

第二章 基礎理論介紹

本章的主要目的是在介紹 SIP 協定的基礎原理，包含 SIP 的主要功能、元件架構、呼叫單元以及如何透過 SIP 訊息建立會議的流程等，方便我們對本篇論文有初步的認識。

2.1 SIP 的主要功能

SIP 允許用戶或裝置透過訊息的傳遞產生互動，這些訊息可以滿足以下四種目的：

1. 使用者向系統註冊
2. 邀請使用者參加互動會議
3. 協商會議媒體內容的格式
4. 建立兩端點或更多人之間的媒體位元流
5. 結束會議

如同所有其它協定，SIP 協定一樣也包含了許多不同型態的 requests 和 responses 訊息，這些訊息的型態和詳細描述將會在之後提到，以下則會先提及有關註冊、邀請和結束會議的訊息。

1. 使用者向系統註冊

在以 SIP 為基礎的系統裡，使用者可能是位於相同乙太網路的兩個點對點裝置，或者是位於分散式系統裡，由伺服器和終端裝置所擴展開來的 SIP 系統，但是當我們的系統變得越來越大，邀請群組參加新會議的方法將會變得更加複雜，除此之外，大部分的人都是移動性強，並且同一個使用者可能會從各種不同的位置和裝置回覆會議的邀請訊息。

SIP 提供了註冊的機制將使用者和系統做連結，當 SIP 電話或終端裝置為線上狀態時，通常第一件要做的事就是傳送 REGISTER 訊息給 Registration 伺服器，告知目前所在的聯絡位址和相關資訊，而 Registration 伺服器則會將使用者 ID 和 IP 位址結合起來記錄在 Location Server 的資料庫裡，以供日後使用，註冊的機制可使使用者具可移動性的優點，使用者只需隨時向伺服器更新自己的 IP 位址，便可使用相同 ID，在不同地區或裝置上讓其它連絡人找到自己，而不需固定在某一 IP 位址上。

註冊的機制並不是必要的，SIP 電話在知道雙方 IP 位址的條件下，也可以不用透過任何伺服器達成建立連線的目的。註冊的機制通常都有時效的限制，而使用者則必須每隔一段時間和伺服器聯絡，以隨時更新目前的位址。

2. 邀請使用者參加互動會議

在 SIP 的系統裡，建立會議的定義就是透過傳送 INVITE 訊息給一到多個裝置建立雙方連結，等建立連線後雙方便可傳送語音、影像或其它數據資料。在 SIP 裡定義的位址可能是傳統的電話號碼、直接的 IP 位址或 SIP URIs 等，當使用者想邀請另一方加入會議，使用者只需在裝置裡鍵入對方的位址，並等待對方的回應。

在等待對方回應的過程中，SIP 裝置會建立 INVITE 訊息並將它傳送到網路上，透過 Proxy 或 Redirection Server 到 Location Server 尋找對方真實的 IP 位址，有了 Location Server 提供的資訊，系統便可以判斷如何將邀請傳達給對方。

3. 協商會議的媒體內容格式

建立 SIP 會議的主要目的是用來傳送雙方的媒體內容，所以在 SIP 傳送的訊息內容中會附加可提供的媒體格式供對方參考，和 email 服務相同，SIP 協定只關心所傳送的訊息內容，並不管所附加的內文是什麼，而在 SIP 裡最常使用到的附加內文則採用 Session Description Protocol (SDP, RFC 2327)協定格式。

當接收方收到 INVITE 訊息的時候，接收方可以選擇接受或拒絕這次的邀請，有時候接收方拒絕邀請的原因是因為它沒有足夠的能力接受對方所提供的媒體需求，例如接收方可能不提供視訊串流服務，這時候只能拒絕對方的邀請。

4. 建立兩端點或更多人之間的媒體位元流

當對方接受了邀請，代表著此次會議已成功建立，接下來進行傳送的便是媒體位元流(media stream)，在 SIP 裡，媒體位元流和 SIP 協定的訊息是分開建立，並且使用 Real-time Transport Protocol (RTP, RFC 1889)進行傳送。

和 SIP 訊息傳遞不同的是，媒體位元流直接使用得到的 IP 位址進行點對點的傳輸，而不需要再次經過 SIP 訊息用來建立會議的 servers。

5. 結束會議

當使用者結束會議掛斷電話的時候，此時裝置便會傳送 SIP 的 BYE 訊息給對方用來結束媒體位元流，連線便正式終止。

2.2 SIP 的元件架構

在 SIP 的標準裡定義了幾個 SIP 的元件，用來建立和傳送 SIP 訊息，其中主要包含兩大類，SIP User Agents 和 SIP servers。

SIP User Agents 是用來建立和接收呼叫的終端裝置，它們可能是實體 IP 電話，或是在 PC 上執行的軟體，SIP Servers 則包含了許多會議控制系統，用來提供 location、proxy、redirection 和 registration 服務，以下則是這些元件的介紹。

1. SIP User Agents

在 SIP 協定裡，電話裝置被定義為 User Agents (UAs)，UAs 可視為 user agent clients (UACs)和 user agent servers (UASs)的通稱，當開始建立會議時，UAC 是用來建立和傳送原始 request 的一方，而 UAS 則為所有 server 型態的泛稱，可用來接收 requests 訊息，並回傳 response 訊息。

SIP UAs 可用硬體實現，例如 IP 電話，也可以是在個人電腦裡的軟體電話。

2. SIP Server

雖然 UA 裡包含了 server 的元件，但是一般當我們提及 SIP servers 時，我們通常指的是在分散式系統裡扮演主要控制角色的 server，在這裡我們討論四種在

RFC 所定義的 SIP server 類型：

a. Location server

Redirect 或 Proxy server 可使用 Location server 獲得被呼叫端可能的位址資訊

b. Proxy server

Proxy server 扮演著中介者的角色，它可以同時是 client 端也可以是 server 端，Proxy server 的目的在於接收並轉送收到的訊息到其它的 servers，當 proxy server 收到訊息後，必須設法解釋收到的訊息，甚至如果必要的話也可以重新定義 request 訊息，並轉送到其它 servers。

c. Redirect server

Redirect server 負責將收到的 SIP request，找出對應於舊位址的新位址，並回傳給 client 端，和 Proxy server 不一樣的是，Redirect server 並不能處理對話，但是可以產生 SIP response 訊息，指示 UAC 連絡其它的 SIP 位址。

d. Registrar server

Registrar server 負責接收 REGISTER 訊息，並將使用者所傳送的位址資訊記錄在 Location server 裡，以供 Proxy 或 Redirect server 查詢用戶位址資訊。

2.3 SIP 的呼叫單元

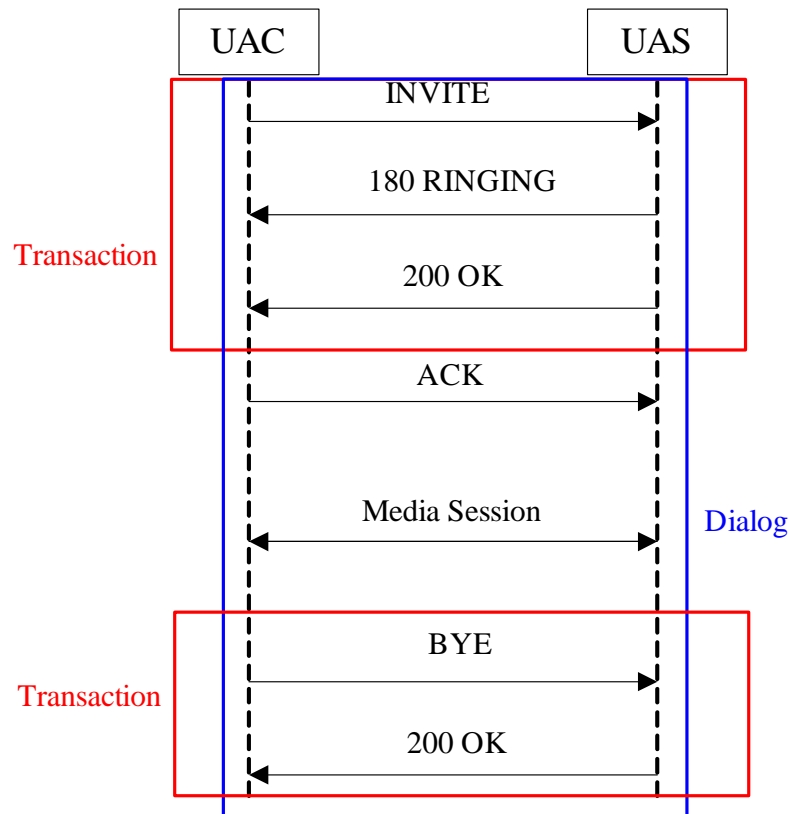


圖 2.1 SIP 基本呼叫建立和結束流程

在我們進一步探討 SIP 訊息的功能時，我們必須先定義 SIP 組成呼叫的單元，其中包括有最基本的訊息(Message)，以及由訊息所組成的交易(transaction)和對話(Dialog)。

1. 訊息 (Message)

訊息是指在 UAs 之間傳遞的資料，它是 SIP 協定的一部份，可以用來建立、修改、結束會議，並分成由 client 端傳送到 server 端的 request 訊息，以及 server

端傳送到 client 端的 response 訊息。

2. 交易 (Transaction)

SIP transaction 發生在 client 和 server 之間，它包含了單一的 request 和針對此 request 訊息所回覆的所有 response 訊息，不同的 request 訊息形成不同的 transaction。

圖 2.1 的建立對話過程包含了三個 transaction，其中步驟 1-3 是第一個 transaction，包含一開始 UAC 傳送的 INVITE request，以及接下來 UAS 傳送的 180 Ringing 和 200 OK response，步驟 4 的 ACK request 則自己成為一個 transaction，步驟 6-7 是第三個 transaction，其中包含了 UAC 傳送的 BYE request 和 UAS 回覆的 OK response。

3. 對話 (Dialog)

我們剛剛定義了 Transaction，知道 Transaction 是包含了單一的 request 和針對此 request 訊息所回覆的所有 response 訊息，而不同的 request 訊息形成不同的 Transaction，由圖 2.1 可知包含 INVITE 訊息以及所有 UAS 回傳的 response 為一個 Transaction，而包含由 UAC 傳送的 BYE request 和 UAS 回覆的 OK response 則為另一個 Transaction，但是這兩個 Transaction 並不是完全沒有關係，因為這兩

個 Transaction 同屬於一個 Dialog。

SIP 將 Dialog 定義為兩個相同用戶從會議開始到結束的這一段連續時間裡所傳遞的所有訊息，我們可以使用 Call-ID、From tag 和 To tag 這三個不同的 header 代表 Dialog，如果兩個 Message 裡的 Call-ID、From tag 和 To tag 都相同，則代表它們屬於同一個 Dialog，藉由圖 2.1 可以清楚的知道 Message、Transaction 和 Dialog 三者之間的關係。

2.4 SIP 訊息和簡單的呼叫建立

上一節提到了 Message 的定義，其中 Message 又可分為 request 和 response，SIP 定義了許多用來傳遞資訊的 request 和 response 訊息，以下介紹幾個常見的訊息，以及如何使用這些訊息達成會議的建立。

1.SIP Request 訊息

Request 訊息是指從 client 端傳送到 server 端的訊息，SIP 協定將 request 訊息定義為各種不同的型態，而每一種型態都可視為一種動作，用來要求 user agent 或 server 特定的服務，以下介紹幾個常見的 request 訊息。

- a. INVITE：初始會議
- b. ACK：確認會議建立
- c. BYE：結束會議
- d. CANCEL：取消尚未建立的會議
- e. REGISTER：向 SIP server 登記使用者位址

2. SIP Response 訊息

Response 是指從 server 端傳送到 client 端的訊息，用來回應 client 端所發出的 request 訊息，response 訊息可能包含額外提供的資訊，但也可能只是簡單的回應，用來防止 client 端再次發送 request 訊息。

在 SIP 協定裡將 response 訊息分成六大類，其中前五大類借用了 HTTP 協定，第六大類則是由 SIP 協定建立，以下介紹這六大類 response 訊息。

a. 1xx Responses: Information Response

代表 request 已收到，並且正在執行 Request 訊息所要求的事件。Example: 180

Ringling

b. 2xx Responses: Successful Responses

代表 Request 要求的動作已被成功接收、了解並接受。Example: 200 OK

c. 3xx Responses: Redirection Responses

表示為了完成 Request 訊息需要進行額外的動作。Example: 302 Moved

Temporarily

d. 4xx Responses: Request Failure Responses

代表使用者所發出的 Request 訊息包含了語法上的錯誤，所以 server 端無法滿足此 Request 訊息的要求。Example: 400 Bad Request

e. 5xx Responses: Server Failure Responses

代表 Server 端發生錯誤而無法滿足 Request 要求。Example: 500 Server Internal

Error

f. 6xx Responses: Global Failure Responses

代表使用者的要求在所有的 Server 端皆無法被滿足。Example: 600 Busy

Everywhere

3. 基本呼叫建立

圖 2.1 代表 SIP 基本呼叫建立和結束的流程，UAC 藉由撥打對方的 IP 位址建立新的連線，SIP 裝置則會傳送 INVITE 訊息給 UAS。

以下為圖 2.1 所有的訊息描述：

a. INVITE

當使用者拿起電話開始撥打對方的 IP 位址時，這時候 UAC 便會傳送 INVITE 訊息給 UAS，INVITE 訊息通常包含 SDP 協定的內文，用來告訴 UAS 可接受的媒體格式。

b. 180 Ringing

當 UAS 端接收到 INVITE 訊息之後，這時候 UAS 的電話鈴聲便會開始響起，並且回傳 180 Ringing 給 UAC 端，此回傳的 Response 訊息包含了 SDP 格式的可接受編解碼媒體格式。

c. 200 OK

當 UAS 端的使用者接起電話，此時 UAS 便會回傳 200 OK 的 response 訊息給 UAC 端。

d. ACK

在 UAC 接收到 180 Ringing 和 200 OK 的 Response 訊息之後，為了回應 200 OK 訊息，必須回傳 ACK request 訊息給對方，此時便完成訊息傳遞的三方握手機制，而連線也正式建立。

e. Media Session

一當 SIP 的連線建立完成，雙方使用者的媒體通道也會立刻建立，在這裡使用 Real-time Transport Protocol (RTP, RFC 1889)達成即時媒體傳送的目的。

f. BYE

當雙方通話結束時，UAC 端會傳送 BYE request 訊息給 UAS 端，代表 UAC 要掛斷電話。

g. 200 OK

當 UAS 端接收到 BYE request 訊息則會回傳 200 OK 訊息，此時會議結束，並且媒體通道也會跟著中斷。

2.5 訊息檔頭

每一個有意義的 SIP 訊息皆包含了許多檔頭欄位，這些檔頭都有其重要的意義，下面是一個由 UAC 產生 INVITE 訊息的檔頭資訊：

To: "The Little Blister" <sip:LittleGuy@there.com>

From: "The Master Blaster" <sip:BigGuy@here.com>;tag=12345

Call-ID: a4a08ea5f3563c128b4c1bbf219ca9b3@127.0.0.1

CSeq: 1 INVITE

Via: SIP/2.0/UDP 127.0.0.1:5060; branch=z9hG4bK1d32hr4

Max-Forwards:70

a. To

代表訊息預期接收的位址，在這個例子裡是：

To: "The Little Blister" <sip:LittleGuy@there.com>

其中 The Little Blister 代表接收端的顯示名稱，sip:LittleGuy@there.com 則是接收端的 SIP URI，To 檔頭的值不會隨著經過的 server 改變。

b. From

所有的 request 和 response 訊息都需要 From 的檔頭欄位，它代表了訊息產生者的位址，在這裡的例子是：

From: "The Master Blaster" <sip:BigGuy@here.com>;tag=12345

其中 The Master Blaster 代表傳送端的顯示名稱，而 sip:BigGuy@here.com 則是傳送端的 SIP URI，From 檔頭由使用者事先提供，並有代表傳送端的 tag 欄位

c. Call-ID

Call-ID 提供了一個全世界獨一無二的識別碼，用來辨別每一次的會議，當 UAC 開啟會談時就會建立一個 Call-ID 以便與其它會談區隔，而所有在同一個會談中傳遞的 requests 和 responses 訊息皆擁有相同的 Call-ID，在這裡的 Call-ID 的例子是：

Call-ID: a4a08ea5f3563c128b4c1bbf219ca9b3@127.0.0.1

其中 a4a08ea5f3563c128b4c1bbf219ca9b3 是用 32-bit 隨機密碼所組成的，而 127.0.0.1 則是邀請端的 IP 位址。

d. Cseq, or command sequence

包含 sequence number 和 request 訊息型態，在這裡的例子是：

CSeq: 1 INVITE

其中 1 為 sequence number，INVITE 則為 request 訊息的型態，所有的 request 和 response 訊息都需要 Cseq 檔頭，sequence number 的初始值不一定要是 1，但一定要是整數，而在同一個 Call-ID 的情況下，每傳送一個新的 request 訊息，Cseq 值便會增加，接收端則可藉由不同的 Cseq 值分辨這是一個新的 request，而不是重新傳送的 request，重新傳送的 Cseq 值是相同的，Cseq 可以用來區隔和排序每一次的 request 訊息，而 response 訊息則可藉由 Cseq 檔頭得知所回應的 request 訊息是什麼。

如果 ACK 和 CANCEL request 訊息是接著 INVITE request 傳送的訊息，則傳送 ACK 或 CANCEL request 的 Cseq 值會和傳送 INVITE 訊息的 Cseq 值相同。

e. Via

Via 檔頭欄位紀錄使用的 transport 以及所有 request 訊息經過的位址，以供 response 訊息找回 request 訊息發出的地方，在這裡的例子是：

Via: SIP/2.0/UDP 127.0.0.1:5060; branch=z9hG4bK1d32hr4

從以上的例子可以知道，此訊息協定為 SIP，版本號碼是 2.0，並透過 UDP 協定傳輸，傳送端的位址和 port number 則為 127.0.0.1 和 5060，同一個訊息的 Via 檔頭可能不只一個，如果有兩個以上的 Via 檔頭，則所有 Via 順序是非常重要的，因為 response 必須沿著這些位址才能回到 request 產生的地方，假設 request 訊息經過 SIP Proxy，則此 Proxy 會將位址記錄在 Via 欄位的最上端。

檔頭裡的 branch 欄位是由 Request-URI, To, From, Call-ID 和 CSeq number，利用 hash function 所產生的，branch 參數可用來當作 transaction id，每產生一個新的 transaction，branch 值都會不同。

f. Max-Forwards

Max-Forwards 檔頭是用來限制 request 訊息到目的地之前可以經過的 hop 數

目，每經過一個 hop，max-forwards 的值便會依次遞減 1，當 proxy 收到一個 max-forwards 的值為 0 的 request 訊息，便會將此 request 訊息丟掉，並回傳 483 Too Many Hops response，Max-forwards 的建議預設值為 70。