Cross domain VM migration over TWAREN OpenFlow testbed

李慧蘭　劉德隆　胡仁維　陳 敏
國家高速網路與計算中心
{gracelee, tlliu, hujw, minchen}@narlabs.org.tw

摘要

隨著雲端運算的快速發展，雲端應用服務的使用需求激增，為提高雲端服務的可靠度，因此對虛擬機器遷移的需求也因應而生。尤其，為確保雲端服務不中斷，首當其衝要面對的是虛擬機器跨不同網域遷移時的網路問題。OpenFlow 可程式化開放架構之特性，應用在雲端資料中心可以簡化對網路需求的複雜性與提高基礎架構的靈活性。在本文中，利用 OpenFlow 可以彈性依據使用者的定義快速過濾和導向封包的特性，設計一套 Layer3 跨網域之虛擬機器遷移機制，並在 TWAREN OpenFlow testbed 上實驗證明其可行性。

關鍵詞：虛擬機器遷移、跨網域、雲端資料中心、OpenFlow。

Abstract

While the cloud computing technology significantly progresses, it has been quickly gaining popularity at the same time. To ensure its service reliability against hardware and facility failure, the demand for live migrating virtual machines between different network domains is getting increasingly higher. OpenFlow's high programmability and its open architecture can be used to reduce the complexity of the cross domain IP migration and increase the overall architecture flexibility. This study proposed an OpenFlow based cross domain virtual machine migration mechanism and verified it in the TWAREN SDN testbed.

Keywords: VM Migration、VM Migration 、Cloud Data Center、OpenFlow.

1. 前言

近年來雲端服務的崛起， 各式各樣以雲端技術建構的基礎架構服務(IaaS)、平台服務(PaaS)及軟體服務(SaaS)等皆相繼推出，在這波熱潮下，帶動企業及相關業者快速建置雲端資料中心的腳步。根據 CISCO 發佈的 Global Cloud Index [1]報告，預估雲端資料中心的流量在 2016 年將達 4.3ZBs (Zettabytes)，屆時雲端資料中心的建置數量將比傳統資料中心多 1.5 倍，估算自 2011 年至 2016 年的網路流量的成長率將達 44%。

雲端服務業者利用虛擬化技術透過網際網路以服務的型態動態提供給使用者，讓每個使用者依需求取得虛擬資源，可降低資料中心建置成本並提高設備使用率。然而在營運過程中，數量龐大的虛擬機器可能面臨實體伺服機器超載、異地備援、資源分配和負載平衡[2]等問題。為解決上述問題，虛擬機器必須隨時可以遷移，而遷移的範圍不侷限於同一機櫃內、相同網域之資料中心內或跨資料中心之 Layer2 網域內，更擴及跨網域之資料中心，因此凸顯虛擬機器遷移是雲端服務管理上重要的議題。

在跨網域的遷移中需要打破不同網域異質網路架構的藩籬，為確保服務不中斷的無縫遷移，虛擬機器在來源網路的設定必須適用於搬遷至目地機器的網路設定，在[2]中提到利用 IP-in-IP Tunnel 技術可滿足上述虛擬機器跨網域遷移。虛擬機器遷移的許多問題已被正視，大部分虛擬機器遷移機制僅限 Layer2 網域間，如：VXLAN[3]、NVGRE[4]、TRILL[5] 、PortLand[6]等，並存在商業化的解決方案，但須全面佈署新功能之網路設備，成本過高且無法客制化。本文將探討以 OpenFlow[7]支援 Layer3 跨網域遷移的作法。

OpenFlow 技術是允許使用者自由設計創新的網路應用，經由傳送層和控制層分離的核心精神，將自由操控封包的工作交由獨立的控制平台，交換器則專責於封包的傳遞，基於 flow table 上記載的 flow entry 來傳送封包，因此只要將自行開發相關功能載入控制平台，如此即能打造一應用 OpenFlow 技術作到跨網域之虛擬機器遷移。

2. 研究背景

許多研究針對雲端資料中心之虛擬機器遷移提出相關的解決方法，其中也針對雲端資料中心的網路拓撲架構探討，隨著軟體定義網路(Software-Defined Network, SDN[8]) 的崛起，OpenFlow 技術獲得重視，如何應用 OpenFlow 技術，降低雲端資料中心之虛擬機器遷移管理的複雜度，將是未來熱門的研究議題。

2.1 相關研究

傳統網路技術在鋪建新興的雲端資料中心時，應用在雲端運算時略顯不足，例如：STP (生成

樹協議)收斂時間在大型資料中心就顯得過長；此外，STP 為確保無迴圈，而將有些路徑阻斷，也降低頻寬使用率和遏止虛擬機器遷移的便利性。有鑒於此 IETF 制定 TRILL (Transparaent Interconnection of Lots of Links)[5]標準，來取代 IEEE 802.1D 在 LAN 環境避免迴圈的角色。TRILL 是利用路由技術作為 Layer 2 網路的控制平台，在資料中心內的各交換器彼此交換鏈路狀態路由協議(link-state routing protocol)。當有封包須要傳送時，交換器透過路由表做最佳路徑選擇到達目地交換器。另 FabricPath[9]技術與 TRILL 協定相近。

PortLand[6]提出一隨插即用、易於管理和容錯性高的雲端資料中心架構，為進行有效率的封包轉送，每個虛擬機器會根據本身在拓樸中的位置分配到一個 PMAC (Pseudo MAC)位址編碼，交換器根據自行定義的 PMAC 位址查找傳送，當封包要往出口交換器轉送時再由中央控管系統 fabric manger 進行 PMAC 位址和傳統 AMAC (Actual MAC)位址的轉換。因此 PorLand 支援同 Layer2 網域內的虛擬主機遷移。

VL2[10]也採用新定址方式，主機上的應用編址稱為 AA (Application Address)，位置編址為 LA(Locator Address)。在主機上必須額外安裝代理軟體，以將要送出的封包封裝在 LA 位址中，導向負責管理的目錄系統(Directory System)，由該系統進行 LA 位址轉換成 AA 位址並計算路由。當虛擬機器遷移到他處時，雖 LA 位址更動但 AA 位址不變，虛擬機器遷移後仍可正常存取。

VXLAN(Virtual eXtensible LAN)[11]的作法是將虛擬機器的乙太訊框封裝在 IP 封包內(MAC-in-UDP)，新增 24 位元的 VNI 欄位來擴充可使用的 VLAN 個數，僅有相同 VNI 的虛擬機器可以彼此互通，虛擬機器唯一的代號是 VNI 加 MAC 位址，因此即使同一台虛擬機器由於搭配不同 VNI 而屬於不同的 VXLAN，藉此區隔多個租用戶(multi-tenancy)的資料流。資料中心之間以 Layer2 Tunnel 串連， 提供相同 Layer2 網域之虛擬機器遷移。而 OTV(Overlay Transport Virtualization)[12]是 MAC-in-IP 的作法，利用 MAC 位址路由建立 MAC 位址與 IP 位址對應表，其技術與 VXLAN 相仿，但支援 Layer3 網域之虛擬機器遷移，缺點是成本過高和缺乏彈性。

2.2 雲端資料中心網路拓樸

Fat Tree[13]如圖 1，是典型的資料中心三層式(three-tiered)網路架構，該架構自底層往樹根的鏈路頻寬越來越大，有別於一般 tree 架構的鏈路頻寬階相同。PortLand[6]採用 Fat Tree 網路架構。VL2[10]採用 Clos Netwoks 架構，作法是存取層(access layer)交換器以兩條鏈路介接匯聚層(aggregation layer)交換器，每一台匯聚層交換器與每一台核心層(在論文中稱 Intermediate Switch)介接，後者為多對多鏈路，該架構也是三層式網路架構，如圖 2所示。

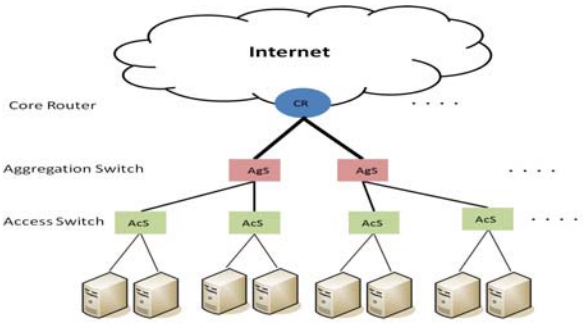

圖1 Fat Tree 架構

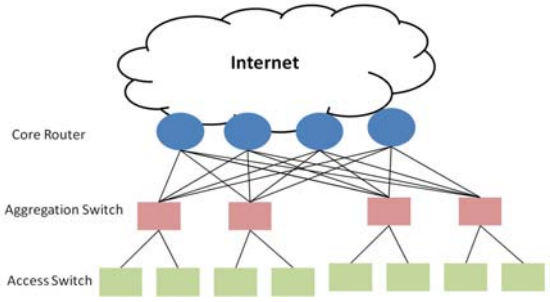

圖2 Clos Network 架構

3. 系統設計與架構

在本節中我們將展示在 TWAREN OpenFlow testbed 上應用 OpenFlow 建構跨網域之虛擬機器遷移的網路架構、方法設計和以實驗情境證明其方法可行性。

3.1 實驗架構

本研究所設計的架構圖如圖 3所示。網域 1 和網域 2 為不同網域且具 L2+L3 混合型網路架構的雲端資料中心。核心層通常是路由器或高階交換器，為資料中心介接至網際網路的邊界，負責 Layer 3 IP 路由查詢與交換工作。匯聚層與存取層則負責 Layer 2 封包轉送，匯聚層把存取層交換器連結匯合聚集後與核心層連結，存取層交換器供伺服器介接。我們設計匯聚層與存取層的交換器為 OpenFlow 交換器，兩個網域之間透過匯聚層供裝之 Tunnel 技術串連成 TWAREN OpenFlow 網路。

本實驗所建置的 OpenFlow 交換器是搭建在 NetFPGA[14]平台上，採用由美國史丹佛大學(Standford University)為了研究下一世代網路技術而開發的實驗平台 NetFPGA。NetFPGA 包含了四個 1Gbps 高速乙太網路通訊埠，使用 Xilinx Virtex-II Pro 50 場域可程式化閘陣列(Field-Programmable gate array, FPGA)為系統晶片，配備兩個 18Mb 的 SRAM 和 64MB 的 DDR2 DRAM，此外也提供兩個 Serial ATA(SATA)接口，可以串連多張 NetFPGA 卡板達成多埠數的交換器與路由器系統設計。本實驗是將 NetFPGA 卡板安裝於 CentOS 5.6 以上作業系

統，該 NetFPGA 是一張 PCI 介面卡，載入 openflow 1.0.4 軟體之後就是一台四埠網路介面 OpenFlow 交換器。

3.2 控制平台

本實驗控制平台採用的控制器是以 Floodlight Controller[15]建置。提供兩種 flow entry 建立方式：

- 主動式 flow 插入(Proactive Flow Insertion, PFI)：為 OpenFlow 標準所支援。可事先撰寫 flow entry 於 Static Flow Pusher 模組中，啟動該模組即可將 flow entry 派送至 OpenFlow 交換器。或以 REST API 提供之介面，手動新增 flow entry 到 OpenFlow 交換器。如表 2中 flow1 和 flow2。
- 回應式 flow 插入(Reactive Flow Insertion, RFI)：為 OpenFlow 標準所支援。當封包到達 OpenFlow 交換器時，比對 flow table 結果為沒有符合的 flow entry，則送往控制平台控制管理，控制平台載入 Forwarding 模組，再依據該模組定義即刻派送適用的 flow entry 到 OpenFlow 交換器，目前於 Forwarding 模組中預設定義有 ARP 請求和 ARP 回應封包皆以回應式 flow 插入(RFI)處理。

本文中除了使用上述兩種 flow entry 建立方式之外，我們也設計 On-demand Flow Insertion(OFI)方式：

- 隨選式 flow 插入(On-demand Flow Insertion, OFI)：利用 Floodlight Controller 內的 LearningSwitch 模組，如圖 4右下方塊所示，可將目的端 IP 位址、MAC 位址和 Switch Port 資訊記錄到 Switch Table。當收到封包後，擷取封包中 DIP 欄位(DIP：目的端 IP 位址)當成搜尋字串，便可獲知 MAC 位址和交換器介接之埠號。最後以 REST API 方式將撰寫好的 flow entry 派送至交換器。

在本架構圖圖 3中，存取層之 OpenFlow 交換器控制端由各別區域內的控制平台控制控：控制平台 1(controller1)和控制平台 2(controller2)，匯聚層之 OpenFlow 交換器則由共同之控制平台控制：控制平台 0(controller0)。

3.3 實驗情境

本實驗將展示位於異地跨網域之資料中心內的虛擬機器遷移到目標主機後，為保持對外服務不中斷，必須維持原有的 IP 位址、閘道和 MAC 位址不變，我們應用 OpenFlow 達到無縫遷移，為便於描述以下網路實驗的三種情境。本文定義以下概念：

- Home Domain (HD)：虛擬機器原本所在的網域，又可稱為虛擬機器的 Home Network。
- Foreign Domain (FD)：虛擬機器遷移到新的網域，又可稱為虛擬機器的 Foreign Network。
- Home Domain VM (HDVM)：位於 Home Network 的虛擬機器。
- Foreign Domain VM (FDVM) ：位於 Foreign Network 的虛擬機器。

下文以圖 3為實驗情境說明，位於網域 1 的虛擬機器有 R 和 S；位於網域 2 的虛擬機器有 U 和 T，當原來位於網域 1 的虛擬機器 VM-S 遷移至網域 2 後，網域 1 對虛擬機器 VM-S 而言是 HD，網域 1 內未曾遷移過的虛擬機器群均屬 HDVM。然而，網域 2 對虛擬機器 VM-S 而言是 FD，虛擬機器 VM-S 在網域 2 內被視為 FDVM。表 1為實驗架構中使用到的設備 IP 位址、MAC 位址資訊，CR1 為網域 1 的閘道；CR2 為網域 2 的閘道等。

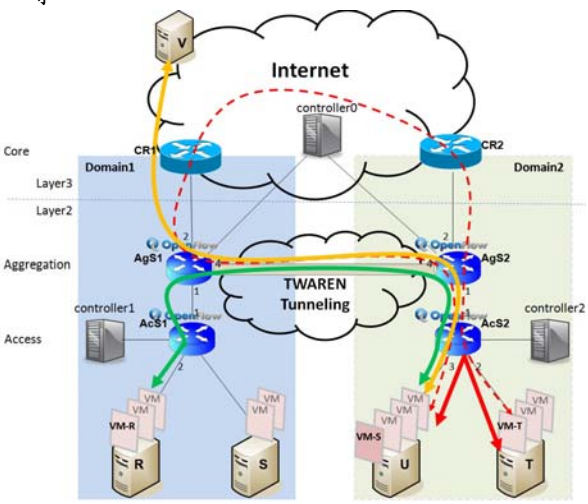

圖3 網路架構與情境示意圖

表1 實驗設備資訊

網域 1		
CR1	IP:192.168.1.254	MAC:00:15:fa:fe:34:41
AgS1	Switch ID: 11:11:11:11:11:11	
R	IP:192.168.1.2	MAC:08:00:27:94:9e:3d
網域 2		
CR2	IP:10.27.82.254	MAC:00:15:fa:fe:34:42
AgS2	Switch ID: 22:22:22:22:22:22	
AcS2	Switch ID: 04:96:52:21:3d	
VM-S	IP:192.168.1.101	MAC:00:15:17:8f:ca:37
VM-T	IP:10.27.82.101	MAC:08:00:27:9b:b3:12

情境一：

虛擬機器 VM-S 遷移至網域 2 後，VM-T 要與 VM-S 通訊時，由於分別屬於不同網域之 IP 位址，因此封包被傳遞到網域 2 的閘道 CR2，如表 1紅色虛線所示，再經由 Internet 上路由表查找，最後送到網域 1 的入口路由器 CR1，再往介接其下之 Layer2 交換器轉送。在本架構中匯聚層交換器 AgS1

和 AgS2 以 Tunnel 連接後，網域 1 的 AgS1 和 AcS1 交換器與網域 2 的 AgS2 和 AcS2 交換器，均屬於同一個 Layer2 廣播網域，於控制平台 2 上啟動預設 Forwarding 模組，當有 ARP 封包經過時，觸發 RFI 建立 VM-S 的 MAC 和對應埠號的 flow table，因此封包一路傳送到目的端。這樣的繞路就是三角路由(Triangle Routing)，這種非對稱性的路徑對於通訊延遲與網路資源浪費帶來了顯著與額外的負擔。尤其資料中心的流量特徵以東西向流量居多，更凸顯其封包傳送延遲造成的效能不彰。

我們在 AcS2 上應用 OpenFlow 可自由操控網路封包流向的技術，縮短通訊延遲與改善頻寬使用率。以下詳述當 VM-S 開始遷移到目的機器 Y 時，步驟如圖 4所示，說明如下：

❶網域 2 的控制平台 2 接收到虛擬機器遷移的 Event 通知，該通知包含 VM-S 的 IP 位址、MAC 位址和閘道的 MAC 位址。

❷將設計好的 flow entry 資訊寫進 Static Flow Pusher 模組。

❸flow entry 被派送到 AcS2 交換器(switch id: 04:96:52:21:3d)的 flow table，即表 2的 flow1 和 flow2。

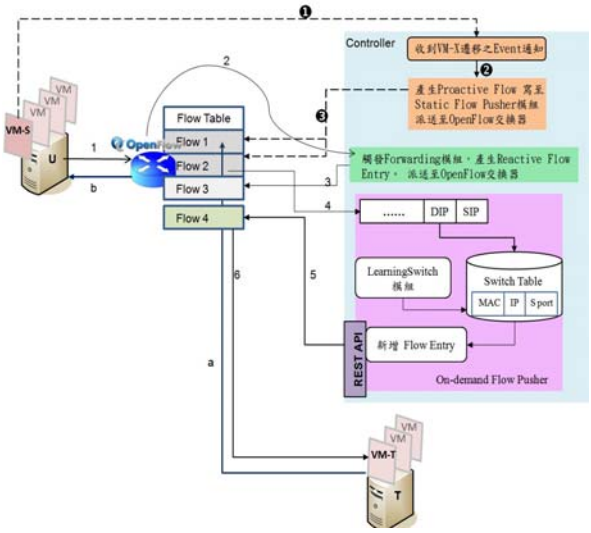

圖4 封包傳送示意圖

flow1 可將目的位址是 VM-S ("dst-ip":"192.168.1.101")，但必須送給網域 2 路由器 CR2("dst-mac":"00:15:fa:fe:34:42")的封包轉向直接給埠號 3。然而，VM-S 要丟往網域 2 的封包，由於目的位址無法事先得知，因此設計 flow2 規則為：來源位址是 VM-S("src-ip":"192.168.1.101")送出的封包轉向給控制平台處理。

接著， VM-S 送出第一個封包給目的機器 VM-T 如圖 4所示，步驟如下：

1. VM-S 發送 ARP 詢問 CR1 之閘道 MAC 位址，封包進到 OpenFlow 交換器，Flow Table 內無符合的 flow entry，送至控制平台處理。
2. 控制平台觸發 Forwarding 模組，以回應式 flowt 插入(RFI) 產生 ARP 廣播用之 flow3，以幫助 VM-S 取得 CR1 之閘道 MAC 位址。
3. VM-S 取得 CR1 之閘道 MAC 位址之後，VM-S 開始送出的封包目的 IP 是 VM-T 的位址("dst-ip":"10.27.82.0/24")，目的 MAC 是 CR1 之閘道 MAC ("dst-mac":"00:15:fa:fe:34:41")，符合 flow2 規則，操作行為(action)是將封包轉向送至控制平台 2。
4. 進入隨選式 flow 插入(OFI) 模組，擷取封包之目的 IP 位址(DIP)當成關鍵字串，進到 Switching Table 的資料庫進行搜尋，便可獲知 MAC 位址和與交換器 AcS2 介接之埠號。
5. 透過 REST API 寫一筆 flow entry 如**表 2** 中的 Flow4 到 AcS2 交換器。
6. 最後傳輸路徑依據 flow4 定義，原應送到網域 1 閘道的封包，直接修改為 VM-T 之目的 MAC 位址並轉向至埠號 2。

VM-S 與目的機器 VM-T 通訊的第一個封包將被送至控制管道的控制平台處理，依上述流程(step1~5)轉送。之後的封包到達 AcS2 交換器時，由於可成功比對 flow4，則不需再送至控制平台處理。亦即忽略優先順序較低 flow2 (priority=3000)。

當 VM-T 反向傳送封包給目的端 VM-S 時，由於目的端 VM-S 資訊在上述步驟❶~❸時，已根據 Event 資訊建立好 flow1，流程如圖 4：

a. 封包進到 AcS2 交換器，比對 Flow Table，符合 flow1 的 flow entry，操作行為是修改目的 MAC 位址為 VM-S，並轉向送到 VM-S 所介接的埠號。
b. AcS2 交換器送出封包。

最後，AcS2 交換器的 Flow Table 有三筆 flow entry，為 flow1、flow2 和 flow4，而 flow3 僅在 ARP 請求時使用，固定時間後會被移出 flow table。因此 VM-S 送包封給 VM-T 的路徑(datapath)最後只會比對 flow4；反向 VM-T 送包封給 VM-S 的路徑(datapath)最後只會比對 flow1，如圖 3紅色實線所示，改善了三角路由的問題。

表 2 控制平台 2 之 Flow 資訊

```
flow1= {
'switch':"00:00:00:04:96:52:21:3d",
"ether-type":“0800",
"priority":"32768",
"src-ip":"10.27.82.0/24",
"dst-mac":"00:15:fa:fe:34:42",
"dst-ip":"192.168.1.101"
"active":"true",
"actions":"set-dst-mac=00:15:17:8f:ca:37, output=3"
}
```

```
flow2= {
'switch':"00:00:00:04:96:52:21:3d",
"ether-type":“0800",
"priority":"30000",
"src-ip":"192.168.1.101",
"src-mac":"00:15:17:8f:ca:37",
"dst-mac":"00:15:fa:fe:34:41",
"dst-ip":“10.27.82.0/24",
"active":"true",
"actions":" output=controller2"
}
```

```
Flow3 = {
 'switch':" 00:00:00:04:96:52:21:3d ",
"ether-type":“0806",
"dst-mac":"ff:ff:ff:ff:ff:ff",
"dst-ip":"192.168.1.254"
"actions":"output=1"
}
```

```
flow4 = {
'switch':"00:00:00:04:96:52:21:3d",
"ether-type":“0800",
"priority":"32768",
"src-ip":"192.168.1.101",
"src-mac":"00:15:17:8f:ca:37",
"dst-mac":"00:15:fa:fe:34:41",
"dst-ip":"10.27.82.101",
"active":"true",
"actions":"set-dst-mac=08:00:27:9b:b3:12,output=2"
}
```

情境二：

與情境一相同的網路架構，如綠色線所示，VM-S 與 VM-R 均屬相同網域，因此與 VM-S 通訊時，傳輸路徑是 AcS1-AgS1-AgS2-AcS2。在情境二中，控制平台 0、控制平台 1 和控制平台 2 載入 Forwarding 模組之後，可自動以回應式 flow 插入模式(RFI)正常通訊。但當數量龐大的虛擬機器需要遷移時，Layer2 廣播網域將拖垮效能，因此本實驗也撰寫主動式 flow 插入(PFI)如圖 5、圖 6所示，並證明可運行。

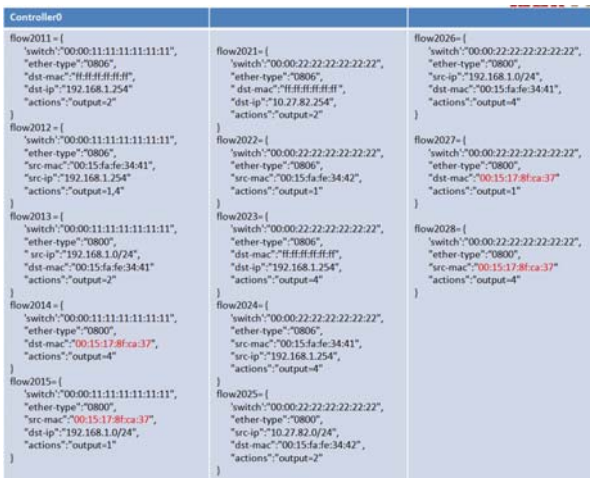

圖5 控制平台 0 之 Flow 資訊(應用於情境二、三)

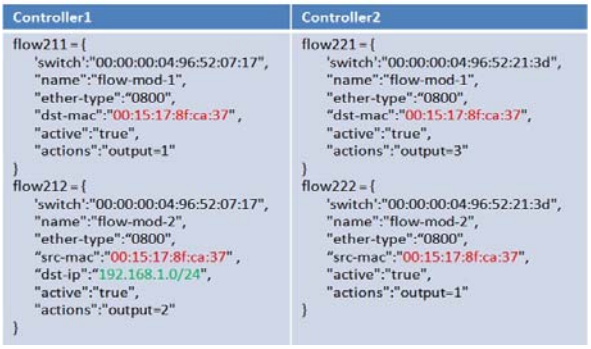

圖6 控制平台 1、2 之 Flow 資訊 (應用於情境二、三)

情境三：

如圖 3黃色線所示，其他網域的機器 V 要與 VM-S 通訊時，由於 VM-S 的網路 IP 位址未異動，經路由表上查找最佳路徑是往網域 1 的邊界路由器 CR1 走，再往 AgS1-AgS2-AcS2 到達目的端。flow entry 設計與情境二相同。

4. 架構分析與評估

由第三章跨網域虛擬機器遷移的情境中所述，我們設計的機制是混合式控制，每個網域之存取層有一專屬的分散式控制平台，因為網域之間的距離可能達數百公里，且虛擬機器遷移後伺服器對伺服器間的東西向流量佔 75%[16]，若由同一個控制平台分送 flow entry 到交換器，將有通訊延遲(latency)過長、彈性不足和服務中斷問題。但為維持 IP tunnel 的建立，匯聚層控制平台則採集中控制，面臨的問題是每一個控制平台之間必需彼此同步。IETF 工作小組提出 SDNi 協定草稿[17]，是 SDN 控制平台同步的初步構想，以 BGP 和 SIP over SCTP 協定的概念作同步交換訊息。

本設計架構以一階層的一對一網域為例，虛擬機器由 HD 遷移至 FD。並經實驗證明可運行。當有兩個以上的雲端資料中心需要交互進行多對多虛擬機器遷移時，如圖 7四個網域為例說明，網域 1 內有 VMx 和 VMa 遷移出去，也會有其他虛擬機器遷移進來。而多階層式的遷移，例如，兩階層的遷移：VMb 位於網域 1 遷移至網域 2(VMB)再遷移至網域 3(VMB')。三階層的的遷移：VMc 位於網域 1 遷移至網域 2(VMC)再遷移至網域 3(VMC')最後再遷移到網域 4(VMC")。多階層式遷移中網域內匯聚層交換器須建立 fully mesh tunnel 連線。

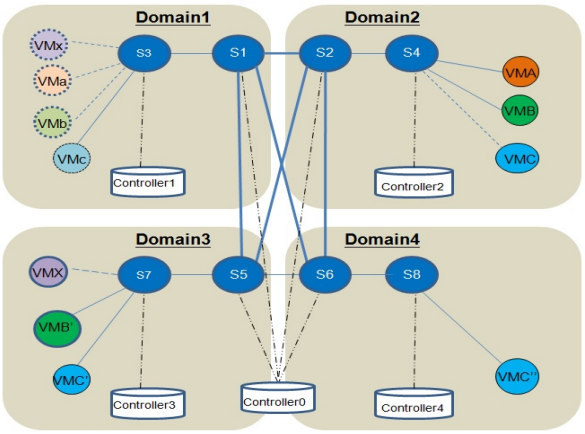

圖7 多對多虛擬機器遷移示意圖

由 3.3 節實驗情境結果可知，情境一時僅須在位於目的端之控制平台 2 撰寫 flow entry，由實驗統計當 FDVM=1 時，新增 2 主動式 flow 插入(PFI)，新增 1 筆隨選式 flow 插入(OFI)。情境二和三：FDVM=1 時，在控制平台 0 上新增 4 筆主動式 flow 插入(PFI)，新增 9 筆回應式 flow 插入(RFI)。在控制平台 1 和 2 上新增 1 筆主動式 flow 插入(PFI)，新增 2 筆回應式 flow 插入(RFI)。表 3和表 4為須要新增到控制平台的 flow 個數。

表3 情境 1 - 控制平台 0 的 flow 個數

Flow Insertion Type	Flow entry number
Proactive Flow Insertion	N+1
On-demand Flow Insertion	N

N = 網域中 FDVM 的虛擬機器個數

表4 情境 2 和 3 -控制平台 0、1、2 的 flow 個數

	Con0	Con1	Con2
Flow Insertion Type	Flow entry number		
Proactive Flow Insertion	3*K+1	K	M
Reactive Flow Insertion	9	2*K	2*M

K = 網域中被遷移出去的虛擬機器個數

M = 網域中 FDVM 的虛擬機器個數

5. 結論與未來工作

在本文中，我們檢視了目前虛擬機器遷移的各種解決方案；PortLand 解決了 Layer2 網域內遷移，VXLAN 解決了跨資料中心的 Layer2 網域內遷移，OTV 解決了跨網域的遷移。而廠商所提供的商業化解決方案，一致性的均須全面佈署支援新協定的網路設備，建置成本過高且缺乏彈性。OpenFlow 可以彈性的根據使用者的需求快速定義網路應用服務，最著名的成功案例是 Google 研發之 OpenFlow 交換器和 OpenFlow 軟體控制平台，可依流量工程計算出最佳路徑，提升 Google 全球資料中心之間傳輸效能至近乎 100%[18]。OpenFlow 應用將會越來越廣泛。而我們在 TWAREN OpenFlow Testbed 上驗證所設計之主動式 flow 插入(Proactive Flow Insertion,PFI)和隨選式 flow 插入(On-demand Flow Insertion, OFI)機制滿足跨網域之虛擬機器遷移的最短路徑。未來將依目前實作之結果，發展出一套自動化的 flow entry 派送軟體安裝於控制平台上，以減少人為設定失誤。尤其，如果遷移的規模更大，需要管理的控制平台複雜度將更高。

參考文獻

[1] http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns1175/Cloud_Index_White_Paper.html

[2] http://datatracker.ietf.org/doc/draft-khasnabish-vmmi-problems/

[3] Mahalingam, Mallik, et al. "VXLAN: A framework for overlaying virtualized layer 2 networks over layer 3 networks." draftmahalingam-dutt-dcops-vxlan-01. txt (2012).

[4] Sridharan, Murari, et al. "NVGRE: Network virtualization using generic routing encapsulation." (2013).

[5] Touch, Joe, and Radia Perlman. "Transparent interconnection of lots of links (TRILL): Problem and applicability statement." (2009).

[6] R. Niranjan Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, and A. Vahdat. Portland: a scalable fault-tolerant layer 2 data center network fabric. In SIGCOMM, 2009.

[7] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson,J. Rexford, S. Shenker, and J. Turner, “OpenFlow: enabling innovation in campus networks,” ACM SIGCOMM Computer Communication Review, vol. 38, no. 2, pp. 69–74, 2008.

[8] Lantz, Bob, Brandon Heller, and Nick McKeown. "A network in a laptop: rapid prototyping for software-defined networks." Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks. ACM, 2010.

[9] Scarfo, Antonio. "The evolution of Data Center networking technologies." Data Compression, Communications and Processing (CCP), 2011 First International Conference on. IEEE, 2011.

[10] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta. VL2: a scalable and flexible data center network. In SIGCOMM, 2009.

[11] Mahalingam, Mallik, et al. "VXLAN: A framework for overlaying virtualized layer 2 networks over layer 3 networks." draftmahalingam-dutt-dcops-vxlan-01. txt (2012).

[12] http://www.cisco.com/en/US/netsol/ns1153/index.html

[13] C. E. Leiserson. Fat-Trees: Universal Networks for Hardware-E_cient Supercomputing. IEEE Transactions on Computers, 34(10):892- 901, 1985.

[14] J. W. Lockwood, N. McKeown, G. Watson, G. Gibb, P. Hartke, J. Naous, R. Raghuraman, and J. Luo. NetFPGA-An Open Platform for Gigabit-Rate Network Switching and Routing. In Proceedings of the 2007 IEEE International Conference on Microelectronic Systems Education, pages 160-161, Washington, DC, USA, 2007. IEEE Computer Society.

[15] http://docs.projectfloodlight.org/display/floodlightcontroller/Static+Flow+Pusher+API

[16] Benson, Theophilus, Aditya Akella, and David A. Maltz. "Network traffic characteristics of data centers in the wild." Proceedings of the 10th ACM SIGCOMM conference on Internet measurement. ACM, 2010.

[17] http://tools.ietf.org/html/draft-yin-sdn-sdni-00

[18] http://opennetsummit.org/talks/ONS2012/hoelzle-tue-openflow.pdf