

SIP Phone 狀態式網路監測系統之設計

蘇暉凱 王建壹* 古文彬 游千冊 吳承崧
中正大學 電機工程研究所 通訊與電腦網路組
m9216@cn.ee.ccu.edu.tw

摘要

網路監測議題一直是網路管理所重視的技術，在網路上監控與量測使用者封包，不僅在網路管理及資源監控上，可提供網路服務提供者網路使用狀態之相關訊息，進而可作為為網路資源調整之依據。由於網路應用程式及通訊協定的更新與發展，許多網際網路應用並不是簡單地從 IP 層與 TCP 從即可得知使用者是使用哪一種應用，因此監測的需求已經有由網路層與傳輸層的量測往上提升到應用層量測之趨勢。現今網路量測系統雖然宣稱可以量測應用層內容，但多數都是使用非狀態式方式量測，這對於目前許多以 Session 為基礎的多媒體應用程式，是會有量測誤差與量測不準確的問題。因此我們提出狀態式量測技術來量測具有 Session-Based 的應用服務，並以目前最熱門的應用層協定，也是未來多媒體應用可能會依據的基礎協定，SIP 協定為實作範例；在不影響使用者通訊的狀況下，進行 SIP-Phone 通話過程的量測，以即將通訊之語音多媒體資料分析與儲存管理，提供網管人員更多有效且正確之量測資料。

關鍵詞： 狀態式，網路量測，Session，SIP

1. 前言

網際網路 (Internet) 以及全球資訊網 (World Web Site) 的興起造成許多網路服務如雨後春筍般的推出；然而，對於網際網路服務提供者來說，也延伸許多網路管理問題。無論是網路量測、網路安全監控、計費與網路資源分配，皆與傳統線路交換網路不同，這些都有賴於準確且有效率之網路監測技術。

隨著網際網路的應用服務，量測所欲觀察之網路現象也不斷更新。早期網際網路服務，一般從封包中之 IP 層標頭與 TCP 層即可得知使用者與應用服務資訊。之後，隨著 WWW 的盛行，透過 HTTP 協定傳輸各種多媒體資料，因此在分析封包時，必須解析到應用層 Content 部分。例如：透過應用層 Content Analysis，我們可以分析出 HTTP 所傳輸的內容是文字、圖片或文字，FTP 所傳出的檔案名稱、檔案大小...等[2]。

近年來，所發展之網際網路應用服務皆是以 Session 的概念為基礎，如：Instant Message、多媒體串流服務、VoIP (Voice over IP)。Session-Based 應

表 1 Session-based 的網路應用服務

Software	Application Protocol	Control Channel	Data Channel
Windows	MMS	1775/TCP	1775/TCP
Media player	MMS	177/TCP	1024-5000/UDP
	HTTP	80/TCP	80/TCP
Real One Player, Real Player, Quick Time Player	RTSP	554/TCP	554/TCP
	RTSP	554/TCP	6970-32000/TCP
	HTTP	80/TCP	80/TCP
FTP	FTP (active)	21/TCP	20/TCP
	FTP (Passive)	21/TCP	Dynamic/TCP
H.323 IP Phone	H.323	1720/TCP and Dynamic/TCP	Dynamic/UDP
SIP IP Phone	SIP	5060/TCP	Dynamic/UDP

用服務具有一大特性，其邏輯連線具有 Control Channel 與 Data Channel，Control Channel 是使用 TCP/UDP 之 Well-known Port 來通訊，主要是負責做 Signaling 的訊息交換動作，但 Data Channel 之 TCP/UDP Port Number 是在 Control Channel 中交換連線訊息時動態決定的，主要是在 Signaling 確認連線建立完成後作為資料傳輸。因此 Data Channel 之 TCP 層之通訊埠是無法事先決定與預測的[3]。

表 1 是目前網際網路常使用且以 Session-Based 來建構的網路應用服務。但對於 Session-Based 應用服務來說，如表 1 斜粗體部分的運作模式，當前網路量測系統大多無法正確量測，因為這些都是使用非狀態性 (Stateless) 的量測技術，只從 IP Header, TCP/UDP Header 與 Application Content 來分析資料，例如：AW-NETMON、NETUSS 和 Sniffer 等。以 SIP Phone 為簡例，如果我們採用 Stateless 之量測技術，我們只能量測出某個來源到某個目的之 RTP 封包，但如果某個來源到某個目的具有兩條 Session 在通訊，Stateless 之量測技術將無法識別出來。

由於傳統上非狀態性的量測會有量測錯誤與無法準確分析資料，因此我們提出狀態式 (Stateful) 的量測技術，根據 Control Channel 中的訊息來管理與監控整個 Session 的通訊狀態，進而可以避免量測錯誤或訊息資料錯失的情形發生，以及取得更多使用者 Session 之資訊。本論文以 SIP Phone 為例，設計 VoIP 監控系統，分析與統計使用者通話資訊以及通訊內容，進而說明 Stateful 監測系統之重要以其可行性。

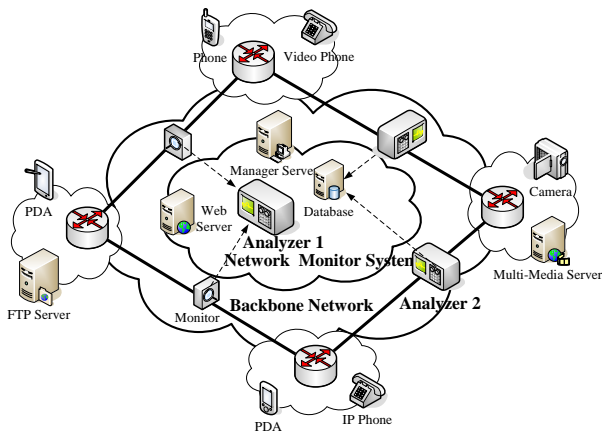


圖1 Network Monitor System

2. 背景

在過往 TANET 和 NBEN (National Broadband Experiment Network) 的網路管理策略中多是採擷取網路上的資訊作為管理的依據，由圖 2 我們可以清楚地看出量測的系統架構，主要可分為 Monitor、Analyzer、和 Database 三個部份。其中 Monitor 主要是負責做資料擷取的動作，Analyzer 主要是負責將擷取到的資料作分析後將有用的資訊存至 Database 中，Database 主要是負責存取資料以便管理者日後作分析。

雖然量測技術都是以擷取資料的原理來建構的，不過根據網管人員的需求和網路環境的改變，量測系統還是有很多種的作法和型態。以下我們將量測系統從不同角度作分類，包含以量測系統所能分析的層次、網路量測的時機和分析的即時性。

- (1) 依資料分析的層次：此分類主要可分為基本型和應用型，基本型的量測只能擷取 OSI Layer 1 至 Layer 4 的網路資料，例如：RMON 和 Sniffer 等網管上常見的量測工具，多只能擷取到 IP 位置、通訊埠等較屬於網路下層的資料。但基本型的量測隨著網路增值服務的不斷擴增，在網管的應用上已漸漸無法滿足網管人員的需求。故應用型被提出來可擷取到 OSI Layer 4 以上甚至是應用層的資訊，例如：NETUSS 便可以擷取到運用的 Session Layer 通訊協定和傳送的資料檔名及格式等等。應用型的量測雖可滿足現今網管的需求，但在設計上去顯得較為複雜，且在速度上還是不及基本型來得快。
- (2) 網路量測時機：從網路量測的時機可分為被動式量測 (Passive mode) 和主動式量測 (Active mode)，被動式量測只單純擷取並分析封包，而這也是大部分傳統量測系所採取的一種方式，但此種方式較適用於像窄頻這類頻寬較小處理速度不快的環境。主動式量測方式主要則是利用本身所發出的測試封包來對整個網路狀況做探測，但主動式量測卻會有著因發出多餘

的封包造成網路的負擔及爭議。不過在現今網路頻寬日益擴增的情形下，測試性封包所佔用的頻寬也就相對性的減少很多。雖然兩者的做法不盡相同，但在網路管理的策略應用上確有相同的結果及目的。

- (3) 分析的即時性：我們可依據分析的即時性將量測技術分為即時分析 (On-line Analysis) 和非即時分析 (Off-line Analysis)，即時性分析的量測顧名思義便是直接針對我們所需量測之特殊應用連線做直接的過濾跟分析，如中的 Analyzer 2 便是直接在骨幹網路上做過濾跟分析的動作，在將所得資料存至 Database 中。反之非即時分析性的量測便是先將先收集起來後再做後續分析和過濾的動作，如中的 Analyzer 1。而兩者在量測應用上最大的不同便是在於速度上的差別，但速度快卻會造成其所量測的資料只能針對特定的網路服務而無法做整個網路的量測，因此採用何種量測完全只是應用和網路環境上的考量。

以網路多媒體服務成長的速度來看，現今絕大部分都使用 Session-Based 的架構，所以針對以 Session-based 為架構的網路服務量測有實質上的需要。由於傳統量測技術大都是屬於基本型、被動性的非即時分析的方式，因此在面對以 Session-based 為架構的網路增值服務時會產生無法量測和量測到錯誤的訊息，而採取應用型的方式量測同樣也會有量測錯誤的情形發生。因此我們提出狀態式量測的概念，可隨時根據控制訊息的改變來調整量測的方式，得到更詳細且更準確之監測資料。

3. 狀態式量測技術

由於傳統量測技術的缺點，無法針對 Session-Based 的應用作監測，因此我們提出具有狀態式的量測技術以適應越來越廣受歡迎的多媒體網路傳輸應用。

3.1 Session-Based 通訊協定

在點對點的通訊應用中，為了確保雙方通訊的完整性及交換彼此通訊的能力，會在建立媒體通道 (Media Channel 或 Data Channel) 前，先作某種程度上的溝通及協調。而在這個過程中，若監測系統不追蹤與管理 Session Control Channel 的狀態與溝通內容，會導致擷取媒體資料 (Media Data) 的錯誤判斷或不能解讀的問題，甚至會遺漏真正的使用者語音資料。

一般用來傳輸多媒體資訊的應用程式，是為了因應多種網路狀態及終端應用軟體對各類多媒體資料格式之支援，因此都會使用具有建立 Session 的 Control Channel 來提高雙方通訊的可靠度，例如：SIP、H.323[4]、RTSP (Real-Time Stream Protocol) 和 MMS (Microsoft Multimedia Service) 應用層之

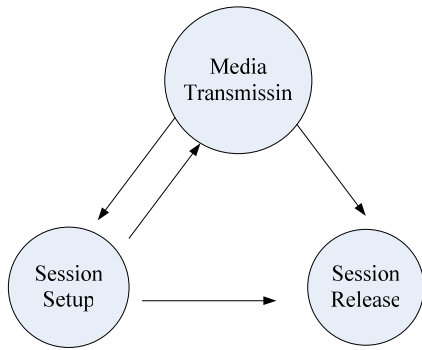


圖2 Session 的三個狀態

通訊協定。這些 Session-Based 的應用服務，會維持各個狀態的切換來強化通話的建立，我們可將整個通話過程分為三個狀態，分別是建立 Session (Session Setup)、媒體傳輸 (Media Transmission)、結束 Session (Session Release)，如所示。

Session-Based 的通訊協定要起始一個 Session 時，會從 Session Setup 狀態開始，在這個狀態中，雙方會交換彼此的通訊與媒體編/解碼支援能力，這些資訊包括能夠接受的頻寬、所支援的媒體編解碼器 (Media Codec)、相互的認證許可及傳輸層的位址 (Transport Address)。當 Session Setup 將 Media Channel 成功建立後，就會進入 Media Exchange 狀態，在此狀態中，雙方會根據 Session Setup 所給的資訊設定參數來傳送多媒體資料，在這過程中，若需要重新設定媒體參數，則會回到 Session Setup 狀態重新建立或修改之前所建立的 Session。而當通話結束後，會進入 Session Release 來釋放建立通話時所給定的資源。

從這三個狀態可知，若我們想正確的監測雙方傳送的多媒體資料，必須要瞭解兩端點的 Session 狀態。若以 SIP VoIP 的應用服務為例，其使用 INVITE 與 OK 之 Control Message 來建立啟始 Session 與交換終端機能力 (Capacity Exchange)。因此針對 Session-Based 之應用程式，一定必須追蹤與管理 Session 之通訊狀態，監測系統才有能力抓出 Media 真正傳送的 Channel。

3.2 狀態式量測技術

針對 Session-Based 的狀態式監測技術主要是監視通訊過程中的控制通道 (Control Channel) 以作為媒體資料歸類的依據，因此如何監視控制通道將會是整個狀態式監測器的設計核心。監視控制通道有兩個目標需要達成：(1) 追蹤 Session 通訊狀態。(2) 取得媒體通道的資訊。

狀態式監測技術須有能力隨著 Session 狀態的改變作狀態更新，因此監視控制通道的第一個目標是需要擷取關鍵的訊息，並依據訊息追蹤適當的 Session 狀態。監視狀態和 Session

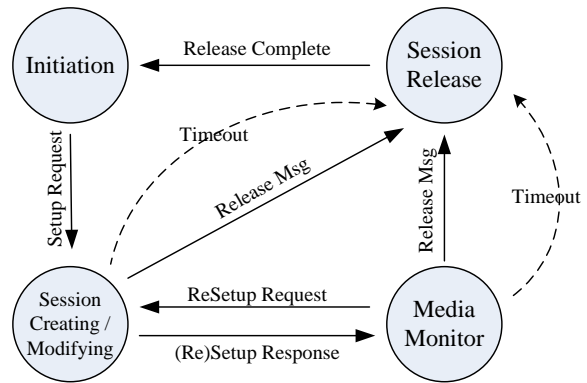


圖3 Stateful Monitor Design Model

Application 的協定標準狀態不一定需要完全吻合，如果追蹤太多狀態，雖然可以完全掌控 Session 的建立、釋放以及更多資訊，但可能需要更多理程序 (Processing Power) 及不必要的時間耗費。相反的，太鬆散的監視狀態設計，會有 Session 狀態被遺漏而無法準確監測到所需的資訊，但其優點在於節省封包檢測 (Packet Inspection) 的時間及數量。因此效能良好的監測器需能夠掌握 Session 的關鍵訊息，花費最小的監測成本。

監視控制通道的第二個目的是取得媒體通道的資訊，其所要取得的資訊須紀錄媒體資料的來源端 (SA)、目的端 (DA)、來源傳輸位址 (SP)、目的傳輸位址 (DP)、協定型態 (PID)。我們可利用這些資訊來過濾媒體資料並將資料作正確的存放或描述，好讓後端的機器作後續處理。監視控制通道可瞭解媒體通道建立及釋放的時機，讓監測器在適當的時間擷取正確的媒體資料。

3.3 狀態式量測機制之設計

在本狀態式監測系統的設計中，我們預計監測使用者資訊、通訊連線時間與通訊資料資料格式與內容。因此，我們設計四個監測狀態來維持每個 Session 的監視過程，此四個狀態為 Initiation、Session Creating / Modifying、Media Monitor 和 Session Release，如。任何具有 Session-Based 的應用都可使用這四個狀態來監測，而其他可能特殊的狀態會依據不同特別的應用而增加。

Initiation 狀態主要是在 Session 沒被建立前監視各種封包，並且抓出需建立 Session 時會發出的關鍵訊息，我們稱此訊息為 Setup Request。當監測器收到由請求者或發話端 (Caller) 送出 Setup Request 此類的封包後，會跳到 Session Creating 狀態，在此訊息中，Caller 會將欲建立的媒體資訊夾帶在訊息裡，因此監測器須將此資訊解析並且記錄下來以作為 Media Monitor 狀態時參考的依據，Session Creating 狀態主要是已確定有 Session 被建立但被請求者或受話端 (Callee) 還

沒接受請求，在此狀態會持續監測封包，且以期待收到 Setup Response 或 Release Message 為主。Setup Response 是 Callee 回送給 Caller 的訊息並帶有同意 Caller 的請求，反相的媒體通道資訊也會被夾帶在此訊息裡，當監測器收到此訊息後，會相對的解析出媒體通道資訊並且跳至 Media Monitor 的狀態以監測多媒體封包。Release Message 則會夾帶 Callee 拒絕建立 Session 的訊息，若收到此訊息則監測器會跳至 Session Release 狀態並且釋放為此 Session 所存取的資源，然後再跳回初始狀態。而在剛進入 Session Creating 的狀態時，會啟動一個計時器以防止遺漏關鍵訊息時還可以跳到 Session Release 狀態並將資源釋放。

當進入 Media Monitor 狀態後，會針對控制通道所監視到的資訊作監測，若監測到封包是屬於此 Session 的媒體資料，則會將此資料儲存或傳遞到相對的裝置，並且將附加對此媒體資料的描述資訊，此資訊可包含 Caller、Callee、Codec 及其他相關資訊。若在此狀態監測到屬於結束此 Session 的 Release Message，則監測狀態會跳至 Session Release 並且將系統不必要的資源釋放掉。在 Media Monitor 狀態也會維持一個計時器以防止結束 Session 的訊息被遺漏掉。而處於 Session Release 狀態時，可以將整個 Session 建立的起始時間、歷時多久等資訊描述出來並且和媒體資料存在一起，而當所有資源都釋放後，則會跳回至 Initial State 回到初始狀態。

4. 狀態式量測技術應用於 SIP Phone 之設計

SIP 為 Session-Based Protocol[5]，所以必須以狀態式的量測技術來進行監測。首先聽取 SIP(Control Channel) 封包，分析出 Media Channel 資訊後，再監聽 Media Channel 中的 Media data，把得到的 Media data 存到資料庫給後端程式處理，必要時可讀取 Media Data 並依格式播放監聽。

4.1 SIP 應用程式式

SIP 為應用層的一種 Session-Based 協定，主要應用在影像、語音的即時通訊，目前有許多此類的軟體，包括 SJphone、Linphone 及 MSN。使用狀態式監測技術在監測 SIP-Phone 上，先必須了解 SIP 的架構及協定的溝通，其中包括 SIP Header、SDP (Session Description Protocol) 訊息、RTP (Real Time Transport Protocol) Header。SIP Call 程序為圖 4 所示，圖中的 INVITE 的 body 部份是 SDP 訊息帶有溝通 Media Port、Audio/Video Codec 的重要資訊。Callee 回應的 OK 資訊也帶有 SDP 用來溝通 Media Data 傳輸的方式。在建立起連線後，媒體資料會開始以 RTP 封包傳遞，這段期間可能有一些 SIP 的控制信號來改變 Session 的狀態，若要結束時會發出 BYE 來結束整個 Session。

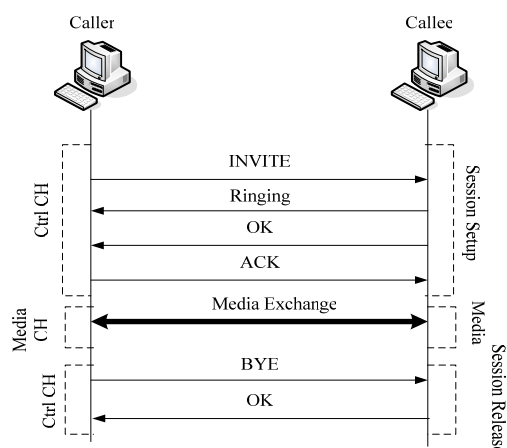


圖4 SIP Phone Call Flow

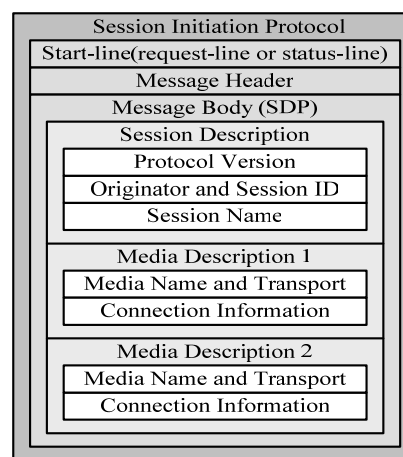


圖5 SIP Message Architecture

4.1.1 SIP 標頭

SIP 與 SDP 為 Text-based Protocol，message 分為三部份，(1) Start-line。(2) Message Header。(3) Message Body。如所示。

Start-line 目的在於明確表明此 SIP 封包的種類與功能，使 User-Agent 能看 Start-line 就知道封包的類型。Start-line 分為 Request-line 與 Status-line 兩種，Request-line 是請求的訊息，包括要求建立通話 (INVITE)、要求結束通話 (BYE or CANCEL)、要求更改 Session 參數 (Option) 等。Status-line 用來表明本身狀態，如：100 Trying、180 Ringing 與 200 OK 等，Status-line 都是由 Status Code 與 Reason Phrase 組成。Message Header 用來描述 Session 的資訊，重要且強制性的 Message Header 包含 To (Callee 的 URI)、From (Caller 的 URI)、Call-ID (Session Identifier)。CSeq 由 Command Sequence 與 Method 組成。其它非強制性的 Header 包含 User-Agent 代表使用何種的 SIP 軟體。Message Body 部份除了對 Session 作詳細描述外，還描述了第二部份沒描述的媒體資訊，如 RTP Codec、Media Port 等，通常在這個部份使用 SDP 來描述，固接下來對 SDP 作介紹。

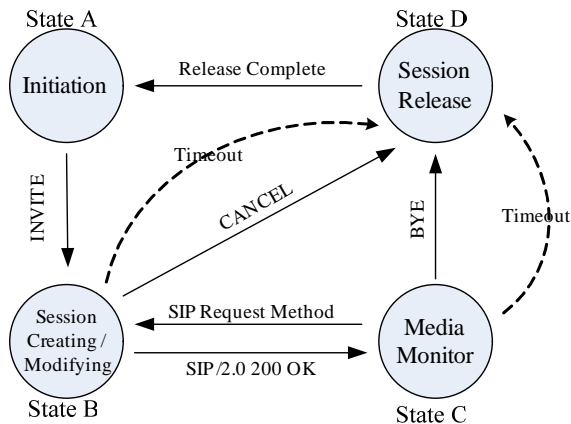


圖6 SIP Session Monitor State

4.1.2 SDP 標頭

SDP[6]用來描述一個 Session 的詳細資訊，包括 IP-address、RTP Codec、接收 RTP 封包的 Media Ports 等。SIP 使用 SDP 來建立一個 Answer/Offer Mode：Caller 將多媒體傳輸能力夾帶在 INVITE 封包中，Callee 可從 INVITE 的資訊挑出自己支援的項目帶在 SIP/2.0 200 OK 資訊並回應。SDP 中又細分為兩部份：(1) Session Level (2) Media Level。我們將只介紹 SDP 重要且必要的欄位。Session Level 中的「v」代表 Protocol Version。描述 Username、Session-ID、Version、Network Type (IN 表示 Internet)、Address Type(IPv6 or IPv4)、Address (IP)。Media Level 中的「m」描述 Media Type (audio/video)、RTP port、RTP Codec。Session Level 與 Media Level 的分界點為第一次出現 m 的欄位為分界。除此之外，非強制性欄位 a 則可能出現在 Session level 或 Media level 中，做為欄位的附加描述。

4.1.3 RTP 標頭

用 RTP 做為即時影像/語音傳送格式，第四層使用 UDP (User Datagram Protocol)，基本的 RTP Header 為 12Bytes，第 2 個 byte 表示 RTP 的 Codec，若傳送時經過 Mixer 處理，則會多加 4 Bytes 的 CSRC (Contribution Source identifier) 欄位或有 Extension Header[7]。

4.2 設計原理

為監測 SIP Call 的系統狀態圖，為了兼顧減少處理程序與確實掌握 Session 的重要訊息，系統的設計只有四種 State，設計的重點在於 (1)如何判斷進入下個狀態的條件。(2)每個狀態該執行的任務。我們檢查 SIP 封包的 Start-line 來判斷各個狀態的切換時機。從 SDP 中取得 Media channel 的資訊，監聽 Media Channel 並判別出 RTP stream，將媒體資料存檔並加入描述檔。

表 2 SIP 監測狀態的改變

狀態改變	判斷準則
A → B	Start-line 為 INVITE sip: (callee URI) SIP/2.0。
B → C	Start-line 為 SIP/2.0 200 OK、Call-ID 為此 Session 的 Call-ID，Message Header 中 CSeq: 1 INVITE。這三個條件目的在確定 OK 是回給同一個 Session 的 INVITE response。
C → D	收到 SIP 封包 start-line 為 BYE sip: (URI) SIP/2.0，當然，Call-ID 也必須符合此 Session 的 Call-ID 才能確定為結束 Session。
B → D	Start-line 為 CANCEL sip: (URI) SIP/2.0、Call-ID 符合此 Session 的 Call-ID，符合這兩個條件才能確定為結束 Session。

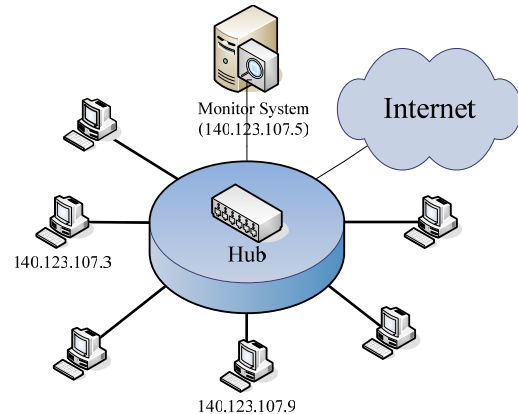


圖7 實作環境架構

此節主要說明監測系統會對每個 SIP Session 維持狀態紀錄並且依 Session 狀態切換，然後從控制訊息可得知媒體資訊，並將媒體資料判斷出來，狀態的切換須依照表 2 的判斷準則作狀態的切換。

SDP 中 Media Port 判斷系統設計上，我們從 Media Description (MD) 得到的資訊必須考慮三個部份才能確保不遺漏媒體通道資訊，而會有此問題是由於各家廠商設計的 SIP-Phone 對於媒體通道建立方式的差異造成的結果。第一部份為 MD 的分析，不同的 SIP 應用程式所填寫的 MD 有所差異，故需針對各種 MD 作解析。第二部份為應用程式選擇的 Media Codec 準則不一，這關係到抓取媒體資料的正確性。第三部份則是可能同時開啟多個媒體通道，因此需能夠判別這些通道的資訊並作正確的抓取。

由於不能直接看應用層的 Header 來判定是否為 RTP 封包，因此對於 RTP Stream 的判斷是依照之前監測控制通到所得知的 Media Port、SA、DA 等資訊來抓取多媒體封包。

4.3 實作環境與監測結果

我們測試的環境如所示，所有的電腦用 Hub 連接起來並安裝有 SIP-Phone 程式，Monitor System 啟動監測程式，並使用電腦互打電話來測試是否有正確的監測到 SIP 資訊與媒體資料，測試軟體包含 SJphone、Linphone、MSN 三種 SIP-Phone。

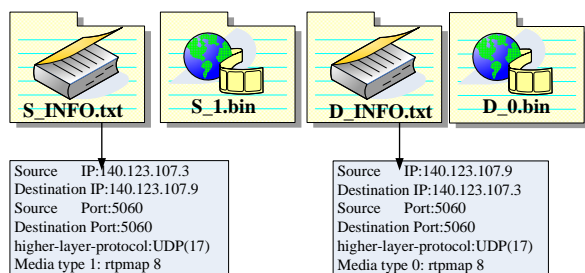


圖8 Session 監測紀錄檔（描述檔與媒體資料檔）

如圖 8 所示，SJphone 的通話過程經過 Monitor System 擷取，產生四個檔案 (1) S_INFO: 紀錄 Caller 端資訊。(2) S_1.bin: 為 Caller 送給 Callee 的 RTP stream, 此為去掉 RTP Header 後儲存的 binary 檔案。S_1.bin 由名稱可知為使用 Media type 1 所傳遞的 RTP stream, 在 S_INFO 檔中會找到 Media type 1 所使用的 RTP Codec 為何。(3) D_INFO: 紀錄 Callee 端資訊。(4) D_0.bin: 為 Callee 端傳給 Caller 的 RTP stream, 由名稱可知為使用 Media type 0 所傳遞的 RTP stream, 在 D_INFO 檔中可以找到 Media type 0 所使用的 RTP Codec 為何。經過三種 SIP-Phone 軟體的測試，我們都能很正確的判斷 Session 狀態，也能抓取到 Media Channel 的 RTP Stream。

5. 結論

本論文中，我們先分析傳統非狀態式的量測技術有遺漏資料和錯抓資料的缺點，再提出我們所設計的狀態式量測機制。具有狀態的監測可適應目前廣受歡迎的多媒體傳輸，其優點在於追蹤與管理 Session 之狀態，可正確地擷取真正該 Session 所傳輸之媒體資料，而不發生誤判或遺漏資料的問題。

我們分析 Session-Based 協定建立的三個主要必經狀態，並依據此結果設計四個量測 Session 的監測狀態，任何要監測 Session-Based 的技術，都可由此模式作擴增及刪減。而在最後，我們以 SIP 通訊協定為實作成果的展現，並根據 SIP 的信令來決定媒體通道的資訊，再利用此資訊來監測媒體封包。在實作過程中，我們發現監測到的語音品質和實際兩方對話的語音品質有所差異，故未來會著重於監測效能的議題。另一方面，為了解決 Stateful 量測所會面臨的效能問題，我們預計使用 NP-Based 的平台來實作狀態式監控系統。

參考文獻

- [1] B. Goode, "Voice over Internet protocol (VoIP)," Proceedings of the IEEE, vol. 90, pp. 1495–1517, Sept. 2002.
- [2] 洪正雄, 高宏政, 李政修, 黃淑華, 方存偉, "骨幹網路上的應用層流量監控系統-NETUSS", 2003 Symposium on Digital Life and Internet Technologies, 2003.9

- [3] Hui-Kai Su, Cheng-Shong Wu and Kim-Joan Chen, "Session Classification for Traffic Aggregation", 2004 IEEE International Conference on Communications (ICC 2004), June 23, 2004.
- [4] ITU-T Recommendation H.323: Packet-based multimedia communications systems," International Telecommunication Union, 1997.
- [5] J. Rosenberg, H. Schulzrinne, Camarillo, Johnston, Peterson, Sparks, Handley, and Schooler, "SIP: Session initiation protocol v.2.0," IETF RFC 3261, 2002.
- [6] M. Handley and V. Jacobson, "SDP: Session description protocol," IETF RFC 2327, 1998.
- [7] <http://www.networksorcery.com/enp/protocol/rtp.htm>