第一章 介紹

隨著網際網路的蓬勃發展，網路的使用者不斷的攀升，各種異質性的網路，如：無線網路(Wireless Lan)、電信網路(Telecommunication)與衛星網路(Satellite Network)，不斷的推陳出新，無非是希望能夠提供各種平台的使用者都能夠隨時隨地無障礙地存取網際網路。許多研究不斷的朝向將新的寬頻網路技術，如：衛星網路，整合至目前的網際網路中，希望能提供網際網路備援的頻寬，以及資料傳遞的替代路徑。

DVB-RCS 是一套支援雙向溝通的衛星網路系統，DVB-RCS 以中央控管方式做資源管理，在 DVB-RCS 系統中有許多研究議題，比如服務品質（QoS, Quality of Service）、動態頻寬機制（BoD, Bandwidth on Demand）、以及網路效能等，很多網路研究者陸續的在這議題上作探討。由於目前尚無公開且完整模擬 DVB-RCS 規格 [1] 的網路模擬器供研究者做研究，因此我們選擇在公開且高品質的 NCTUns（交大網路模擬器）[2] 作爲開發 DVB-RCS 衛星系統的模擬平台。我們在論文中針對DVB-RCS的規格來設計且實做出對應的網路協定堆（Network Protocol Stack），並經過嚴謹的功能驗證及效能測試後，終於提出來讓其他網路研究者共同研究並改善。

在本論文綱要如下：第二章，我們將介紹 DVB-RCS 規格，第三章簡介 NCTUns 及說明 DVB-RCS 在 NCTUns 上的模組設計，第四章詳細介紹每個模組的實作，第五章是模擬結果與數據驗證。最後的第六、第七章則是未來展望與結論。

第二章 背景

數位視訊廣播（DVB, Digital Video Broadcasting）目前是由 DVB 計畫組織所維護的一系列國際承認的數位電視公開標準 [3] 。這套標準在歐洲使用的也相當的普及，而在本章節中，我們遵照了歐洲電信標準協會（ETSI, European Telecommunications Standards Institute）DVB-RCS 相關的規格書，並在 NCTUns 上開發這套系統，而在往後的章節中我們也會驗證這套系統的正確性，並且提供相關的模擬數據以便供日後的參考。

而在本節內，我們會漸循式的介紹 MPEG-2 [4] 系統、DVB 及 DVB-RCS，這是由於 DVB 系統基本上是架構在 MPEG-2 系統上，並且遵循了很多 MPEG-2 的基本規格，而最後我們再引進返回通道的概念，以真正建構出整個 DVB-RCS 的架構。

2.1 規格介紹

2.1.1 MPEG2 規格

MPEG-2 是於 1994 年由 MPEG 工作團隊所發佈的視頻及音頻壓縮的國際標準。而 MPEG-2 通常用來爲廣播信號提供視頻及音頻編碼，其應用包括了數位衛星電視、有線電視等。而 MPEG-2 在經過適當的修改後，現今也成爲了 DVD 產品的核心技術。

MPEG-2 並非是對 MPEG-2 編碼器進行標準化，而是爲經過 MPEG-2 編碼的位元流提供了一種標準格式，另外一方面，它也爲 MPEG-2 解碼器提供了一個標準模式。而 MPEG-2 標準的音頻部份大致上以 MPEG-1 爲標準，因此兩者的相容性很強。這一點也使得現有的 MPEG-1 設備可對 MPEG-2 信號中相容 MPEG-1 的部份信號進行解碼，而 MPEG-2 的設備也可解碼 MPEG1 信號，從而

實現向前相容。

在經過 MPEG-2 系統編碼過後，大致上會分成以下兩種形式的資料流：

I. 傳輸流（Transport stream）

- 主要應用爲數位視訊及音訊傳輸在不可靠的傳輸媒介上。
- 通常都使用在廣播的應用上，例如 DVB。

II. 程式流（Program stream）

- 主要應用在可靠的傳輸媒介上，例如 Disk。

藉由下圖 2-1，我們將對 MPEG-2 系統作簡短的介紹。以下我們可以看到視訊資料及音訊資料是分別輸入的，並且分別作各自的編碼（此處編碼的用意就是將本來較大的檔案經由適當的編碼成較小的檔案，如此一來便可降低頻寬的浪費）產生單元傳輸流（elementary stream），接著再將各自產生的單元傳輸流作封包切割（packetize）並產生一連串的封包切割單元傳輸流（PES, Packetized Elementary Stream），而在多工（multiplexing）中可以合併多個單元傳輸流，並全部統一切割成最小大小爲 188 位元組的傳輸流封包（TS packet），並且一個個輸出傳輸流封包，進而形成傳輸流或是程式流，最後再交給實體層作通道編碼（channel coding）及調變（modulation）的處理。

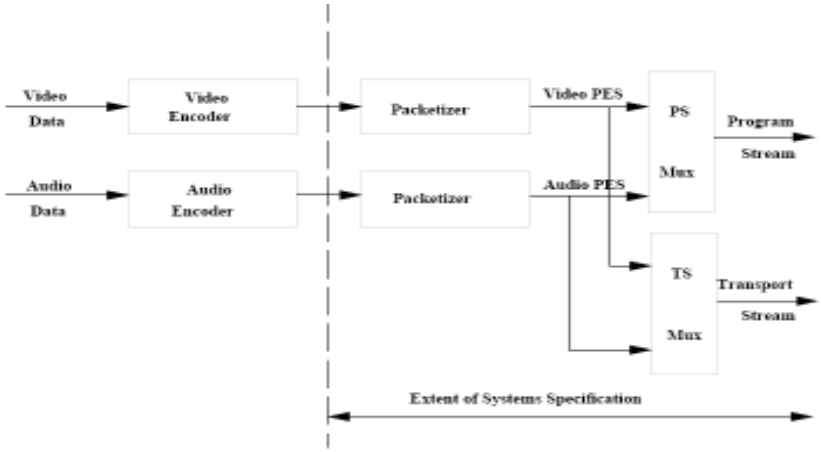

圖 2-1 視訊及音訊在經過編碼及多工處理流程圖

MPEG-2 是一個規模相當龐大的系統，我們在這裡只是稍微提到幾個重要部份，而 DVB 基本上就是架構在這樣的系統上，並且承襲了很多 MPEG-2 本有的機制，而在往後的小節中，我們也會介紹 DVB 是如何建構在 MPEG-2 之上的。

2.1.2 DVB 規格

DVB 是由歐洲電信標準協會、歐洲電子標準化組織（CENELEC, European Committee for Electrotechnical Standardization）及歐洲廣播聯盟（EBU, European Broadcasting Union）聯合組成的「聯合專家組」(JTC, Joint Technical Committee) 發起的。而這些標準定義了傳輸系統的實體層及資料鏈結層。設備通過同步並行介面（SPI, Synchronous parallel interface），同步串列介面（SSI, Synchronous serial interface），或非同步串列介面（ASI, Asynchronous serial interface）與實體層交互。而資料是以 MPEG-2 傳輸流的方式傳輸。

DVB 系統傳輸方式有以下幾種，而本章節主要是針對 DVB-S2 來做討論：

- 衛星（DVB-S 及 DVB-S2）。
- 有線（DVB-C）。

- 地面無線（DVB-T）。
- 手持地面無線（DVB-H）。

下圖 2-2 中定義出 DVB 協定的架構，針對不同的應用類型，DVB 亦定義了不同的模式，而所有資料的傳輸都是架構在 MPEG-2 TS 的系統上。而以下我們將會針對各個模式作說明：

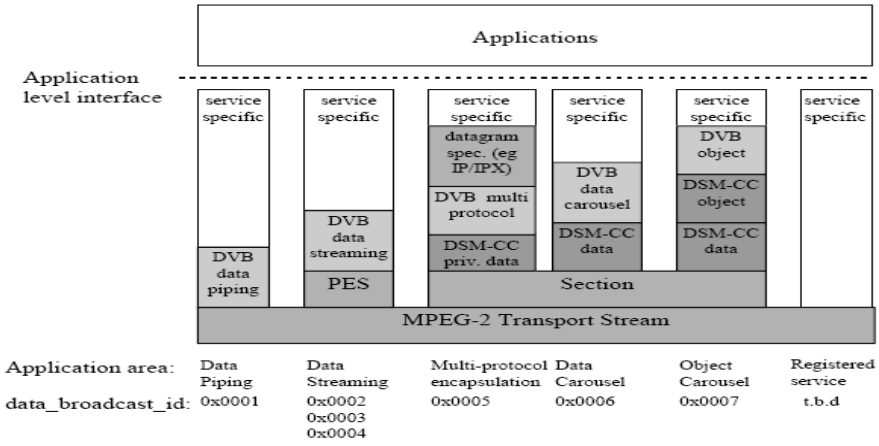

圖 2-2 DVB 協定在不同應用類型的架構圖

DVB 根據不同的應用程式類型提供了以下幾種資料廣播的模式：

- 資料管道（Data Piping）。
- 資料流（Data Streaming）。
- 多協定封裝（MPE, Multi-protocol Encapsulation）。
- 資料傳送帶（Data Carousels）。
- 對象傳送帶（Object Carousels）。

資料管道：

- 資料廣播規範中規定資料管道模式支援在 DVB 兼容的廣播網路中傳送簡單的、非同步的（asynchronous）、端對端（end-to-end）的資料。其中亦規定直接將所需廣播的資料插入到 MPEG2-TS 處理（也就是說不用再做其他的封裝及切割的動作）。

資料流：

- 資料廣播規範中規定資料流模式支援在 DVB 兼容的廣播網路中傳送資料流導向的（streaming-oriented）、端對端的、同步（synchronous）或非同步的資料。而需要廣播的資料必須做分割封包成爲分割封包單元流並插入到 MPEG2-TS 處理（也就是說會做至少一次切割的動作）。
- 而對於非同步資料流，如 RS-232 資料並沒有任何定時的要求。
- 同步資料流及同步化（synchronized）資料流均是要求定時的資料流。而同步資料流在接收端可以恢復其時脈（clock）及資料，例如 E1, T1。而同步化資料流可以實現與其他資料流的同步回放（play back in synchronization）。

多協定封裝：

- 多協定封裝模式支援使用通訊協定（例如 TCP, UDP）傳輸資料的資料廣播，而傳輸的資料都必須按照數位儲存體控制命令（DSM-CC, Digital Storage Media-Command and Control，而這部份規格是定義在 ISO/IEC 13818-6，並且符合 MPEG-2 private section 格式）格式封裝。
- 多協定封裝提供了在 MPEG2-TS 之上傳送使用其他通訊協定的一種機制。對傳送 IP 協定做了最佳化，但是也可以透過邏輯鏈結控制層／子網附著點（LLC/SNAP, Logical Link Control/Sub-network Attachment Point）封裝模式傳送其他協定的資料。它包含了單點傳播／多點傳播

／廣播（unicast/multicast/broadcast），並且利用 48 位元紀錄收端的 MAC 位址。但是由於 DVB 網路是廣播性質的，所以資料的安全是十分重要的。而封裝協定支援對資料的加密以及動態變換 MAC 位址以確保傳送數據的安全。

資料傳送帶：

- 資料傳送帶模式支援週期性的資料傳輸，且資料的長度已知並且可以從資料傳送帶中及時的更新、添加或刪除內容。而資料亦可以切割成更小的模組，或是多個模組亦可以組成一個超級模組。而在接收端，如果想要獲得特定模組中的內容，僅僅只需再等待該模組被再次廣播就可以了。
- 而資料傳送帶上的傳輸資料皆必須符合 MPEG-2 DSM-CC 中所定義的資料傳送帶模式來進行傳送。

對象傳送帶：

- 對象傳送帶模式支援需要週期性廣播 DSM-CC 用戶到用戶的資料。而對象傳送帶模式將對一組對象進行廣播，實際的目錄及內容皆存放在伺服器中，而伺服器只需要週期性的將需要廣播的對象放入 DVB 兼容的 MPEG-2 TS 中即可。

DVB-S 是使用空氣作爲傳輸媒介並使用衛星作爲轉送器（Transponder）的一套 DVB 系統。在近十年中，DVB-S 已經成爲最受喜好的數位電視廣播系統，尤其是在地域廣大的歐洲。用來取代 DVB-S 的新標準，稱爲 DVB-S2 [5]，已經在 2005 年制訂出。DVB-S2 標準引進了適應編碼調製（ACM, Adaptive Coding and Modulation），使得衛星系統的利用更爲有效率。在本論文中，我們將著重於「多

協定封裝」的討論。這是因爲我們希望能夠觀察 UDP 或是 TCP 在這樣的系統下所得到的傳輸率，而不是著重於視訊傳輸及音訊傳輸。

2.2 DVB-RCS 架構

在介紹完了 MPEG-2 及 DVB 後，我們可以發現服務都只有單向的，也就是說資料只透過正向通道（Forward Link）經由服務供應商到客戶端。而在這一節我們會引進反向通道（Return Link）的概念，也就是我們這節要介紹的數位視訊廣播-衛星反回通道（DVB-RCS, DVB-Return Channel Satellite）。

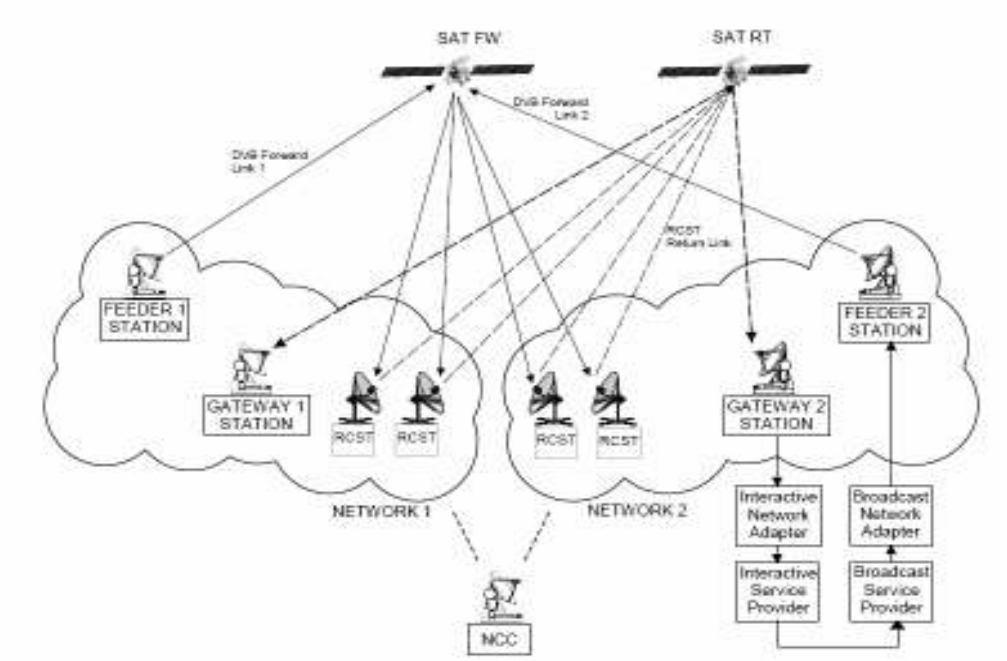

圖 2-3 DVB-RCS 系統架構圖

在上圖 2-3 中，我們先列出在 DVB-RCS 中幾個重要的節點：

- 衛星節點（Satellite）。
- 饋送者節點（Feeder）。
- 閘道節點（Gateway）。

- 網路控制中心節點（NCC, Network Central Control）。
- 服務提供商節點（SP, Service Provider）。
- 衛星地位接收站（RCST, Return Channel Satellite Terminal）。

在上圖 2-3 爲 ETSI 的 DVB-RCS 規格中互動式衛星的網路示意圖，在 DVB-RCS 的網路系統中，所有送給終端使用者的都是透過正向通道來傳送，反之由終端使用者傳回的封包都是透過反向通道來傳送，一個 DVB-RCS 的網路系統是由一個或多個子網路所組成，每個子網路都有一個饋送者用來將封包送入正向通道以及一個閘道用來將封包送入反向通道，使這個子網路能利用衛星資源。整個 DVB-RCS 網路系統的資源都是由統一的網路控制中心來統一管理，使此網路中的每一個衛星地面接收站可以分配到適當的網路資源。爲了讓 DVB-RCS 網路可以與 Internet 網路互動，網路控制中心可以將其所擁有的資源下放給服務提供者，如此一來這些服務提供者就可以利用其原本 Internet 網路的資源及 DVB-RCS 網路中的饋送者和閘道將 DVB-RCS 網路中的封包轉出到 Internet 網路或是將 Internet 網路中的封包轉入到 DVB-RCS 網路。

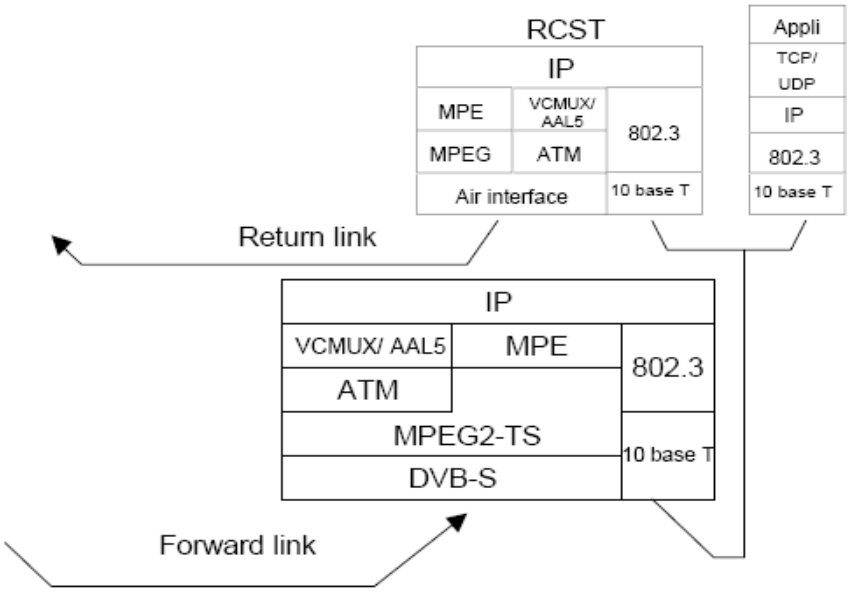

圖 2-4 正向通道及反向通道的協定堆

如果我們今天使用的是 IP 通訊協定，在前一節裡面我們有提到是利用到「多協定封裝」的模式運作。而在 DVB-RCS 的規格中有明確的定義在此通訊協定底下正向通道及反向通道的協定推。如上圖 2-4 所示，我們可以清楚的看到，資料在正向通道的傳送是利用了多協定封裝及 MPEG-2 TS，而正向通道的接收則是用相同的協定堆處理。而資料在反向通道的傳送則是利用了非同步傳輸模式（ATM, Asynchronous Transfer Mode）及多頻分時多工（MF-TDMA, Multi-Frequency Time Division Multiple Access），而反向通道的接受也是用相同的協定堆去處理。

而在以下的章節中，我們將詳細介紹 DVB-RCS 在 NCTUns 中實作部份及模擬驗證，並提供所模擬的數據及日後可能會遇到的問題及解決方法。

第三章 設計

3.1 NCTUns 協定堆疊架構

我們的衛星網路開發平台是在 NCTUns 交大網路模擬器上。NCTUns 主要是由以下三個原件組成的：simulation engine、modified kernel 及 user GUI，以下主要介紹 simulation engine。simulation engine 主要包含了許多 protocol modules，每種模組都是爲了模擬某種特定的網路行爲；然而多個協定模組（Protocol Module，下文中統稱模組）堆疊起來便可形成某種特定的網路協定（Network Protocol），我們稱之爲協定堆疊（Protocol Stack）。一般的使用者或模組開發人員都可以很輕易的加入或修改某特定的模組來達到自己想要的功能。

以下展示 802.3 協定堆疊的例子：

在這個小型的網路中，兩台主機透過 Switch 連在一起，每個主機的協定堆疊是由以下模組組成的：Interface 模組、ARP 模組、FIFO 模組、802.3 模組、PHY 模組及 LINK 模組。對於一個傳送中的封包，當主機接收到封包時，會由最下層的模組開始往上接收，直到 Interface，Interface 會把封包導入 kernel 適當的 tunnel interface，由 kernel 繼續其他 layer（TCP/IP）的處理，直到 kernel 處理完後，會再把封包導到 Interface，封包經由 Interface 下面的模組繼續模擬下面的協定。一般來說協定堆疊中，LINK 是最底層的模組，是用來連接其他網路節點的 LINK 模組，節點間藉由彼此 LINK 模組的連接，在送出封包時才能送到正確的節點上。而 Interface 是最頂層的模組，在封包還沒進到 IP layer 之前，將收進來的封包導到 kernel 的 tunnel interface。

每個模組皆有其特定的功能，例如 ARP 模組來講做兩件事情，一是把 IP 位置對應到 MAC 位置，此動作也就是一般瞭解的地址解析協議（Address Resolution Protocol）；二是把 MAC 位置填入到 Ethernet header。

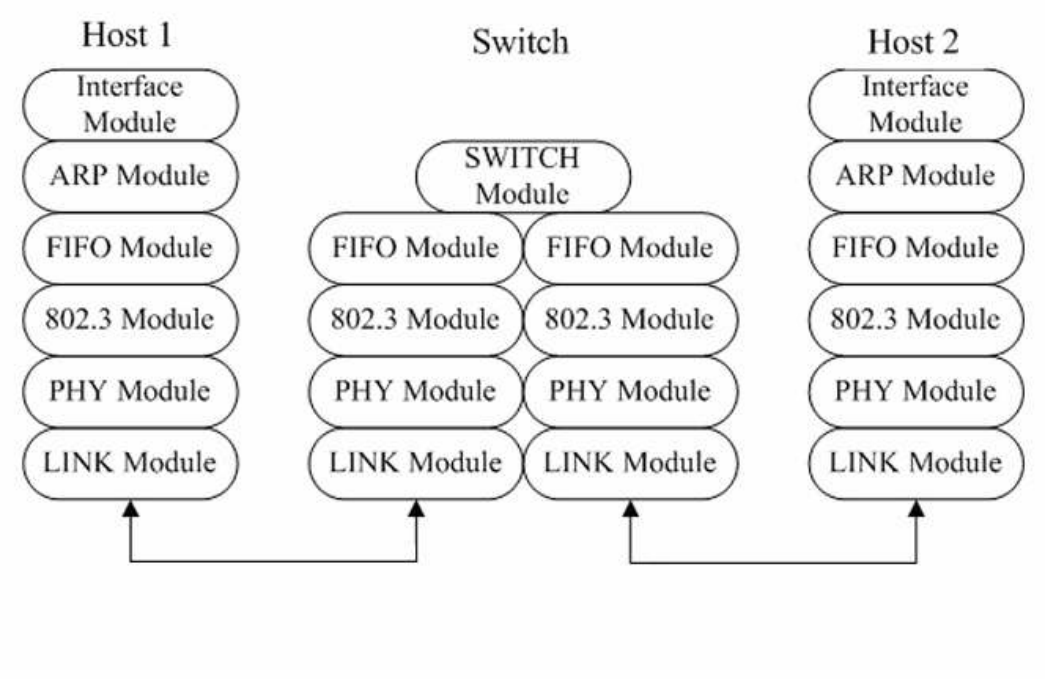

圖 3-1 Module-based platform in the NCTUns

模擬器的實作上是用 C++建立起來的，而且提供了一個基本的 prototype module NslObject。每個在模擬器中的模組在實作中都必須繼承 NslObject。圖 3-2 顯示 NslObject class 的宣告，其中最重要的兩個函式是 send()、recv()，所有模組的運作都是由此兩個函式開始的。當接收一個來自其他節點的封包時，最底層的模組（LINK）的 recv()函式會被最先呼叫到，然後 LINK 的 recv()函式會再呼叫上層模組的 recv()函式，由下往上呼叫一直到最頂層的模組。反觀於 send()函式，當一個封包從 kernel 下來要送出去，最頂層的 Interface 模組的 send()函式會被最先呼叫到，然後 Interface 模組的 send()函式會再去呼叫下層模組的 send()函式，由上往下呼叫，一直到最底層的模組。圖 3-3 顯示模組堆疊間 send()函式與 recv()的呼叫流程。

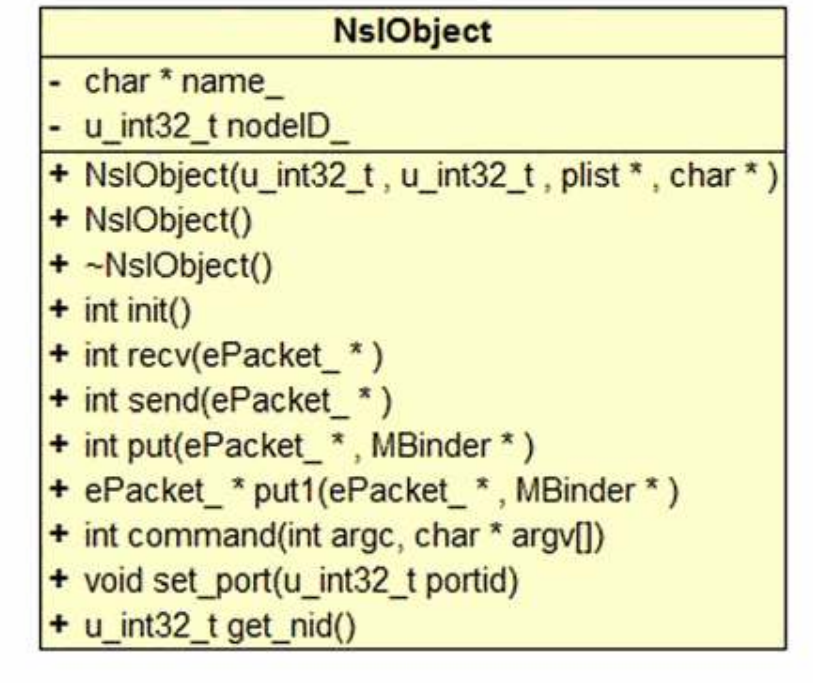

圖 3-2 NslObject class

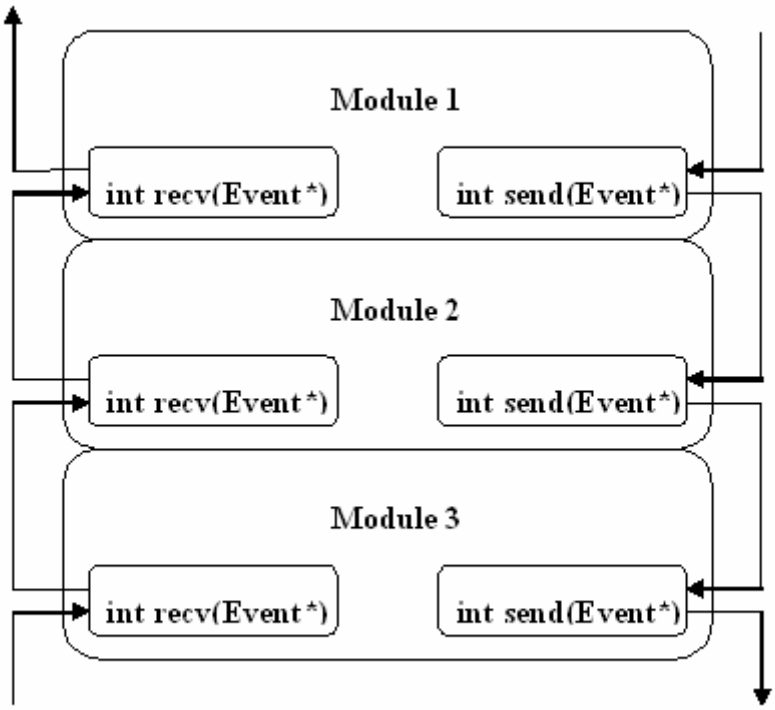

圖 3-3 模組間的封包流向架構

3.2 網路拓樸（Network Topology）

爲了在模擬器上支援衛星網路，我們新增了六種節點分別是：

- 衛星節點（Satellite，NodeID：1）：

  衛星節點接收饋送者節點由正向通道傳上來的資料，並且廣播下去給所有的衛星地位接收站；或者接收衛星地位接收站由反相通道傳過來的資料，並且傳給閘道節點。

- 饋送者節點（Feeder，NodeID：2）：

  饋送者節點接收來自網路控制中心節點及服務提供商節點傳過來的資料，將收集的資料存在緩衝區中，每過一段時會會自緩衝區中取一段固定長度的資料量送到衛星節點。

- 閘道節點（Gateway，NodeID：3）。

  接收由衛星自反相通道傳過來的資料，同時把資料傳給服務提供商節點及網路控制中心節點。

- 服務提供商節點（SP, Service Provider，NodeID：4）。

  服務提供商節點有兩組 interface，一邊是用來與 802.3 的 Internet 網路溝通，一邊是與衛星網路溝通。在衛星網路，接收來自閘道節點的資料，並將資料（控制表格或來自 Internet 的資料封包）送到饋送者節點。

- 網路控制中心節點（NCC, Network Central Control，NodeID：5）。

  此節點送出控制表格到饋送者節點，並接收閘道節點送過來的控制封包。

- 衛星地位接收站（RCST, Return Channel Satellite Terminal，NodeID：6）。衛星地位接收站有兩組 interface，一邊是用來與區域網路溝通，一邊是與衛星網路溝通。在衛星網路部分，接收來自衛星節點的資料，並將資料（控制封包或來自後方區域網路主機的資料封包）送到衛星節點。

以上的六種節點依據其 NodeID 分別可以在下圖中找到對應的圖像

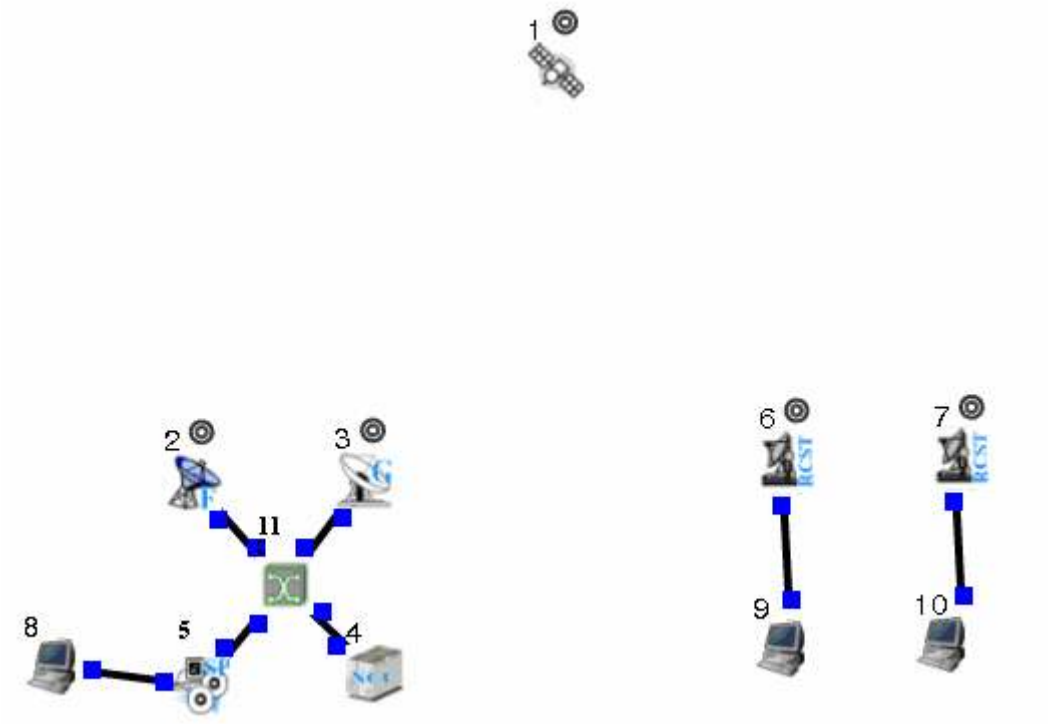

圖 3-4 DVB-RCS 衛星系統拓樸

以上的拓樸是根據圖 2.3 DVB-RCS 系統架構圖所建構出來的，爲了實作方便，在一個 DVB-RCS 系統中我們只實作一個饋送者節點及一個閘道節點。其中 NodeID：11 是虛擬 switch，以及連接虛擬 switch 的四條 link，都是爲了方便產生模擬網路描述檔（tcl 檔），虛擬 switch 在模擬過程中並沒有實際上的用途。服務提供商節點左邊有連接一個 HOST，這裡我們用 HOST 及一條連線來代表連接到一整個 Internet 網路，實際上可以是一個交換器，交換器再連接外面的 Internet

網路。衛星地位接收站右邊也連接一個 HOST，這裡也是用 HOST 來代表及一條連線來代表連接到一個區域網路，實際上可以是一個交換器，交換器再連接區域網路

3.3 DVB-RCS 節點協定堆及模組設計

在開始介紹每個節點的協定堆之前，可以先看下面的圖，這是一張整個 DVB-RCS 協定處理流程圖，在此圖中會看到許多類似的模組名稱，例如 MPEG2_TS_SP、MPEG2_TS_RCST 及 MPEG2_TS_NCC，這些原本都是 MPEG2_TS 模組，只不過後來爲了考慮到每個節點之間的獨立性以及模組的可讀性，所以將它拆開。例如 MPEG2_TS_RCST 只會用到 MPEG2_TS 模組中的 recv()函式，所以 MPEG2_TS_RCST 等於從 MPEG2_TS 模組將 send()函式抽出，而同樣 MPEG2_TS_SP 及 MPEG2_TS_NCC 只會用到 MPEG2_TS 模組中的 send()函式，所以將 recv()函式抽出。這是爲了防止後進的模組開發人員對模組函式的混淆，所作的區別，其他模組也基於相同的考量所以拆開。

除此之外，在圖上使用實線代表封包在正向通道的流向，以及使用虛線代表封包在反向通道的流向。詳細的協定堆及模組會在下面的小節介紹。

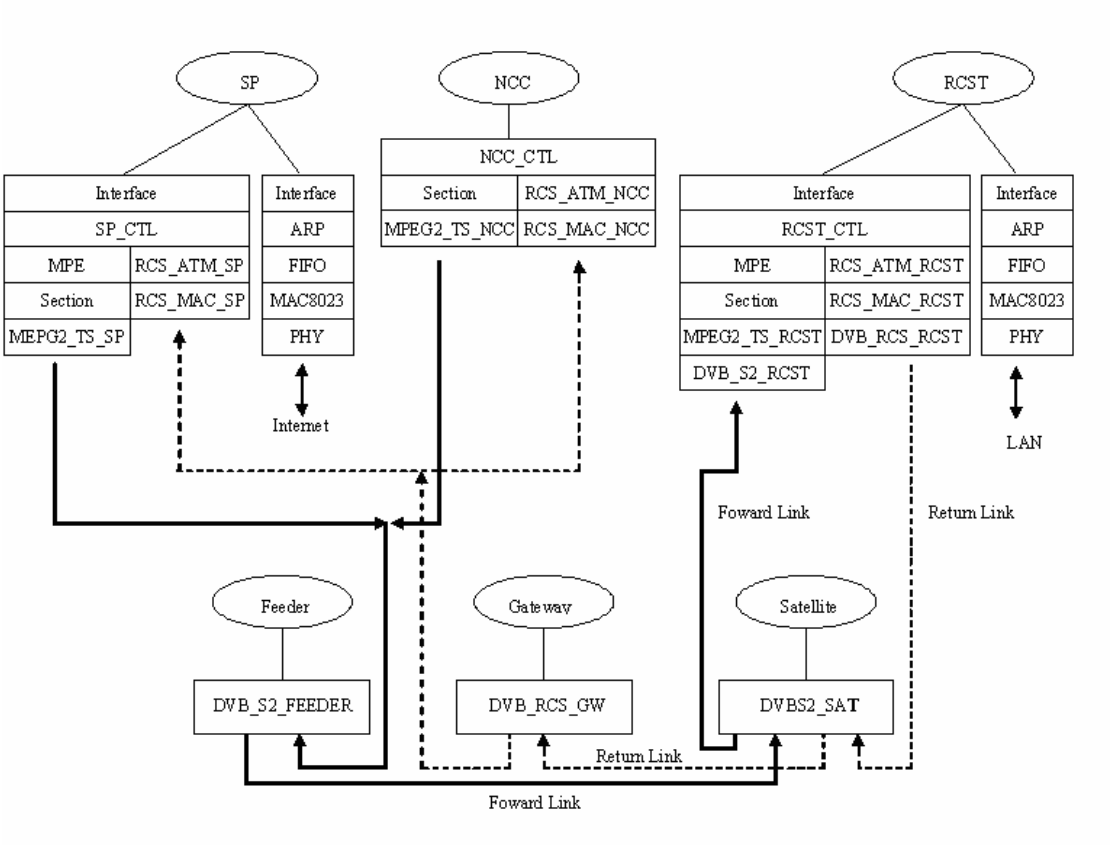

圖 3-5 協定處理流程圖

3.3.1 衛星節點（Satellite）

3.3.1.1 協定堆

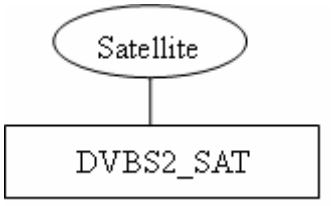

圖 3-6 衛星節點協定堆

3.3.1.2 DVBS2_SAT 模組

衛星節點用來模擬位於太空中衛星的行為，它會負責將某一個地面站所送的訊號廣播給其訊號範圍下所有的地面站。在設計上衛星節點就像是一個

repeator，收到訊號就往下廣播，而接收端在接收到了以後再自行判斷是否爲該資料要收進來或丟掉。

3.3.2 饋送者節點（Feeder）

3.3.2.1 協定堆

饋送者節點主要的功能是提供網路控制中心節點與服務提供者節點一組實體層界面，使其可利用這個界面將控制封包與資料封包經由正向通道送到衛星，這個實體層的規格定義在 ETSI EN 302 307 這份文獻中，此規格爲名稱爲 DVB-S.2，它定義了功能更大且更有彈性實體層。我們將 DVB-S.2 所定義的實體層包裝成一組模組，稱爲 DVB_S2 模組，其協定堆如下圖 3-7。

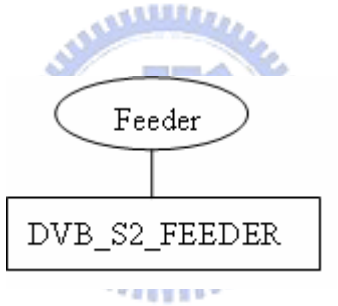

圖 3-7 饋送者節點協定堆

3.3.2.2 DVB_S2_FEEDER 模組

在饋送者節點中的 DVB_S2_FEEDER 模組有三個主要的工作：提供一組傳送界面、正向糾錯編碼（Forward Error Correction, FEC Encoding）及調變，最後再將封包送到衛星，此模組的架構如圖 3-8。

爲了簡化設計，在目前的設計下 DVB_S2_FEEDER 模組只接受 MPEG2 Transport Stream 封包，此外 DVB_S2_FEEDER 模組還支援多個界面，每一個界面都有專屬的緩衝空間，所以當網路控制中心節點或服務提供者節點想要將封包

透過 DVB_S2_FEEDER 模組傳送到衛星之前，除了要先將其包裝成 MPEG2 Transport Stream 封包的格式，還須要指定一個傳送界面。在 DVB_S2_FEEDER 模組收到一個封包之後，它會先將這個封包放入相對的緩衝區，等待一段時間後即會被發送到衛星。DVB_S2_FEEDER 模組從一開始動做後，就會一直反覆的檢查每一個介面的緩衝空間中是否有未送出的封包，若是沒有就會發送虛設訊框（Dummy Frame），反之若有等待發送的封包，則 DVB_S2_FEEDER 模組會將某個緩衝空間中所有的封包封裝成 DVB-S.2 所規定的格式，接著送到正向糾錯編碼系統做處理。

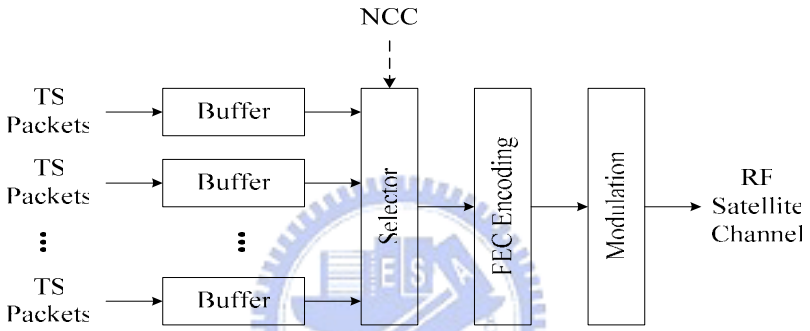

圖 3-8 DVS_S2_FEEDER 模組系統架構圖

DVB_S2_FEEDER 模組支援高彈性的正向糾錯編碼系統和調變系統，其中正向糾錯編碼系統是由 BCH 碼加 LDPC 碼（Low Density Parity Check Code）所組成，BCH 碼會依不同的編碼率（Coding Rate）和調變系統而配用 8、10 與 12 bits 的錯誤更正能力，有支援的編碼率有 1/4、1/3、2/5、1/2、3/5、2/3、3/4、4/5、5/6、8/9 與 9/10；調變系統可使用的有 QPSK、8PSK、16APSK、32APSK，另外 DVB_S2_FEEDER 模組的封包長度可以是 64800 bits 或 16200 bits，饋送者節點的管理者可以視通道狀況來改變這兩組系統的模式以及封包的長度。

3.3.3 閘道節點（Gateway）

3.3.3.1 協定堆

閘道節點的功能與饋送者節點類似，提供網路控制中心節點與服務提供者節點一組實體層界面，但是這個界面是用來接收反向通道中由衛星送下來的控制封包與資料封包，這個實體層的規格定義在 ETSI EN 301 790 這份文獻中。我們將此文獻中所定義的實體層包裝成一組模組，稱爲 DVB_RCS 模組，其協定堆如下圖 3-9。

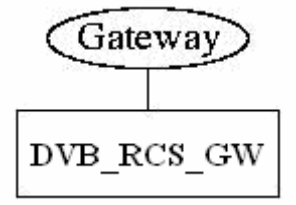

圖 3-9 閘道節點協定堆

3.3.3.2 DVB_RCS_GW 模組

閘道節點中的 DVB_RCS_GW 模組主要的工作，是將在反向通道收到由衛星送下來的封包做解調變及通道編碼解碼的動作，經過這些處理後會還原成一個 Burst，之後就會將其送到對應的網路控制中心節點以及服務提供者節點。DVB_RCS_GW 模組支援具彈性的通道編碼系統，它的通道編碼系統可使用里德所羅門碼（RS code）加上迴旋碼（Convolution Code）或是單使用渦輪碼（Turbo Code），目前我們已實作的部份是前者，然而渦輪碼已經在開發中，希望可以下一個版本能支援兩種模式的通道編碼系統。里德所羅門碼的部分使用的錯誤更正能力爲 8 bytes；編碼率可支援 1/2、2/3、3/4、5/6、7/8。另外 DVB_RCS 模組所使用的調變系統是 QPSK。

在 DVB_RCS_GW 模組在反向通道上收到訊號後，它會向網路控制中心節點詢問在這個時間點的訊號是用那一種編碼率，如此一來 DVB_RCS_GW 模組

就可以用正確的編碼率來做通道編碼解碼。

3.3.4 網路控制中心節點（NCC）

3.3.4.1 協定堆

網路控制中心節點必須週期性的透過正向通道發送控制表給所有的衛星地面接收站節點，另外一方面又必須接收由每一個衛星地面接收站節點透過反向通道所送回來的控制訊框（Frame），所以在網路控制中心節點中必須要有兩串協定堆如圖 3-10，分別處理送入正向通道的封包以及接收反向通道的訊框，在這兩串協定堆之上爲 NCC_CTL 模組，其會負責網路控制中心節點的運作。

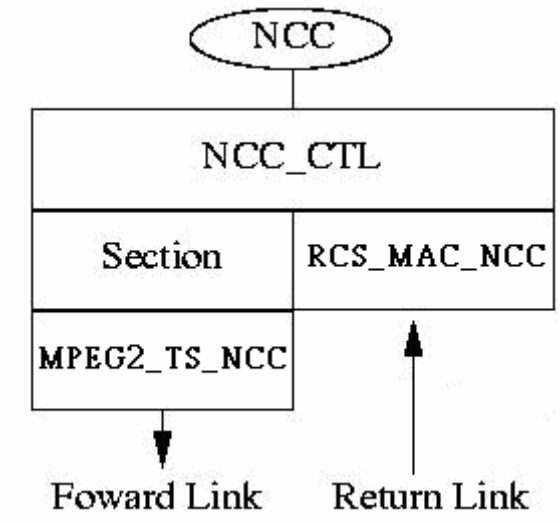

圖 3-10 網路控制中心節點協定堆

3.3.4.2 NCC_CTL 模組

網路控制中心節點主要的功能是管理個 DVB-RCS 網路中所有的資源，NCC_CTL 模組這個節點中負責這個任務的模組。

在 DVB-RCS 網路中的資源爲整個反向通道可使用的通訊頻帶，另外反向通道的存取協定是使用多頻帶分時多工協定，所以在我們的設計下，會將整個可用

的反向通道切割成多個頻帶與時槽。

除了要能管理整個網路的資源外，NCC_CTL 模組還必須視每一個衛星地面接收站的要求，依據某一個準則來分派這些資源給每一個衛星地面接收站，所以在此模組中我們使用了一個排程系統（Scheduler）來負責這個工作。

3.3.4.2.1 多頻帶分時多工

DVB-RCS 規格中是使用多頻帶分時多工的機制來切割反向通道，在這個機制中會對反向通道做三層的切割，第一次切割出的單位稱爲超級訊框；由超級訊框再切割一次所得到的單位稱爲訊框；再對訊框切割後的單位稱爲時槽。切割超級訊框及訊框之前，NCC_CTL 模組須要先定義一個基本的頻寬 w，這個值在第二層及第三層的切割時，會被拿來當作一個最小頻帶單位。

圖 3-11 爲將反向通道切割成超級訊框的示意圖，在頻域上會依照需求切割成不同大小的頻帶，每個頻帶都會有它自己的 SuperframeID；而在時域上會用一個適當的時段來切割，切割出來的每一個時槽都會用一個 SFcounter 來表示它。由 SuperframeID 與 SFcounter 就可以唯一決定一個超級訊框。

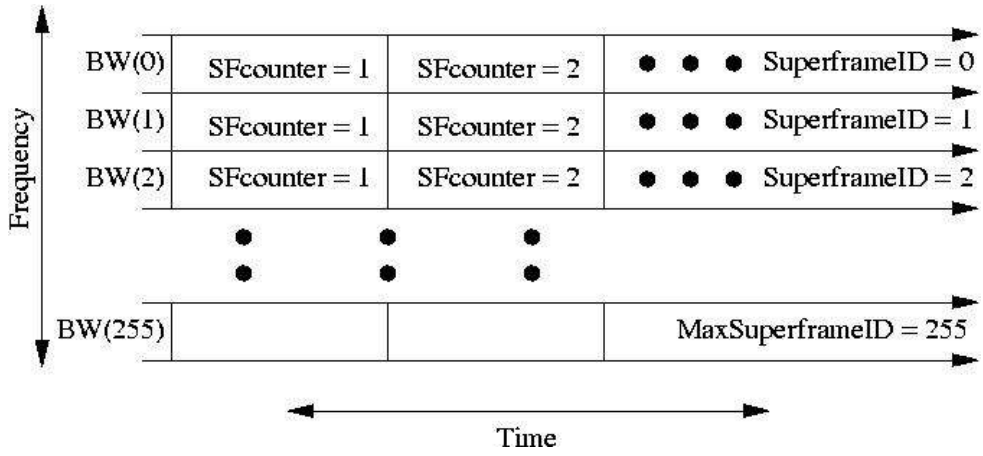

圖 3-11 多頻帶分時多工中切割反向通道成超級訊框的示意圖

切割超級訊框機制，須要先從時域切割一個超級訊框，在我們的設計下，會將它切割成 0 到 31 個時槽，用一個 5 bits 的欄位來辨識這個時槽。之後再針對每一個切割出來的時槽在頻域上切割成一個或多個單位的 w，如圖 3-12 所示，因此由此機制切出來的每一個訊框可能會占有不同大小的頻帶。

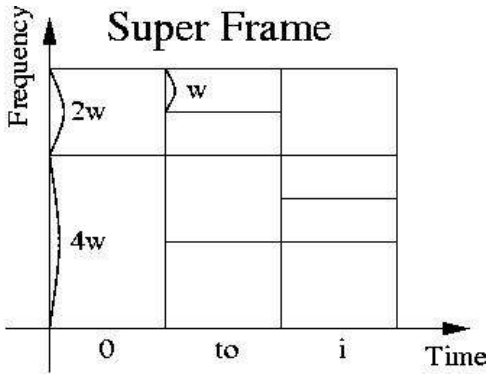

圖 3-12 多頻帶分時多工中切割超級訊框成訊框的示意圖

最後每一個訊框又可以再做一層的切割，這次的切割次序與第二層相反，必須先在頻域上將一個訊框分成一個或多個單位的 w，之後再對切出來的每一個頻帶切割，如圖 3-13 所示，最後切割出的單位即爲時槽，爲分配資源時的最小單位。

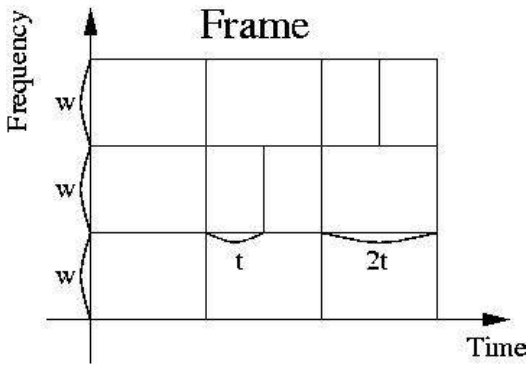

圖 3-13 多頻帶分時多工中切割訊框成時槽的示意圖

3.3.4.2.2 排程系統

爲了降低排程系統的複雜度，在這個版本中我們在以多頻帶分時多工來分割

資源時，在第二層和第三層切割時只在時域上做切割，即我們將最小頻帶單位 w 定義爲一個超級訊框所占有的頻帶，圖 3-14 爲我們第二層和第三層切割的示意圖，其中超級訊框會切割成 32 個訊框；訊框會切割成 2000 個時槽。

NCC_CTL 模組中的排程系統會擁有兩個資訊：所有衛星地面接收站可使用的資料傳送速率（Data Rate）的清單以及所有可分派的時槽，因此排程系統就可以用這些資訊算出每一個衛星地面接收站所需要的時槽個數，以滿足其所要求的資料傳送速率。

最後 NCC_CTL 模組用之前計算出的資訊來分配一個訊框中每一個衛星地面接收站可使用時槽，然後將這些訊息包裝成 SCT 控制表以及 TBTP 控制表，由正向通到送給所有的衛星地面接收站，使其可以依據這兩張控制表的內容來使用反向通道。

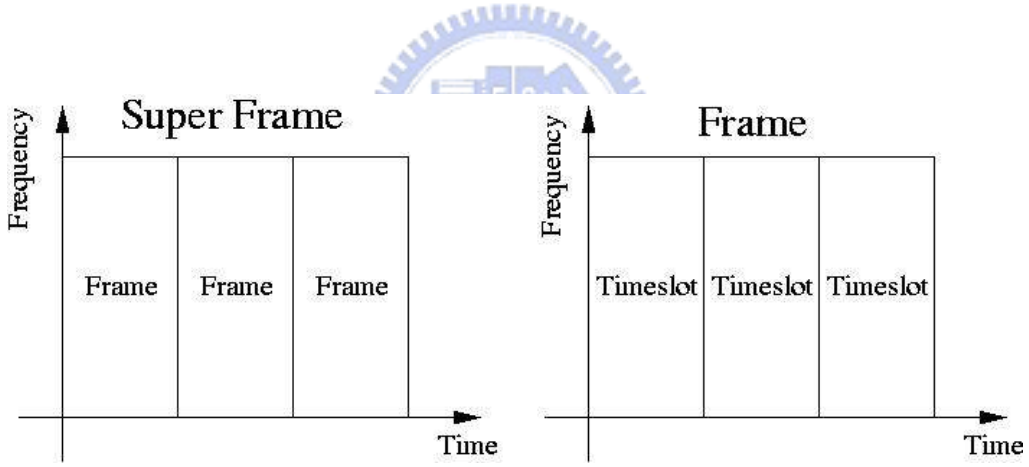

圖 3-14 多頻帶分時多工中切割訊框成時槽的示意圖

3.3.4.3 SECTION 模組

由於 NCC_CTL 模組所發送的控制表格的大小不一，可能會超過 DVB 規格中所定義的最大 Section 長度，所以在衛星地面接收站節點的 SECTION 模組會將這些控制表格包裝成一個或多個 Section（s），並加上 Section 表頭及 CRC 的資訊。

從上層收到一個控制表格時，也會同時收到這種控制表格所對應的 MPEG2 Transport Stream Packet ID，SECTION 模組在把每一個 Section 送往 MPEG2_TS_NCC 模組時，也會把這個 Packet ID 的訊息告訴 MPEG2_TS_NCC 模組。

3.3.4.4 MPEG2_TS_NCC 模組

在 DVB 規格中，在把控制表格或是資料送給饋送者之前，都要將其包裝成 MPEG2 Transport Stream 的封包，這即是 MPEG2_TS_NCC 模組的功能，它會把所有送到這個模組的 Sections，依照 ISO/IEC 13818-1 中所定義的格式，包裝成 188 bytes 的 MPEG2 Transport Stream 封包，前 4 bytes 爲 MPEG2 Transport Stream 的表頭，其中會有一個欄位爲 Packet ID，這個欄位的會填入由 SECTION 模組所帶下來的 Packet ID。包裝完成後就會將這個封包送到指定的饋送者節點。

3.3.4.5 RCS_MAC_NCC 模組

由於反向通道是使用多頻帶分時多工的存取機制，這個模組要能模擬同時接收多個頻帶訊號的行爲，因此當收到從閘道節點送過來的訊號時，它會模擬將這個訊號中每一個頻帶的訊號取出，並給與每一個頻帶各一個獨立的計時器來計算接收時間。爲了模擬碰撞的情況，在接收時間結束之前若是又收到具相同頻帶的訊號，則該頻帶的訊號就會被破壞，必須將其丟棄；反之在接收時間結束之前都沒有收到同頻帶的訊號，則在接收時間結束時就會將這個 Burst 轉換回控制封包。

3.3.5 服務提供商節點（SP）

3.3.5.1 協定堆

服務提供者節點主要是要處理 DVB-RCS 網路與 Internet 網路之間的溝通，

所以它的協定堆分爲兩部分，分別是 DVB-RCS 網路的協定堆和 Internet 網路的協定堆如圖 3-15，因爲 Internet 網路的協定堆並非這次計劃的重點，故在本文獻中不做詳細介紹。

在這兩組協定堆的上方都有一個 Interface 模組，這個模組是 NCTUns 中用來將封包導進或導出 Linux Kernel 的介面，如此一來在這個節點所需要支援的路由功能就可以直接借用 Linux Kernel 中原本的路由機制。

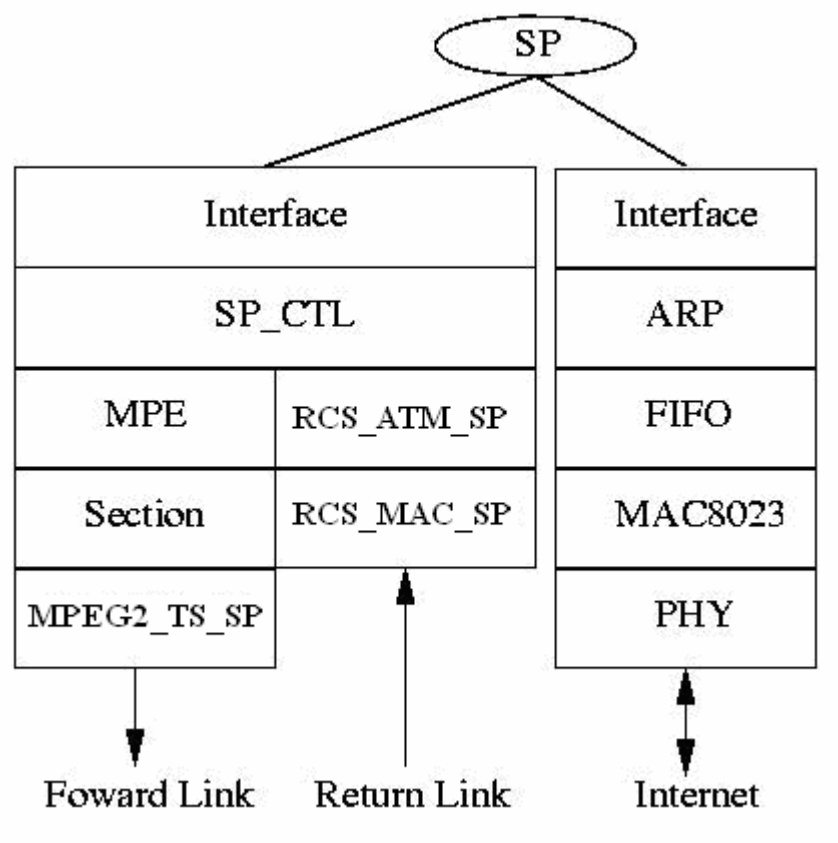

圖 3-15 服務提供者節點協定堆

3.3.5.2 SP_CTL 模組

SP_CTL 模組與 NCC_CTL 模組類似，在它之下掛有兩串不同的協定堆，分別是將控制表格或是資料發送到正向通道，另一個則是接收反向通道上面的資料封包，將其接收後送往 Interface 模組，使該封包能被路由到正確的目的。

在正向通道的這個部分，SP_CTL 模組在一啓動時會由設定檔取得所有衛星

地面接收站節點後方的 IP 位址，所以當 SP_CTL 模組由 Interface 模組收到一個 IP 封包時，它能由該封包的目的位址查表決定這個封包應該要送到那一個衛星地面接收站節點，並決定這個 IP 封包在封裝成 MPEG2 Transport Stream 封包後的 Packet ID 爲多少，這些訊息 SP_CTL 模組會用 PAT、PMA、NIT 以及 INT 這四種控制表格週期性的透過正像通到送給所有的衛星地面接收站節點並將這個由 Interface 模組收到的 IP 封包送給 MPE 模組。

另外一方面，當 SP_CTL 模組在回饋通道這串協定堆收到一個資料封包時，它會直接把它轉收到 Interface 模組，這樣一來 Linux Kernel 就會依照預先設定好的路由表來轉送這個封包，若是這個封包的目的位址爲 Internet 網路中的某一台主機，則這個封包就會被轉到服務提供者節點中對應到 Internet 網路的那串協定堆；若是目的位址爲 DVB-RCS 網路中某台衛星地面接收站節點後方的主機，則這個封包就會被轉到服務提供者節點中對應到 DVB-RCS 網路的那串協定堆，之後就會依照上一段所敘述的機制把這個封包送到特定的衛星地面接收站節點。

3.3.5.3 MPE 模組

MPE 模組主要功能是把資料封裝成 DSM-CC 格式封裝。在這裡主要會接收到兩種封包格式，資料封包及控制表格封包。由於控制表格封包原本已經符合 DSM-CC 格式，所以只需要對資料封包作格式封裝。

3.3.5.4 SECTION 模組

服務提供者節點與網路控制中心節點的 SECTION 模組幾乎是相同的，唯一的差別是在服務提供者節點中的 SECTION 模組若收到已經封裝成 Datagram Section 的封包就不會對其做任何處理，直接把封包送往 MPEG2_TS_SP 模組。

3.3.5.5 MPEG2_TS_SP 模組

服務提供者節點與網路控制中心節點的 MPEG2_TS_NCC 模組是相同的，故在此不再贅述。

3.3.5.6 RCS_ATM_SP 模組

RCS_ATM_SP 模組在此主要功能是將收到的 ATM 封包組合成一個 IP 封包，當這個模組成功的組合出一個 IP 封包，則就會把這個 IP 封包送往 RCS_MAC_SP 模組。

3.3.5.7 RCS_MAC_SP 模組

由於反向通道是使用多頻帶分時多工的存取機制，這個模組要能模擬同時接收多個頻帶訊號的行爲，因此這當收到從閘道節點送過來的訊號時，它會模擬將這個訊號中每一個頻帶的訊號取出，並給與每一個頻帶各一個獨立的計時器來計算接收時間。爲了模擬碰撞的情況，在接收時間結束之前若是又收到具相同頻帶的訊號，則該頻帶的訊號就會被破壞，必須將其丟棄；反之在接收時間結束之前都沒有收到同頻帶的訊號，則在接收時間結束時就會將這個 Burst 轉換回 ATM 封包送給 RCS_ATM_RCST 模組。

3.3.6 衛星地面接收站節點（RCST）

3.3.6.1 協定堆

爲了讓衛星地面接收站節點可以被一台以上的主機使用，所以我們將這個節點設計爲一個第三層的節點，讓衛星地面接收站節點後面可以接一個區域網路（Local Area Network），所以它就必須要有路由功能，使得 DVB-RCS 網路與節

點後方的區域網路可以溝通。

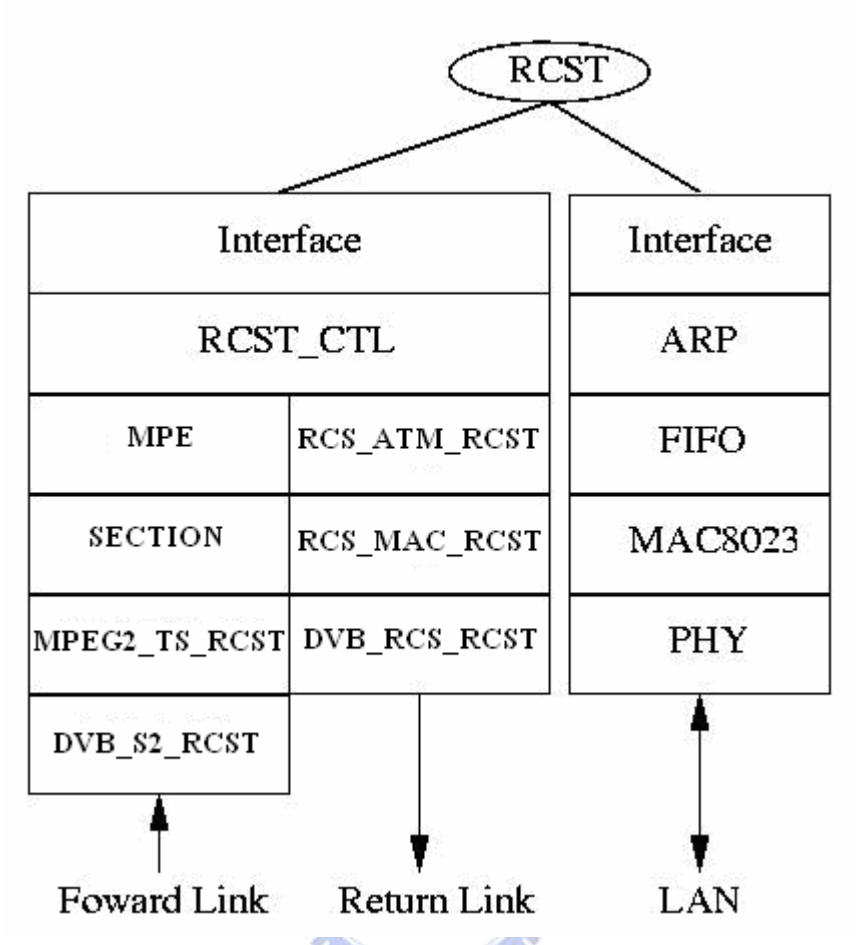

圖 3-16 衛星地面接收站節點協定堆

因此衛星地面接收站節點採用與服務提供者節點類似的協定堆如圖 3-16，不同的地方在於衛星地面接收站節點是接收正向通道的封包而透過反向通道來傳送它的資料封包，這點與服務提供者節點正好相反。

3.3.6.2 RCST_CTL 模組

RCST_CTL 模組之下掛有兩串不同的協定堆，一串是用來接收正向通道中的控制表格或是資料封包，另一串則是用來處理欲透過反向通道發送的資料封包。RCST_CTL 模組主要有幾個工作：

1. 收集服務提供商節點送過來的四種控制表格，藉由分析這四種表格得知它必須接收資料封包的 Packet ID。因爲對每個衛星地面接收站節點來說都是不斷接收衛星節點所廣播下來的封包，所以衛星地面接收站節點必須自己判斷哪些封包才是給自己的。
2. 收集網路控制中心節點送過來的四種控制表格，藉由分析這四種表格得知它所得到哪些衛星資源，或者是說在未來的時間中，有哪些時槽是屬於它的。有了這些資訊後，透過與 RCS_MAC_RCST 模組中的 Queue Manager 溝通，才能正確地將時槽分配給 RCS_MAC_RCST 模組來傳送資料。
3. 透過與 RCS_MAC_RCST 模組中的 Queue Manager 溝通，得知下層 Queue 的需求後，發送需求封包給網路控制中心節點要求時槽。

3.3.6.3 MPE 模組

當 MPE 模組收到 SECTION 模組的封包時，會先檢查這個封包的內容是否爲控制表格，若是則直到送到 RCST_CTL 模組，否則收到的應該是一個以 Datagram Section 格式封裝的封包，MPE 模組會用 Datagram Section 的表頭的資訊將這個封包還原成 IP 封包，將其送往 RCST_CTL 模組。

3.3.6.4 SECTION 模組

當 SECTION 模組收到 MPEG2_TS_RCST 模組送上來的 Section 時，會先檢查這個 Section 的內容是否爲一個 Datagram Section，若是則直到送到 MPE 模組，否則由這個封包的表頭取得這個 Section 的類別，之後會把這個 Section 放到該類別的緩衝空間中。

在 SECTION 模組中緩衝空間都會對應到一個特定類別的 Section，用來暫存

那些還無法組成某一種控制表格的 Section（s），在收到 Datagram Section 以外的 Section 時，會嘗試著將那些在這種 Section 的緩衝空間中的 Section 組合一個控制表格，當一個控制表格成功的被組合回來時，SECTION 模組就會把這個控制表格送往 MPE 模組。

3.3.6.5 MPEG2_TS_RCST 模組

MPEG2_TS_RCST 模組在此主要是把收進來的 MPEG2 Transport Stream 封包組成 Section，把完整的 Section 送往 SECTION 模組。目前設計中，不支援亂序的 MPEG2 Transport Stream 封包，也就是說當一個 Section 切成數個 MPEG2 Transport Stream 封包送出，若 MPEG2_TS_RCST 模組無法依序接收那些數個 MPEG2 Transport Stream 封包時，則丟棄所有的數個 MPEG2 Transport Stream 封包，重新從第一個數個 MPEG2 Transport Stream 封包開接收。

3.3.6.6 DVB_S2_RCST 模組

DVB_S2_RCST 模組主要的工作是處理由衛星經由正向通道發送的訊號，這些訊息必須經過解調變、正向糾錯解碼與解多工（De-multiplexing）的處理，最後在把處理後的封包與其所對應的界面送到 MPEG2_TS_RCST 模組。

3.3.6.7 RCS_ATM_RCST 模組

RCS_ATM_RCST 模組在衛星地面接收站節點中有兩個工作，分別是切割封包與包裝封包。把上層送下來的 IP 封包先加上 8-byte 的表尾（trailer），然後以 48-byte 爲一單位作切割並封裝成 53-byte 的 ATM cell。

3.3.6.8 RCS_MAC_RCST 模組

RCS_MAC_RCST 模組的工作是要把一個或多個 ATM 封包封裝成一個 Burst，讓它能夠透過反向通道送上衛星。在 RCS_MAC_RCST 模組中有一個緩衝空間，在這個緩衝空間中沒有任何 ATM 封包時，RCS_MAC_RCST 模組會處於閒置的狀態，直到有 ATM 封包由 RCS_ATM_RCST 模組送下來，就會觸動 RCS_MAC_RCST 模組的傳送機制。

3.3.6.9 DVB_RCS_RCST 模組

衛星地面接收站節點的 DVB_RCS_RCST 模組的功能與閘道節點中的相對稱，它會對 DVB_MAC_RCST 模組送下來的 Burst 做通道編碼與調變的動作，詳細的敘述請參照閘道節點的 DVB_RCS_SP 模組章節，最後會將處理過後的封包透過反向通道送上衛星。

第四章　實作

4.1　衛星節點（Satellite）

4.1.1　DVBS2_SAT 模組

DVBS2_SAT 模組在實作上就是一個 repeator，將收到的封包往下面廣播，因此在 recv()函式中，收到封包除了做紀錄之外，就是馬上去呼叫 send()函式將此封包往下送。而在往下送的時候會去判斷收到封包的類型作不同方向的廣播，目前會有兩種類型：

1. PKT_DVB_S2，此種類型是代表此封包是從正向通道傳過來的，也就是從饋送者節點傳過來的，所以此封包的目的地是衛星下面的衛星地位接收站。因此衛星在傳送此種類型的封包時會去判斷目前要去傳送的節點是否屬於衛星地位接收站，是屬於衛星地位接收站才去傳送，否則不傳。
2. PKT_DVBRCS，此種類型是代表此封包是從反向通道傳過來的，也就是從衛星地位接收站傳過來的，所以此封包的目的地是衛星下面的閘道節點。因此衛星在傳送此種類型的封包時會去判斷目前要去傳送的節點是否屬於閘道節點，是屬於閘道節點才去傳送，否則不傳。

4.2　饋送者節點（Feeder）

4.2.1　DVB_S2_FEEDER 模組

這個模組是屬於 physical layer 的，所以主要是作循環冗餘校驗（CRC Check）、位元攪亂（Bit Scramble）及編碼（Encoding）。

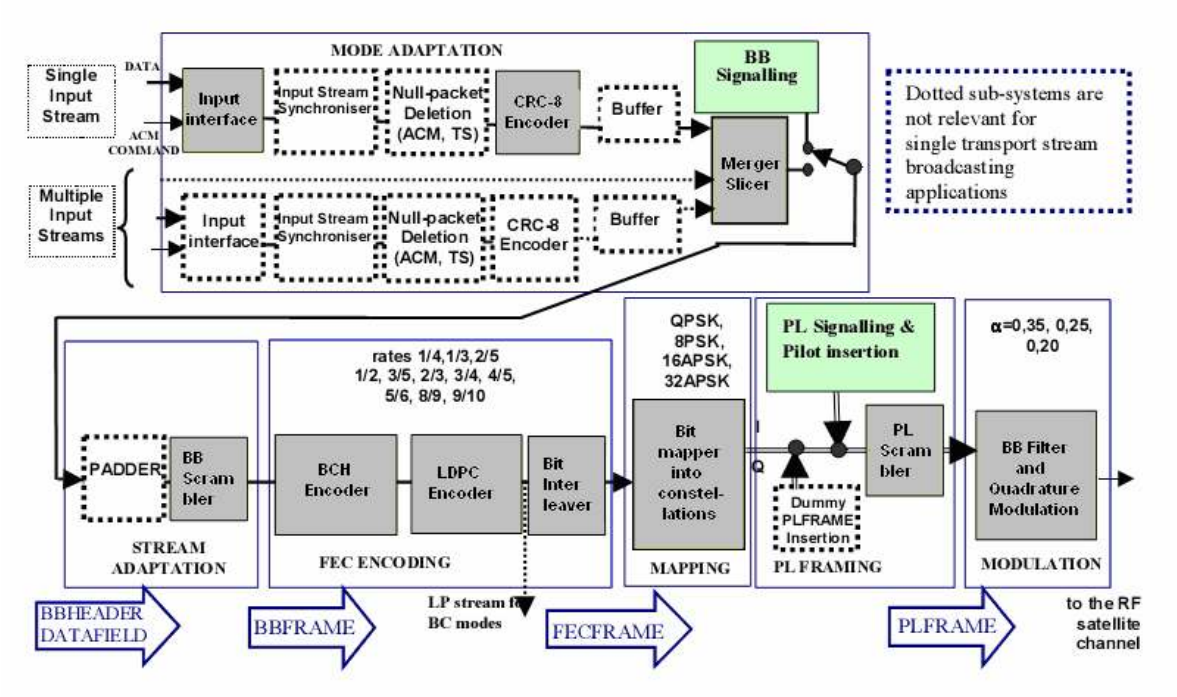

圖 4-1 DVB-S2 的功能流程圖

在圖 4-1 我們可以清楚的看到在 DVB_S2_FEEDER 模組內做的事情，灰色區塊是我們有實作出來的部份，白色及綠色是其他沒有實作的功能。其主要的工作是前面的 CRC-8 Encoder、Bit Scramble 及 FEC Coding，後面的 Modulation 只是在模擬一個 symbol 帶幾個 bit 而已。以下簡單介紹灰色區塊的實作：

- Input Interface：存放進來 DVB_S2_FEEDER 還沒送出去的封包。

- CRC Check：在規格中有規定使用的位元數及生成多項式，所以我們採用 CRC-8，生成多項是爲： $X^8 + X^7 + X^6 + X^4 + X^2 + 1$。其演算法可參考網路上有的現成程式碼可套用。

- Merger Slicer：每一次從不同的 interface 抓取資料出來，作爲 BBFRAME 的 Data Payload，並且加上 10 bytes 的表頭。

- Bit Scrambler：此功能是把原本的資料作攪亂的動作，主要是爲了防止資料內有連續一連串的”1”或者一連串的”0”。如果有一連串的”1”或”0”，這樣對編碼來說是比較不利的。而位元攪亂的程式碼是根據規格上所給的硬體實作方式來完成程式碼的。

- FEC Encoding：在這裡需要兩種編碼 Outer encoding (BCH)、Inner encoding (LDPC)，編碼所需要的編碼參數表及 BCH 多項式在規格中都有提供，詳細的程式碼在網路上也有現成的可以套用，在此不多做介紹。

- Bit Interleaver：把編碼完畢的 BBFRAME 作 bit 交錯，以減少連串錯誤帶來的影響。。作法是準備一個 M x N 的位元矩陣（M、N 視調變方式而定，M x N = 64800 or 16200），然後把資料依序 column-wise 的方式寫入矩陣，再把矩陣的資料依序 row-wise 的方式讀出來，讀出來的資料便是一個完整的 FECFRAME。

- Bit mapper into constellation：在這裡我們只有模擬 FECFRAME 調變厚轉成 symbol 後所需的傳輸時間，所以不同的調變方式算出來的傳輸時間便不同。

4.3 閘道節點（Gateway）

4.3.1 DVB_RCS_GW 模組

當封包經由反向通道到達閘道節點時，只會呼叫到 DVB_RCS_GW 模組的 recv()函式。在 recv()函式中根據收到的封包類型作不同的解碼處理，目前主要支援兩種封包格式，資料封包與控制封包兩種，根據規格的定義，控制封包目前只做 CRC 確認，而資料封包除了作 CRC 確認還要作里德所羅門碼加上迴旋碼的解

碼過程。解碼過程結束後，再分別呼叫服務提供商節點及網路控制中心節點的接收函式。

4.4 網路控制中心節點（NCC）

4.4.1 NCC_CTL 模組

NCC_CTL 模組主要的功能是管理 DVB-RCS 網路中所有的資源，所以 NCC_CTL 會定期的廣播以下四種控制表格到衛星地位接收站，讓衛星地位接收站知道有哪些時槽是屬於這個衛星地位接收站的，並且在何時可以傳送資料到頻帶上。

SCT（Superframe Composition Table）：

此表格會帶有一個 superframe 的基本資訊，比如說 superframe_id、superframe_start_time、superframe_duration、superframe_counter 以及帶有哪些 frame 等，主要功能是清楚的描述一段 superframe 的資訊。

FCT（Frame Composition Table）

此表格會帶有 frame 的基本資訊，從此表格內可以知道此 FCT 內的 frame 是隸屬於哪個 superframe，此表格會帶有一連串的 frame 的資訊，每個 frame 有 frame_id、frame_duration 以及 timeslot 的資訊。收到此表格後，對整個 superframe 又有了更詳細的瞭解。

TCT（Timeslot Composition Table）

此表格中帶有一個 superframe 中所有時槽種類的基本資訊，在目前的設計中，只有兩種時槽種類，分別是資料時槽（Data Timeslot）及需求時槽（Request

Timeslot），其中衛星地位接收站是用資料時槽來傳送資料的，而用需求時槽來發需求要下一次的資料時槽。

由於這三種表格是有時效性的，這三種表格代表的是一段 superframe 時間的整體架構，所以 NCC_CTL 模組必須在時間軸還沒執行到這個 superframe 時就提早送出這三種表格，而 NCC_CTL 模組是在程式起來後定期每隔一段 superframe 時間送一次這三種表格。

TBTP（Terminial Burst Time Plan）

此表格用來分配一個 superframe 中的時槽給衛星地位接收站，依照衛星地位接收站對時槽的需求去做配置。在規格中允許一份 TBTP 表格帶有一整個 superframe 中所有時槽的配置，其中保含了資料時槽及需求時槽。而在我們的程式中，我們 TBTP 表格一次僅針對一個 frame 中的時槽做配置，因此此種表格是每隔一段 frame 時間送一次。

NCC_CTL 在送端會定期送出以上的四種控制表格讓衛星地位接收站知道在未來的時間軸上時槽的分配，而在收端主要爲接受 SAC（Satellite Access Control）的表格，此種表格是衛星地位接收站用來發需求用的。衛星地位接收站若有需要時槽來傳送資料，則將自己對時槽的需求量填在 SAC 表格中，並在需求時槽時發送 SAC 的表格到網路控制中心節點。而 NCC_CTL 在收到了 SAC 表格後，便作判斷是否接受此需求量，若接受的話並在下次分配時槽時，滿足每個衛星地位接收站的需求量。

4.4.2 SECTION 模組

此模組在實作部分架構很簡潔，在送端主要工作是把上層模組傳下來的控制

表格作切割，切割成 section 的格式；並且在收端把收到的 section 組回成控制表格往上層模組送。而每個控制表格的格式不同所以切割方式也不同，因此我們根據每個控制表格寫各自的切割 section 的函式，在 SECTION 模組送端的時候，只要呼叫此各自控制表格切割 section 的函式就可以將控制表格切成 section 往下傳。同樣地，每個控制表格也會有各自的將 section 組回成控制表格的函式，在 SECTION 模組的收端，也只要呼叫各自控制表格的組 section 的函式就可以組回控制表格了。

所以每個控制表格必須提供切割 section 的函式與將 section 組回控制表格的函式，例如 FCT 表格必須提供 fct_table_to_section()與 to_table()的兩個函式分別爲切割 section 與組回控制表格的函式。每種控制表格的此兩種函式寫法都不同，過程也很繁瑣，但其主要工作是將結構複雜的控制表格作記憶體搬移的動作，將不連續的記憶體區塊搬移到連續的記憶體空間，在此便不贅述。對 SECTION 模組來說只需要知道在送端呼叫切割 section 的函式，在收端呼叫組回表格的函式這樣就可以了。

4.4.3 MPEG2_TS_NCC 模組

在網路控制中心節點中，此模組只用到 send()函式，因此在 MPEG2_TS_NCC 中的 recv()函式是直接 assert(0)出來的。在 send 的函式中，主要功能是把上層送下來的 section 包裝成 188 bytes 的 MPEG2 Transport Stream（簡稱 MPEG2_TS）封包。在實做上，會將接收到的 section 的封包長度取出，算出要切成幾個 MPEG2_TS 的封包

4.4.4 RCS_MAC_NCC 模組

在網路控制中心節點中，此模組只有用到 recv()函式，主要做下列兩件事情：

1. 模擬接收多個頻帶的資料以及模擬碰撞發生
2. 將接收到的 Burst 轉成 ATM Cell 傳到 RCS_ATM 模組。

爲了完成第一點，在程式實作中用 timer 來實現這個機制。每當收到一個 Burst 進來，則去看目前的 timer list 中察看目前使用的頻帶中，是否有存在其他的 timer：

- 若沒有，則表示此時沒有其他封包正在接收，然後建立一個新的 timer，並且將 timer 的終止時間設爲此封包的接收時間。假設 timer 的終止時間到了，則此 timer 會去呼叫另一個函式，將 Burst 轉成 ATM Cell 傳到 RCS_ATM 模組，並且把在 timer list 中終止時間到了的 timer 拿掉，以代表此封包已經成功被 RCS_MAC_NCC 接收，因此 timer list 中的所有 timer 代表的是目前有幾個封包正在被 RCS_MAC_NCC 接收中。

- 若有，則表示此時進來的封包與之前尚未接收完畢的封包發生碰撞，此時將進來的封包丟掉，並且將找到的 timer 的終止時間延長，以模擬接收碰撞封包的接收時間。

爲了完成第二點，這裡的工作比較簡單，只是將原來的 Burst 切割成 ATM Cell，所以將 Burst 依據 ATM Cell 的長度（53 bytes）去切割，由於在送端封裝 ATM Cell 成 Burst 的時候並沒有加其他額外的 header，所以在這裡切割的時候也需要考慮其他額外的 header，只需要依據 ATM Cell 長度去作切割。切割完的 ATM Cell 再一個個往上層的 NCC_CTL 模組傳送。

4.5 服務提供商節點（SP）

4.5.1 SP_CTL 模組

在前面的第三章設計章節，提過 SP_CTL 模組會定期的送四種控制表格給衛星地位接收站，因此在實作方面，程式裡面有四種分別爲_PAT_circleq、Pmt_circleq、Nit_circleq 及 Int_circleq 指標型態的變數，這四種控制表格是爲了讓衛星地位接收站知道在衛星節點不斷廣播下來的東西中，有哪些是屬於這個衛星地位接收站的，並且會定期每隔一段 superframe 時間送一次這四種表格。

4.5.2 MPE 模組

當封包由 SP_CTL 模組送下來時，有可能是控制表格或是 IP 封包，如果是控制表格 MPE 模組會直接將控制表格送給 SECTION 模組做處理，然而若送下來的是 IP 封包，則 MPE 模組會將這個封包依照 DVB 規格的定義，加上特定的表頭以及 CRC，將其封裝成 Datagram Section 的格式，以便 MPEG2_TS 模組可以再進一步的把這個封包封裝成 MPEG2 Transport Stream 封包。

從上層收到 IP 封包時，也會同時收到當這個 IP 封包要被包裝成 MPEG2 Transport Stream 封包時的 Packet ID，MPE 模組在把 Datagram Section 送往 SECTION 模組時，也會把這個 Packet ID 的訊息一起送下去。

4.5.3 SECTION 模組

此模組在網路控制中心節點的模組介紹已經介紹過，在此不再贅述。

4.5.4 MPEG2_TS_SP 模組

在 SP 節點中，此模組只有用到 send()函式，其功能行爲與 MPEG2_TS_RCST 相同，請參考 MPEG2_TS_RCST，在此不再贅述。

4.5.5 RCS_ATM_SP 模組

RCS_ATM 模組收到 RCS_MAC 模組送上來的 ATM 封包時，它會判斷其是否爲控制用的 ATM 封包，若是則將其丟棄；否則就會將此 ATM 封包放到 RCS_ATM 模組中的緩衝空間。

這個緩衝空間的用途是暫存那些還無法組成一個 IP 封包的 ATM 封包，在 RCS_ATM 模組收到一個非控制用的 ATM 封包時，都會去檢查該 ATM 封包表頭的資訊，由 ATM 表頭可以知道這個 ATM 封包是否爲這個欲組合之 IP 封包的最後一個 ATM 封包，若是則會嘗試著將這些在緩衝空間中的 ATM 封包組合成一個 IP 封包，當這個模組成功的組合出一個 IP 封包，則就會把這個 IP 封包送往 SP_CTL 模組。

4.5.6 RCS_MAC_SP 模組

在服務提供商節點節點中，此模組只有用到 recv()函式，其功能行爲與 RCS_MAC_NCC 相同，請參考 RCS_MAC_NCC，在此不再贅述。

4.6 衛星地面接收站節點（RCST）

4.6.1 RCST_CTL 模組

爲了方便敘述，我們可以將 RCST_CTL 模組概略分成以下三部分論：

4.6.1.1 控制表格管理

RCST_CTL 模組爲了要能讓衛星地面接收站節點使用 DVB-RCS 網路，所以它必須要接收並處理正向通道的控制表格，因此在 RCST_CTL 模組中有一組控制表格管理系統，在這個系統中，每一種收到的控制表格都會被存放在我們設計的資料結構中。

在 RCST_CTL 模組在收到 MPE 模組送上的封包，判斷這個封包是帶有控制表格之後，RCST_CTL 模組會先由表格的表頭取得此控制表格的類別，並準備將這個控制表格存到該類別的資料結構中。在要把控制表格存入所對應類別的資料結構之前，RCST_CTL 模組會檢查這個控制表格是否爲更新的版本，若是則會將這個新版的表格存放在一個新的資料結構中，準備將來所有表格接更新後可以動態的瞭解 DVB-RCS 網路的狀況。

爲了要讓 RCST_CTL 模組在接受資料封包時，可以知道帶有那些 Packet ID 的 MPEG2 Transport Stream 封包是該衛星地面接收站節點要接收的，它必須要分析由服務提供者節點所送過來的控制表格，即 PAT、PMT、NIT 和 INT，藉由分析這幾種表格，RCST_CTL 模組可以得知它必須接收的 Packet ID，而這種查表的動作，會在這四種控制表格皆收齊之後被觸發，之後在每次這些控制表格有更新的時後也會重新在做查表的動作。

當衛星地面接收站節點要使用反向通道時，必須先知道反向通道的時槽有那些是分派給這個節點的，要得到這個訊息必須要分析由網路控制中心節點所送過來的控制表格，即 SCT、FCT、TCT 和 TBTP，如此一來 RCST_CTL 模組可知反向通道的特性、可以使用的時槽以及在 RCS_MAC 模組傳送時一個 Burst 所包含的 ATM 封包個數，這些資訊都保存在 RCST_CTL 模組中。

4.6.1.2 資料封包收送

RCST_CTL 模組處理由正向通道送上來的資料封包的機制如上一小節所敘述，在其接收到 MPE 模組送上來的 IP 封包時，會先檢查這個封包的 Packet ID 是否爲送給這個衛星地面接收站節點的封包，若不是則將其丟棄，反之就表示這個 IP 封包是送給此衛星地面接收站節點後方區域網路中的某一台主機，所以 RCST_CTL 模組會將這個 IP 封包轉收到 Interface 模組，讓這個封包依照設定好的路由表來轉送，在這裡路由封包的方法與之前服務提供者節點的小節的機制相同。

當有 IP 封包由 Interface 模組送下來時，表示節點後方區域網路中的某一台主機有封包要藉由衛星地面接收站節點送出，這時候 RCST_CTL 模組會將這個 IP 封包送給 RCS_ATM_RCST 模組，準備由反向通道送出。

4.6.1.3 發送需求封包

RCST_CTL 模組會透過與 RCS_MAC_RCST 模組中的 Queue Manager 溝通，得知下層 Queue 的需求後，發送需求封包給網路控制中心節點要求時槽。在此發送需求封包與分配時槽的問題也是實作的一大重點之一，因爲 RCS_MAC_RCST 模組中的 Queue 有四種之多，每種 Queue 的特性不同，要求時槽及分配時槽的方式也不一樣，因此在此不做詳述。

4.6.2 MPE 模組

此模組在服務提供商節點節點的模組介紹已經介紹過，在此不再贅述。

4.6.3 SECTION 模組

此模組在網路控制中心節點的模組介紹已經介紹過，在此不再贅述。

4.6.4 MPEG2_TS_RCST 模組

MPEG2_TS_RCST 模組收到 DVB_S2_RCST 模組送上來的 MPEG2 Transport Stream 封包時，它會由該封包的表頭取得 Packet ID，之後會把這個封包放到對應到該 Packet ID 的緩衝空間中。

在衛星地面接收站節點中的 MPEG2_TS_RCST 模組，每一個 Packet ID 都會有一個對應的緩衝空間，其用途是暫存那些還無法組成一個 Section 的 MPEG2 Transport Stream 封包，在 MPEG2_TS_RCST 模組收到每一個 MPEG2 Transport Stream 封包時，都會由該封包表頭的取出 Packet ID，嘗試將這個封包與那些在緩衝空間中，有相同 Packet ID 的 MPEG2 Transport Stream 封包組合成一個 Section，當一個 Section 成功的組合回來時，MPEG2_TS_RCST 模組就會把這個 Section 送往 SECTION 模組。

4.6.5 DVB_S2_RCST 模組

DVB_S2_RCST 模組收到正向通道上的訊號後，會先分析同部訊號（Preamble），由同部訊號中可以得知這個訊號的格式，包含調變與正向糾錯碼類型、訊框長度和指示區塊（Pilot Block）。若收到的是一個虛設訊框，則將其丟棄，反之則要依據由同部訊號所得到的資訊來做相對應的處理。

解調變與正向糾錯解碼的部分與饋送者節點中 DVB_S2_FEEDER 模組的部分相對稱。在正向糾錯解碼中 LDPC 碼本身的特性，會因爲解碼過程中解碼階段（Decoding Phase）重複次數的不同，而會有不同的解碼能力，但是在真實世界中會因爲重複次數的增加而使解碼的運算時間延長，但是在我們的模擬中，解碼的運算時間是不做計算的，這部分在重複次數很多的設定下，可能會與真實世界有所出入，但是一般真實世界中爲了良好的系統反應時間，解碼階段的重複次數

都不高，所以在我們的設計中，忽略這個部分的模擬是可以接受的。在 BCH 碼的部份，可更正錯誤的能力會依照設定而有 8、10 與 12 bits 的差別。

在 DVB_S2_RCST 模組完成解調變與正向糾錯解碼處理後會得到原始的資料訊匡，即可由此訊框的標頭中取得該訊框的長度和一些用來把此訊框還原成 MPEG2 Transport Stream 封包的訊息，之後 DVB_S2_RCST 模組就會將這個訊框依照表頭中的訊息切割成數個 MPEG2 Transport Stream 封包並且送往 MPEG2_TS_RCST 模組，這部份的動作即爲解多工的動作。在把 MPEG2 Transport Stream 封包送給 MPEG2_TS_RCST 模組前會依照訊框的標頭中的界面欄位的值，將這個封包相對應的介面告訴 MPEG2_TS_RCST 模組。

4.6.6 RCS_ATM_RCST 模組

在收到 RCST_CTL 模組送下來的 IP 封包後，RCS_ATM_RCST 模組會依照 AAL5 規格中的定義在這個 IP 封包後面加上一個 AAL5 的表尾，其中包含這個 IP 封包原始的長度以及 CRC32 的訊息，如圖 4-2。之後把這段資料填滿成 48 bytes 的倍數，並以每 48 bytes 爲一個單位切割這個封包。在切割完成後，每一個 48 bytes 的封包會依據 ATM 規格中的定義加上 5 bytes 的 ATM 表頭，如圖 4-3，在其中有一個 bit 是用來表示該 ATM 封包是否爲這一串 ATM 封包的最後一個，另外 CRC8 的訊息也會在 ATM 表頭中。

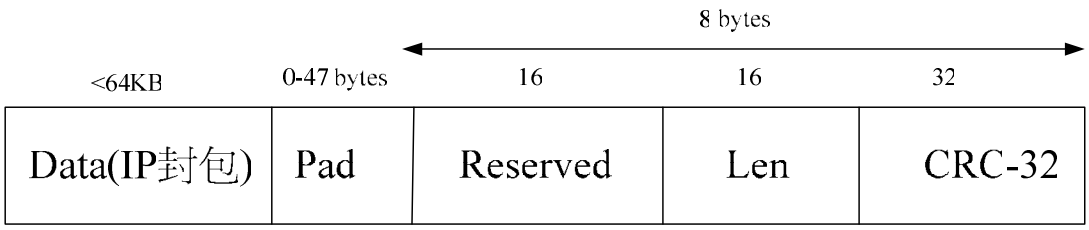

圖 4-2 （AAL5）ATM Adaptation Layer 5 封包格式

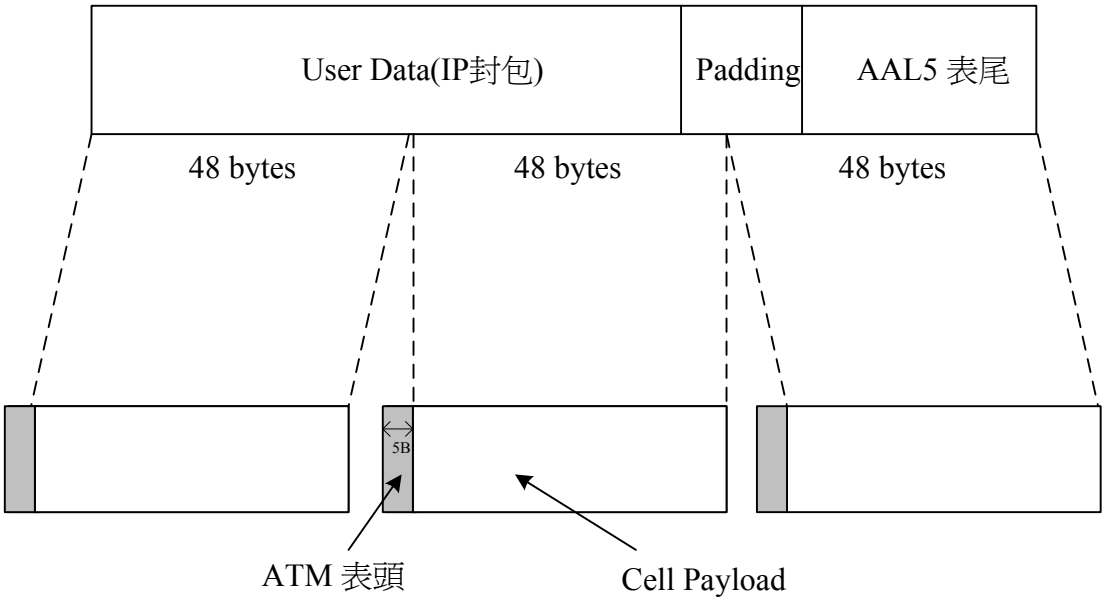

圖 4-3 AAL5 的切割與封裝

4.6.7 RCS_MAC_RCST 模組

RCS_MAC_RCST 模組會先向 RCST_CTL 模組取得一個 Burst 所包含的 ATM 封包個數，這可能是 1、2 或 4 個 ATM 封包，然後再向 RCST_CTL 模組取得反向通道的特性與可以使用的時槽，有了這些資訊後 RCS_MAC_RCST 模組就可以由緩衝空間中取出適當數量的 ATM 封包，在這些封包集合的前面加上同部訊號之後即爲一個 Burst，最後等到剛剛取得之時槽的時間點到來，就可以把這個 Burst 送到 DVB_RCS_RCST 模組做實體層的處理。

在一個 Burst 傳送結束時，RCS_MAC_RCST 模組會檢查緩衝空間，若是裡面還有未送出的 ATM 封包，就再進行一次傳送的程序，若是該緩衝空間已經沒有任何待傳送的 ATM 封包，那 RCS_MAC_RCST 模組會回到閒置的狀態。

當 RCS_MAC_RCST 模組向 RCST_CTL 模組要求可以使用的時槽時，若是拿不到任何的時槽，RCS_MAC_RCST 模組就會等待一個訊框的時間之後向 RCST_CTL 模組再次要求。這個動作會一直重複，直到取得可用的時槽爲止。

此模組在實做時，需要有一塊緩衝空間來存放要傳送出去的封包，我們在這裡實

作了四種在規格中所定義的 Queue，執行時可以依據不同流量樣式來導入到不同的 QUEUE。RCS_MAC_RCST 模組透過 Queue Manager 來管理所有的 Queue，RCST_CTL 也是透過 Queue manager 來與 QUEUE 溝通

4.6.8 DVB_RCS_RCST 模組

DVB_MAC_RCST 模組會在 RCST_CTL 模組取得可以使用的時槽時，也會得知這些時槽所應該要使用的編碼率，這個訊息在 DVB_MAC_RCST 模組把 Burst 送到 DVB_RCS_RCST 模組時，也會把這個 Burst 所使用的編碼率一同帶給 DVB_RCS_RCST 模組，所以 DVB_RCS_RCST 模組就可以用這個資訊以正確的編碼率來做通道編碼。

第五章 模擬結果

5.1 正向通道

5.1.1 參數設定

在正向通道主要的參數：

UDP Payload Size：1472 bytes

此封包大小是指應用程式送下來的資料大小，未包含 UDP 及 IP header

Symbol Rate：16 Mboud

正向通道的 symbol rate。

Modulation Type：16APSK

調變方式影響到的是每個 symbol 可以表示多少個 bit，若採用 16APSK 則表示每個 symbol 可表示出 4bits 的資料量。

LDPC Code Rate：9/10

底層的編碼率會影響到整個通道可用總頻寬中，被標頭（header）或是其他額外資料（overhead）所佔據的比例，9/10 的編碼率代表每 10 個 bits 資料有 9bit 是真正會被使用的資料。

FECFRAME Size：Normal（8100 bytes）

此大小爲在做完FEC編碼後的FECFRAME大小，規格中定義有normal（8100

bytes）、short（2025 bytes）。

5.1.2 傳輸率理論推算

以下的理論推導是根據上述的參數設定來作推導，在正向通道中，從應用程式開始送下來的資料一直到加表頭及切割成 MPEG2_TS 封包的流程圖可參考下圖 5-1。

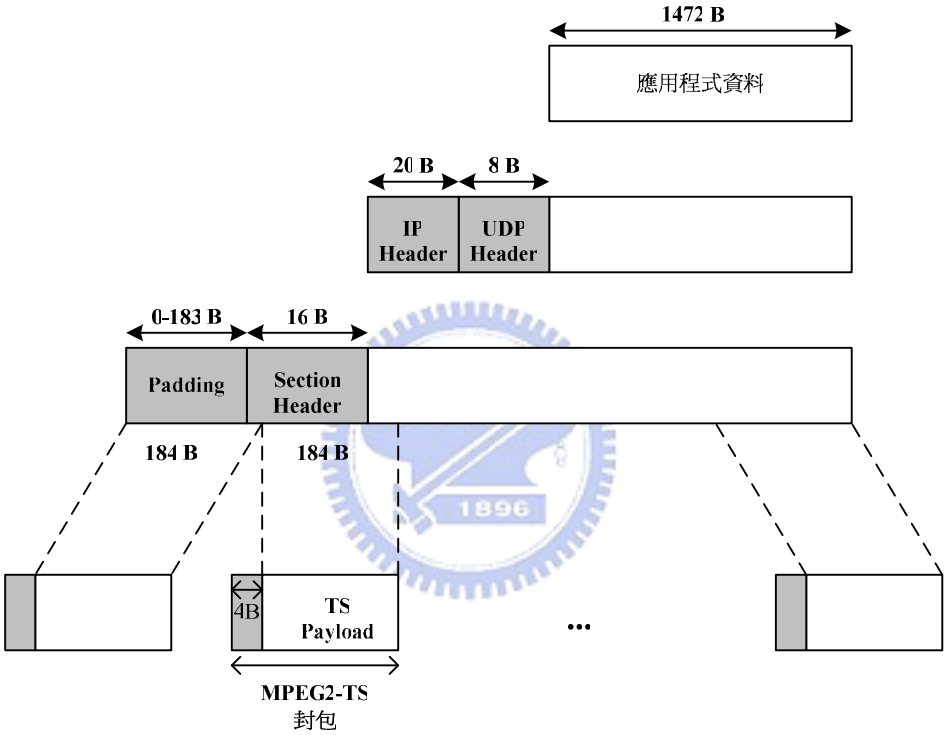

圖 5-1 正向通道應用程式資料切割圖 1

在正向通道方向資料從應用程式送下來 1472 bytes 的資料，到了 UDP/IP 以後加上 28 bytes 的表頭。資料從 Interface 往下層模組傳送至 MPE 模組時會加上 Section 的標頭 16Bytes，再往下送到 MPEG2_TS_SP 模組進行切割成 184Bytes 爲單位的資料並加上 4Bytes 的 MPEG2-TS 標頭，因此每 1500Bytes 的 IP 封包會有 16Bytes 的 MPE 標頭資料，封包大小爲 1516Bytes，MPEG2-TS 切割成 9 個 MPEG2-TS 封包，作了 140Bytes 的 Padding，每個 MPEG2-TS 封包有 4Bytes 標

頭資料，所以額外資料或是標頭資料佔據的資料量如下：

表頭額外資料：28 + 16 + 140 + 9 x 4 = 220Bytes。

UDP Payload 所佔比例：1472 / (1472+220) = 87%

因此到 MPEG2_TS_SP 這層模組時總資料量就已經有 13%的比例是屬於封包標頭或是填補的資料，真正 UDP Payload 所佔比例爲 87%。

從 MPEG2_TS Stream 開始取出資料作爲 BBFRAME Payload 開始，經過中間的編碼、調變等，一直到組成一個完整的 PLFRAME，這中間的加標頭切割的動作可以參考下圖 5-2。

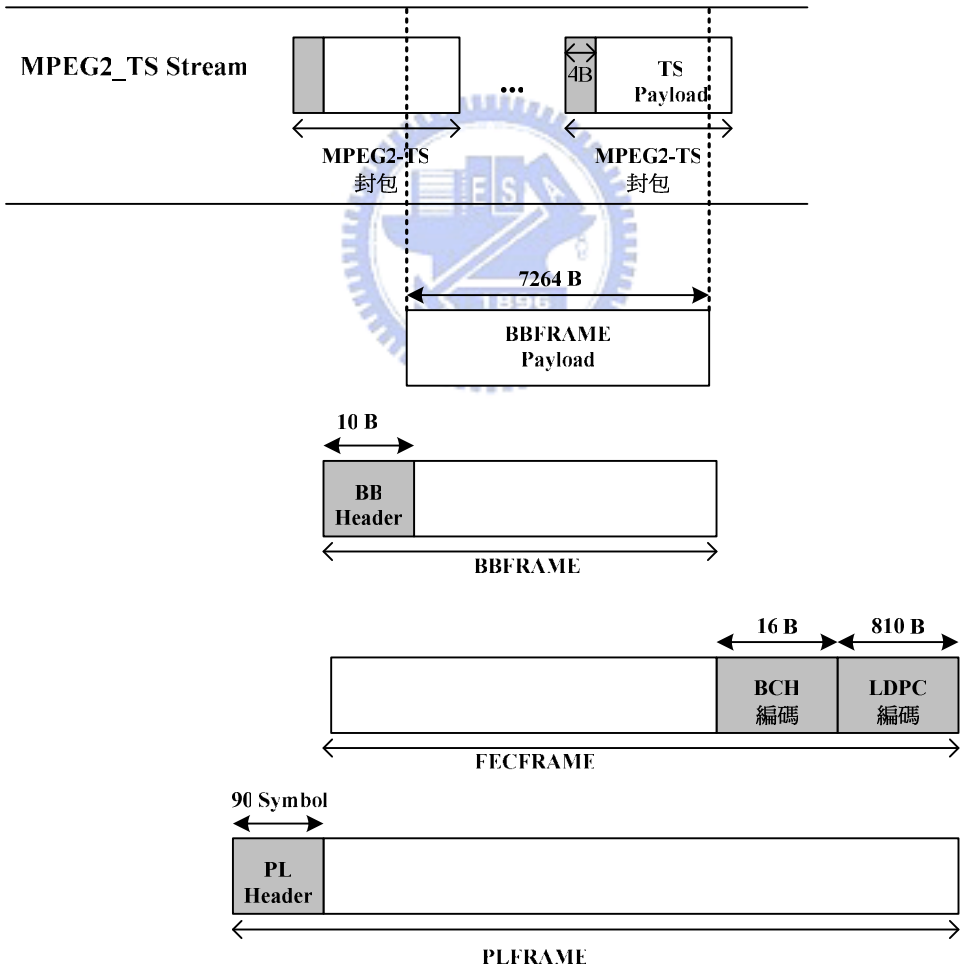

圖 5-2 正向通道應用程式資料切割圖 2

接著在底下實體層會根據編碼率及調變方式來算出應該取出多少 bit 的 MPEG2-TS 封包資料量來做編碼。根據查表得知必須取出 7264 bytes 的 BBFRAME Payload 來作編碼。在編碼之前會先加上 10 bytes 的 BBFRAME Header。經過 BCH、LDPC 編碼後得到 FECFRAME，資料長度爲 64800 bits（8100 bytes）。FECFRAME 在送出之前要先轉換成 Symbol，所以經過調變後封裝成 PLFRAME，並且增加 90 Symbol 的 PLFRAME Header 才會送到通道上，因此以編碼率 9/10 及 16APSK 的調變方式的話會得到下面的額外資料：

*[10*8 + (8100 – 10 - 7264)*8]/4 + 90 = 1762 (Symbol)*

BBFRAME Payload 所佔比例：14528 / (14528+1762) = 89.18%

因符號傳輸率爲 16MBaud 加上 16APSK 的調變方式，因此得到資料傳輸率爲 64Mbits，

1. 考慮圖 5-1 的額外表頭資料，使得通道真正可以用來傳輸資料剩下 87%
2. 考慮圖 5-2 的額外表頭、編碼資料，使得通道真正可以用來傳輸資料剩下 89.18%

因此相乘會得到下面的最高資料傳輸率：

*64Mbits * 87% * 89.18% = 49.655Mbits*

根據以上的理論推導得到應有的傳輸率爲 49.655Mbits，而在我們的模擬數據中的傳輸率爲 49.391Mbits，已經相當接近。其餘不同的調變方式搭配不同編碼方式的模擬設定的理論及模擬數據都在表 5-1 中。

Normal FECFRAME		
Modulation -Coding	**理論傳輸率 (Mbits/s)**	**模擬傳輸率 (Mbits/s)**
QPSK-1/4	6.824	6.562
QPSK-1/3	9.208	9.009
QPSK-2/5	10.988	10.726
QPSK-1/2	13.765	13.503
QPSK-3/5	16.541	16.278
QPSK-2/3	18.405	18.142
QPSK-3/4	20.705	20.443
QPSK-4/5	22.093	21.830
QPSK-5/6	23.032	22.770
QPSK-8/9	24.589	24.325
QPSK-9/10	24.897	24.633
8PSK-3/5	24.777	24.514
8PSK-2/3	27.570	27.306
8PSK-3/4	31.015	30.751
8PSK-5/6	34.501	34.237
8PSK-8/9	36.832	36.567
8PSK-9/10	37.294	37.030
16APSK-3/4	41.296	41.030
16APSK-4/5	44.064	43.799
16APSK-5/6	45.937	45.672
16APSK-8/9	49.041	48.776
16APSK-9/10	49.660	49.391
32APSK-3/4	51.549	51.282
32APSK-4/5	55.004	54.737
32APSK-5/6	57.342	57.075
32APSK-8/9	61.217	60.949
32APSK-9/10	61.984	61.717

表 5-1 正向通道理論及模擬數據表

5.2 反向通道

5.2.1 參數設定

在反向通道主要的參數：

UDP Payload Size：1472 bytes

符號傳輸率：9.935 MBaud

編碼率：1/2

調變方式：QPSK

Preamble Length：16 Symbols

Guard Time： $2*10^{-7}$

每個 burst 的 ATM cell 個數：2

每個 frame 的 traffic burst 個數：99

每個 frame 的 request burst 個數：1

5.2.2 傳輸率理論推算

以下的理論推導是根據上述的參數設定來作推導，在反向通道中，從應用程式開始送下來的資料一直到加表頭及切割成 ATM Cell MPEG2_TS 封包的流程圖可參考下圖 5-3。

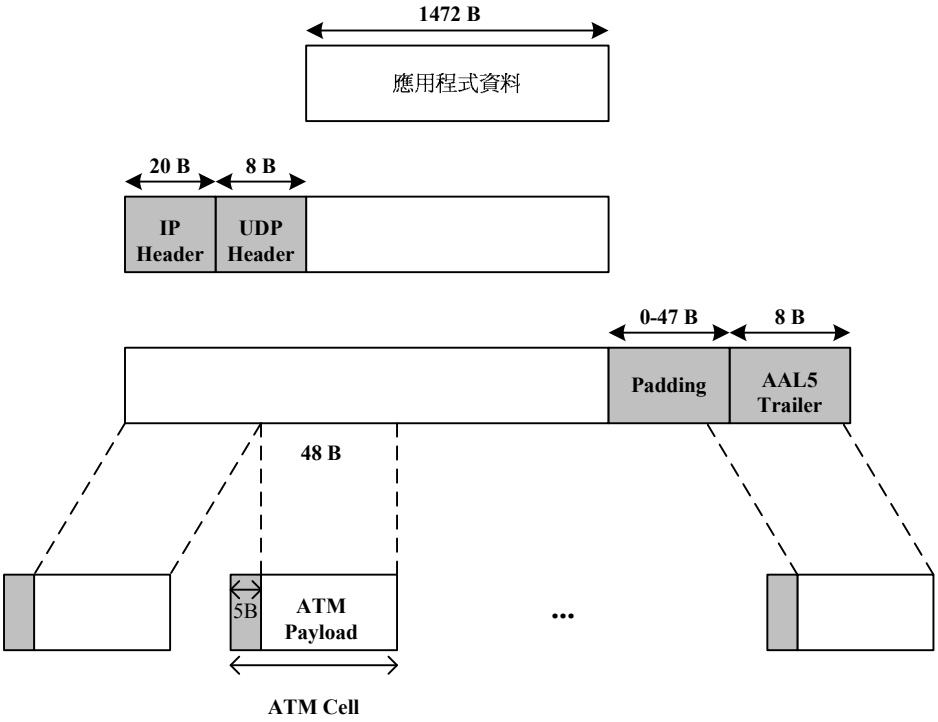

圖 5-3 反向通道應用程式資料切割圖 1

在反向通道方向資料從應用程式送下來 1472 bytes 的資料，到了 UDP/IP 以後加上 28 bytes 的表頭。資料從 Interface 往下層模組傳送至 ATM 模組時會加上 Trailer 8 bytes 並加上 28 bytes Padding，再進行切割成 48 bytes 爲單位的資料並加上 5 bytes 的 ATM 標頭。所以額外資料或是標頭資料佔據的資料量如下：

*表頭額外資料：28 + 8 + 28 + 5*32 = 224 bytes。*

UDP Payload 所佔比例：1472 / (1472+224) = 86.79%

從開始取出 ATM Cell 作爲 ATM Burst 開始，經過中間的同位元、編碼、調變等，一直到組成一個完整的 traffic burst，這中間的加標頭的動作可以參考下圖 5-4。

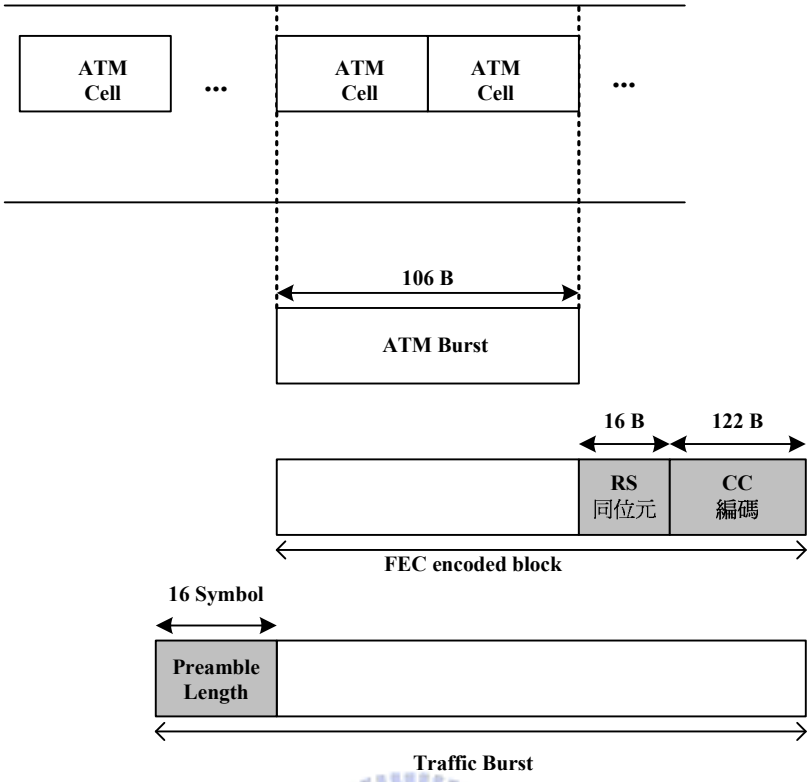

圖 5-4 反向通道應用程式資料切割圖 2

在 RCS_MAC 抓兩個 ATM Cell 組成一個 ATM burst，接著在底下實體層會根據編碼率及調變方式來做編碼。在編碼之前會先加上 16 bytes 的 RS 同位元檢查碼，並使用編碼率 1/2 的迴旋碼（Convolutional Code）連同同位元檢查碼一起進行編碼。在送出之前要先轉換成 Symbol，所以經過調變後會增加 16 Symbol 的 Preamble Length，才會送到通道上，因此以編碼率 1/2 及 QPSK 的調變方式的話會得到下面的額外資料：

*(16*8 + 122*8)/2 + 16 = 568 (Symbol)*

*ATM Burst Payload：53*2*8/2 = 424 (Symbol)*

ATM Burst Payload 所佔比例：424 / (424+568) = 42.74%

因符號傳輸率爲 9.935MBaud 加上 QPSK 的調變方式，因此得到資料傳輸率

爲 19.87Mbits/s，

1. 考慮圖 5-3 的額外表頭資料，使得通道真正可以用來傳輸資料剩下 86.79%
2. 考慮圖 5-4 的額外同位元檢查碼、編碼資料及 Preamble Length，使得通道真正可以用來傳輸資料剩下 42.74%

因此相乘會得到下面的最高資料傳輸率：

*19.87Mbits/s * 86.79% * 42.74% = 7.37 Mbits/s*

最後還必須考慮兩點：

1. 在 RCS_MAC_RCST 層只會得到 99 個 traffic slot，所以使用率爲 $99/(99+1)=99\%$
2. 在一個 traffic slot 中送出 Traffic Burst 時候，我們還會增加 200ns 的 Guard Time，所以扣除掉 200ns 所佔的比例來看，使用率爲 99.649%

   根據以上的理論推導得到應有的傳輸率爲 7.37 Mbits/s * 99% * 99.649% = 7.27Mbit/s。而在我們的模擬數據中的傳輸率爲 7.213Mbits/s，已經相當接近理論值。其餘不同參數的設定的理論及模擬數據都在表中。

CC Coding 模式	**理論傳輸率(Mbits/s)**	**模擬傳輸率(Mbits/s)**
CC-1/2	7.27	7.213
CC-2/3	9.642	9.568
CC-3/4	10.819	10.722
CC-5/6	11.976	11.876
CC-7/8	12.571	12.459

表 5-2 反向通道理論及模擬數據表

5.3 模擬效能測試

在經過以上的數據驗證後，我們希望對模擬時間對測試，

以下是我們的測試硬體設備及模擬環境設定：

硬體設備：

中央處理器：Intel 3.0G

記憶體：1G

主機板：ASUS P4B533

模擬環境：

RCST 個數：一個 RCST

資料流量：UDP traffic

資料流向：Forward link，從 SP 後端的 HOST 傳送到 RCST 後端的 HOST

模擬時間：100 秒

我們將紀錄當 UDP traffic 從一條增加到兩條、三條…的時候，所需的模擬時間。

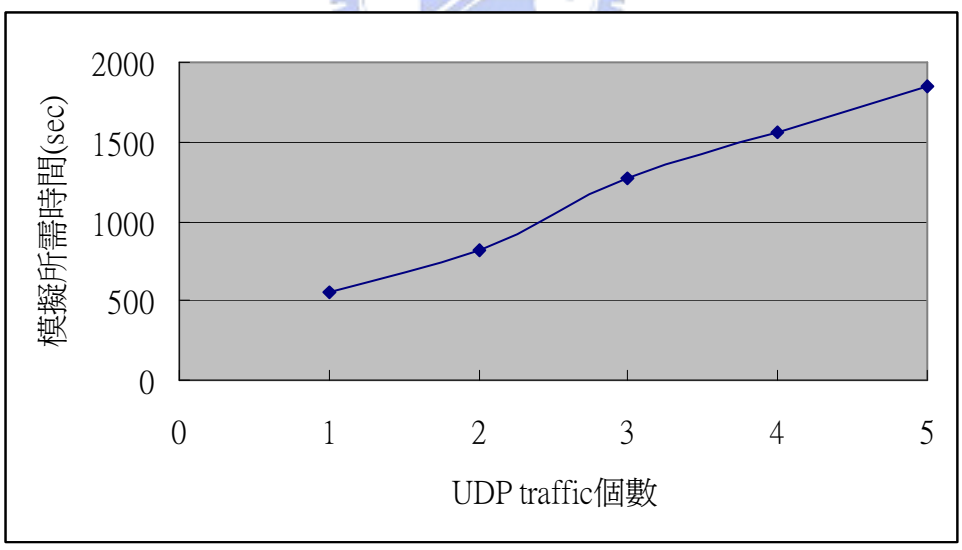

圖 5-5 模擬效能測試圖

由以上數據可看出當 UDP traffic 越多時，所需的模擬時間越長，其中底層的 Symbol rate 也隨著 traffic 的個數成線性成長，8、16…、40 MSymbols/s

第六章 未來展望

以上介紹完了 DVB-RCS 的主要設計與實作後，也驗證了系統的正確性，然而定義在規格中的功能太多了，我們並沒有完全實作出來，以下是未來有待開發的部份：

1. 多個饋送者節點及多個閘道節點

   當初設計爲了方便實作在不影響效能的情況下，我們只實作單一個饋送者節點及閘道節點。爲了更符合一般的使用情況，可以考慮設計多個饋送者節點及多個閘道節點的架構。

2. On board processing

   目前的衛星只是一個 layer 1 的 repeator，並沒有任何的 switching 的能力，爲了可以達到更有效率的使用頻寬，可讓衛星具有 switching 的能力。或者是提供 TCP proxy on board 在衛星上，也是可以有效的改善效能。目前網路也找的到相關的 paper [6][7] 在探討以上 on board processing 的能力，可以參考別人提出的作法。

第七章 結論

在此論文中，我們模擬及分析 DVB-RCS 衛星網路之效能。由於目前並無實體衛星可供實驗及量測效能，我們先在交大網路模擬器(NCTUns)上開發實做此種網路的模擬平台並仔細驗證其正確性。之後，我們在這些平台上進行應用測試來觀察與分析網際網路(Internet)上常用的一些通訊協定及應用是否適用在這種衛星網路上。

針對所開發出來的模擬平台，我們先進行了一系列嚴謹的效能及功能驗證。驗證結果顯示，此模擬平台所模擬出來的效能結果符合理論分析值，是個能令人信服的模擬平台。接下來，我們根據我們所表達感興趣的效能項目，開始進行模擬研究，研究這些效能項目在不同網路參數下會有什麼樣的變化。由於重要的衛星網路參數眾多，其各種組合所產生出的組態及情況也非常多。我們因此選了幾種較具有代表性的組合，對其進行模擬研究，並分析及探討其趨勢變化。

希望本論文能夠帶給在 NCTUns 研究 DVB-RCS 的研究人員由淺入深的介紹，且且完成自己的研究。

參考文獻

[1] ETSI EN 301 192 V1.4.1. Digital Video Broadcasting (DVB); DVB specification for data broadcasting. November 2004.

[2] S.Y. Wang, C.L. Chou, C.C. Lin, “The Design and Implementation of the NCTUns Network Simulation Engine”, Simulation Modelling Practice and Theory, 15 (2007) 57-81. (SCI)

[3] ETSI EN 301 192 V1.4.1. Digital Video Broadcasting (DVB); DVB specification for data broadcasting. November 2004.

[4] ISO/IEC 13818-1. Information technology - Generic coding of moving pictures and associated audio information: Systems. November 1994

[5] ETSI EN 302 307 V1.1.1. Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications. March 2005.

[6] Nicolas Courville AND Hermann Bischl, German Aerospace Center (DLR) Jingdi Zeng, Lipens –Lyon, Ffrance “CRITICAL ISSUES OF ONBOARD SWITCHING IN DVB-S/RCS BROADBAND SATELLITE NETWORKS”

[7] M. Luglio, M. Y. Sanadidi, M. Gerla, “On-Board Satellite “Split TCP” Proxy”, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 22, NO. 2, FEBRUARY 2004

附錄A 中英對照

交大網路模擬器（NCTUns）

數位視訊廣播-衛星反向通道（DVB-RCS）

正向通道（Forward Link）

反向通道（Return Link）

多頻分時多工（MF-TDMA, Multi-Frequency Time Division Multiple Access）

網路控制中心（NCC, Network Control Center）

媒體存取控制層（MAC, Media Access Control）

硬體層（PHY, Physical）

網路層（Network layer）

傳輸層（Transport Layer）

時槽（Time Slot）

傳輸率（Throughput）

地面站（Ground Station）

廣播（Broadcast）

傳輸時間（Transmission Time）

協定堆（Protocol stack）

衛星媒體存取控制（Satellite Media Access Control）

地面站媒體存取控制（Ground Station Media Access Control）

調變（Modulation）

解調變（De-modulation）

迴旋編碼（Convolution Encode）

解迴旋編碼（Convolution Decode）

上傳通道（Uplink Channel）

下傳通道（Downlink Channel）

控制通道（Control Channel）

符號傳輸率（Symbol Rate）

核心（Kernel）

傳輸率（Uilization）

時槽長度（Time-slot）

分時多工（TDMA, Time Division Multiple Access）

數位視訊廣播（DVB, Digital Video Broadcasting）

歐洲電信標準協會（ETSI, European Telecommunications Standards Institute）

流量樣式（traffic pattern）

傳輸流（Transport stream）

程式流（Program stream）

單元傳輸流（elementary stream）

封包切割（packetize）

多工（multiplexing）

傳輸流封包（TS packet）

通道編碼（channel coding）

多協定封裝（MPE, Multi-protocol Encapsulation）。

資料傳送帶（Data Carousels）。

對象傳送帶（Object Carousels）。

非同步的（asynchronous）

資料流導向（streaming-oriented）

同步（synchronous）

同步回放（play back in synchronization）

數位儲存體控制命令（DSM-CC, Digital Storage Media-Command and Control）

邏輯鏈結控制層／子網附著點（LLC/SNAP, Logical Link Control/Sub-network Attachment Point）

廣播（broadcast）

超級訊框（Superframe）

時槽（Timeslot）

控制表（Control Table）

虛設訊框（Dummy Frame）

編碼率（Coding Rate）

渦輪碼（Turbo Code）

訊框（Frame）

排程系統（Scheduler）

資料傳送速率（Data Rate）

ATM 封包（ATM Cell）

解多工（De-multiplexing）

同部訊號（Preamble）

表尾（Trailer）

標頭（Header）

符號（Symbol）

填補（Padding）

網路拓樸（Topology）