

# 於 TWAREN 上 DTN 自動供裝系統的設計與建置案例

黃文源<sup>1</sup>, 周大源<sup>1</sup>, 劉德隆<sup>1</sup>, Fei I Yeh<sup>2</sup>, Jim Hao Chen<sup>2</sup>, Joe J. Mambretti<sup>2</sup>

<sup>1</sup>財團法人國家實驗研究院國家高速網路與計算中心

<sup>2</sup>International Center for Advanced Internet Research, Northwestern University, USA  
{wunyuanyuan, 1203053, tliu}@narlabs.org.tw,  
{fyeh, jim-chen, j-mambretti}@northwestern.edu

## 摘要

本論文提供了一個 TWAREN 上高速資料傳輸網路 (Data Transfer Network, DTN) 自動供裝系統之建置案例。於硬體部份, 我們使用 TWAREN 的骨幹線路, 並搭配最後一哩的 10G 頻寬, 將本中心與合作單位 iCAIR 成功地建立高達 10G 頻寬的實體線路。另外軟體部份, 設計讓 OpenNSA 與 DTN 結合, 讓 DTN 的專有線路, 能夠隨時根據需求來自動供裝。利用這套系統, 使用者便能隨時隨地依需求來建立屬於自己的高速傳輸專有線路。

**關鍵詞:** OpenNSA, DTN, 高速資料傳輸網路

## 1. 簡介

最近這幾年來, 網際網路與電子產品等科技技術已經完全融入人類的生活中。Gartner[1]在先前分析與預測過, 電腦、手機、攜帶型電腦等等產品的銷售量都會有極高的成長。而各大產業的公司組織為了要充分滿足眾多使用者的喜好, 一一的推出不同的客製化服務或是蒐集使用者眾多資訊等等, 以便能取得更多的商機。為了達到前述目的, 這些公司組織也開始投入成本試圖在產業中導入人工智慧科技, 藉由更高階的科技來解決產業上的問題。針對眾多資料來分析與訓練, 以便讓機器深度學習, 而這是人工智慧的基礎。

同時, 一些與全球氣候變遷的相關議題越來越重要, 例如: 國土觀測、天文觀測與農業觀測等, 除上述議題外, 全世界也有許許多多領域也需要進行大量資料的分析, 而許多重大議題也已無法獨立的由單一領域和單一團隊完成。故跨團隊合作研究上的資料與資源共享也變得極為重要。隨著科技技術的進步, 資料計算的資源也同樣得更加進步, 分析問題時更能夠有更精細與更多的資料來分析, 而此情況也造成資料量從 Giga 往 Tera 與 Peta 等級邁進。

這些大量資料的傳輸, 最簡單的方式便是採用傳統的網路網路。FTP、HTTP 等傳統的網路傳輸方式均能上傳與下載資料。但是, 這些方式往往會受傳輸速率所限制, 對於巨量資料的傳輸, 傳統協定的資料傳輸速率將會浪費許多的時間。假若使用者數量遽增, 或者資料量更加龐大, 可能會發生網路壅塞的情況。另外, 公眾通道的傳輸也會有資訊安全疑慮, 例如: 資料外洩等。故巨量資料傳輸網路的建置將是較為適當的方式。

巨量資料傳輸網路的研究上有個較熱門的技

術, 此技術是以專用且高速的線路將多個存取節點 (Access Point) 串聯, 名為資料傳輸網路 (Data Transfer Network, DTN)。這些串聯的專用線路通常都是採用 VPLS、Tunneling 或是 VPN 等技術來達成。藉由前述的技術使專用線路與一般網路隔開來, 如此一來, 將能避免使用相同傳輸線路時的造成網路壅塞, 而且也能夠解決資訊安全的問題。在資料傳輸上, 有兩件必需考量的事情: 一、骨幹線路: 用來讓存取節點互相連接的部份。二、最後一哩線路 (last mile): 存取節點與使用者端連接的部份。

硬體骨幹線路部份, 我們先前將 DTN 建置於台灣先進學術研究網路 (Taiwan Advanced Research and Education Network, TWAREN) [2]上, 且 2016 年升級完成 100G 的學研網路 [3], 故此部分我們將無須再花費成本建置。最後一哩部份, 先前本中心和合作單位僅使用 1G 的線路與骨幹連接, 因此考量與合作單位再設計與規劃新的 DTN 系統, 提升最後一哩的頻寬。

原先建置的 DTN 系統, 有著自己專有的線路, 無法讓別的系統或應用程式使用, 使得頻寬使用上的彈性較缺少, 而有新的合作連線需求, 又必須花費許多時間來溝通與建置新的專有線路, 因此我們考量讓 DTN 能與自動供裝系統結合, 讓 DTN 系統適應於暫時且多變的連線要求, 省卻大量的建置時間成本。本中心先前也於 TWAREN 上建置了一套 OpenNSA [4][5][6], 主要用來提供合作單位的短暫連線需求, 故我們設計讓 DTN 與 OpenNSA 結合, 打造一個符合能即時連線與充滿彈性連線需求的 DTN 系統。

本論文的組織架構如下。相關技術將於第 2 節說明。本系統的架構設計與實作將於第 3 節陳述。第 4 節將展示我們的成果。第 5 節則是本文的結論與未來展望。

## 2. 相關技術

### 2.1 高速傳輸技術

對於資料於 MB 至 GB 等級的大小, 都可以於一般常用的資料傳送方法上正常運作。但是, 當有巨大的資料量必須傳送時, 就得考量可能的問題, 例如: TCP 或 UDP 傳輸、封包延遲、或是以平行化 TCP 的方法傳送資料。

Ping time 主要是用以測量初始節點到目的主機去回的時間 (roundtrip time, RTT), 也就是一個封包從傳送端送出後, 到達目的地再回傳送端的時間。依據網路架構與真實地理環境的不同, 使

用 traceroute 測試工具，不同地方送出測試的封包需要經過的裝置數通常都不相同。根據裝置的多寡、線路品質的優劣往往會造成不同的延遲情況，而延遲也會使得光傳輸的速率降低。

TCP 必須確保每個 packet 的送達，其花費許多的時間於 packet 來回之間的溝通與確認，假若每個時間點所確認的 packet 數量急遽降低，這就造成 ping time 變長，因此 TCP 協定應用於長距離的網路環境下，其傳輸的效率會較緩慢。但若你傳輸一串 TCP 封包的串流，含 FTP 等等協定，當 ping time 大幅增加時，則送封包、接收確認或其他資料等等的傳輸速率便會急遽減少。此限制就是大家所說的頻寬時延乘積 (Bandwidth-delay product)，傳輸效能會於長距離網路環境上降低。

當然，於資料傳送前，先將資料壓縮，就能傳送更多的資料了，藉由此方式，便能避開改善上述的問題。除了此方式外，也能採用變更傳輸單元 (Maximum Transmission Unit, MTU) 的方式來避開，此方式就是將封包容量加大，盡可能多存放傳輸資料。然而，一般市面的網路產品，例如：伺服器、個人電腦或是筆記型電腦，其預設的 MTU 都設定為1500位元組，此數字是偏低。一些學研網路或是高速網路環境，為了提高傳輸速度，會將 MTUs 拉高至9000位元組(區域網路則拉高至64K 位元組)，以便允許巨大封包的傳輸，來增加傳輸效率。

由於前面陳述 TCP 需要花大量時間來確保封包的正確性，故有些會採用 UDP 來代替 TCP，來節省許多的時間，提高傳輸效能。然而，UDP 因為少了封包確認等機制，是屬於不可靠的傳輸方式，因此使用者就必須想辦法減少 UDP 缺點的影響。

而同時傳輸多個 TCP 串流的方式，是最後一個解決長距離資料傳輸限制的方法，而這個機制又沒有 UDP 的缺點，也確保 TCP 的缺點，故此機制逐漸廣為被使用，或者是將現有的應用包裝而達成這種機制。

## 2.2 OpenNSA

北歐國際研網組織 (NORDUnet) 依據 NSI(Network Service Interface)協定開發了一套開放原始法軟體控制器，名為 OpenNSA。NSI 是一個網路服務層(Network Service Layer)介面協定標準 [7]，主要用途是動態網路資源調配、網路拓樸交換與跨網域之間分享網路服務資源。

NSI 架構最主要的部份是網路服務控制器 (Network Service Agent, NSA)，許多學研單位有開發了 NSA 的平台，OpenNSA 就是其中一個，其它的有 OSCARS[8]、G-Lambda[9]和 AutoBHAN[10]等。OpenNSA 是較容易擴充且屬於輕量級的控制器，主要是由 Python 語言所開發，OpenNSA 與前述的 NSA，全部都採用 NSI 來互相交換資料與溝通。

NSA 扮演著三種角色，這三種角色的劃分主要是以運作模式來區分，他們分別是提出網路服務請求(uRA ; ultimate Request Agent)、媒介網路

服務請求(AG ; Aggregator)和提供網路服務請求(uPA ; ultimate Provider Agent)的角色。

接下來使用圖1的架構來陳述，此情境是使用者想藉由 NSA 建置跨網域的專線與頻寬，一開始使用需透過網域內的 uRA 提出供裝專用線路以及頻寬等需求，AG 收到 uRA 的要求後，便會從記錄的資源庫中查找符合要求的線路與頻寬，下一步便會通知被選定的 uPA 並設定線路與頻寬，假若，此 AG 所通知的非 uPA，而是同屬 AG 角色，那就將需求交由此 AG，讓此 AG 繼續執行類似的行為，直到目的地的專用線路與頻寬能被建立和預約為止。

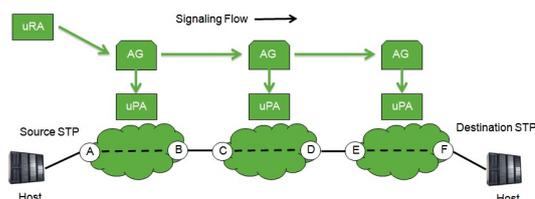


圖 1 NSA 使用情境[11]

## 3. 系統架構與實作

### 3.1 網路環境設計

我們欲建置的自動連線供裝 DTN 系統分為兩大部分，第一部分為 DTN 服務與實體線路建置，第二部分為 OpenNSA 系統建置與設定。圖1為我們設計的 DTN 服務的實體網路架構圖，此架構由 A、B 與 C 三個部分組成，A 部分為 NCHC 的線路規劃，我們會部署一台 DTN Server，此 Server 裝了兩張10G 網卡，分別是 p3p1與 p3p2，p3p1的用途是讓管理員可以登入設備，以便維護 DTN Server，而 p3p2將使用10G 光纖線 Brocade Switch 10G port 連接，同樣，Brocade Switch 也使用10G 光纖線與 Cisco ASR 9K 路由器連接，讓最後一哩能與骨幹線路連接。

接著 B 部分為 TWAREN 骨幹，先前 DTN 僅使用1G 的線路，為了能讓 DTN 能擁有完整10G 頻寬，也考量合作單位 International Center for Advanced Internet Research (iCAIR)測試時均採用 VLAN 1779~1800的 ID，故與 TWAREN 骨幹維運人員申請10G 的線路與 VLAN ID 1779~1800來使用。最後 C 部分，iCAIR 同樣將最後一哩的線路升級至10G，並且請將至骨幹的光纖線路改接至設定好的10G 卡板之上，至此，所有線路重新改接與升級便完成。

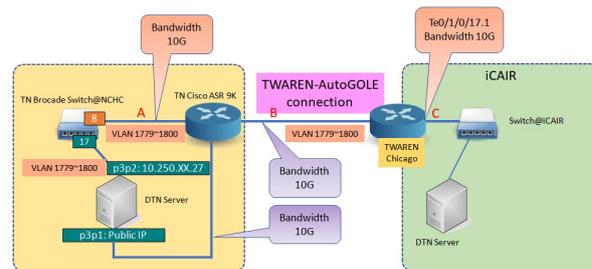


圖 2 實體線路建置設計

### 3.2 自動連線供裝系統建置

前述第一部分建置完成後，我們接下來要描述我們設計的自動連線供裝系統。如圖3所示，我們部署一台裝有 OpenNSA 及擁有兩個網路介面的 Server，一個網路介面與公眾網路連接，目的是透過公眾網路介面和 iCAIR 的 OpenNSA 連線並交換連線供裝訊息，另一張網路介面與 Brocade Switch 的 Management port 連接，目的是讓 OpenNSA 能連線至交換器，並針對目標 port 設定 VLAN ID，同理，iCAIR 那一側也採用同樣設計來部署 OpenNSA Server。

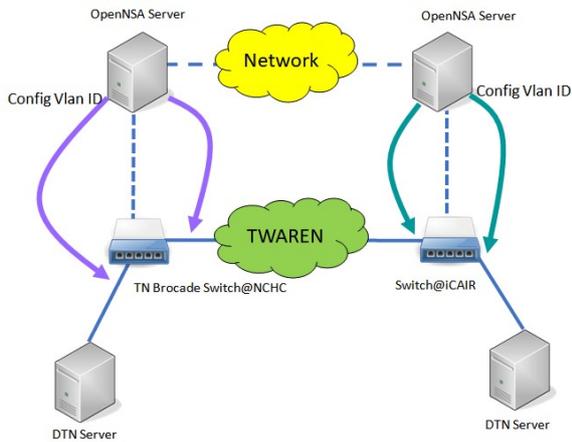


圖3 OpenNSA 建置設計

接下來，我們設定 OpenNSA 的檔案參數，以便能順利運行於我們所設計的環境之上。OpenNSA 內所需設定的檔案有兩個，分別為 `opennsa.nrm` 與 `opennsa.conf`，`nrm` 檔案主要是設定虛擬的介面名稱以及相關參數，圖 4 為我們設定好的參數。圖 4(c) 為完整的設定，而(a)為(c)設定的前半部，(b)為後半部，`nrm` 的欄位說明如下：

- NetworkType: 設定此介面的連線類型。
- Interface\_name: 設定此介面的名稱。
- NetworkId: 此介面所連接的 NetworkId。
- Vlan: 設定能使用的 Vlan ID 範圍。
- Bandwidth: 設定此 port 的多少頻寬。
- PHY\_Interface: 設定此介面應對的實體網路介面

```
opennsa@nsi: ~/opennsa
#networkType interface_name networkId
ethernet starlight icair [redacted] (-in|-out)
ethernet nchcdtn -
```

```
vlan bandwidth PHY_interface
vlan:1779-1800 10000 e1/1/8 -
vlan:1779-1800 10000 e1/1/17 -
```

```
opennsa@nsi: ~/opennsa
#networkType interface_name networkId vlan bandwidth PHY_interface
ethernet starlight icair [redacted] 1779-1800 10000 e1/1/8 -
ethernet nchcdtn - 1779-1800 10000 e1/1/17 -
```

圖4 Web-UI 系統架構圖

`opennsa.conf` 為 `opennsa` 基本設定的檔案，圖 5 是我們使用參數的範例，檔案內區分為 `service` 與 `switch backend` 區塊，檔案內我們使用的參數如下：

- network: OpenNSA 的網路 ID。
  - logfile: log 檔案的名稱。
  - nrmmap: nrm 檔案的名稱。
  - host: 主機的 IP 或 Domain Name。
  - port: 主機使用的 Port。
  - peers: 設定要與那些 OpenNSA 節點連線。
  - database: OpenNSA 使用的資料庫名稱。
  - dbuser: 資料庫的使用者名稱。
  - dbpassword: 資料庫的使用者密碼。
  - key: Server 使用的 key file。
  - certificate: Server 使用的證書。
  - certdir: 證書使用的資料夾。
  - verify: 是否啟動驗證。
  - allowedhosts: 允許那些 host 可以使用 OpenNSA。
- 上述均為 Service 區塊的設定參數，`switch backend` 區塊的參數是根據使用的 backend 來決定的，使用者可以客製化自己的 backend，以下列出我們使用的 `switch backend` 的參數作為範例供大家參考：
- [brocade]: [”和”] 填入使用的 backend 名稱
  - host: 交換器的 IP 或 Domain name。
  - user: 交換器登入名稱
  - password: 交換器登入密碼
  - fingerprint: 交換器的 fingerprint

```
This is a configuration file for running an OpenNSA service directly from the
development directory

[service]
network=opennsa.nchc.org
logfile=opennsa.log
nrmmap=opennsa.nrm
host=1.2.3.4
port=9443
peers=https://opennsa.test1.com:9443/NSI/discovery.xml
      https://opennsa.test2.com/nsa-discovery
database=opennsa
dbuser=opennsa
dbpassword=test
tls=true
key=/home/opennsa/certs/test.key
certificate=/home/opennsa/certs/test.crt
certdir=/home/opennsa/certs
verify=true
allowedhosts=testtest@nchc.org.tw,opennsa.test.edu

[brocade]
host=5.5.5.5
user=test
password=testtest
fingerprint=11:11:11:22:22:33:33:33:44:44:55:55:55
```

圖5 OpenNSA 建置設計

至此，我們的 OpenNSA 就設定完成，當我們與合作單位有 DTN 連線需求，便能將 Request 送至我們的 OpenNSA，然後我們的 OpenNSA 會設定我們的 Brocade 交換器，並且告知合作單位的 OpenNSA，請求設定底層的連線設定，如此一來，我們的 DTN 便能夠開始與合作單位的 DTN 傳送大量的流量。

## 4. 系統測試

### 4.1 OpenNSA 供裝測試

DTN 連線自動供裝系統的測試，分為兩部分，第一部分為 OpenNSA 的供裝測試，目的是確認與合作單位的 OpenNSA 能連線並提供所需的 VLAN 專有線路。

OpenNSA 的連線供裝需要藉由 `onsa` 這個程式來申請，圖 6(a)(b) 為我們針對 `onsa` 程式設定的參數範例與命令執行結果。圖 6(a) 的範例是根據圖 5 來設定，參數說明如下：

- bandwidth: 設定 onsa 要提出多少的頻寬需求。
- host: onsa 所在的 IP 或是 Domain name
- port: 開始 onsa 要監聽的 port



## 參考文獻

- [1] Gartner：全球各種裝置（個人電腦，平板、手機）出貨量估 2018 年增 2.1%，達 23.2 億台  
<http://iknow.stpi.narl.org.tw/Post/Read.aspx?PostID=14166>
- [2] 台灣先進學術研究網路 TWAREN, <http://www.twaren.net/>
- [3] 10G 變 100G 台灣新學研網路啟用  
<http://www.chinatimes.com/realtimenews/20161006004628-260405>
- [4] OpenNSA, <https://github.com/NORDUnet/opennsa>
- [5] 黃文源, 李慧蘭, 劉德隆, “NSI 於軟體定義式網路(SDN)上資訊傳遞之應用與實作”, TANET 2015 論文集, 南投, 2015
- [6] 黃文源, 李慧蘭, 劉德隆, “使用 NSI 達成 L2 與 SDN 的異質網路之介接”, TANET 2016 論文集, 南投, 2016.
- [7] NSI project, <https://redmine.ogf.org/projects/nsi-wg>
- [8] OSCARS, <https://www.es.net/engineering-services/oscars/>
- [9] Atsuko Takefusa, et. al, “G-lambda: coordination of a grid scheduler and lambda path service over GMPLS Source”, Future Generation Computer Systems ,Vol.22 , Issue 8, pp. 868 – 875, October 2006
- [10] M. Büchli, et al., “Deliverable DJ.3.3.1:GÉANT2 Bandwidth on Demand Framework and General Architecture”, GÉANT, 2005
- [11] John MacAuley et. al, “Network Service Interface Signaling and Path Finding”, GWD-I NSI-WG, December 2014