第二章

文獻探討

本研究之主要目標在於解決無線網路的環境中的有關資源存取等議題。因此，在文獻探討的部份，如圖所示2.1，將會針對問題面從不同的方法面加以研究。並整理出相關資料及應用方法，希望能提出一個更具彈性的解決方法。

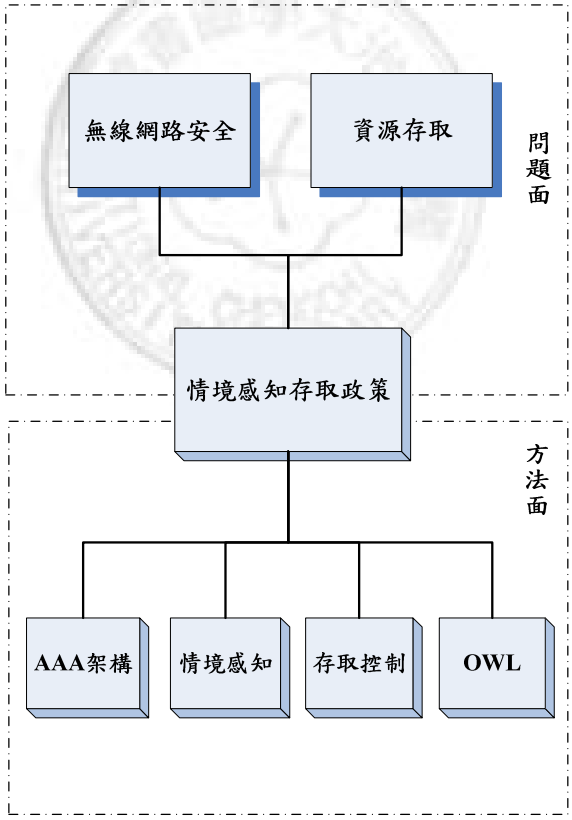

圖 2.1: 文獻架構

2.1 情境感知

情境就一般性的解釋而言，它是指一個事件的相關環境或狀態。在Schilit 等學者(1994)所提出的情境感知運算系統中，首先將情境分為三大類：個人所處的地點、周圍的人以及週遭的資源。因此情境不單只是只地點而言，更代表一個持續變化的環境。若再將此分類做延伸，可以將情境定義為任何可以描繪有關個體環境特性的資訊。在使用者與應用系統的互動過程中，一些可差異化的特徵或是會影響到對話過程的資訊，都被視作情境(Abowd et al., 1999)。

一個系統若具備情境感知(context-aware)的能力，也就表示能夠使用情境資訊來提供給使用者相對的服務 (Abowd et al., 1999)。當地點、周圍所聚集的人或是當下可取得的設備或資源隨著時間產生改變時，情境感知的應用軟體就會根據這些改變來做調整與適應(Schilit et al., 1994)。Schilit (1995)將這類型系統的運作流程分為三大階段：

- 發現(discovery)

  發掘是否有資源增加或移除 (Schilit, 1995)，使用者的地點或其他屬性使否改變。當環境改變時，系統會透過偵測器(sensor)自動地偵測到情境的變化，進而在從這些偵測的資料上，發現有用的情境資訊(Chong et al., 2005)。另外Mostefaoui 等學者 (2004)與Chong 等人 (2005)並指出，從不同偵測器所得到資料屬於第一級情境(primary context)，通常都還需要再作進一步的分析及解譯才能得到有用的次級情境(secondary context)。

- 選擇(select)

  根據前一階段中所發現到的情境，透過系統中的情境推理的機制 (Chen et al., 2004)，推論出相對應的服務或資訊提供給使用者。

- 使用 (use)

  根據情境推論後的結果，系統做出回應或調整。Schilit 等人 (1994) 介紹了四種情境的使用：

  - 距離最近的選擇 (proximate selection)

    根據目前所在地，選擇最近的物件 (如 : 加油站或餐廳) 呈現在使用者介面上。

  - 自動重新配置 (automatic contextual reconfiguration)

    自動地調整組態的設定，新增或移除現有元件，已滿足目前的情境環境。

  - 情境關聯的資訊與控制 (contextual information and command)

    由於使用者的行為可以根據環境來被預測。利用這些可被預測的行為，當情境改變時系統便能即時地調整控制項目或呈現不同的資訊以供使用。

  - 以情境來驅動不同的動作 (context-triggered actions)

    透過 if-then 敘述，來定義系統應該如何做調整與適應。當偵測到的情境滿足某些條件時，開始驅動透定的動作，執行相關的活動 (Chong et al., 2005; Gwizdka, 2000)。

綜合來說，情境感知的應用系統能夠根據情境，自動地執行服務、呈現給使用者相關資訊與服務、並針對一般的訊息添加額外的情境資訊，做情境的增強 (contextual augementation)，供後續使用 (Dey, 2000)，希望達到資訊的適切性與服務的適地性 (location-based service)。

2.2 AAA 架構

AAA 架構是由 IETF 下的 AAA 工作小組 (AAA Working Group) 於 2000 年所提出，代表著由多個 AAA 伺服器所組成網路架構，而不同的伺服器

之間可以透過標準的協定進行溝通與合作。一個基本的AAA伺服器，必須要能處理客戶端的請求，首先針對使用者進行身分的驗證，決定授權的形式，給予適當的回應，最後依服務的使用情況進行計費(Laat et al., 2000)。

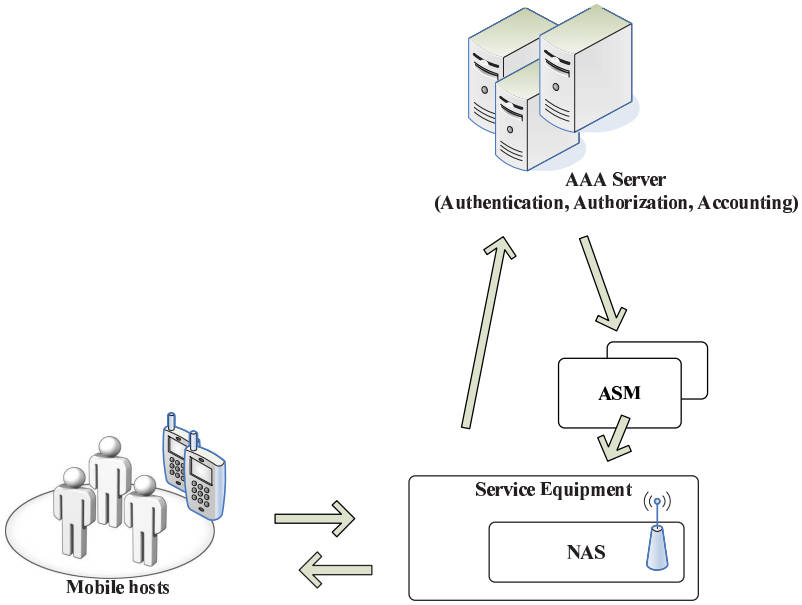

圖 2.2: AAA 架構(參考自 Generic AAA Architecture)

2.2.1 組成元件

為了提供網路上驗證、授權與計費之服務，AAA架構中必須包含下列組成元件：

AAA伺服器(AAA server)

AAA伺服器可執行驗證、授權與計費等三大服務，RFC2903 (2000)對此三項服務有以下的定義：

- 驗證(authentication)

允許終端使用者(end user)存取網路服務之前，先進行身分的檢驗。使用者必須提供有關他個人身分的資訊，如:帳號/密碼、憑證等，AAA伺服器則會將這些驗證資訊與資料庫做比對，決定要接受或拒絕網路的存取(Metz, 1999)。因此透過驗證的程序，系統可以確保只有合法的使用者才能有網路的存取權。

- 授權(authorization)

  當使用者通過身分的驗證後，AAA伺服器便開始針對使用者進行授權。系統內部建立著一套有關授權的政策，因此會限制只有滿足特定屬性或條件的使用者，才能存取特定的服務或資源(Metz, 1999; Jähnert, 2003; Prasad et al., 2004)。

- 計費(accounting)

  指蒐集、整合使用者資源或服務使用(service usage)的資訊。根據這些使用情形，可以建立帳單、審查或資源規劃等分析工作。RFC3334 (2002)提出一個以政策為基礎的計費架構(policy-based architecture)，可與AAA架構(Laat et al., 2000)整合，形成完整的驗證、授權與計費的機制。

政策儲存點**(policy repository)**

政策儲存點也就是用來存放有關授權決策與安全政策的資料庫。AAA伺服器必須根據所接收的使用者驗證資訊來存取相對應的政策、並根據決策做出授權的決定(接受/拒絕服務請求)。

特殊應用系統模組**(application specific module)**

特殊應用系統模組(ASM)也是在AAA架構中必要的元件之一，扮演AAA伺服器與特殊應用系統之間的橋樑。由於每個應用系統都有其

特殊的資料格式或參數等設定，而標準的AAA伺服器不能夠同時滿足這些多樣化的需求。因此需要透過特殊應用系統模組將已被標準化的特殊型態的資料(application specific information)，依據不同應用系統的需求，來進行資料格式上的轉換。最後根據所接收到的設定檔案，對目標資源、應用系統或是服務來設定相關參數。

網路存取點**(network access point)**

在無線通訊的環境中，行動用戶可以透過網路存取點(NAP)，也就是無線網路基地台(base station)與網路溝通。因此在AAA架構中，網路存取點會接受使用者服務請求的訊息，並負責將此訊息與相關的身分資訊傳遞給相關的AAA伺服器進行驗證。一旦身分驗證後，行動用戶便可以透過基地台來與其他伺服器進行資料封包的交換或對話。

2.2.2 AAA協定

在分散且異質型網路環境中，不同的網路服務業者會有各自的管理範圍、各自的AAA伺服器。網際網路以AAA的架構為基礎，可以整合跨網路技術與平台的規範，分散的網路元件可以互相地溝通與合作，提供使用者漫遊等服務。因此個別的AAA伺服器接受客戶端的請求後，將請求傳送到適合的AAA伺服器進行身分的驗證與授權等。最後再整合來自多個AAA伺服器有關授權的決策，來對終端使用者提供服務與資源的存取等，

因此對於網路服務業者而言，不同的業者之間可以傳遞有關使用者身份或服務使用的資訊，確保網路的安全、限制網路的使用、有效地控制網路資源的使用與分享、服務的最佳化，最後再透過計費的機制增加服務提供者的收益等(Laat et al., 2000; Jähnert, 2003)。目前實作AAA協定的有：遠端認證撥接使用者服務(Remote Access Dial-In User Service, RADIUS)與DIAMETER。

2.3 存取控制

存取控制的目的是用來限制一個合法的使用者所能執行的操作或活動。因此將存取控制與其他安全機制整合運用，可以確保資訊與系統的安全(Sandhu and Samarati, 1996)。而在網路的環境中，使用者或電腦系統等可以透過網路的便利性，即時地存取在任何地點上的資料或服務。管理者更必須要透過存取控制的機制，限制各種資源或是服務的使用權限，防止不當的存取。

Sandhu與Samarati (1994)則介紹了三種不同形式的存取控制機制：

- 自由裁決型的存取機制(Discretionary Access Control, DAC)

  資料的擁有者可以自行任意地決定資料的使用權限。此種形式雖然在執行上相當便利且具彈性，但是不能確保系統中資訊的流向。

- 強制型的存取機制(Mandatory Access Control, MAC)

  無論是使用者、資料或是物件等，都會依其屬性授予一個安全等級。因此強制型的存取機制(MAC)會依個人與物件間的安全等級來決定特定的使用權限。例如，資訊流只能由低層級至高層級(由下至上的方向)來進行移轉。

- 以角色為基礎的存取機制(Role-based Access Control, RBAC)

  有關資源的存取權是根據角色(role)而不是依個人來限制的。因此個別的使用者會先被授予不同的角色，而系統則是根據這些角色來決定是否有存取的權限(Ferraiolo and Kuhn, 1992)。