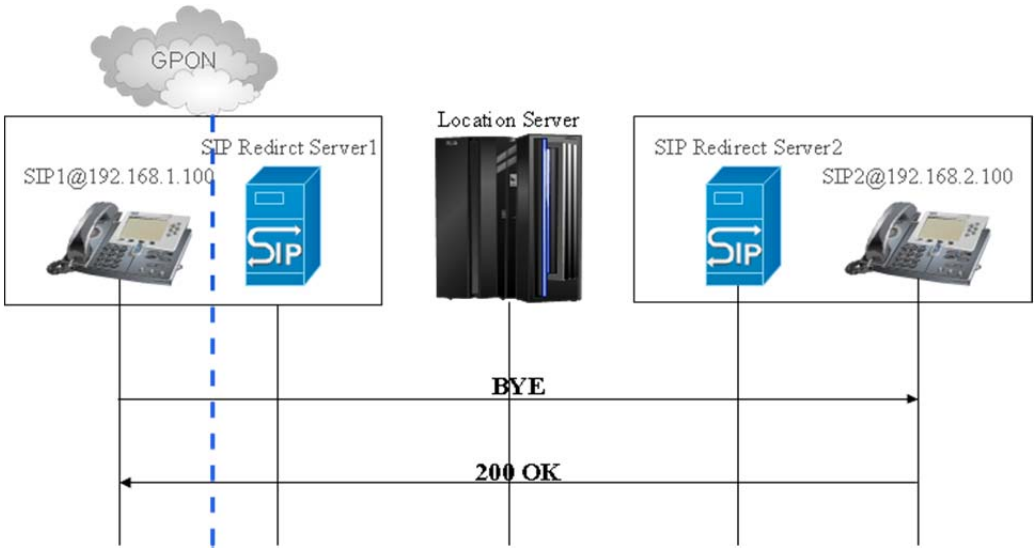

圖 2.2.6 SIP 會談結束通話序列圖

SIP 會談結束通話序列如圖 2.2.6，當通訊中的其中一方欲結束通話時 (在此假設是由 SIP1 欲結束通話)，SIP1 會發送 BYE 訊息告知 SIP2，而當 SIP2 收到訊息後會回覆 200 OK 訊息，此時即代表通話結束。

2.3 數位鑑識

網路鑑識為網路安全中所延伸之重要環節之一，主要著重於擷取封包、紀錄行為、與分析網路數據流量及事件 [17]。網路鑑識乃透過應用各種網路協定技術分析網路之封包 [18]，並透過擷取、檢查、分析、報告…等呈現出網路鑑識之結果 [19]。使用網路鑑識可觀察出隱藏在正常傳輸行為下之異常活動,並可察覺與還原出某一時間之網路行為及內容，以防止攻擊與入侵行為發生，或還原遭受攻擊與入侵後之電腦環境。網路鑑識為防火牆與入侵偵測工具後之安全分析保護系統，透過網路鑑識之結果可提供網路安全系統模組規劃，提升整體網安能力；並提供網安管理員針對網安警告之分析與回應能力；並可透過收集保留之網路傳輸原始資料，提供歷史資訊檢索之功能。

網路鑑識之蒐證與調查皆於遭受攻擊後才啟動，而目前網路鑑識之蒐證方式大多以監聽網路流量為主，當鑑識人員監聽所流過之封包時，所有封包將被完整記錄並複製傳送到分析系統中，儲存封包流量資料之用意在於保證資料內容不會在過程中被更改或是因為時間流逝而遺失，之後透過分析及檢驗便可達到網路鑑識蒐證之目的。

網路鑑識技術於目前所遭受到之阻礙包括以下：

1. 由於高速網路之普及與網路流量日漸上升，導致網路鑑識之效能與負載相對變重。
2. 由於網路流量提升，非具攻擊性之網路資料複雜度高，其干擾將增加鑑識之困難度。
3. 由於駭客之攻擊手段不斷翻新，隱藏在正常行為下之攻擊行為可能潛伏於電腦很長時間而不被發現，故網路鑑識記錄時需要龐大儲存空間。

傳統 GPON 之鑑識行為大多透過監聽局端 OLT 並將所有封包資訊複製傳送至分析伺服器中，由於 GPON 結構之單點流量最高可達 2.5Gbps，故容易造成系統之效能瓶頸與分析上之困難。由於多媒體串流會談之 Control Channel Port Number 為已知埠號 (Well-Known Port)，而 Data Channel 為非已知埠號 (Unknown Port)，故本研究設計一兩層式分散負載之架構，透過設計於 GPON 用戶端 ONU 元件 Snooping Agent 先監聽多媒體串流會談之 Control Channel，並利用 UDP Tunnel 封裝技術傳送至後端本研究所設計之 Analyzing Server，經由 Analyzing Server 分析出會談所使用之 Data Channel Port Number，並告知 Snooping Agent 針對分析出之埠號擷取會談內容封包，並利用 UDP Tunnel 封裝技術傳送至後端本研究所設計之 Media Processing Server，透過重組還原出原始會談內容之檔案。

第三章、系統設計

本章節主要將介紹本論文之系統架構，第一節將介紹本論文所設計之系統架構，並於第二小節介紹並說明本研究於系統架構中所設計之各元件之功能與運作流程。

3.1 系統架構

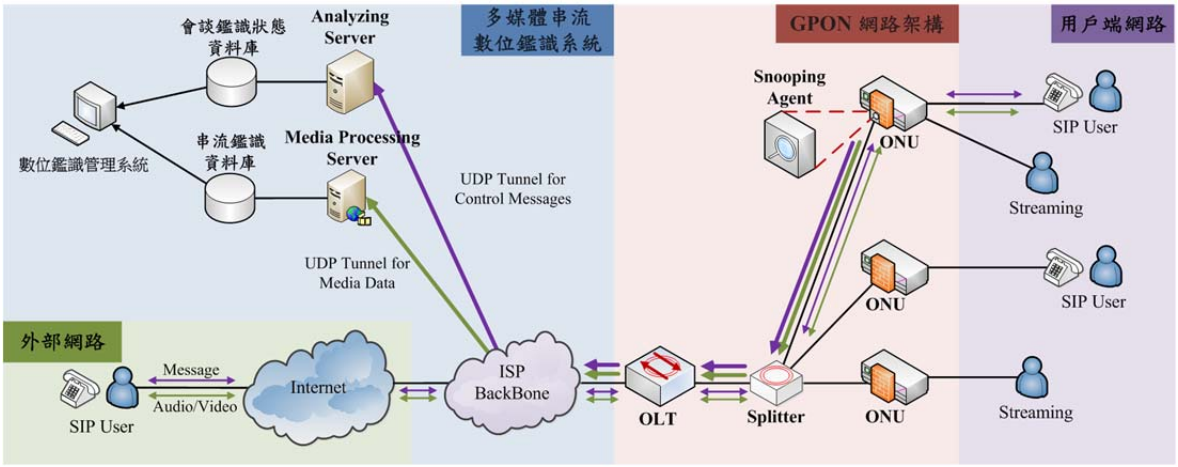

圖 3.1.1 系統架構圖

本論文所提出之系統架構圖如圖 3.1，底層為 GPON 之網路架構，包含局端元件 OLT 與用戶端元件 ONU，其中用戶端元件將架設本研究所設計之 Snooping Agent 元件擷取監聽封包，OLT 與 ONU 兩元件之間將利用光纖網路與被動式分光器 Splitter 串接；後端多媒體串流數位鑑識系統則包含本研究所設計之 Analyzing Server、Media Processing Server、資料庫、與數位鑑識管理系統。

1. OLT (Optical Line Termination)：光纖網路局端 OLT 放置於機房中心，提供網路服務供應商光纖信號介面與 GE (Gigabit Ethernet) 介面間以及裝置設備之轉換，並協調與用戶端 ONU 間之多工。

2. ONU (Optical Network Unit)：光纖網路用戶端 ONU 主要採用乙太網路協定，可接收由 OLT 所發送之廣播數據以及控制命令，並做出相對應之調整;由於 ONU 與用戶間採用乙太網路協定，故通訊過程中無須再進行協定之轉換,故可實現 ONU 與用戶間之透明傳送，進而達到 OLT 與 ONU 間高速資料轉發。
3. POS (Passive Optical Splitter)：被動式分光器 POS 其功能為連接局端 OLT 與用戶端 ONU 之無電源設備,透過 Splitter 可將光纖中一束光轉發出 2 至 64 條光束。當下行傳輸 (由 OLT 至 ONU) 經過 Splitter 時，光纖會透過 Splitter 將傳輸資料分發；當上行傳輸 (ONU 至 OLT) 經過 Splitter 時,Splitter 會將傳輸資料集中上傳。Splitter 更可依照網路需求選擇所需之設備，並進行多層級連接。

3.2 元件設計

本小節將介紹本論文所設計之系統元件,包含 GPON 端用來擷取封包之 Snooping Agent，於後端系統中分析 Control Channel 之 Analyzing Server、分析串流會談封包之 Media Processing Server、紀錄分析內容之資料庫、以及數位鑑識管理系統。

3.2.1 Snooping Agent

本論文所研究之系統架構將 Snooping Agent 元件設計於用戶端 ONU,透過會談分類將本研究所需之封包傳送至後端 Analyzing Server，本系統所設計之元件架構圖如圖 3.2.1。

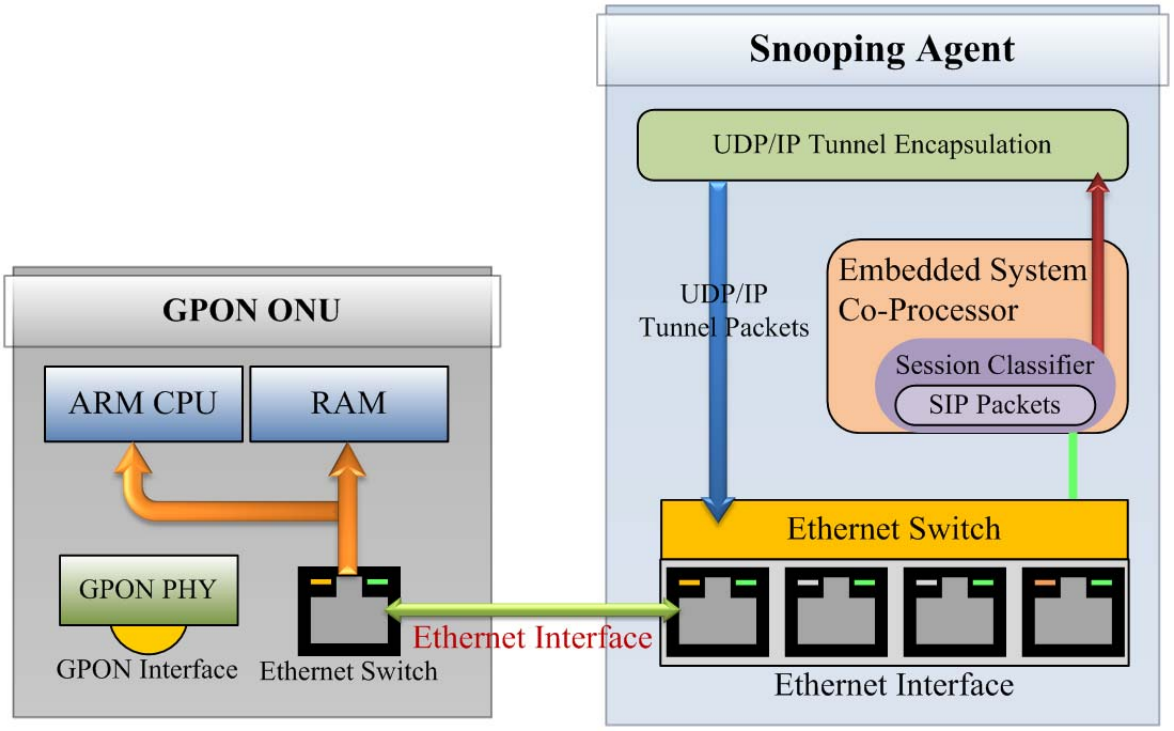

圖 3.2.1 Snooping Agent 元件架構圖

本系統於 GPON ONU 利用嵌入式系統 EP80579 實作 Snooping Agent 元件，本元件為用來對 SIP 網路電話進行監聽 (Snooping) 工作，在不影響正常封包傳送之情況下，將本研究所需之 SIP 串流會談封包擷取複製，透過 Snooping Agent 中 Encapsulation 封裝技術將封包偽裝為 UDP 傳送至後端 Analyzing Server 進行分析與儲存。

當 SIP 封包流過用戶端 ONU 時，會通過與 ONU 連接之 Snooping Agent，由於 Snooping Agent 開啟雜湊模式 (promiscuous)，故所流經之封包會進入 CPU 中進行監聽擷取。本論文所研究 SIP 串流會談之 Control Channel Port Number 為 Well-Known Port，而 Data Channel Port 為 Unknown Port，故須先進行 Control Channel 之分析後才能得到 Data Channel 資訊。

當封包流入 Snooping Agent 之 Ethernet interface 後，透過 Pcap Lib 所撰寫之程式利用會談分類之方式以 Session Classifier 先行對 SIP

Control Channel Port Number 監聽擷取。當 CPU 擷取封包後，Snooping Agent 將利用 Encapsulation 功能將原始封包透過 UDP Tunnel (將原始內容加上 IP Header 與 UDP Header 偽裝成 UDP 封包) 傳送，經由與 GPON ONU 所相連之 Ethernet Switch 由 GPON 對外之光介面送出並轉送至後端 Analyzing Server 進行分析，找出性質為 Unknown Port 之 Data Channel。

當後端 Analyzing Server 找出 Data Channel Port 後，便將 Port Number 以 Control Message 透過 UDP Tunnel 傳送至 Snooping Agent，告知 Snooping Agent 可依照所得到之 Port Number 擷取 SIP 之串流會談資訊，並將所擷取之封包同樣以 UDP Tunnel 封裝技術送至後端 Media Processing Server 還原儲存。

3.2.2 Analyzing Server

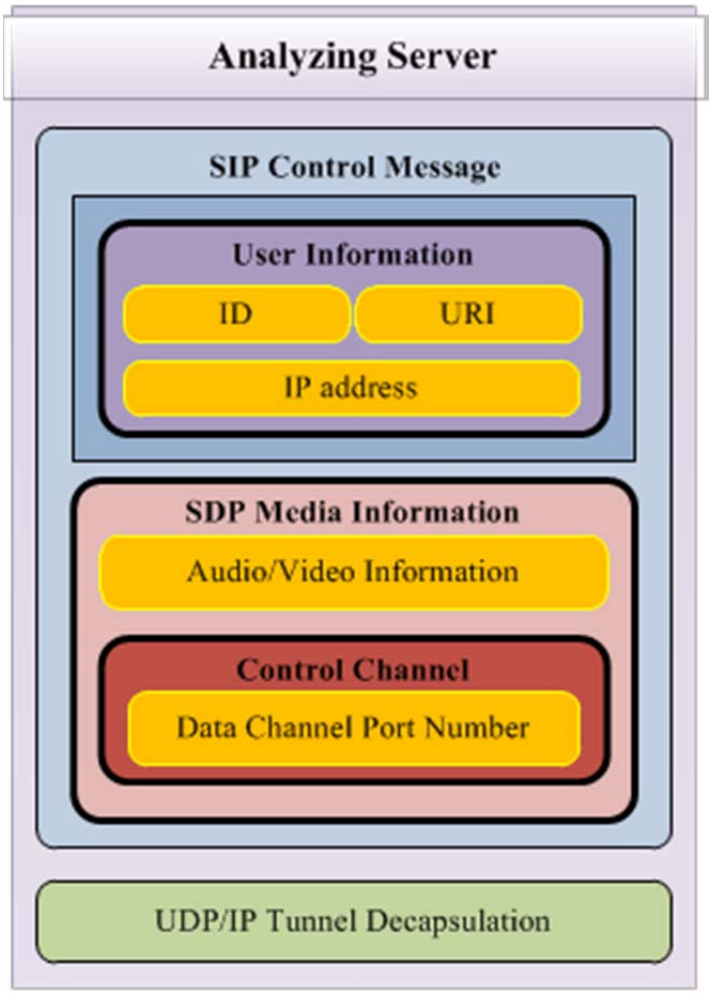

圖 3.2.2 Analyzing Server 功能方塊圖

本論文所設計之 Analyzing Server 為利用一般桌上型主機所架設之會談分析伺服器，其功能方塊圖如圖 3.2.2。Analyzing Server 為用於分析 Control Message，並找出串流會談之 Data Channel Port。

當 Analyzing Server 收到 SIP 封包時，首先會透過 Decapsulation 將其解封還原成原始之 SIP 封包並分析封包內容，透過分析 Session Description Protocol (SDP) 資訊可擷取出串流會談所使用之 Data Channel Port Number，並可觀察出使用者資訊。當分析完成後，Analyzing Server 會將所得到之 Port Number 回傳告知 Snooping Agent 以擷取會談串流之封包，並將所得到之使用者資訊與多媒體資訊儲存至資料庫中，

並透過數位鑑識管理系統以 Web Interface 方式呈現資訊。

3.2.3 Media Processing Server

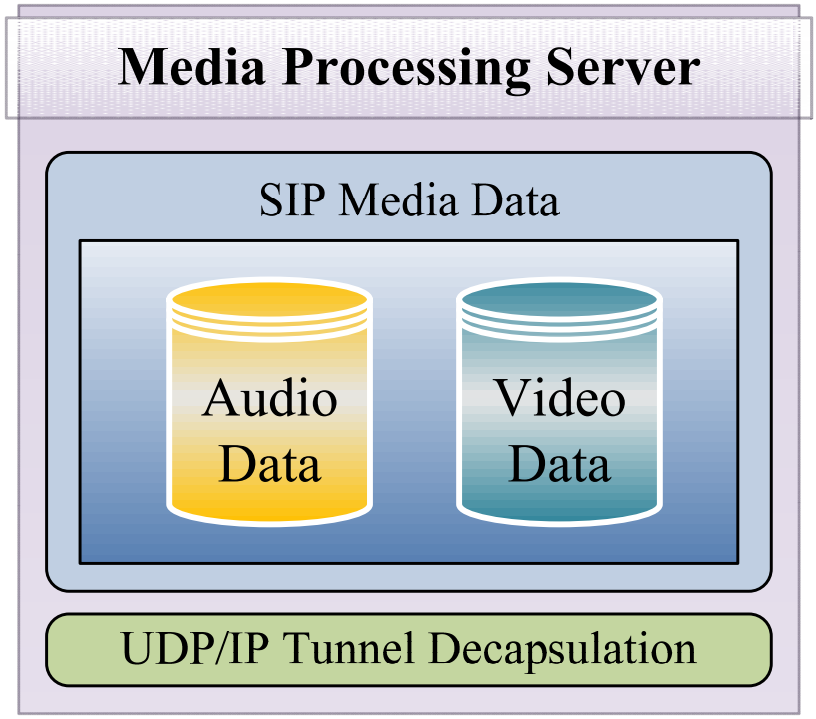

圖 3.2.3 Media Processing Server 功能方塊圖

本論文所設計之 Media Processing Server 為利用電腦主機所設計之多媒體串流分析伺服器，其功能方塊圖如圖 3.2.3。Media Processing Server 為用來分析由 Snooping Agent 擷取之會談串流封包，依照其內容可分為音訊資料 (Audio Data)、以及影像資料 (Video Data)，並可將分析後之內容依照其類別分別儲存至資料庫中，透過數位鑑識系統之 Web Interface 呈現資訊。

Media Processing Server 首先會透過 Decapsulation 功能將由 Snooping Agent 收到以 UDP Tunnel 封裝傳送之會談串流封包解封，還原至原始 SIP 封包，並將所得到之 SIP 封包依照其內容分別分析出音訊與影像之封包。影像與音訊之封包將利用 SIP Control Message 決定 Port Number 後透過 RTP 傳輸。

當封包分類完成後，便將得到之 RTP 封包串聯還原後可得到使用者影音會談內容之可撥放檔案，並將上述還原後之檔案儲存至資料庫中，利用數位鑑識管理系統呈現鑑識結果。

3.2.4 數位鑑識管理系統

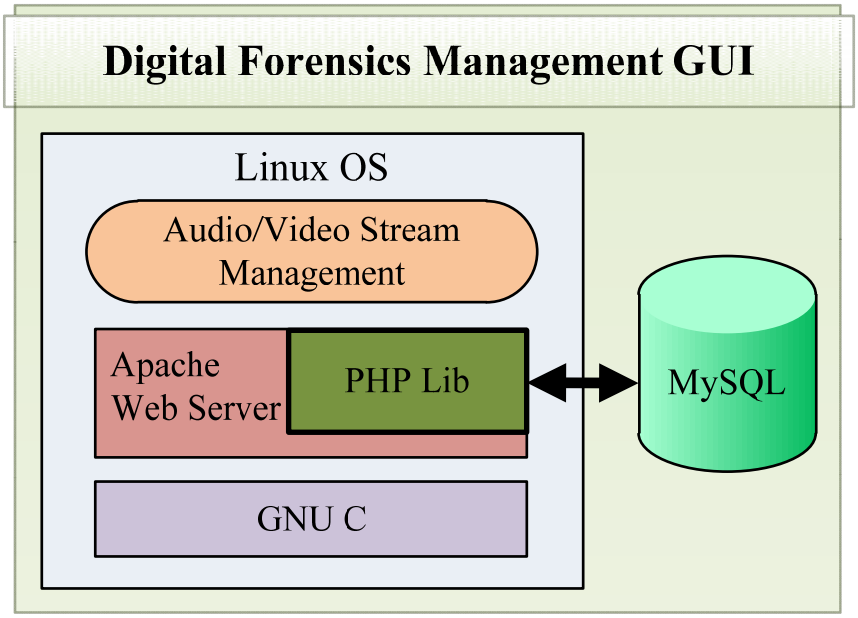

圖 3.2.4 數位鑑識管理系統

本論文所設計之數位鑑識管理系統之圖形使用者介面 (Digital Forensics Management GUI) 建構於 Linux 作業系統之伺服器上，乃透過以 PHP 網頁程式語言為基礎，提供鑑識人員與管理者查詢與維護串流會談之鑑識資料。本元件之功能方塊圖如圖 3.2.4。

數位鑑識管理系統將可查詢包含使用者之狀態資料與串流會談之影音資訊…等。本系統以 Linux (Ubuntu) 作業系統平台為基礎，透過 GNU C、Apache HTTPd/PHP 與 MySQL…等 Open Source 開發環境建置出數位鑑識管理系統之 Web Application。

1. SIP 通訊資料庫

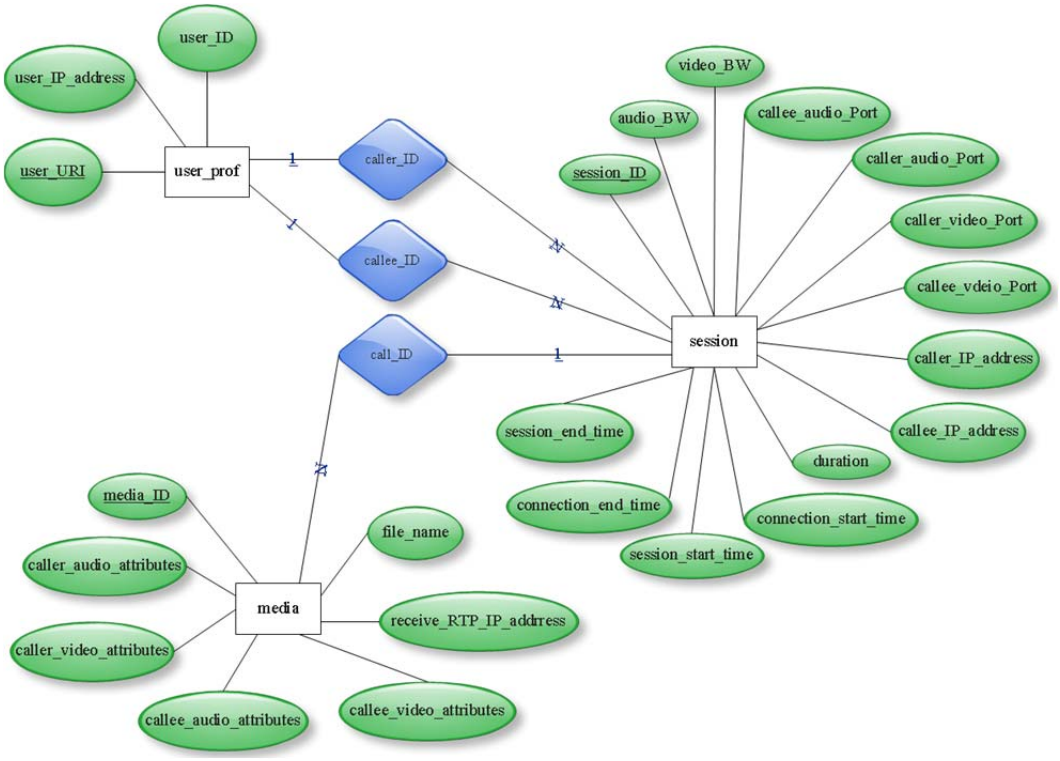

圖 3.2.5 SIP 通訊資料庫實體關聯圖

本系統之 SIP 通訊資料庫實體關聯圖如圖 3.2.5，並以此圖說明個資料表之間之關係。關聯中 1 與 N 為一對多之關係；user_prof 資料表與 session 資料表之關聯為，一個撥話者或一個受話者有多個通訊資料，故 user_prof 與 session 為一對多之關係；而 session 資料表與 media 資料表之間為一通會談會使用兩種多媒體格式 (視訊語音訊)，故 session 與 media 為一對多之關係。

2. SIP 通訊資料庫實體關聯圖

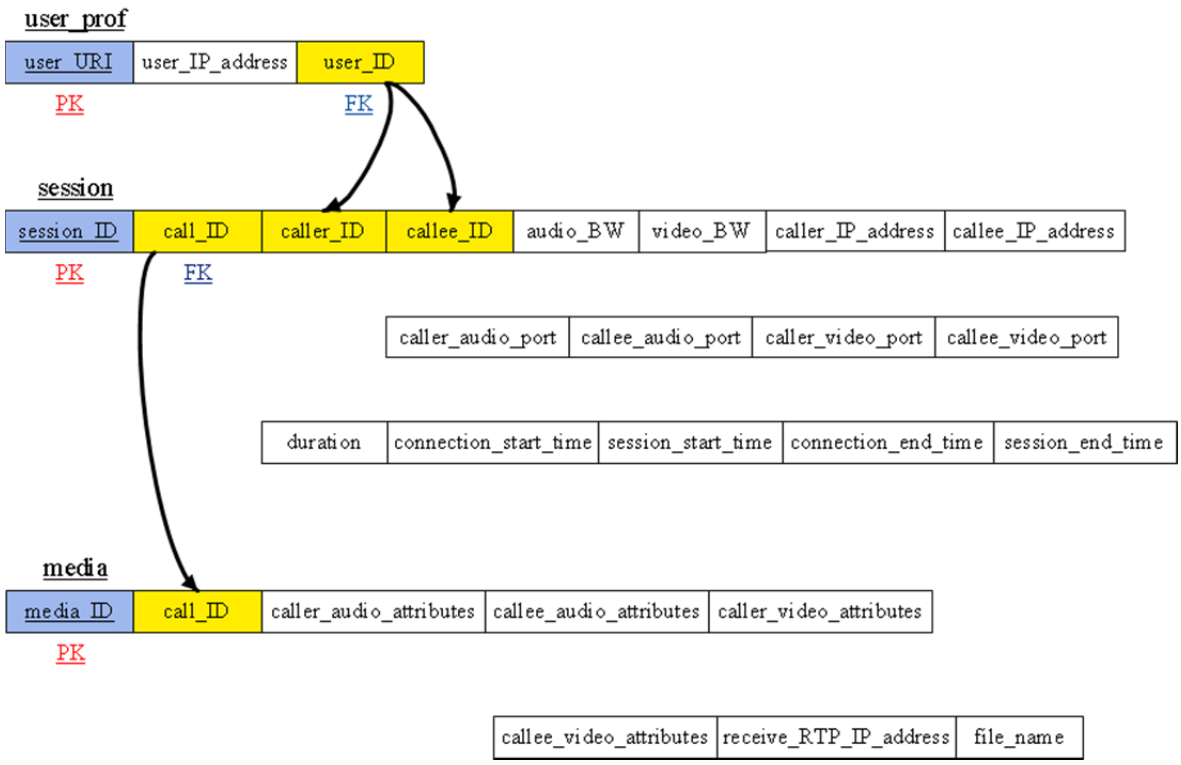

圖 3.2.6 SIP 通訊資料庫實體關聯圖

SIP 通訊資料庫實體關聯圖如圖 3.2.6 所示，圖中標示 PK (Primary Key) 之欄位為主索引鍵，標示為 FK (Foreign Key) 之欄位為外部索引鍵；本資料庫主要分為 3 張資料表，以下將詳細敘述。

2.1 user_prof 資料表中，user_URI 為使用者登入所記錄之名稱，故具有唯一性，本資料表以此為主索引鍵；資料表將記錄使用者之 IP 位址 (user_IP_address)、使用者所使用之 ID 名稱 (user_ID)，並以 user_ID 欄位為外部索引鍵，與 session 資料表中之 caller_ID 與 callee_ID 做為關聯性。

2.2 session 資料表中，session_ID 為流水編號之主索引鍵，具唯一性；資料表中記錄一通會談中所包含的資訊，分別為會談通話之編號 (call_ID)、音訊頻寬 (audio_BW)、視訊頻寬

(video_BW)、通話時間 (duration)、連接開始與結束之時間 (connection_start_time, connection_end_time)、會談開始與結束之時間 (session_start_time, session_end_time)；而 user_prof 與 session 資料表之關聯為一對多，即表示一個使用者可能以撥話者或受話者之身分擁有多通會談資訊，故 user_prof 資料表中之 user_ID 欄位將對應至 session 資料表之 caller_ID 與 callee_ID 為關聯性。

2.3 media 資料表中，media_ID 為流水編號之主索引鍵，具唯一性；資料表中記錄會談所使用之多媒體資訊，其中包含：撥話者所使用之音訊與視訊多媒體屬性 (caller_audio_attributes, caller_video_attributes)、受話者所使用之音訊與視訊多媒體屬性 (callee_audio_attributes, callee_video_attributes)、接收 RTP 之 IP address 與 Port Number (receive_RTP_IP, receive_RTP_Port)、以及所儲存之檔案名稱 (file_name)；session 資料表與 media 資料表之關聯為一對一，一通會談會帶有一組多媒體資訊，故 session 資料表中之 call_ID 為外部索引鍵，對應至 media 資料表中之 call_ID。

第四章、系統實作

本章節將探討 SIP 影音會談情境之觀察，以及數位鑑識整合系統之實作。本研究初期先觀察 SIP 網路電話之封包傳輸過程，透過 Wireshark 監聽封包傳輸流程，並將本研究之程式與系統建立於 VMware 虛擬機上測試其運作狀況。

4.1 SIP 影音會談情境說明

本研究所規劃之虛擬機環境主要分為 Snooping Agent (以 Ubuntu 12.04 建置)、Analyzing Server (以 Ubuntu 12.04 建置)、Media Processing Server (以 Ubuntu 12.04 建置)，SIP 網路電話所使用之軟體為 Linphone 3.5.0 (使用於 Microsoft Windows 7 作業環境)。

Time	Source	Destination	Protocol	Length	Info
2.950313	192.168.2.1	192.168.2.110	SIP	372	Request: OPTIONS sip:sip2@192.168.2.110
2.954606	192.168.2.110	192.168.2.1	SIP	442	Status: 200 OK
2.983419	192.168.2.1	192.168.2.110	SIP/SDP	811	Request: INVITE sip:sip2@192.168.2.110, with session description
2.986611	192.168.2.110	192.168.2.1	SIP	334	Status: 100 Trying
2.986612	192.168.2.110	192.168.2.1	SIP	399	Status: 101 Dialog Establishement
3.014066	192.168.2.110	192.168.2.1	SIP	363	Request: OPTIONS sip:192.168.2.1:5060
3.042685	192.168.2.1	192.168.2.110	SIP	414	Status: 200 OK
3.093507	192.168.2.110	192.168.2.1	SIP	385	Status: 180 Ringing
3.816703	192.168.2.110	192.168.2.1	SIP/SDP	704	Status: 200 OK, with session description
3.832041	192.168.2.1	192.168.2.110	SIP	390	Request: ACK sip:sip2@192.168.2.110
5.214742	192.168.2.110	192.168.2.1	SIP	386	Request: BYE sip:sip1@192.168.2.1
5.215054	192.168.2.1	192.168.2.110	SIP	315	Status: 200 OK

圖 4.1.1 SIP 網路電話封包傳輸流程

本研究初期以 Wireshark 觀察 SIP 網路電話封包傳輸流程，如圖 4.1.1 所示。由所觀察到的封包來看，SIP 網路電話傳輸時，主要透過 INVITE 訊息發出邀請，以帶有 SDP 之 200 OK 訊息確定會談所使用之音訊與視訊屬性，最後透過 BYE 訊息結束通話。故本研究將針對 INVITE 訊息、SIP 200 OK 訊息與 BYE 訊息三個封包進行分析研究其中所需之資訊。以下將詳細敘述本論文針對各訊息所需之資訊敘述。

4.1.1 INVITE 訊息：

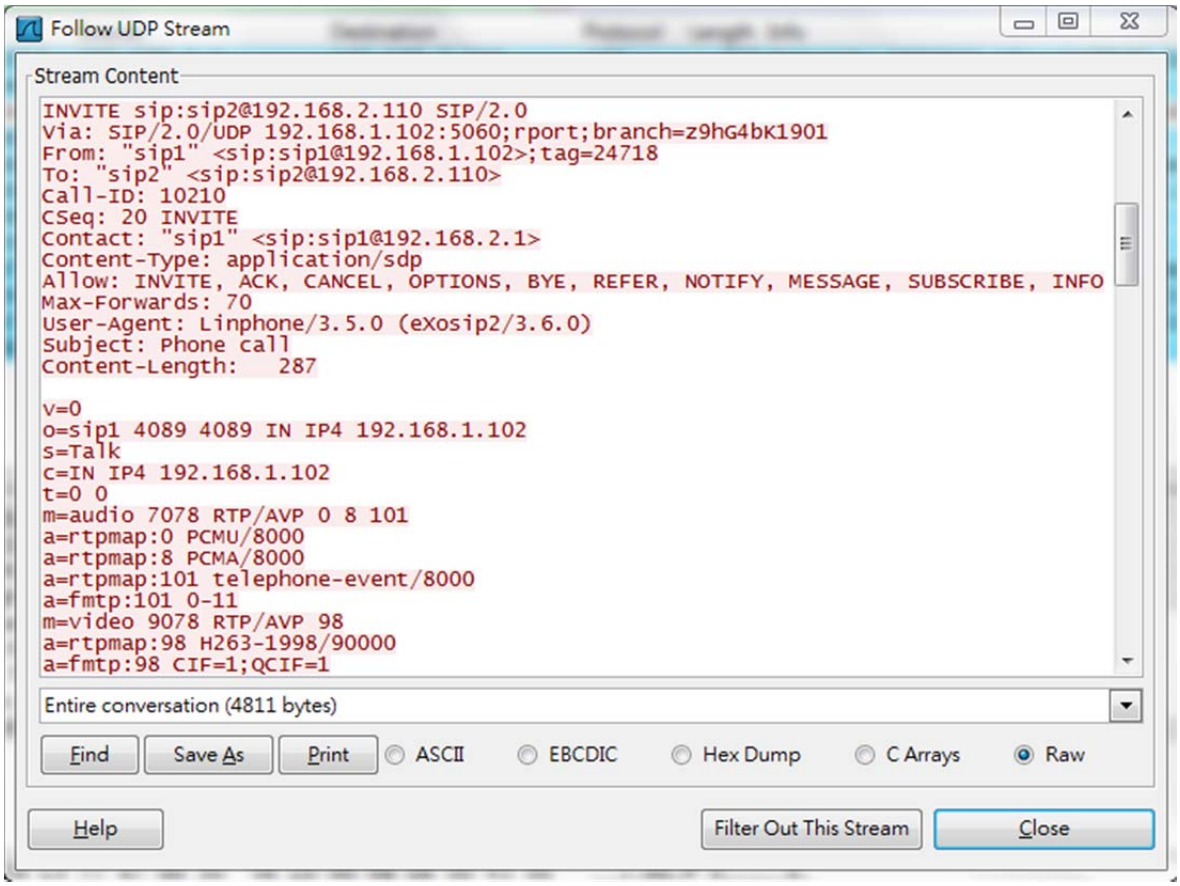

圖 4.1.2 INVITE 訊息內容

INVITE 訊息之內容如圖 4.1.2，在 INVITE 訊息中，主要訊息包含撥話者與受話者之 ID、IP 與 URI、通話 ID…等；並將所擷取到之資訊存入資料庫中。

4.1.2 SIP 200 OK 訊息：

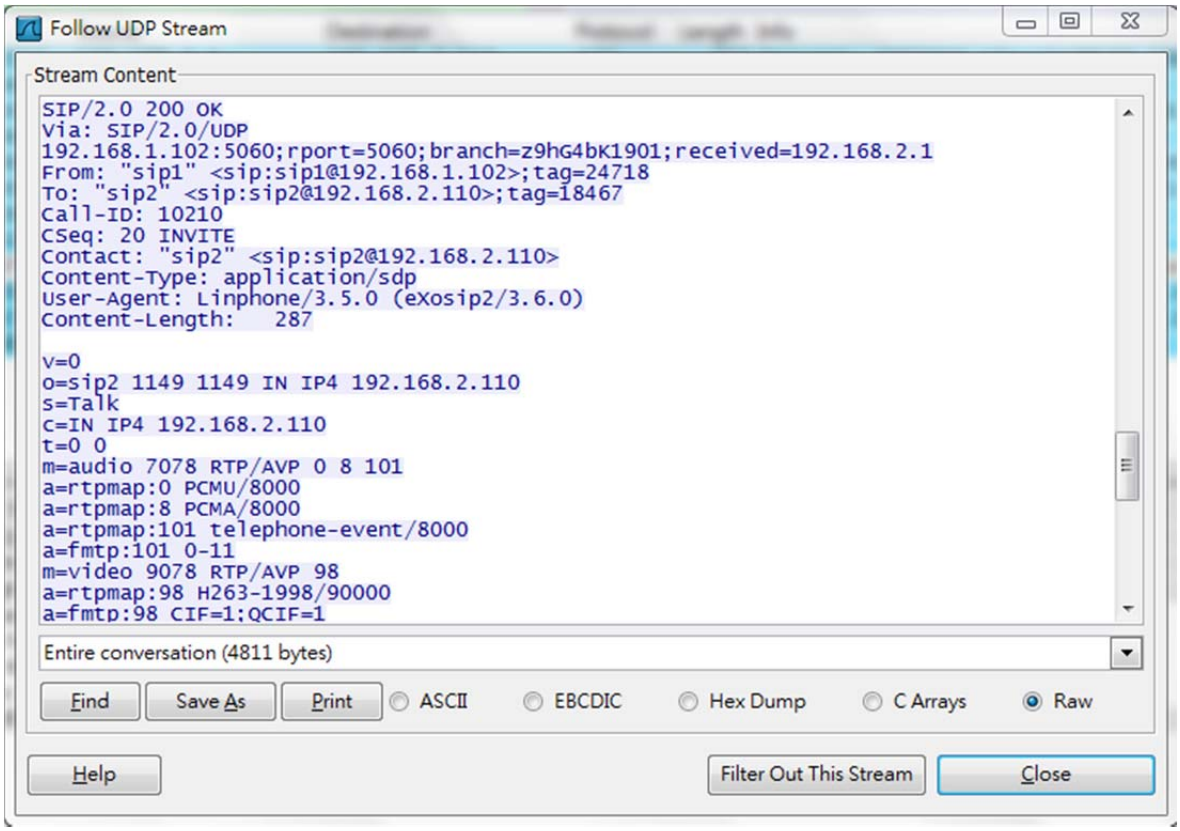

圖 4.1.3 SIP 200 OK 訊息內容

SIP 200 OK 訊息之內容如圖 4.1.3，SIP 200 OK 訊息中，主要擷取之資訊為接收 RTP 之 IP、所使用之音訊與視訊埠號、以及所使用之影音編碼類別…等；並將所擷取到之資訊存入資料庫中。

4.1.3 BYE 訊息

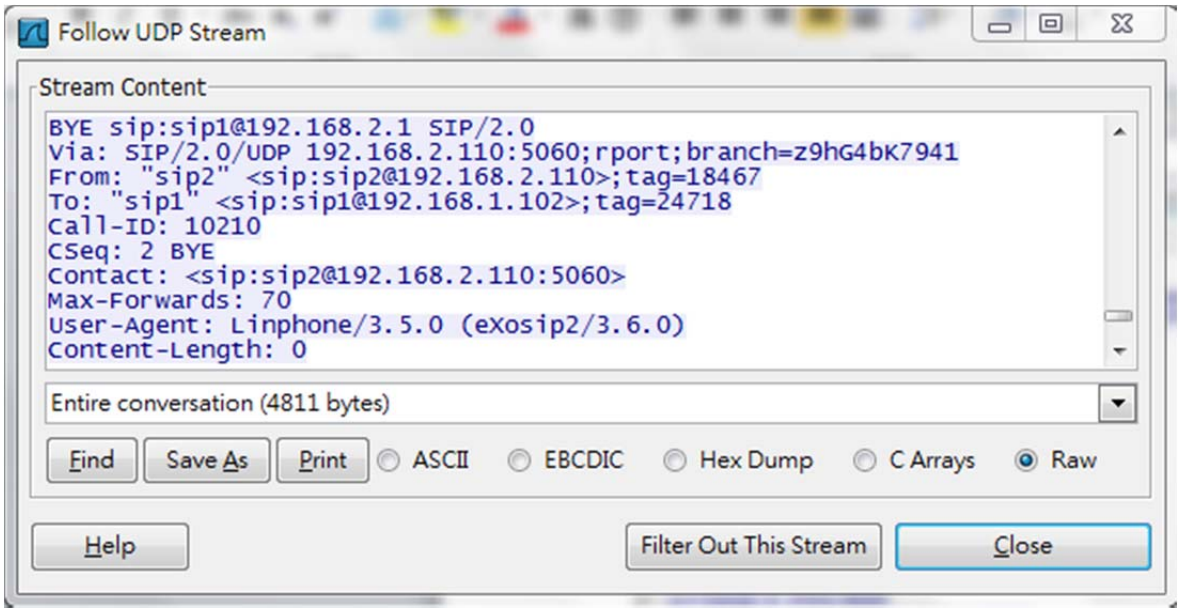

圖 4.1.4 BYE 訊息內容

BYE 訊息之內容如圖 4.1.4，BYE 訊息中，主要擷取會談之 call-ID 訊息，以提供 Media Processing Server 會談結束訊息。

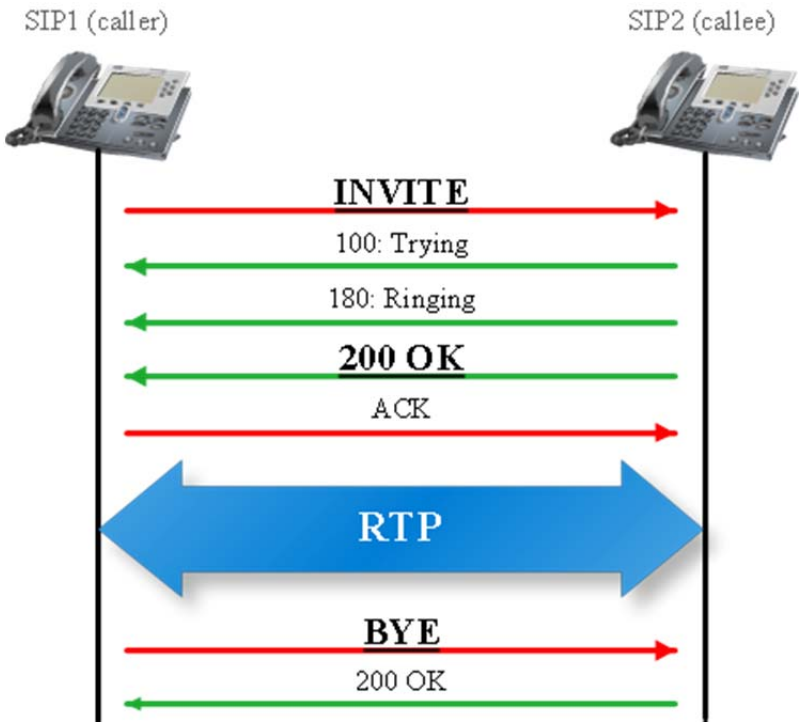

圖 4.1.5 SIP 會談訊息傳輸圖

完整之 SIP 會談資訊傳輸之流程如圖 4.1.5，透過針對以上三個訊息擷取並分析出資訊後，可提供本研究之系統運作所需相關資訊。

下一小節將詳述本系統於 VMware 虛擬機上之實作。

4.2 多媒體串流數位鑑識系統—於 VMware 實作

本系統之測試環境將先建構於 VMware 虛擬機上模擬本系統所設計之元件，並於完成後移植至嵌入式系統與主機上進行測試。本系統自行開發之元件為：Snooping Agent、Analyzing Server 與 Media Processing Server 三個元件。

- Snooping Agent 元件將用於監聽擷取於網路流量中之 SIP 資料流與會談所傳輸之 RTP 封包；本元件以 Ubuntu 12.04 建置。
- Analyzing Server 元件將用於分析所擷取之 SIP 封包資訊並傳送 Control message 至 Snooping Agent 與 Media Processing Server；本元件以 Ubuntu 12.04 建置。
- Media Processing Server 元件用於接收所監聽擷取之 RTP 封包，並將 RTP 封包之 Payload 串接還原成可撥放之音訊與視訊檔案；本元件以 Ubuntu 12.04 建置。
- 本系統之 SIP 網路電話將應用 Linphone 3.5.0 網路電話軟體，本軟體將安裝於 Windows 7 電腦中。

以 Snooping Agent 元件而言，本系統所遭遇之瓶頸，乃於如何在高速網路環境下監聽 SIP 會談封包，並傳送至後端 Analyzing Server 分析出 SIP 網路電話中所使用之動態 Data Channel Port Number，並接收由 Analyzing Server 所回傳之 Control message，使本元件可接收會談所傳輸之 RTP 封包。因此，本系統所設計之 Snooping Agent 元件利用 tcpdump 所提供之 Libpcap Simple Sniffer[4] 來分析並擷取封包[5]，該程式以

Libpcap 為基礎，擷取流經所監聽網路卡資料流所包含之 SIP 網路電話封包，並將擷取之封包透過 UDP/IP Tunnel Encapsulation 方式傳送 UDP socket 至 Analyzing Server 分析；當系統欲以 Libpcap 監聽擷取封包時，由於 SIP phone 傳輸是使用 port 5060 傳輸，故程式開啟所監聽之網路卡介面時會將程式中 pcap_compile 之 filter 設定為監聽 port 5060，當程式擷取到封包後，將進入封包擷取之迴圈，並以 UDP/IP Tunnel Encapsulation 方式將封包傳送至 Analyzing Server 進行會談資訊之分析；其運作方式如圖 4.2.1，相關程式碼如下。

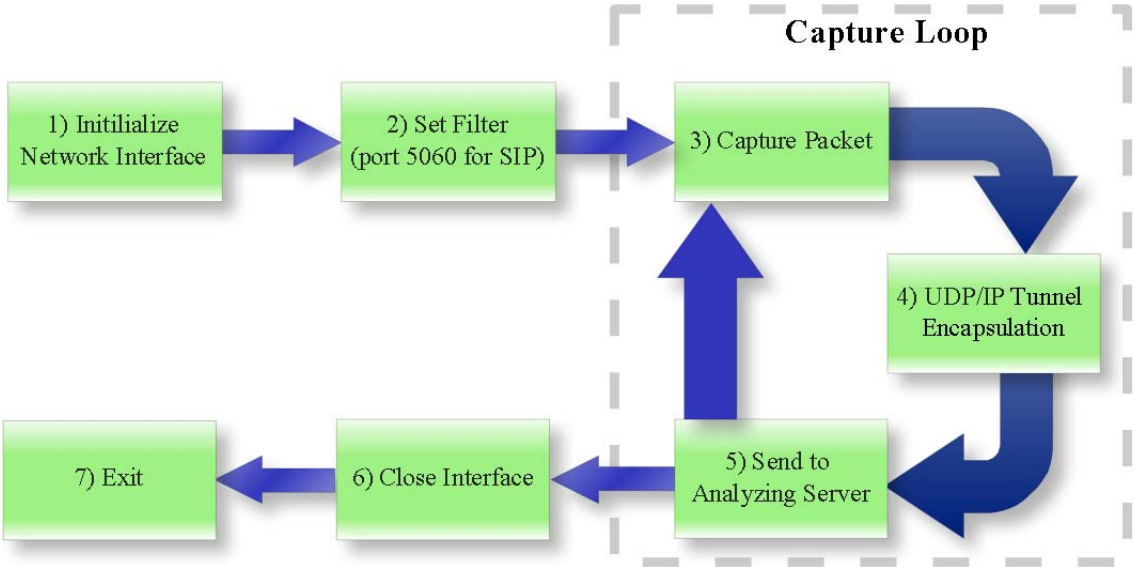

圖 4.2.1 Snooping Agent 元件之 SIP 封包處理流程

```
/* Simple Raw Sniffer
*/
/* To compile: gcc simplesniffer.c -o simplesniffer -lpcap
*/
/* Run as root!
*/
#include <stdio.h>
#include <pcap.h>
```

```
    #include <sys/socket.h>
.
    .
    .
    #define MTU_SIZE 1514
    #define INTERFACE "eth0"
    #define DEST_IP "192.168.2.101"
    #define DEST_PORT 5001

    typedef unsigned short int uint16;
    typedef unsigned char uint8;
typedef unsigned int uint32;
    .
    .
    .
        //設定目的地之 IP address 與 Port Number
    int end_info;
        struct sockaddr_in ma;
        int end_len = sizeof(ma);
        ma.sin_family = AF_INET;
        ma.sin_port = htons(DEST_PORT);
        ma.sin_addr.s_addr = inet_addr(DEST_IP);
        end_info = socket(AF_INET, SOCK_DGRAM, 0);
        connect(end_info, (struct sockaddr *)&ma, end_len);

    int aa;
        aa = send(end_info, packet, hdr.len, 0);
```

```
int aa;
aa = send(end_info, packet, hdr.len, 0);
.
.
.
int main(void){
char errbuf[PCAP_ERRBUF_SIZE];
        pcap_t *phandle;
        char *dev;
        struct bpf_program fp;
        char filter_exp[50] = "port 5060";    //Set filter condition
bpf_u_int32 mask;    //netmask
        bpf_u_int32 net; //ip address
        struct pcap_pkthdr header ;
        const u_char *packet;

        printf("%d\r\n", sizeof(header));
        dev = pcap_lookupdev(errbuf);
        if( dev == NULL){
                printf("Can't get dev %s\r\n", errbuf);
                return -1;
        }
        else printf("Our dev = %s\r\n", dev);

        if( pcap_lookupnet(dev, &net, &mask, errbuf) == -1){
                printf("Can't get mask %s\r\n", errbuf);
                return -1;
```

```
        }
        phandle = pcap_open_live( dev, 1600, 1, -1, errbuf );
        if( phandle == NULL){
            printf("open device failed : %s\r\n", errbuf);
            return -1;
        }

        if( pcap_compile(phandle, &fp, filter_exp, 1, mask) == -1) {
        printf("Paser filter failed %s:%s\r\n", filter_exp,
                pcap_geterr((pcap_t*)errbuf));
        }

        if( pcap_setfilter(phandle, &fp) == -1){
            printf("Install filter failed : %s\r\n", filter_exp);
    }

        pcap_loop(phandle, 0, FilterSIP, NULL);

        pcap_close(phandle);
        return 0;
}//End of Main
```

以 Analyzing Server 元件而言，本系統設計時將遭遇如何分析本系統所需之資訊，以及在高速網路環境下如何不影響 Session 分析結果之準確性，並將其中所包含之 Control Message 傳送至 Snooping Agent 與 Media Processing Server 作為系統擷取封包之設定。因此，本系統設計 Analyzing Server 時，由於使用 pcap 方式擷取封包必需將監聽之網路卡

介面設置為雜湊模式 (promiscuous)，所有網路流量之資料流仍會流經 CUP 中，難免降低鑑識之品質。故為提高系統效能，本元件將以 UDP socket server 方式 (所謂 UDP socket server 即針對一固定 UDP port 接收) 接收由 Snooping Agent 所傳送之 UDP socket；當本元件收到封包時，首先會透過 UDP/IP Tunnel Decapsulation 將封包解封，並透過字串分析之方式，分析出 SIP 與 SDP 資訊，並將 SDP 中所包含之 Control Message 以 UDP/IP Tunnel Encapsulation 方式傳送至 Snooping Agent 與 Media Processing Server，並將分析得到之資訊儲存至資料庫中；當 Analyzing Server 欲接收由 Snooping Agent 傳送之 SIP 封包時，會先開啟 UDP socket Server 接收，當程式接收到封包後會先經過 UDP/IP Tunnel Decapsulation 將封包解封，得到原本之 SIP 封包後會將其中所包含之 SIP 使用者資訊 (如：userID, userIP, userURI…等) 以及 SDP 所包含之資訊分析出來 (如：會談所使用之會談多媒體資訊、會談多媒體所使用之埠號、會談經過時間…等)，並將相關所需資訊儲存至資料庫中，當 control message 分析出來後，會再透過 UDP/IP Tunnel Encapsulation 方式將封包傳送至 Snooping Agent 與 Media Processing Server，以提供該二元件相關擷取資訊；相關程式碼如下，其封包處理流程如圖 4.2.3。

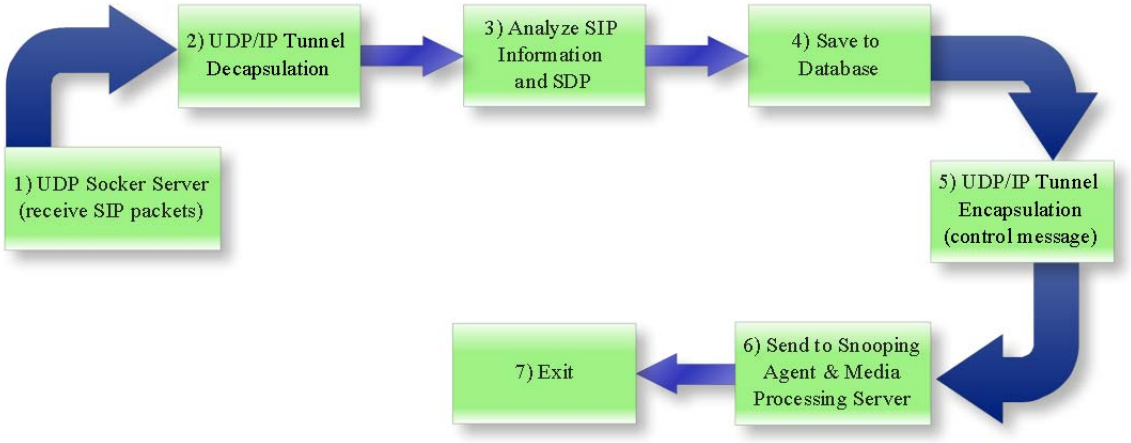

圖 4.2.3 Analyzing Server 元件之封包處理流程

```
#include<stdio.h>
    #include<string.h>
    #include<stdlib.h>
#include<sys/socket.h>
.
.
.
    #define UNKNOW 0
    #define SIP_INVITE 1
#define SIP_OK 2
#define SIP_BYE 3
    #define SIP_UNKNOW 9
#define MAXBYTES2CAPTURE 2048
.
.
.
        //透過structure 儲存所分析之資訊
        struct data
```

```
    {
            char aa[50];
            char stok[7];
        char state1[6];
            char state[6];
    .
    .
    .
            int a1;
        int a2;
            int a3;
            int b1;
            int b2;
            int b3;
    } se;
.
.
.
        // 以 UDP socket server 方式接收 SIP 封包
     si_me.sin_family = AF_INET;
        si_me.sin_port = htons(PORT);
        si_me.sin_addr.s_addr = htonl(INADDR_ANY);

        //bind socket to port
        if( bind(s , (struct sockaddr*)&si_me, sizeof(si_me) ) == -1)
        {
            die("bind");
```

```
    }
.
.
.
            // 針對 SIP 封包中所包含之三個資訊分析
                int msg_type = UNKNOW;

            if( strstr(state,"INVITE"))
                {
                    msg_type = SIP_INVITE;
                }
                else if( strstr(stok,"200 OK"))
                {
                    msg_type = SIP_OK;
            }
                else if( strstr(state,"BYE"))
                {
                    msg_type = SIP_BYE;
                }
                else
                {
                msg_type = SIP_UNKNOW;
            }

                switch(msg_type)
            {
                // 針對 INVITE 資訊分析
```

```
                    case SIP_INVITE:
                        {
                            char callerIP[20]; //20130425_20->30
                            s1=strstr(sipheader,"Contact: ");
                            printf("\n%s\n",s1);
                            s1=strstr(s1,"@");
                            s2=strstr(s1,">");
                        printf("\n%s\n",s2);
.
.
.
                        }
                break;
                    // 針對 SIP 200 OK 資訊分析
                    case SIP_OK:
                    {
                            char callerURI[50];
                            s1=strstr(sipheader,"From: ");
                        s1=strstr(s1,"<");
                            s2=strstr(s1,">");
                            memset(callerURI, 0x00, sizeof(callerURI));
                            strncpy(callerURI,s1+1,s2-(s1+1));
                            memset(se.callerURI, 0x00,
    sizeof(se.callerURI));
                        strcpy(se.callerURI,callerURI);
                        printf("\n%s\n",se.callerURI);
.
```

```
.
.
                    //以 UDP socket 傳送 Control msg 至 Snooping Agent
                        int aa;
                        aa = send(end_info, se.control_msg, 40, 0);
.
.
.
//連接資料庫儲存資料
    mysql_init(&my_connection);
                            //session table
                            if (mysql_real_connect(&my_connection,
    "localhost", "sip", "sip","sip",0,NULL,0))
                        {
                            printf("Connection success\n");
.
.
.
                    }
                    break;
                // 針對 SIP BYE 資訊分析
                case SIP_BYE:
                    {
                            char callID2[20];

                            s1=strstr((char*)sipheader,"Call-ID: ");
                        s1=strstr(s1," ");
```

```
                    s2=strstr(s1,"\n");


.
.
.
                    }
                break;

                case SIP_UNKNOW:
                    printf("\n0\n");
                default:
                    printf("\n00\n");
            }
.
.
.
        close(s);
        return 0;
}
```

當 Snooping Agent 欲接收從 Analyzing Server 回傳之 Control Message 時，本元件將以 UDP socket server 方式接收，透過 UDP/IP Tunnel Decapsulation 方式將 Control Message 解封出來，找出該會談所使用視訊與音訊之 Data Channel Port Number，並將埠號設定至 filter 中；當擷取到 RTP 封包後，將其透過 UDP/IP Tunnel Encapsulation 把 RTP 封包傳送至 Media Processing Server；其程式碼如下。運作方式如圖 4.2.2。

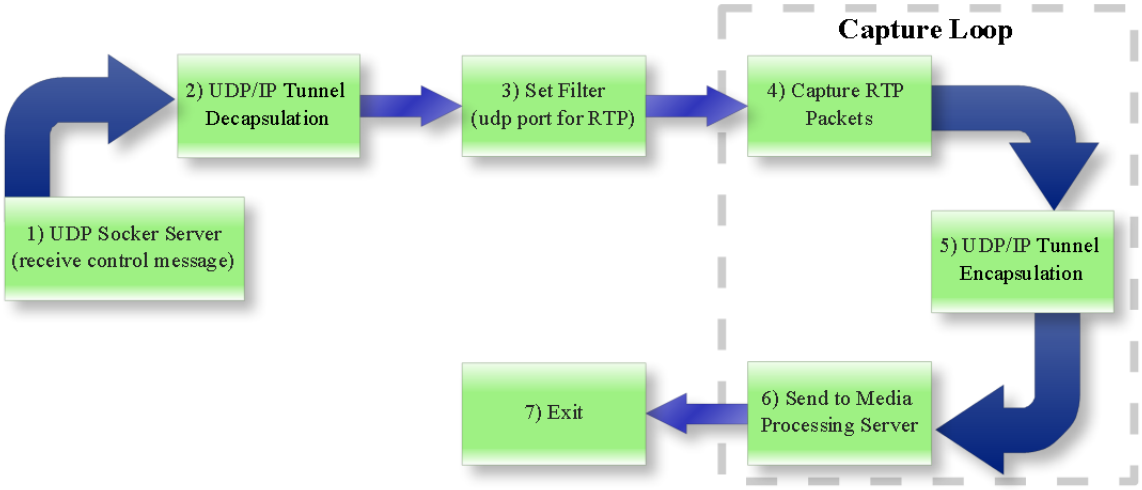

圖 4.2.2 接收 Control Message 後抓取 RTP 封包處理流程

```
/* Simple udp server by Silver Moon    */
#include <stdio.h> //printf
#include <string.h> //memset
#include <stdlib.h> //exit(0);
#include <sys/socket.h>
.
.
.
#define BUFLEN 100    //Max length of buffer
#define PORT 5003      //The port on which to listen for incoming data

#define MTU_SIZE 1514
#define INTERFACE "eth0"
#define DEST_IP "192.168.2.103"
#define DEST_PORT 5005
.
.
.

//利用struct  儲存control message
struct{
      char callerIP[20];
      char calleeIP[20];
      char audio_port[10];
      char vedio_port[10];
      char filter_audio[50];
} se;
```

```
.
.
.
int main(void)
{
.
.
.
//分出control message  並儲存至struct  中
char *msg = strtok(buf, " ");
            printf("\nSIP Control Message :");
            sprintf(se.callerIP, "%s", msg);
            msg = strtok(NULL, " ");
            sprintf(se.calleeIP, "%s", msg);
            msg = strtok(NULL, " ");
            sprintf(se.audio_port, "%s", msg);
            msg = strtok(NULL, " ");
            sprintf(se.vedio_port, "%s", msg);

            printf("\nCallerIP : %s", se.callerIP);
            printf("\nCalleeIP : %s", se.calleeIP);
            printf("\nAudio Port : %s", se.audio_port);
            printf("\nVedio Port : %s\n", se.vedio_port);
      puts("--------------------------------------------------------------------\n");
.
.
.
```

```
        si_me.sin_family = AF_INET;
        si_me.sin_port = htons(PORT);
        si_me.sin_addr.s_addr = htonl(INADDR_ANY);
        //bind socket to port
        if( bind(s , (struct sockaddr*)&si_me, sizeof(si_me) ) == -1)
        {
              die("bind");
        }
.
.
.
            return 0;
            //End of Main
        }
        close(s);
        return 0;
}
```

而 Media Processing Server 元件所遭遇到之困難點為如何將所擷取到之 RTP 封包 Payload 合成為可撥放之檔案，並將其上行與下行之檔案分開儲存並考慮是否混音。當 Media Processing Server 接收到由 Snooping Agent 所傳輸之 UDP socket 封包時，需先透過 UDP/IP Tunnel Decapsulation 將封包解封成 RTP 封包，並分析 RTP 將一通會談中所包含之上下行音訊與視訊之 RTP payload 串成檔案並儲存成一般可撥放之檔案；當 Media Processing Server；相關程式碼如下，其封包處理流程如圖 4.2.4。

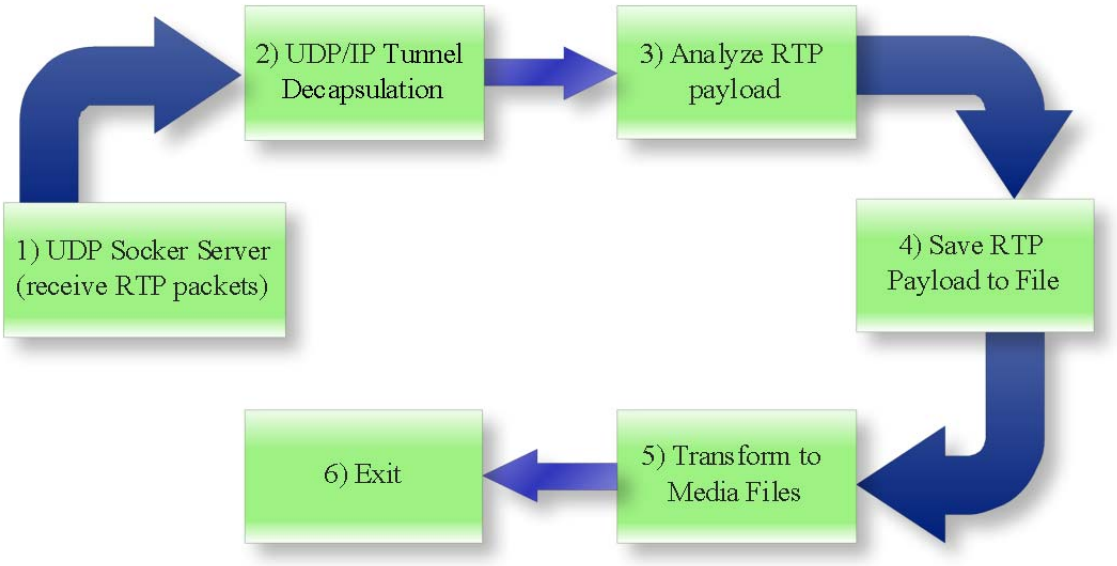

圖 4.2.4 Media Processing Server 元件之封包處理流程

```
#include<stdio.h> //printf
#include<string.h> //memset
#include<stdlib.h> //exit(0);
.
.
.
//印出接收之 RTP 內容
void dumpPacket(unsigned char *buf, int length){
    int i;
    printf("Receive Length = %d\n", length);

    for ( i = 0 ; i < length; i++ )
    {
        printf("%02x ",buf[i]);
        if ( (i+1)%16 == 0 )
            printf("\r\n");
```

```
        else if ( (i+1)%8 == 0 )
                printf(" ");
        }
        printf("\n");

        //寫出成檔案
        FILE *fileOut;

        fileOut = fopen("input.bin", "ba");
        for (i=54;i<length;i++)
        {
             fprintf(fileOut, "%c", buf[i]);
        }
    fclose(fileOut);
    } //End of dumpPacket

    int main(void)
    {
         struct sockaddr_in si_me, si_other;
        char *audio_port = NULL ;
.
.
.
     // 接收 RTP 封包
     si_me.sin_family = AF_INET;
          si_me.sin_port = htons(PORT);
          si_me.sin_addr.s_addr = htonl(INADDR_ANY);
```

```
        //bind socket to port
        if( bind(s , (struct sockaddr*)&si_me, sizeof(si_me) ) == -1)
        {
            die("bind");
    }
.
.
.
        close(s);
        return 0;
}
```

第五章、系統測試與效能分析

本系統之測試項目主要分為兩大項，一為功能性驗證測試，本系統之功能性驗證主要於 VMware 環境下測試，透過一 Web Interface 了解本系統功能之完整性；二為透過 SIPp 網路電話壓力測試軟體測試本系統之會談抗壓能力，本階段之系統測試為建構在實體系統上，測試環境預計將使用兩台桌上型電腦做為 Analyzing Server 與 Media Processing Server 之實作平台，並以 EP80579 之 x86 CPU 嵌入式系統為 Snooping Agent 之實作平台。以下將敘述本系統之測試流程與結果。

5.1 功能性驗證

本系統之功能性驗證主要分為三個元件之各功能驗證，以下將詳細說明。

1. Snooping Agent 元件之驗證內容為是否可監聽擷取 SIP 會談之封包，並傳送至後端 Analyzing Sever 分析，以及接收由 Analyzing Server 傳送之 Control Message 並設定該會談所使用之 RTP Port Number 至過濾條件中。

   結果：本研究所設計之 Snooping Agent 可監聽擷取 SIP 會談封包，並將其傳送至後端 Analyzing Server，並於分析完成後接收由 Analyzing Server 所傳送之 Control Message，並將其設為該會談 RTP 所使用之埠號已進行監聽擷取。

2. Analyzing Server 元件之驗證內容為是否可以接收由 Snooping Agent 所傳送之封包，並分析封包內容提取本系統所需之 SIP user Information 與 SDP Information 資訊儲存至資料庫中，並將

Control Message 發送至 Snooping Agent 與 Media Processing Server。

結果：本研究所設計之 Analyzing Server 元件可透過 UDP socket server 接收由 Snooping Agent 所傳送之 SIP 封包，並分析出本系統所需之會談資訊儲存至資料庫中，並將其中所包含之 Control Message 傳送至 Snooping Agent 與 Media Processing Server。

3. Media Processing Server 元件之驗證內容為是否可接收由 Snooping Agent 所傳送之 RTP 封包，並將其串連後還原為原本之可撥放影音檔案。

   結果：本研究所設計之 Media Processing Server 元件可利用 UDP socket server 接收由 Snooping Agent 傳輸之 RTP 封包，並將其一會談中封包中所包含之 Payload 串聯還原成會談原始通話檔案並可撥放。

5.2 抗壓性測試

本研究之抗壓性測試系統環境乃透過系統實作後，利用 SIPp 網路電話壓力測試軟體進行測試，本系統之實作環境為：於網路環境下，Snooping Agent 利用嵌入式系統 EP80579 實作測試，由於 EP80579 為 x86 CPU 之嵌入式系統，故本研究所使用之作業系統為 CentOS 5.0；Analyzing Server 利用桌上型電腦實作測試，作業系統為 Ubuntu 12.04；Media Processing Server 利用桌上型電腦實作進行測試，作業系統為 Ubuntu 12.04。

第六章、結論與未來展望

本論文設計與實作以 SIP 網路電話為應用之多媒體串流數位鑑識系統，透過本系統之設計，可克服於 GPON 環境上因應龐大流量與 SIP 會談時動態指定之影音埠號之瓶頸，以提供會談鑑識所需之相關資訊。

本論文發展於 GPON 架構上，傳統之鑑識方式大多以軟體於骨幹傳輸時監聽，並以局端 OLT 作為監聽點，容易因單一節點之高速傳輸流量過大而難以負荷，造成封包遺漏不完整；故本論文提出以兩層分散監聽負載之作法，上層先於用戶端 ONU 建構 Snooping Agent 擷取 SIP 封包並傳輸至下層 Analyzing Server 分析出 SDP 資訊，以克服於單一節點流量過大之狀況；並利用 SDP 中所包含之 Control Message，將其訊息傳至用戶端擷取 SIP session 所傳輸之 RTP 封包，並將封包傳送至下層之 Media Processing Server 儲存並還原為原本之影音資訊，以克服會談於開始後才指定影音所使用之埠號問題。因此，以此兩層式架構便可分散本來位於 OLT 單一節點所需監聽之負載流量，於 ONU 端分散掉負載以提升監聽環境之效能，並克服 SIP session 於會談開始前才指定影音 Port Number 之問題。

透過本論文之實作，其鑑識結果可應用於當使用者使用網路電話之影音串流傳輸服務時遭受到攻擊，在事後可透過鑑識系統所擷取之資訊找出攻擊者之主機與身分；或提供鑑識人員查找利用網路會談規劃非法行為之證據，在事後可透過鑑識系統所擷取之資訊找出攻擊者之主機與身分；本研究之設計亦可提供數位鑑識時所需之相關檔案資訊以應用於網路會談管理、監聽蒐證或通訊品質監測…等應用。

未來，本系統於有限資源下，如何因應更複雜之網路環境，並進行

效能優化，將是本系統可繼續研究與探討之方向。

參考文獻

[1] G.983.1 : Broadband optical access systems based on Passive Optical Networks (PON) - http://www.itu.int/rec/T-REC-G.983.1-200501-I

[2] G.984.1 : Gigabit-capable passive optical networks (GPON): General characteristics - http://www.itu.int/rec/T-REC-G.984.1/en

[3] STPI 科技產業資訊室

[4] http://cdnet.stpi.narl.org.tw/techroom/market/eetelecomm/eetelecomm159.htm

[5] FTTH 聚焦中國：宏觀戰略需微觀調控 - C114 中國通信網 http://www.dvbcn.com/2012-05/22-89062.html

[6] 於 Gigabit 被動光纖網路下 MSN/SIP 會談鑑識系統之設計與實作 - 胡勝雄

[7] A. Ra fiq, S.M. H. Zaidi, M. Ramzan, Y. Raja, N. Ghani, "Time quantum based online scheduler (TQOS) for WDM EPON," International Symposium on High Capacity Optical Networks and Enabling Technologies (HONET 2007), pp. 1-6, 18-20 Nov., 2007.

[8] S. Hussain, X. Fernando, "EPON: An extensive review for up-to date dynamic bandwidth allocation schemes," Canadian Conference on Electrical and Computer Engineering (CCWCW 2008), pp. 511-516, 4-7 May, 2008.

[9] S.R. Sherif, A. Hadjiantonis, G. Ellinas, C. Assi, M.Ali, "A novel decentralized ethernetentralized ethernet-based PON access architecture for provisioning differentiated QoS," Journal of Lightwave Technology, vol. 22, no. 11, pp. 2483-2497, Nov. 2004.

[10] A.P. Singh, D. Chadha, "Quality of Service Issues in WDM-EPON Systems," International Conferen Systems," International Conference on Signal Processing, Communications and Networking (ICSCN '08),

pp.194-198, 4-6 Jan., 2008.

[11] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg, “SIP: session initiation protocol,” Request for Comments (Proposed Standard) 2543, Internet Engineering Task Force, Mar. 1999.

[12] J. Rosenberg, H. Schulzrinne, G. Camarillo “SIP: Session Initiation Protocol,”Request for Comments: 3261, Internet Engineering Task Force, June 2002.

[13] J. Rosenberg, H. Schulzrinne,“Reliability of provisional responses in Session Initiation Protocol (SIP). RFC 3262.” Internet Engineering Task Force, June 2002.

[14] J. Rosenberg, H. Schulzrinne, "Session Initiation Protocol (SIP): Locating SIP Servers", RFC 3263, June 2002.

[15] J. Rosenberg, H. Schulzrinne,“An offer/answer model with Session Description Protocol (SDP). RFC 3264.”Internet Engineering Task Force, June 2002.

[16] A. Roach "Session Initiation Protocol (SIP)-Specific Event Notification", RFC 3265, June 2002.

[17] H. Schulzrinne, S. Casner, R. Frederick “RTP: A Transport Protocol for Real -Time Applications” Request for Comments: 1889, Internet Engineering Task Force, January 1996.

[18] A. Almulhem and I. Almulhem, “Experience with engineering a network forensics system,” Lecture Notes in Computer Science, vol. 3391, pp. 62–71, Jan, 2005.

[19] Corey, V. Peterman, C. Shearin, S. Greenberg, M.S. Van Bokkelen, J. “Network forensics analysis,” Internet Computing, IEEE, pp. 60-66, Dec 2002

[20] United States National Institute of Justice, Technical Working Group for Electronic Crime Scene Investigation, 2001.

[21] L. M. Garcia, Programming with libpcap - Sniffing the network from our own application. hackin93 (2008), 39.

[22] simplesniffer.c - Programming with Libpcap, http://www.programming-pcap.aldabaknocking.com/code/simplesniffer.c