Azblink NFV 平台

– Secure Multi-OS Workspaces at the Network Edge

摘要

本手冊介紹 **Azblink 網路功能虛擬化（NFV）平台**。本平台將虛擬機與網路安全功能整合於同一硬體上，可同時在其上安裝與執行 Windows、Linux 等作業系統，並透過多組網路介面提供 防火牆 與 路由服務。Azblink NFV 平台可用於 防火牆虛擬化 與 SBC 虛擬化，並支援以滑鼠點選方式，將 虛擬機 快速部署至不同安全區域，大幅簡化網路架構管理與後續擴充。

透過將多台虛擬或實體設備整合為單一 NFV 平台，企業能有效降低硬體與維運成本、縮短導入時程，並在需求變更時以軟體方式快速調整資源配置。對於中小企業或進階個人用戶而言，Azblink NFV 提供了一個兼具安全性、彈性與成本效益的多功能網路及運算基礎環境，可在有限空間與預算下，同時滿足實驗測試、日常辦公與關鍵服務的運行需求，達到導入 “單一安全主機，企業戰力全開，成本大幅縮減” 的目標。

第一章 Azblink 網路功能虛擬化平台的輪廓 (Azblink NFV Platform Outline)

在過去很長一段時間裡，一個服務就配一台實體機：檔案伺服器一台、郵件伺服器一台、防火牆一台、專用應用程式再一台。這種做法雖然直覺，但硬體閒置率高、維護成本大，而且一旦要擴充或調整網路架構，總是伴隨著搬機、拉線、停機與風險。

如今的環境已經完全不同。CPU 核心數持續增加、記憶體與 SSD 價格快速下滑，連一般 PC 或小型伺服器，都已具備足夠的運算能力；光纖與千兆／多千兆乙太網路也日益普及。對多數組織與個人來說，**硬體效能與頻寬不再是主要限制，如何有效運用資源與確保安全，才是新的課題**。

同時，許多關鍵但老舊的應用程式仍然佔據著一整台實體主機：舊版作業系統、只能在特定環境下運作的業務系統、歷史資料庫……這些系統不能輕易關掉，又難以直接升級或搬遷，還可能帶來安全風險。為了避免「動到就壞」，它們常被孤立在角落，卻持續消耗機櫃空間、電力與維運成本。

Azblink 網路功能虛擬化（NFV）平台，正是為了彌補這個落差而設計：一方面充分發揮現代硬體的效能與成本優勢，一方面又能安全地承載既有與新世代應用。在同一台 NFV 平台上，您可以同時：

- 建立多台 Windows / Linux 虛擬主機，承載新舊應用程式
- 於多組網路介面上提供防火牆、路由、VPN、SBC 等網路安全功能
- 透過「區域（Zone）」與「橋接器」精準劃分內外網、DMZ、訪客網等安全邊界
- 將高風險或老舊系統隔離在受控區域內，降低對整體環境的影響

除了企業級應用之外，Azblink NFV 平台也刻意被設計成**任何 PC 擁有者都能運用的「第二台機器」**：

- 在辦公室環境中，Azblink NFV 平台可作為**一站式的安全網路中心**，集中管理 **VPN、檔案服務、內部應用程式和開發測試環境**等關鍵服務。如此一來，員工的個人電腦（PC）**無需安裝複雜軟體或承載機密資料**，有效實現**工作環境的潔淨與安全**。
- 在住家環境中，Azblink NFV 可以扮演整個家庭的「智慧家用主機」與安全中樞：集中承載各種 IoT 與智慧家電控制服務、監視與錄影系統、門禁與感測器平台，同時為小孩的上網裝置建立獨立且受控的網段，並提供回家的 VPN、私有雲、影音媒體伺服器等自架服務。所有這些都在 NFV 主機上運行，而不是分散在多台 PC 或路由器裡，大幅降低風險並讓管理與擴充更簡單。
- 對於開發者與技術愛好者，它則是隨手可用的實驗平台：在一台小型 x86 主機上，就能同時模擬多網段、多防火牆、多種 OS 與服務的真實情境。

對企業與組織而言，這代表：

- **更高的資源使用率**：多個服務共用同一套硬體，不必再為單一功能準備一台實體機。

- **更低的總持有成本（TCO）**：節省硬體、機櫃空間、電力與維運人力。
- **更清楚的安全邊界**：以 NFV 平台為核心，明確定義誰能存取哪些服務與網段。
- **更靈活的部署策略**：先在虛擬環境中驗證，再視需要遷移、擴充或替換。

對個人使用者、小型工作室與 SOHO 族群來說，NFV 平台則提供：

- 一台專門處理「對外連線與對內服務」的第二台機器，不干擾主要 PC 的日常使用
- 可被快速重建、快照還原的測試環境，適合嘗試新系統、新版本與新服務
- 透過 Web 介面就能完成的大部分設定，無需具備深厚的網路背景，也能循本手冊逐步上手

安全隔離 是 Azblink NFV 平台的核心價值之一。透過區域（例如 net / loc / dmz / road）與橋接器（br0、br1…）的搭配，NFV 能讓高敏感度系統、公開對外服務、內部使用系統與家用設備，各自在不同安全等級的區域中運作；即使共用同一套硬體，也能維持清楚、可控的邏輯邊界，避免風險彼此傳染。

本手冊的目的，是協助您**充份發揮 Azblink NFV 平台的潛力**：

- 從橋接器、防火牆、區域與 NAT 的基本概念開始
- 到虛擬主機的建立、網路介面的指派與隔離
- 再到 Port Forwarding、IP 負載平衡、Web Proxy、IP Policy Routing、HTTP 反向代理等進階功能

只要掌握各章節所介紹的原理與操作，無論您是企業管理者、系統工程師，辦公室專業人員 或是在家中打造個人實驗環境的 PC 使用者，都能：

- 在單一 NFV 平台上安全整合多種服務與舊系統
- 依照風險與需求設計辦公室與家庭的網路架構
- 以更低的成本，完成原本需要多台設備才能達成的網路與安全部署

接下來，讓我們從基礎觀念開始，一步一步帶您將 Azblink NFV 平台打造成辦公室與家庭環境中，值得信賴的「第二台機器」與網路核心。

在後續章節中，我們會先定義本文所使用的術語，接著說明各種網路操作的運作機制。

Bridging and Routing (橋接與路由)

我們首先從橋接（Bridging）和路由（Routing）的概念談起。在 OSI（開放系統互連）模型中，如果數據交換發生在**數據鏈路層，稱為「橋接」**；如果發生在**網路層，則稱為「路由」**。在 TCP/IP／**乙太**網環境中，若是根據乙太網幀中的 MAC 地址來判斷數據的去向，就稱為「橋接」；若是依據 IP 標頭中的 IP 地址來判斷，則稱為「路由」。
當討論在數據鏈路層傳輸與接收的數據時，人們通常使用「幀」（frame）這個術語；而在討論

網路層的傳輸與接收時，則使用「封包」（packet）。然而，在本文件中我們不嚴格區分這兩個用語，而是統一以「network traffic （網路流 或 網路流量）」來稱呼這些數據，無論其位於數據鏈路層或網路層。

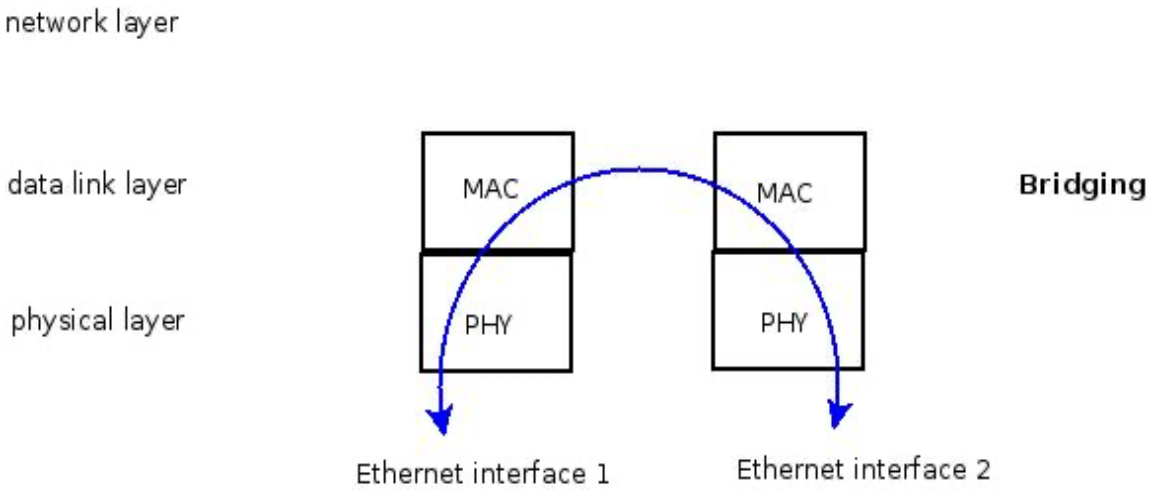


圖 1：橋接操作

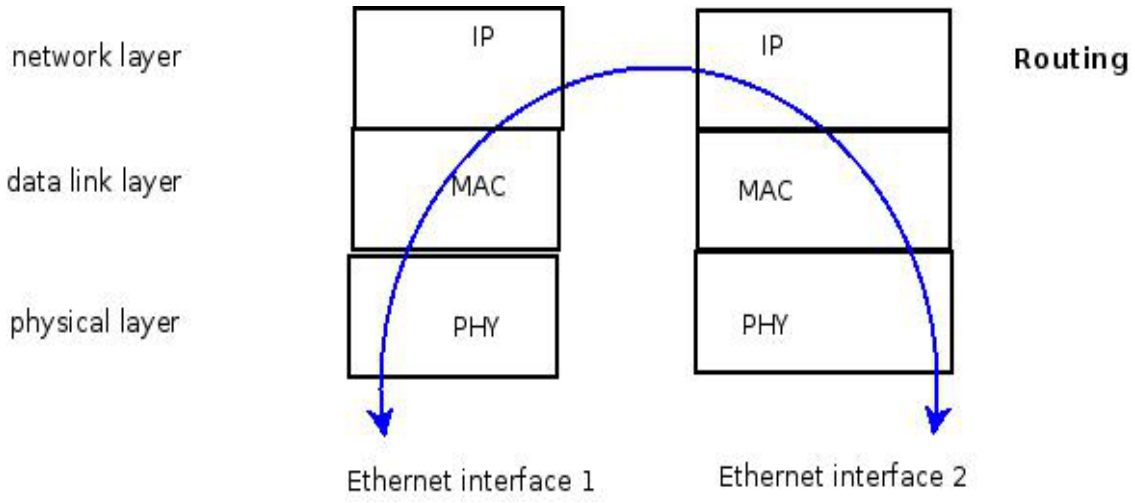


圖 2：路由操作

在本文件中所說的「乙太網」，是指實體層與數據鏈路層（通常分別稱為 PHY 與 MAC）。而 IP（網際網路協議）則屬於網路層。 因此，如果只是根據乙太網幀中的 MAC 位址來判斷網路流該被轉送到哪裡，我們稱之為「橋接」；同樣地，如果是依據 IP 標頭中的 IP 位址來決定路徑，則屬於「路由」的範疇。 對一個網路應用而言，要把網路流送到遠端的另一端點，其實是一個跨越多個層級的複雜過程，當中包含許多細節。

當主機 A 要把 IP 封包送到一台已知 IP 位址的主機 B 時，在乙太網上仍然需要知道目的端的 MAC 位址。這個對應關係是透過 ARP（地址解析協定）來取得的。
主機 A 會先在自己的 ARP 查詢表中，尋找「主機 B 的 IP 位址」所對應的 MAC 位址：

- 如果查詢表裡已經有紀錄，就直接使用該 MAC 位址來送出封包。
- 如果沒有紀錄，主機 A 會在本地子網上廣播一個 ARP 請求。
    - 若主機 B 與主機 A 位於同一個 IP 子網，主機 B 會看到這個請求中的 IP 與自己相符，便回覆自己的 MAC 位址。
    - 若主機 B 位於不同的子網，主機 A 其實是向「預設閘道（路由器）」的 IP 發出 ARP 請求，由預設閘道回覆自己的 MAC 位址，接手後續跨子網的轉送工作。
      本文件不會深入說明 ARP 的所有細節，只會說明與規劃與部署相關的原則。

一般而言，在沒有啟用 VLAN 的情況下，連接到同一台乙太網交換器的網路設備，通常會被配置在同一個 IP 子網中。若需要在不同 IP 子網之間傳送網路流，就必須透過路由器：也就是說，路由器至少要有兩個介面（「兩條腿」），分別連到不同的子網，才能在這些子網之間轉送網路流。

Virtual Bridge and Virtual Host (虛擬橋接和虛擬主機)

Azblink NFV 平台提供「虛擬橋接器」與「虛擬主機」，讓其他作業系統（如 Windows 或 Ubuntu）可以安裝並執行在這些虛擬主機中。

在後續章節中，我們將把 Azblink NFV 平台本身，或任何可以建立虛擬主機的系統，統稱為「基礎平台」、「基礎 OS」或「主機 OS」；而安裝在虛擬主機內的作業系統則稱為「客座 OS」。

在「基礎平台」上，網路操作策略是以「虛擬橋接器」為單位來進行管理。一旦建立一台虛擬主機，與該虛擬主機對應的一個或多個虛擬網路介面也會同步建立。我們會要求您為每一個虛擬主機的虛擬網路介面指定要連接到哪一個虛擬橋接器。換句話說，每個虛擬主機的「虛擬網路介面」都必須被放入某個虛擬橋接器之中。

如果在某個虛擬橋接器中再加入一個實體乙太網介面，那麼該橋接器裡的虛擬主機就可以透過此實體介面，利用「橋接」方式存取「基礎平台」外部的網路。

Virtual Network Bridge v.s. Physical Interface (虛擬網路橋接與實體介面的關係)

在 NFV (網路功能虛擬化) 環境中，主機作業系統（Host OS）上的實體網路介面（例如 eth0）與虛擬網路橋接介面（例如 br0）**是兩個獨立但相互協作的實體**。

它們的常見配置如下：

1. 實體乙太網路介面（例如：eth0）

這是主機伺服器上 實體網路卡 (NIC) 的對應介面。它負責主機與外部實體網路之間的 L1 (實體層) 和 L2 (資料連結層) 通訊。

當它被加入到一個虛擬橋接器時，它通常會移除自身的 IP 位址，轉而成為該虛擬交換機上的一個「埠 (Port)」，專注於傳送和接收 L2 訊框。

2. 虛擬網路橋接介面（例如：br0）

- 這是建立在主機作業系統核心中的一個 **軟體定義 L2 交換機**。
- 它扮演著中央連接點的角色，將所有連接到它的成員埠（包括 eth0 和虛擬網路功能 VNF 的介面）進行 L2 訊框交換。
- 在標準的虛擬化配置中，**主機的 IP 位址會被設定在 br0 上**。這樣，橋接器不僅能為 VNF 提供交換服務，同時也代表主機參與 L3 (網路層) 通訊。

IP 位址配置概覽 (常見情境)

介面	角色	**IP 位址狀態 (一般配置)**
eth0	實體網路存取（橋接器的一個 **埠**）	**無 IP 位址**
br0	軟體 L2 交換機（主機的網路識別身份）	**設定主機的 IP 位址**

總結： eth0 本身~~不再~~擁有 IP 位址，它僅作為 br0 橋接器的一個 L2 成員。因此，主機在該網路區段上的 L3 通訊是透過 **br0 的 IP 位址** 設定來完成的。

當以 admin 登入 NFV 基礎主機後， 在設定頁面 System >> Network >> WAN 上，您可透過 “Ehternet Ports in Bridge br0” 這個設定項目，讓實體網口 例如 eth0, 加入 br0, (這是目前系統預設的)。因此，在 System >> Network >> DHCP 設定頁面上，你只會看到其他橋接器 br1, br2 …br11 的設定項目。 當您在 System >> Network >> WAN 頁上，在. IP Address 項目上 填入 IP, 則意味 您把這個 IP 設定給了 br0。

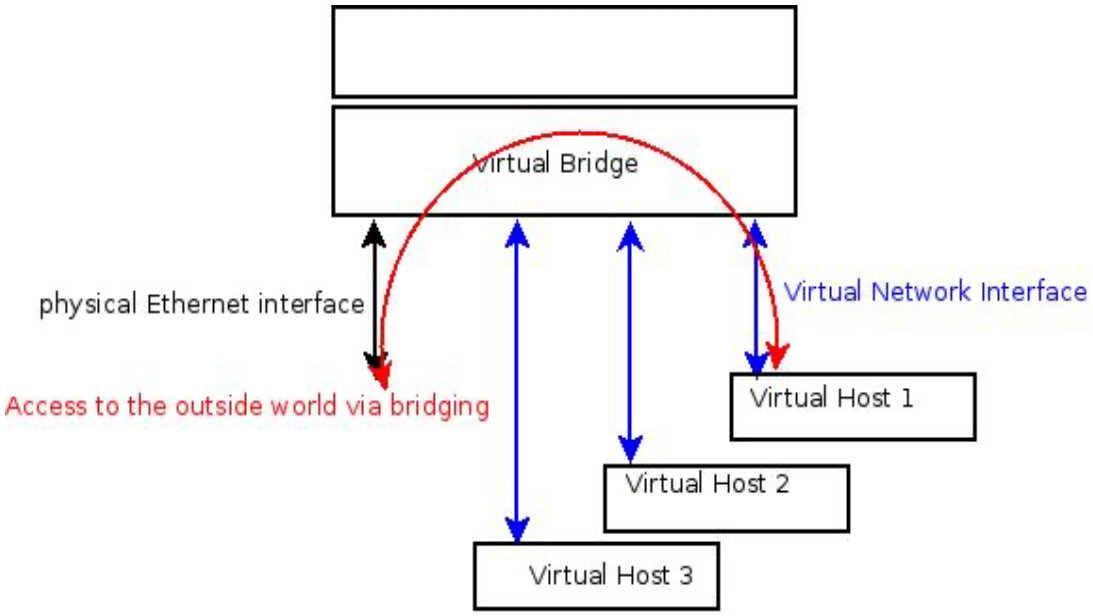


圖 3：橋內交通

在基礎平台上，**「虛擬橋接器」同時也扮演一個「超級乙太網介面」的角色，因此它本身可以設定 IP 位址與子網路遮罩，以標示其所屬的 IP 子網。您可以將橋接器的 IP 位址視為，從外界存取「基礎平台」上本機網路服務時所使用的入口**。從基礎平台的角度來看，橋接器內部的各個網路介面（不論是實體或虛擬）都不會在基礎平台上個別設定 IP 位址。
另一方面，從虛擬主機的角度來看，「客座 OS」會在自己的虛擬網路介面上設定 IP 位址。

具體而言，假設有一個名為「br0」的虛擬橋接器，並已為 br0 設定了 IP 位址。在虛擬橋接器 br0 之內，您可能會看到「eth0」以及其他虛擬乙太網介面——但就基礎平台而言，這些被加入 br0 的乙太網介面本身並不會各自設定 IP 位址。IP 位址只配置在橋接器 br0 上。

在實務上，這種「虛擬橋接器」可以視為一台乙太網交換器。連到同一座交換器的主機，在未啟用 VLAN 的情況下，一般會被配置在同一個 IP 子網中。
將 IP 網路劃分成多個子網，主要目的是把網路廣播與管理範圍限制在較小、相較於 在地化（LAN）的區域，可避免對整體網路造成不必要的影響。
例如，當送出一個 ARP 請求時，所有收到該請求的主機都必須檢查自己的 IP 位址是否與請求中的目標 IP 相符。若這類請求非常頻繁，而實際目標只有其中一台主機，其餘主機花在檢查上的資源就是一種浪費。為了降低這種負擔，IP 網路中的 ARP 廣播被限制在同一個子網內，不會跨越到其他子網。ARP 請求就是透過 IP 廣播送出的；此外，還有其他依賴 IP 廣播的網路協定也是如此。對於必須透過這類協定彼此溝通的主機，應該被放在同一個 IP 子網中。

但在某些情況下，虛擬主機送出的網路封包仍然可能需要跨越子網。如下圖所示：

- 如果目標主機所在子網的閘道，可以經由某一實體乙太網介面到達，那麼來自虛擬主機的封包會先透過該實體介面送到此閘道，再由閘道轉送到目標子網。
- 同時，「基礎平台」本身也在這個橋接器上配置了 IP 位址。當虛擬主機送出的封包，其目的 IP 所屬的子網由基礎平台所連接時，這些封包也可以先送達基礎平台，再由基礎平台執行路由處理，轉送到對應的子網。

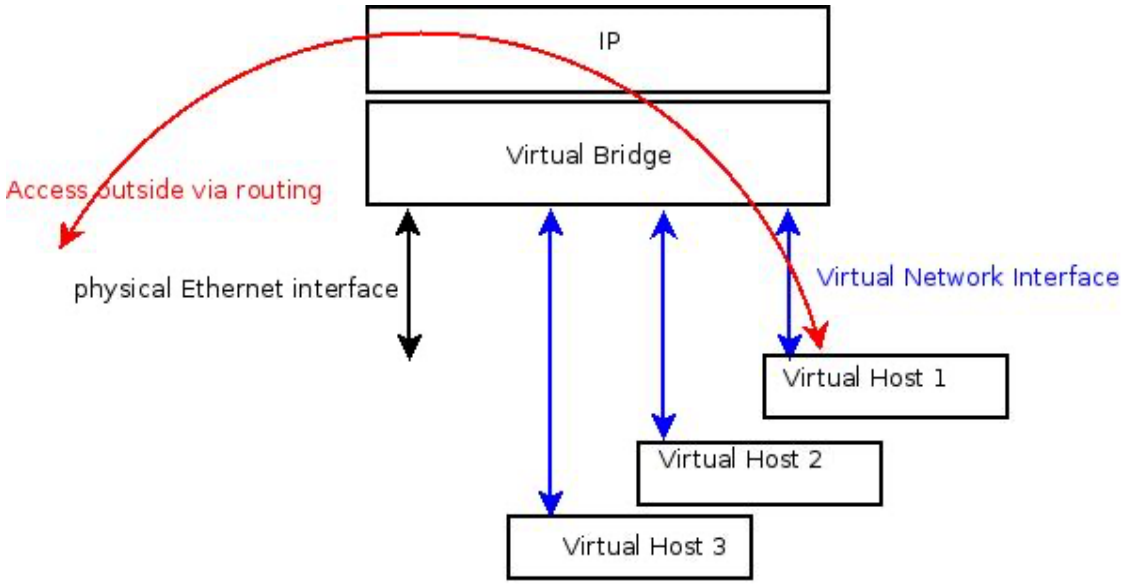


圖 4：交通在另一子網的橋接器上通過

綜合上述，雖然虛擬橋接器與虛擬主機都是由基礎平臺在其內部建立的資源，但從 IP 網路層的角度來看，基礎平臺本身就是一台與各虛擬主機並列的主機。當虛擬主機中的網路應用程式需要存取基礎平臺時，只要將基礎平臺的 IP 位址設定為目的位址並發出網路請求即可。

Intentionally Unsupported Bridge Network Operations (刻意不支援的橋接網路操作)

理論上，在虛擬主機與其於基礎平台上所連結的實體乙太網介面之間，還可以加入各種不同的網路處理方式。本節將列出幾種 **本產品刻意不提供** 的網路操作。
不提供這些做法的原因，是因為 **Azblink NFV 平台是為「伺服器型」應用而設計，而這些操作方式多半並不適合伺服器型場景。** 儘管如此，為了說明完整性，我們仍會在本節簡要討論這些可能的作法。 分別說明如下：

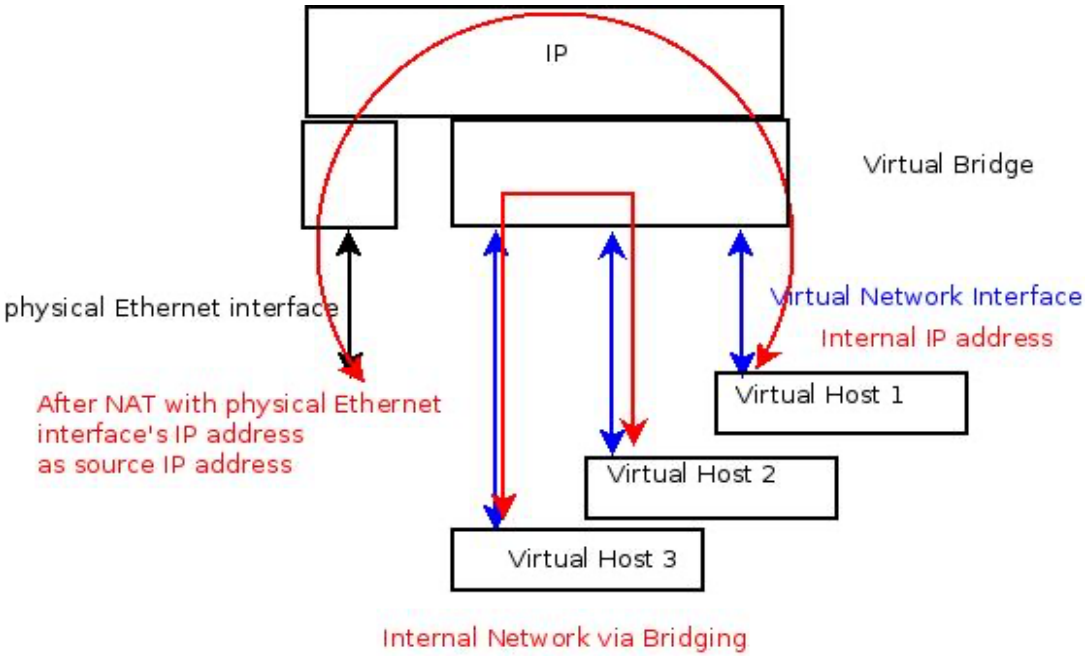


圖 5：按物理埠為基礎的 NAT

上圖說明，在啟用 NAT（網路位址轉換）之後，虛擬主機會被隱藏在後方，外部網路無法直接看見它。這種情況通常出現在：先在「基礎 OS」上，直接在實體乙太網介面設定一組 IP 位址，之後再在其上建立虛擬主機。
當虛擬主機建立完成時，系統會同時建立一個對應的橋接器：

- 原本設定在實體乙太網介面上的 IP 位址會被移除，改由這個橋接器來使用；
- 該實體乙太網介面被加入橋接器，成為橋接器的一個成員介面；
- 虛擬主機的虛擬網路介面則被指派一組私有 IP 位址。
  當虛擬主機需要存取「基礎 OS」之外的主機時，從虛擬主機送出的封包會先經過基礎 OS，在那裡其來源 IP 會被轉換成橋接器的 IP 位址（也就是原本實體乙太網介面所使用的那個 IP 位址），然後再送往外部網路。

反過來說，如果要從「基礎 OS」外部主動連線到這台虛擬主機，因為它使用的是私有 IP 位址並被 NAT 隱藏，就必須在基礎 OS 上設定對應的埠轉發（port forwarding），將外部的連線埠對映到虛擬主機的內部 IP 位址與埠號。

下圖示範一種情境：多台共用同一組對外 IP 位址、位於同一個 NAT 後方的虛擬主機，只要被接在同一個橋接器上，就能直接以各自的私有 IP 位址互相通訊，而不需要再經過 NAT。

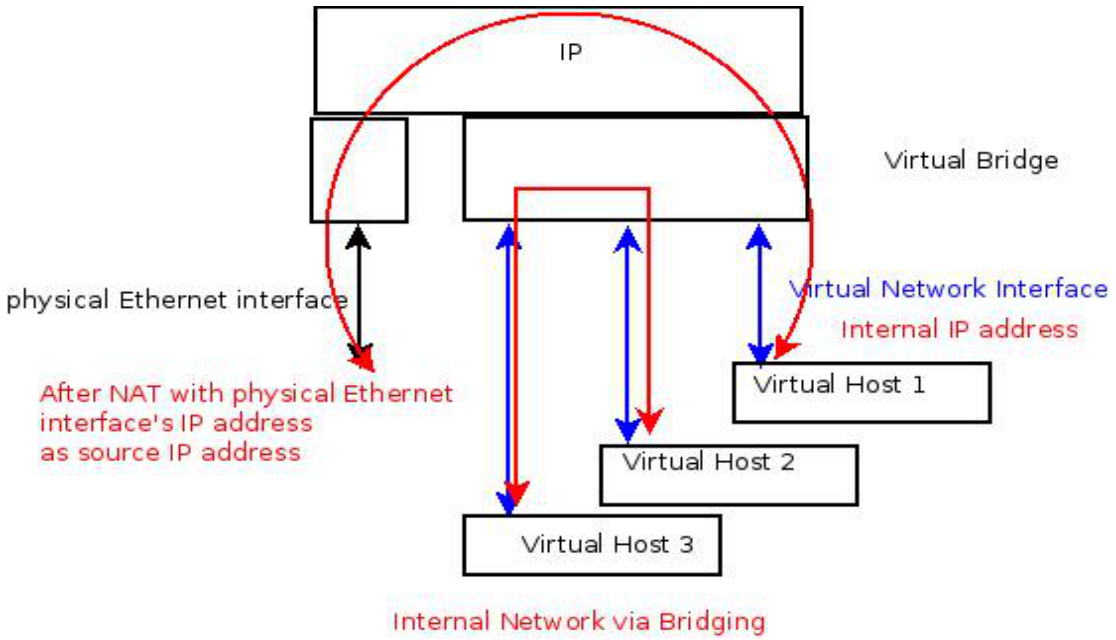


圖 6：允許在同一橋上虛擬主機之間發送流量

另有一種情況是：即使多台虛擬主機位於相同的 NAT 對外 IP 位址之後，它們之間仍無法彼此通訊。這些虛擬主機只能透過 NAT 存取外部世界，卻不能使用自己的私有 IP 位址互相連線。這種設計常見於雲端環境，例如兩台虛擬主機分屬不同公司或租戶時，平台會刻意隔離它們之間的網路存取。

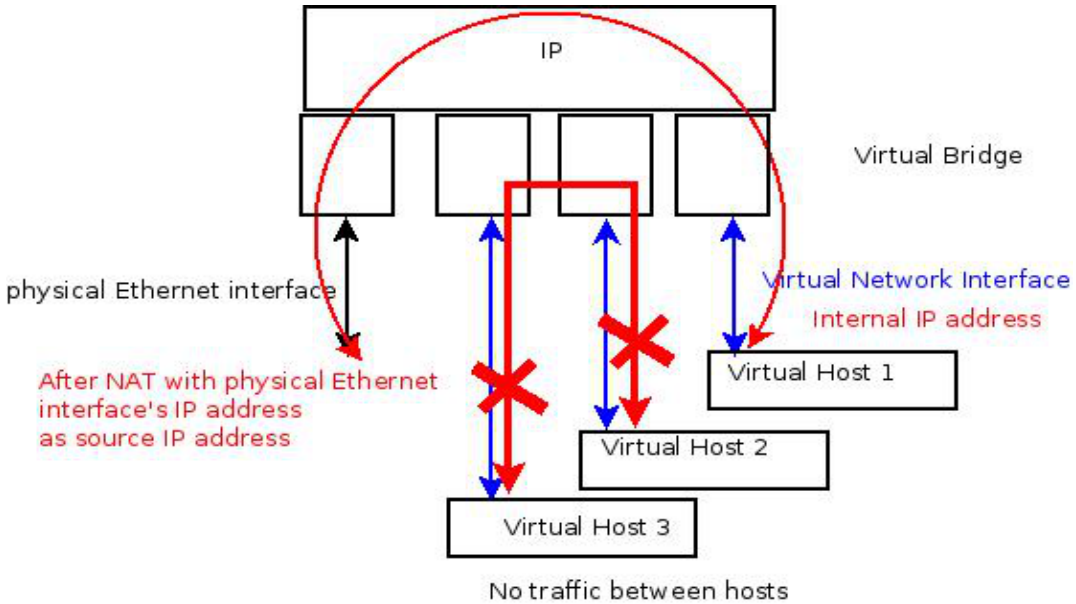


圖 7：在此設定中，即使虛擬主機位於同一個橋接器內，也不允許它們之間直接傳遞網路流。（ 不支援）

下圖說明，在移除 NAT 功能之後，虛擬主機將無法存取外部網路，只能透過同一個橋接器內的本地私有 IP 位址彼此通訊。

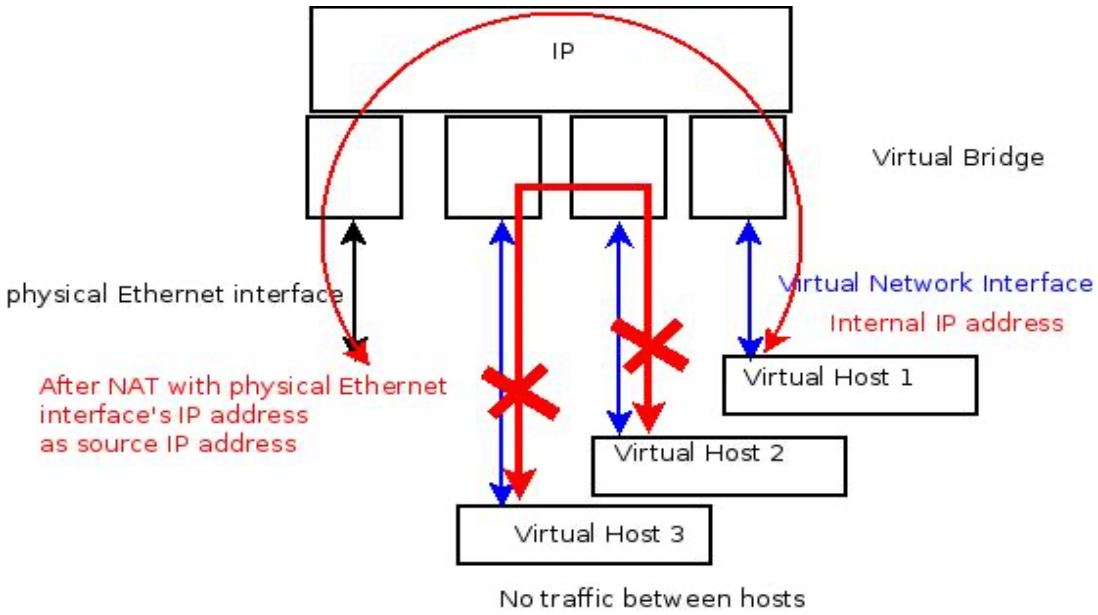


圖8:「所有網路流都被限制在同一個橋接器內部，無法進出該橋接器。」（不支援）

上述這兩類操作並非我們基礎平台所提供的功能。您只需要決定每個虛擬網路介面要連接到哪一個虛擬橋接器即可。

接下來的段落，則會說明 **Azblink NFV 平台實際所提供的功能與操作模式。**

如果在系統部署期間確實需要某些功能（例如 NAT），我們會改由「基礎平臺」上的防火牆與路由規則來管理整體虛擬橋接器的配置； 每一個橋接器都會被賦予預先定義的角色與功能。

例如，當來自子網 A 的網路封包（經由某一橋接器接入）被允許轉送到子網 B（經由另一橋接器接入）時，若在基礎 OS 中設定需進行 NAT，封包的來源 IP 位址就會被改寫為通往目標子網之橋接器的 IP 位址。
相關設定與行為將在下一節中進一步說明。

How the Bridge-based Firewall Works (橋接器防火牆如何運作)

Azblink NFV 平台虛擬橋接器互通與安全設計優化說明

Azblink NFV 平台透過預先定義的虛擬橋接器，將虛擬網路流量劃分為多個安全區域，並在底層統一實施嚴格的存取與路由規則，特別設計 br2 作為 DMZ 區域。

核心設計與規則（以 br0, br1, br2 (DMZ), br3 為例）

1. 廣域網 (WAN) 出口路由與 NAT 機制

- **出口統一 NAT：** 任何來自內部虛擬網路 (br1, br2, 或 br3) 且目標是廣域網 (WAN) 的網路流量，都必須經由 br0 出口。
- **位址轉換：** 在基礎平台上，所有經由 br0 出口的流量都會自動套用 **NAT (網路位址轉換)**。
    - **結果：** 封包的來源 IP 位址會被改寫為 br0 介面的 IP 位址，確保內部虛擬 IP 不暴露於外部網路。

2. 內部橋接器間的通訊規則與安全區域劃分

區域	簡述	對內連線規則 (內部主動發起)	對外連線規則 (外部主動進入)
br1	內部受限區	**不允許** 主動連線到其他 IP 子網。	來自其他子網連往 br1 的網路流，則可以依規則被允許。
br2	**DMZ (非軍事區)**	預設對其他內部區域 br1, br3 **高度隔離**，僅允許有限的服務連線（例如後端資料庫）。	專為**外部服務連入**設計，通常允許來自 br0 /WAN 的特定服務連線進入。
br3	內部信任區	允許與 br1 之間進行 **雙向通訊**。	預設為內部使用，通常不允許外部主動連入。

3. 來自 br0 子網的隔離限制

高隔離性： 任何由 br0 所在子網內（管理或核心服務區）的主機所發起的網路流，不得進入其他虛擬橋接器（如 br1, br2, br3）。

目的： 確保主機管理網路與 VNF 服務網路之間存在嚴格的隔離界線，提升系統安全性。

這種配置帶來的優勢

1. **多層次安全防護：** 透過將服務區分為 br1 (受限)、br3 (信任) 和 br2 (DMZ)，能根據服務的敏感度、信任度及是否面向公網，提供 **精確的安全控制**。
2. **DMZ 隔離外部風險：** 將對外服務（如 Web Server、Mail Server）部署在 br2 (DMZ) 中，即使服務被攻破，由於 br2 對於 br1 和 br3 內部區域是隔離的，也能**有效限制攻擊者對核心內網的滲透**。
3. **全域集中管控 (Centralized Control)：** 只要事先規劃好橋接器之間的互通規則，防火牆或安全政策就能在 **全域層級** 統一管控所有區域之間的存取權限，簡化複雜的多層級安全策略部署。

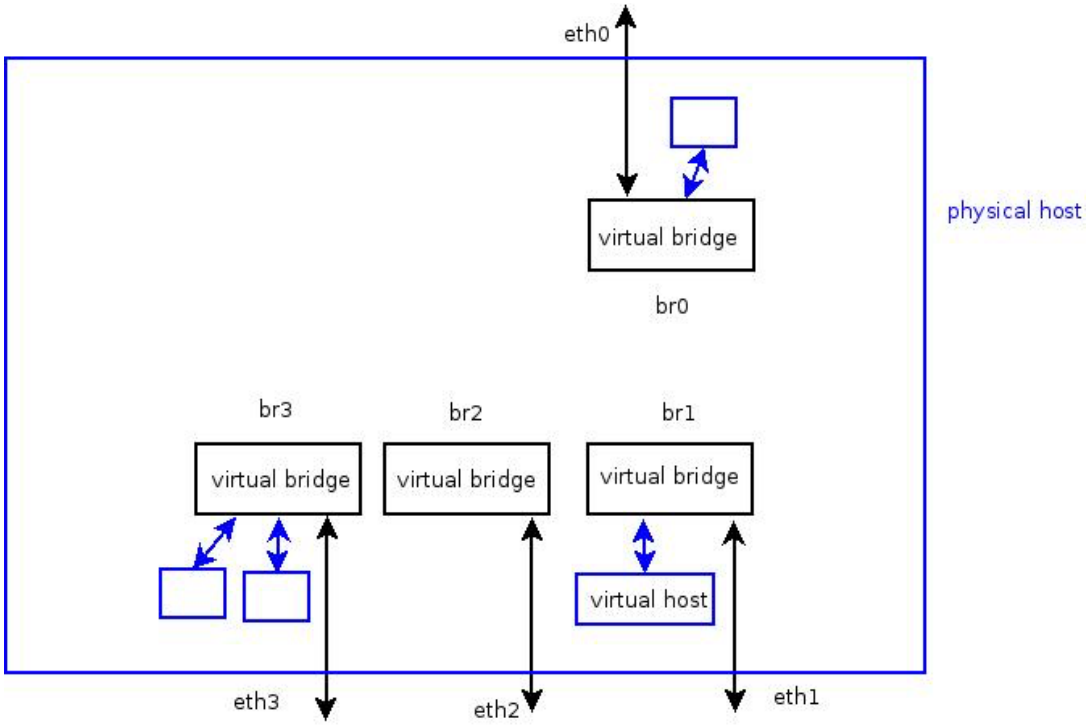


圖 9: 基礎平台內的虛擬橋接器

虛擬橋接器 (br1/br2) 對外通訊與 NAT 機制

下圖所示為 Azblink NFV 平台中網路流經由 br0 出口的統一 NAT 處理機制。此規則確保了所有從內部安全區域 (br1 或 br2) 送往外部廣域網路 (WAN) 的**流量，都能統一且安全地透過單一出口 br0 進行發送。**

核心機制說明：

1. **流量路徑：**
   - 凡是從 br1 區域或 br2 區域發起，且目標是外部網路的網路流量。
   - 必須經由 br0 介面作為**出口閘道 (Egress Gateway)** 送出。
2. **統一的 NAT 處理：**
   - 無論這些網路流原本是來自**虛擬主機 (VNF)**，還是位於 br1 或 br2 橋接器之下的**其他網路介面**。
   - 在離開基礎平台時，都會在 **基礎平臺 (NFVI)** 上層套用 **NAT (網路位址轉換)**。
3. **位址改寫結果：**
   - 封包 IP 標頭中的**來源 IP 位址**，將被改寫為 **br0 介面所配置的 IP 位址**。

優點與意義：

- **網路統一性：** 外部網路只會看到來自 br0 的單一 IP 位址，簡化了外部網路的管理與路由。
- **內部隱藏性：** 確保了 br1 和 br2 內部虛擬子網的 IP 位址不直接暴露於公網，提升了內部網路的安全性與隱私性。
- **資源共享：** 允許多個虛擬網路區域共享同一個公網 IP 位址進行連網。

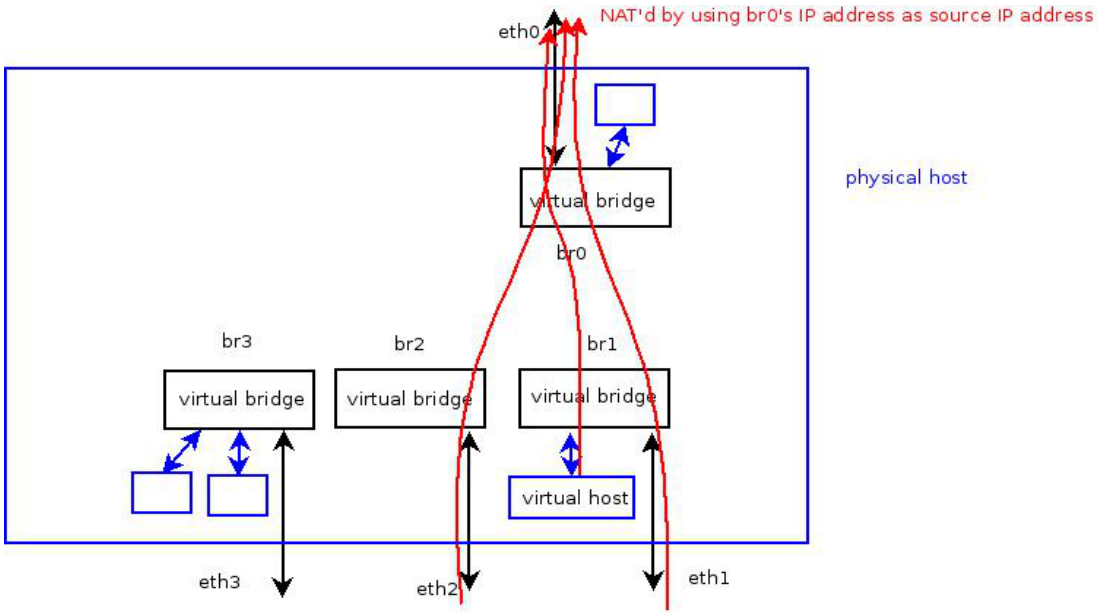


圖 10：跨越 br0 邊界時的 NAT

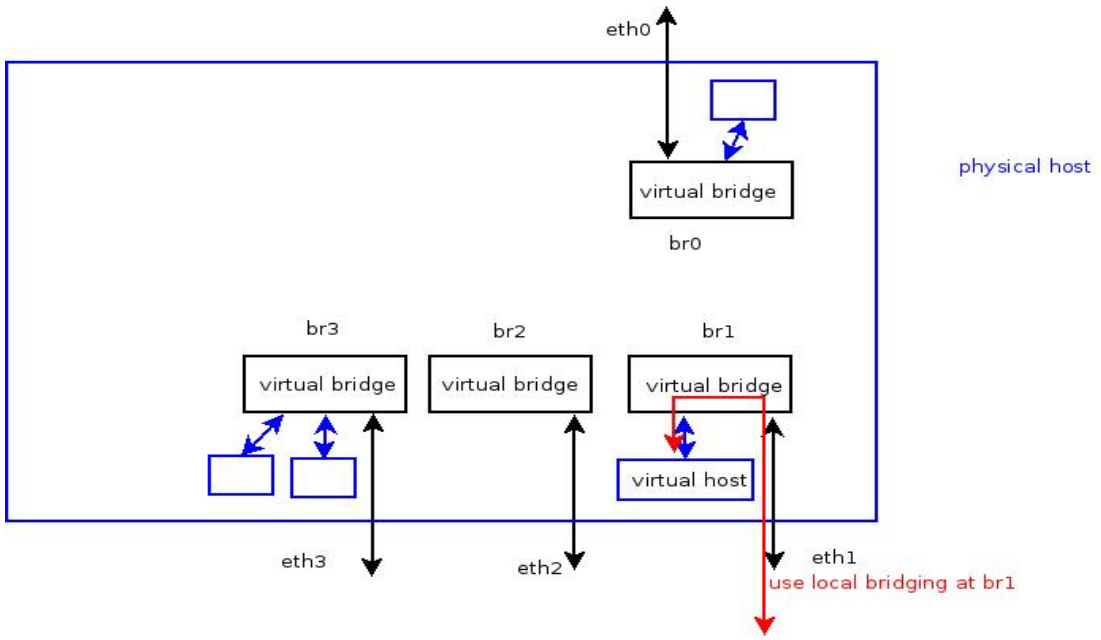


圖 11：橋接 br1 內沒有 NAT 操作。

VNF 與基礎平臺間的存取控制機制

預設政策：連線主要由 VNF 內部控制

在預設情況下，這些連線**並不會**受到「基礎平臺 (NFVI)」的額外限制。連線是否允許存取，主要取決於**虛擬主機 (VNF)** 內部（即客座 OS）本身所實作的**存取控制機制**（例如內建防火牆或安全群組）。

基礎平臺的強制安全控管 (NFVI 防火牆)

若虛擬主機端的管控**缺乏足夠的強度或細緻度**，則「基礎平臺 (NFVI)」上的防火牆可以作為**第二層防護**：

- NFVI 防火牆可施加**額外且強制性的規則**，對這些連線進行更細緻、更高級別的安全控管，確保安全性不會被虛擬主機的錯誤配置所繞過。

實施範例：隔離核心網路

此機制的一個典型應用是保護 br0 區域（通常為核心管理或出口網路）：

- NFVI 可設定強制規則，禁止 br1 或 br3 下的虛擬主機**直接**存取 br0 橋接器下的主機。
- 除非是透過事先設定好的 **埠轉發 (Port Forwarding)** 機制，允許特定服務或埠進行例外連線，否則嚴格隔離此類主動連線。

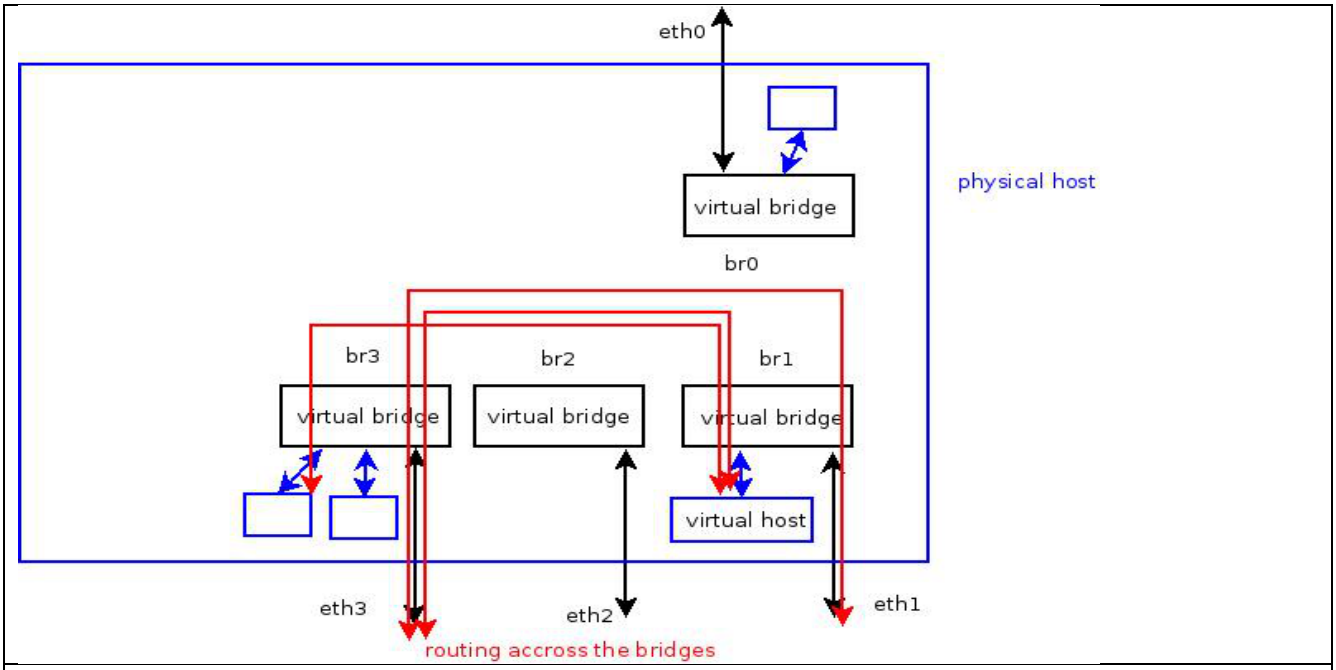


圖 12：在 br1 與 br3 這兩個橋接器之間轉送網路流時，不應套用 NAT（網路位址轉換）。

虛擬主機 (VNF) 的網路獨立性與身份隔離

儘管 NFV 環境中的虛擬主機 br1 與 br3 下的主機共用同一組實體硬體資源，但從網路層面來看，它們是**彼此獨立且擁有自主身份的節點**。

網路流交換與路由

- **透過基礎平台路由：** br1 與 br3 之間的主機可以透過**基礎平台 (NFVI) 所提供的路由流程**來交換網路流。這意味著基礎平台扮演著中介路由的角色，確保不同虛擬區域之間的連線能力。

邏輯網路身份的獨立性

實體層面 (Physical Layer)	**邏輯層面 (Logical Layer)**
運行在**同一台實體設備上**，介面被接在同一個**虛擬橋接器 (Bridge)** 內。	虛擬主機與基礎平台是**彼此獨立的網路節點**。
共享同一組硬體資源。	每台虛擬主機都擁有**自己的網路身份**。

核心原則：IP 身份不可共用

- 虛擬主機的連線，只能透過**自身的 IP 位址**來存取。

- 它們**不會**因為位於同一台實體設備上，就自動**共用基礎平台 (NFVI) 的網路身份或 IP 位址**。

結論：

在邏輯上，這些虛擬主機應被視為與基礎平台**獨立的外部主機**，它們只是恰好共用同一組硬體資源，這種設計是實現 **NFV 隔離性**和 **多租戶環境**的基石。

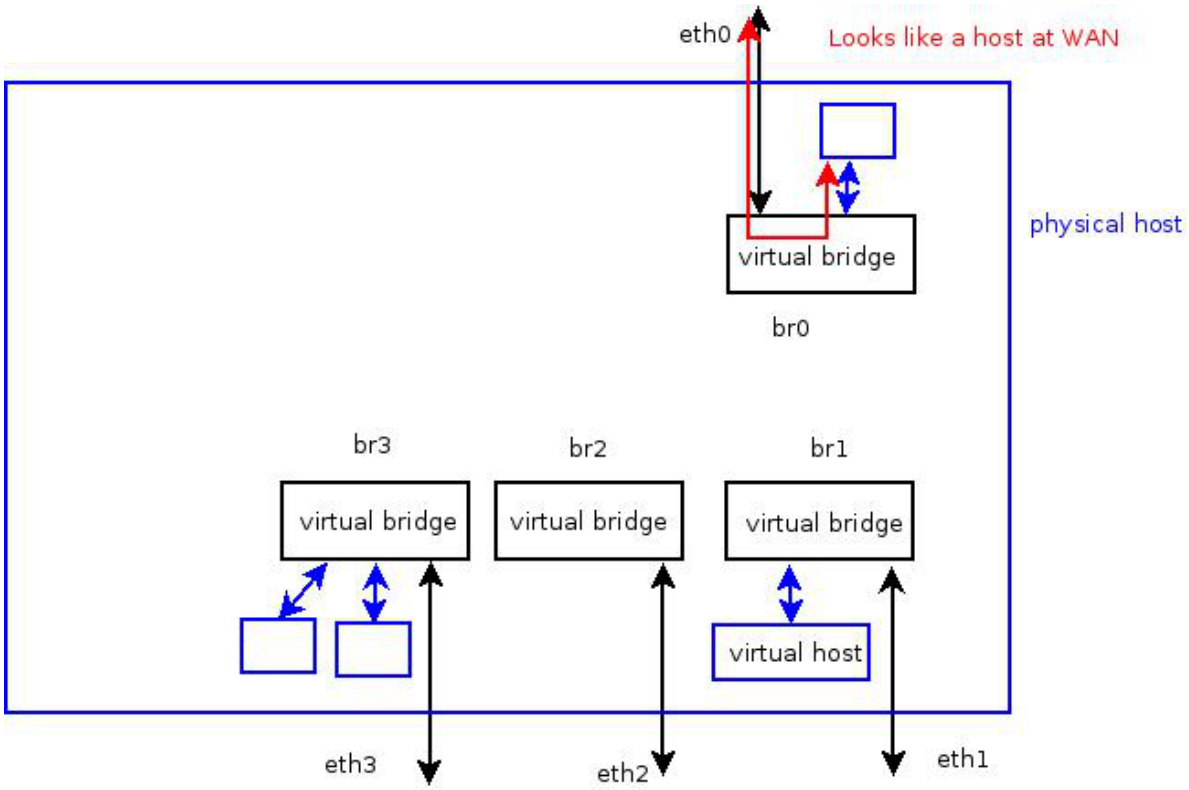


圖 13：虛擬主機連接到橋接 br0

br0 區域虛擬主機的網路安全與隔離

我們以在 br0 底下建立的一台虛擬主機為例，來解釋其網路身份和安全特性。

br0 虛擬主機的網路行為：

- **IP 配置：** 橋接器 br0 本身配置了一組 IP 位址，代表**基礎平臺 (NFVI) 在該子網中的網路身份**；而虛擬主機也配置有自己的 IP 位址，且與 br0 位於**同一個 IP 子網**內。
- **流量路徑（對外）：** 雖然這台虛擬主機對外（通往網際網路）的網路流會經由 br0 送出，但這些封包的處理方式是：
    1. **直接橋接轉送：** 封包**並不會**進入基礎平臺的防火牆流程進行檢查。
    2. **來源位址保持不變：** 封包會以**虛擬主機本身的 IP 位址**作為來源位址，直接透過橋接器轉送出去。

- **安全後果：** 因此，這台位於 br0 下的虛擬主機**沒有受到基礎平臺防火牆的保護**。
- **邏輯視角：** 從 br1 或 br2 所在網段的觀點來看，br0 下的這台虛擬主機等同於**一台位於基礎平臺防火牆之外的外部主機**。

多腳 (Multi-legged) 虛擬主機與直接通訊

一台虛擬主機可以有多個「腳」（legs），意指它可以配置**多個虛擬網路介面 (vNICs)**，同時連接到多個虛擬橋接器，以實現更靈活的網路設計。

以下圖所示為例，一台虛擬主機的 vNICs 分別連到橋接器 br3 與 br4：

1. **直接通訊路徑：** 當這台虛擬主機要存取位於 br3 與 br4 下的兩個不同子網時，它可以**直接透過各自連接的虛擬介面**與這兩個子網進行通訊。
2. **繞過路由：** 在這種配置下，虛擬主機**不需要經過基礎平臺的路由程序**來實現 br3 和 br4 子網之間的通訊，因為這兩個子網是作為主機的「本地」連線而存在的。
3. **協議兼容性：** 由於有些網路協定（例如某些多播或 L2 協定）**無法跨越 IP 子網進行路由**。透過這種多介面的配置，該虛擬主機便能在 br3 與 br4 所屬的兩個子網上，分別運行這類協定與應用，大幅提升應用部署的彈性。

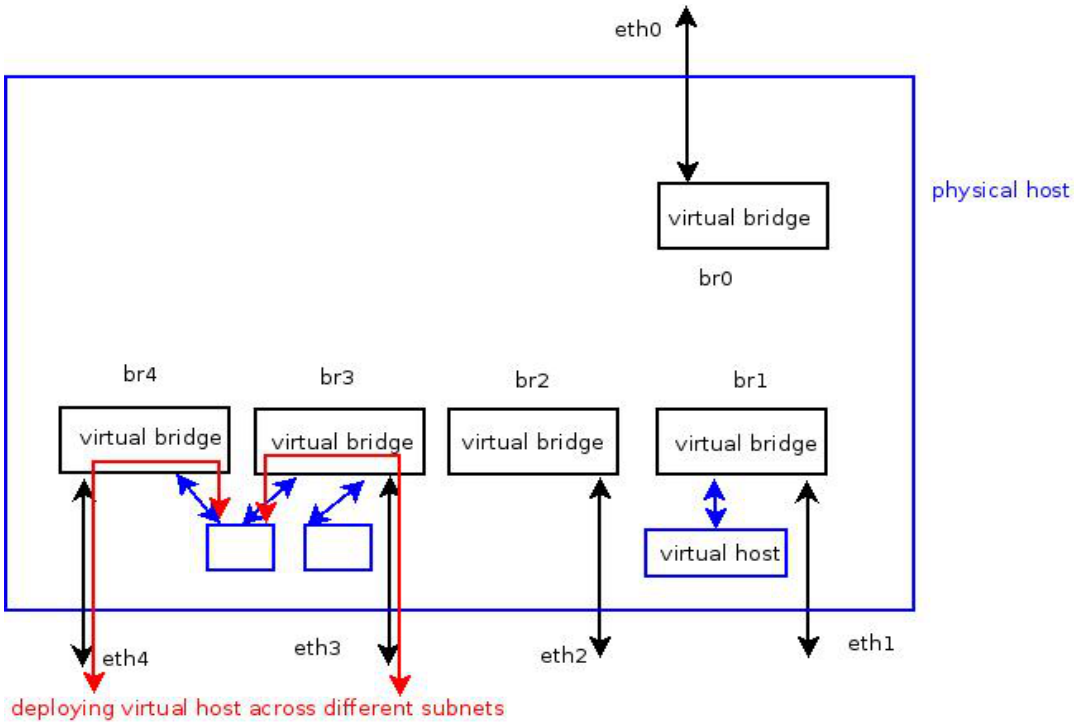


圖 14：虛擬主機連接到多個橋接器

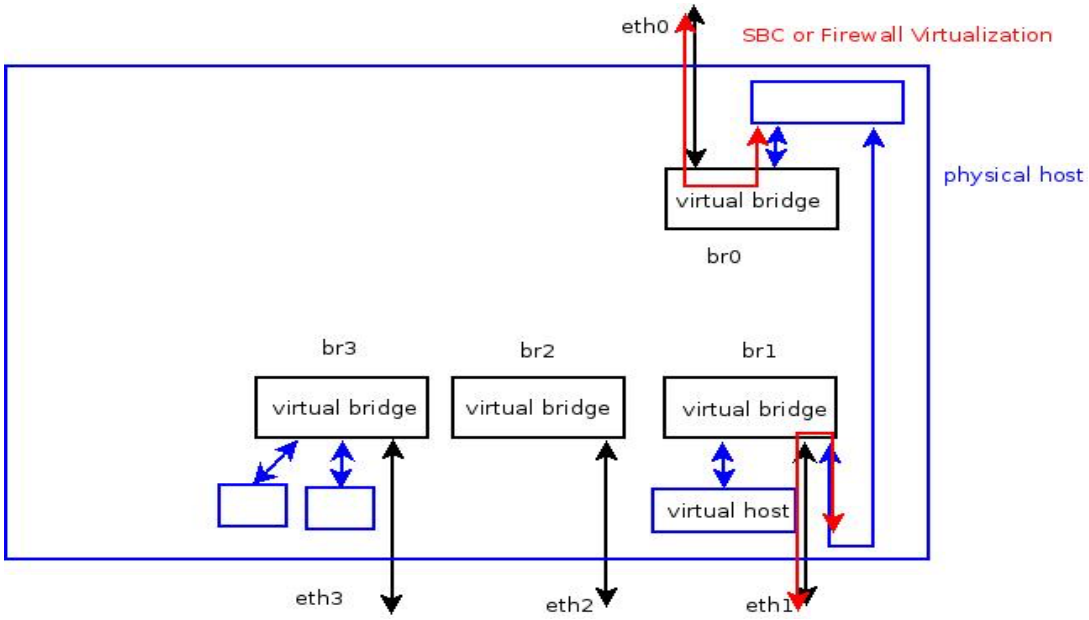


圖 15：虛擬主機連接到橋接器 br0 和 br1

Firewall or SBC Virtualization (防火牆與 SBC 虛擬化)

在 Azblink NFV 平台上，透過為虛擬主機配置兩個虛擬網路介面（分別掛載於不同的橋接器下），即可將其高效地實作為**虛擬防火牆 (Virtual Firewall)** 或**虛擬會話邊界控制器 (Virtual Session Border Controller, SBC)**。

虛擬安全設備的實現架構

若一台虛擬主機同時擁有：

- 一個介面掛載於 **br0** 下（作為廣域網 (WAN) 介面）。
- 另一個介面掛載於 **br1** 下（作為區域網 (LAN) 介面）。

這使得這台虛擬主機非常適合扮演網路邊界控制的角色，例如作為虛擬防火牆或虛擬 SBC。

雙層防火牆的選擇與強制路由

由於**基礎平臺本身也是一套防火牆**，對於掛在 br1 的子網而言，就會出現「流量應該經過基礎平臺防火牆，還是虛擬防火牆？」的選擇問題。

- **預設路由：** 若處於 br1 子網中的主機將該**虛擬防火牆的 IP 位址**設定為其**預設閘道 (Default Gateway)**，則所有前往網際網路的網路流將會先送到虛擬防火牆，再由它轉送到外部。

- **強制路由：** 若希望**強制** br1 子網中的主機一律經由虛擬防火牆上網，可以採取以下兩個步驟：
    1. **關閉 br1 的 DHCP 服務：** 阻止基礎平臺在 br1 子網上發放預設閘道資訊。
    2. **新增基礎平臺封鎖規則：** 在基礎平臺上新增一條封鎖規則，**禁止 br1 => br0 的直接通行**。
    3. **結果：** 如此一來，br1 子網中的主機若要連到網際網路，就只能透過這台虛擬防火牆作為閘道。

虛擬 SBC 部署： 透過相同的方式，也可以將該多介面虛擬主機部署成「虛擬 SBC」，讓相關的 SIP／RTP 網路流統一先經過它再通往 WAN。

虛擬網路架構的邏輯視角

為了避免把「基礎平臺」與「虛擬主機」的身份混淆在一起，我們可以從一個更清晰的視角來理解網路架構：

- **將橋接器視為交換器：** 把每一個虛擬橋接器 (br0, br1 等) 想像成一台**獨立的「乙太網交換器」**。 也就是說，每一個 br0, br1… br11, 每一個就是一台交換器，若硬體主機上有多個網口，則代表 這台基礎平台最多可提供 12 台交換器 並接到了基礎主機了， 而每一個 “硬體實體接口” 就是每個獨立交換器 對外界的 硬體實體 接口。
- **若以一般的理解，其實 Azblink NFV 的這部分，是把一般所見的硬體 “路由器，管理型交換器，防火牆，VPN “，用虛擬軟體方式實現。 對於這四種設備的代表的觀念及使用，基本上不變。只是這些都是以軟體虛擬化了，而不是一台台 分別的硬體主機了。 這是為何 Azblink NFV 可提供如此 節約及提高效率的原因。**
- **節點連接：**
    - **基礎平臺：** 透過多個網路介面（legs）連到這些「交換器」，並在各個橋接器介面上設定對應的 IP 位址。
    - **虛擬主機：** 也透過自己的虛擬網路介面連到這些「交換器」。
- **網路結果：** 如此一來，每一台「乙太網交換器」（即一個橋接器）底下，都會有一群位於**同一 IP 子網**中的主機——其中既包含**基礎平臺**，也包含**各個虛擬主機**。

系統管理與網路維運的優勢

- **配置彈性：** 虛擬化的優點在於，這些「網路介面要接到哪一個橋接器」的關係，都可以純粹透過**軟體重新配置**，不需要實際安裝額外的網路卡或重新佈線。
- **內建路由協定：** 本平臺內建 **OSPF 與 RIPv2 路由協定**，可自動與其他路由器交換路由表，從而簡化多路由器環境中的網路規劃與維運。
- **遠端主控台：** 基礎平臺同時也提供了 **「網路化的虛擬主機主控台」** 的存取功能，管理者可以從遠端直接連線到虛擬主機的主控台，而「客座作業系統」本身無須安裝任何額外軟體，即可讓使用者或管理者直接存取。

- **雙重防火牆選項：** 基礎平臺本身內建一套稱為 「**Border Control**」 的防火牆功能。此外，您也可以依照前述多介面方式，在虛擬主機內自行實作**軟體防火牆**，提供更多的安全選擇。

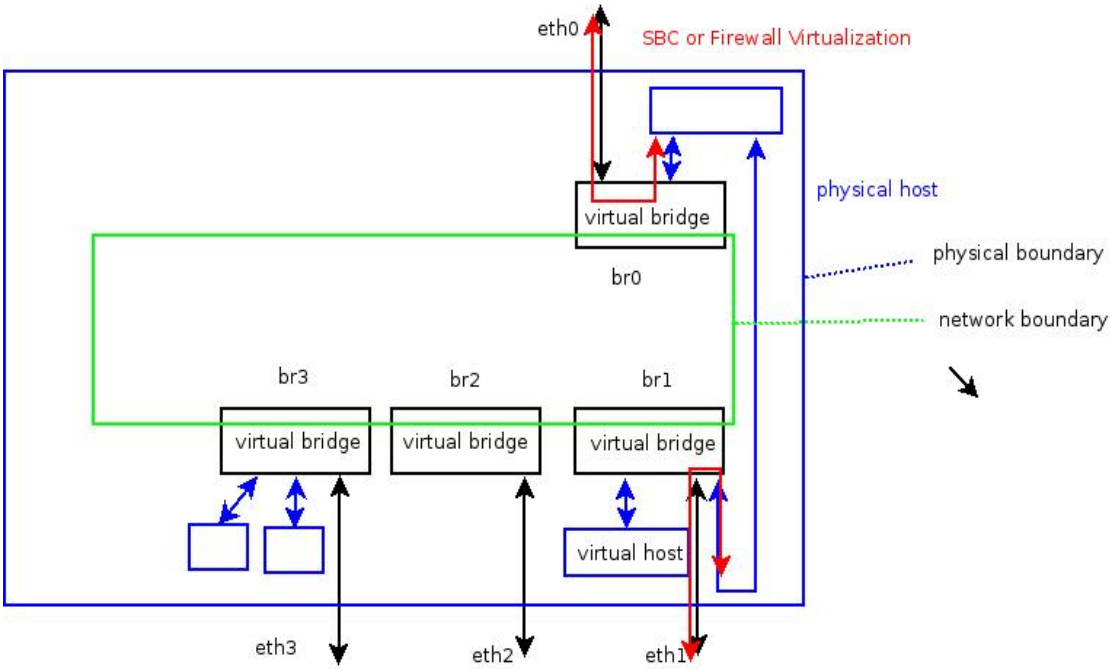


圖 16：防火牆虛擬化中的虛擬主機連接

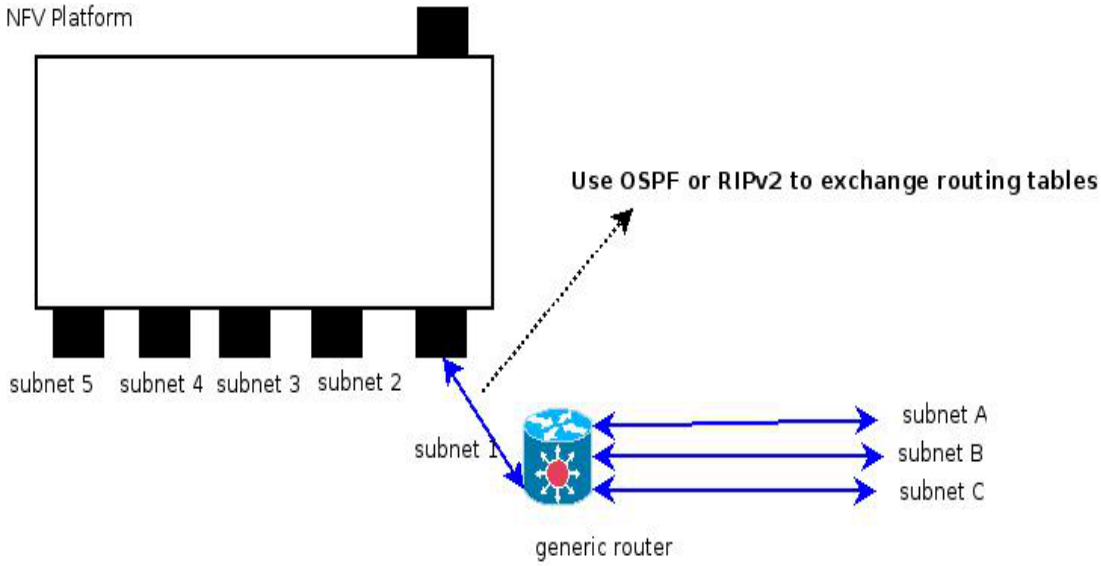


圖 17：基礎平台與其他路由器

在下一章中，我們將介紹如何在基礎平臺上建立與管理虛擬主機。

第 二 章 虛擬主機 (Virtual Host)

使用虛擬主機的目的，是讓它「看起來與實體主機一樣工作」。
許多應用程式仰賴實體硬體介面（例如 RS-232 串列埠、藍牙等），才能完整運作。
然而，本產品在此僅專注於**網路型應用程式**，也就是主要透過網路介面進行通訊的應用。
如果你的應用程式嚴重依賴實體硬體介面，就需要特別留意這一點。

在概念上，「虛擬主機」可以被視為一個**承載作業系統的容器**。
從作業系統的角度來看，虛擬主機提供了 CPU、記憶體、儲存空間、網路介面以及其他周邊裝置，就像一台實體機器一樣。

為了區分不同角色，我們採用以下用語：

- 提供「虛擬主機」能力、負責模擬硬體的平台，稱為
  「host OS」／「host system」／「base platform（基礎平臺）」／「base OS」。
- 安裝在虛擬主機中、實際執行應用程式的作業系統，稱為
  「guest OS（客座作業系統）」／「guest system」。

在一般說法中，「虛擬主機」有時會包含其中的 guest OS；但在本節中，
「虛擬主機」僅指那一組被模擬出來的硬體資源本身（CPU、記憶體、磁碟、網路介面等），不包含 guest OS。

在建立虛擬主機之前，必須先決定要模擬的硬體規格，例如：

- 記憶體容量
- 磁碟／儲存空間大小
- 分配給虛擬主機可使用的 CPU 數量
- 虛擬主機需要多少個乙太網介面

其中有些屬性最好在一開始就決定好，有些則可以之後再調整。例如，**磁碟空間大小可以之後擴充**，但在擴充之後，往往仍需在 guest OS 內重新分割與調整檔案系統，才能實際使用新增空間。這與「虛擬主機可用記憶體大小」不同：記憶體的調整通常會在虛擬機關機後修改設定，下次開機即可生效。這些設計上的限制與差異，都需要在規劃虛擬主機時先行考量。

一個基礎平臺可以同時執行多個虛擬主機實例。
如果「所有已啟動虛擬主機要求的記憶體總量」超過基礎平臺實際可提供的實體記憶體，就可能發生以下狀況：

- 某些虛擬主機無法順利啟動；
- 或者部分虛擬主機在運作過程中遭遇記憶體不足的問題。

因此，在規劃虛擬主機數量與規格時，必須同時考慮基礎平臺本身的資源上限。

從 guest OS 的角度來看，它所看到的「硬體」，全部都是由基礎平臺所模擬的。
模擬的硬體越簡單，對基礎平臺造成的負擔就越小。為了獲得較好的整體效能，建議為每一台虛擬主機選擇的周邊裝置「**剛好滿足 guest OS 與應用程式的需求即可，不必過度預留**」。
這樣既能降低系統負載，也有助於提升整體穩定度與可預測性。

正如前面所提到的，基礎平臺的設計重點之一，就是協助安排虛擬主機之間的網路關係。每一台虛擬主機可以擁有多個乙太網介面，分別掛載在不同的橋接器上，而每個橋接器又對應到不同的 IP 子網。各個橋接器扮演的角色與權限，是由基礎平臺統一管理的。
各個客座作業系統（guest OS）對乙太網介面的使用方式可能不同：有的會啟動 DHCP 用戶端向網路取得 IP 位址，有的則在該子網中運行 DHCP 伺服器，替其他主機分配 IP 位址。若同一個 IP 子網中同時存在兩個 DHCP 伺服器，就可能造成衝突，因此在規劃時必須確保同一子網中只啟用一個 DHCP 伺服器。

如果橋接器 A 的網路流會被基礎平臺 NAT 到橋接器 B，那麼從橋接器 A 底下主機送往橋接器 B 底下主機的網路封包，在 IP 標頭中其來源 IP 位址會被改寫為橋接器 B 的 IP 位址。這些行為與限制，都應在部署虛擬主機內部軟體之前先行評估與規劃。

除了決定虛擬網路介面要掛在哪一個橋接器之外，在建立虛擬主機時，也必須選擇要模擬哪一種類型的乙太網卡。客座作業系統必須具備相容的驅動程式，才能正確使用這些虛擬乙太網介面。

為虛擬主機提供高精度的時鐘來源，對基礎平臺而言是一項額外負擔；時鐘頻率與中斷行為，會直接影響系統對事件的反應效率。如果您發現 guest OS 反應遲緩，有時可能與時鐘來源設定有關。Windows 系統一般採用較低頻率的時鐘來源，以降低相關問題；在 Linux 系統上，如果遇到時間或排程異常，建議檢查並改用 KVM clock 作為時鐘來源。

每一台虛擬主機都會被指派一個 TCP 連線埠，方便您使用 VNC Viewer（或 SPICE 用戶端）連線到該虛擬主機的主控台。這類遠端主控台連線同樣受基礎平臺的存取控制策略所保護，您必須從被允許的網路區段中，才能成功連入該虛擬主機。

若要使用基礎平臺提供的網路管理介面，您可以在瀏覽器中輸入基礎平臺的管理 IP 位址與管理埠號，透過 Web 介面進行集中設定與管理。

http://ip_address:8082/apps/

“admin” 帳戶使用預設密碼 “admin123”。 您可以在網際網路管理介面稍後更改密碼。

以下將開始演示 如何在 基礎平台上，創建虛擬主機。

Upload CD Image（上傳 系統 .iso 檔)

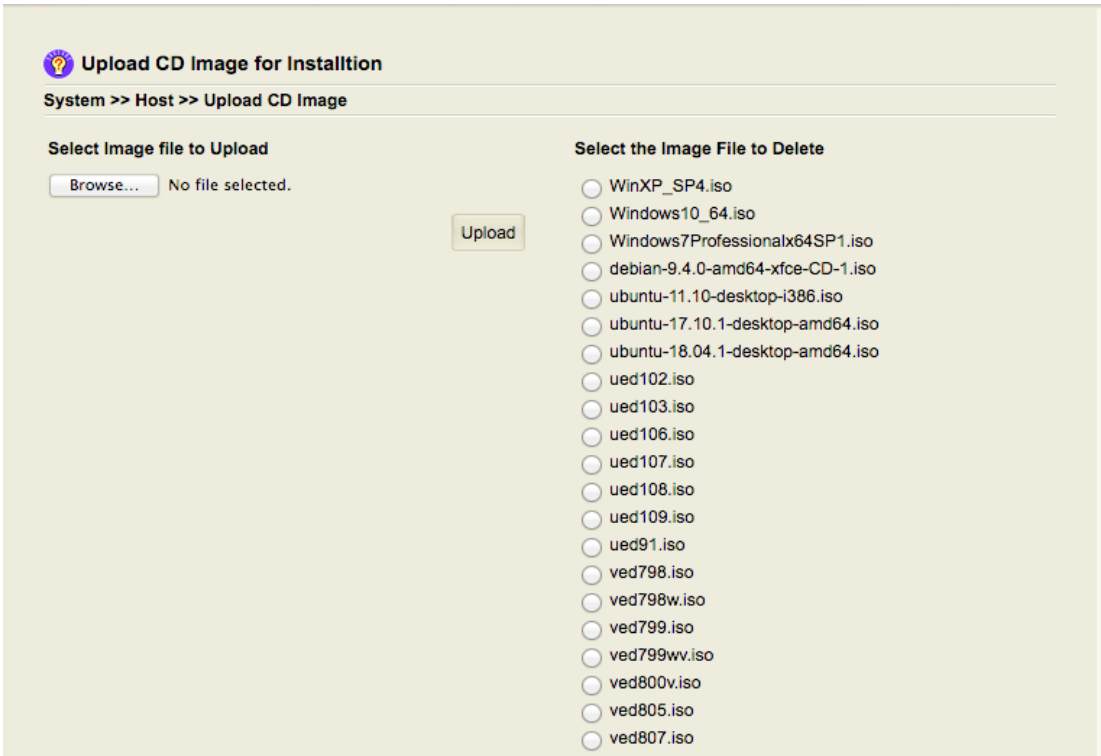


圖 18：上傳客製作業的光碟影像

若要讓虛擬主機從 CD／DVD 開機並安裝作業系統，您需要先準備對應作業系統的 ISO 映像檔，並透過「System → Host → Upload CD Image」功能上傳至基礎平臺。

每個映像檔的大小**不應超過基礎平臺實體記憶體的一半**，否則在上傳後嘗試載入映像檔時可能會失敗。若遇到這種情況，請改用 scp 或其他檔案傳輸方式，將 ISO 檔直接傳送到目標目錄 `/home/qemu/iso`。
請注意：

- 若要從遠端主機使用 scp，基礎平臺上的 sshd 必須正在執行；
- 若是從 WAN 端（在防火牆區域中標記為「net」的一側）連入，則需在防火牆上開放 TCP 埠 22，才能完成 scp 連線與傳輸。

Add Virtual Host (創建虛擬主機實例)

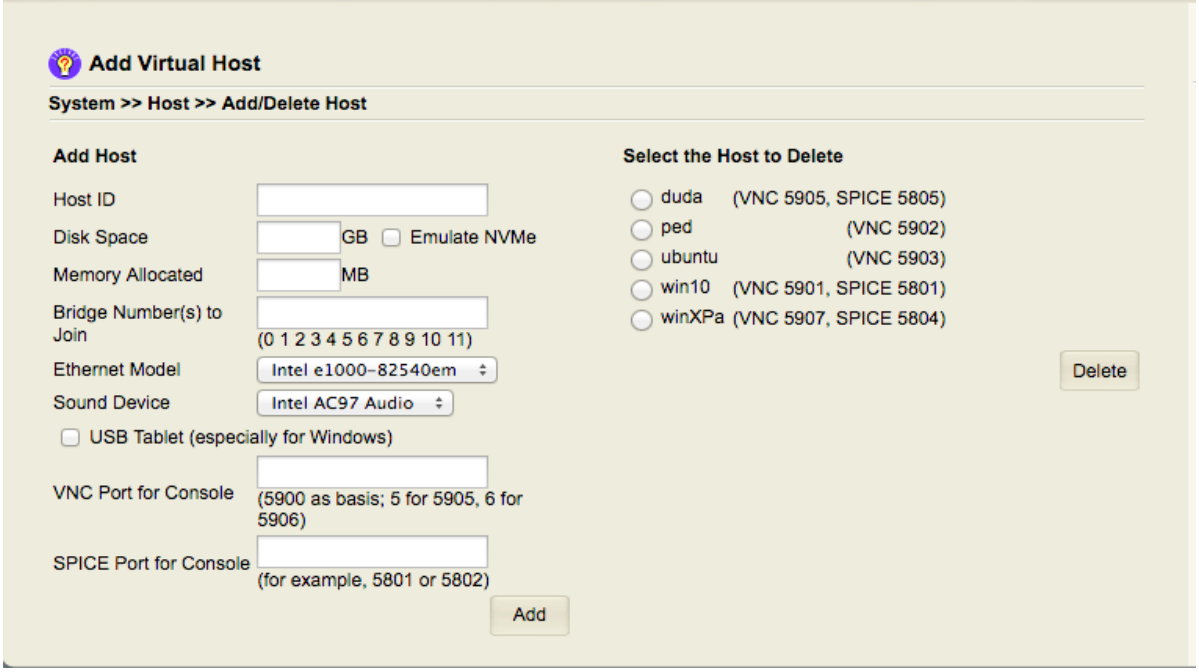


圖 19: 建立虛擬主機實例的螢幕擷圖

為了建立虛擬主機實例，您需要指定「**Host ID**」，以及該虛擬主機的磁碟空間、記憶體容量、所連接的橋接器、乙太網介面型號與 VNC 連線埠等參數。

其中 **Host ID** 僅用於在基礎平臺上識別這台虛擬主機，與 guest OS 內部所使用的主機名稱或其他識別字串無關。

「**USB Tablet**」勾選框用來解決 VNC 與用戶端主機之間的滑鼠指標同步問題。
如果客座作業系統是 Windows，建議勾選此選項。 未勾選時，系統預設提供的是 PS/2 鍵盤與滑鼠。

某些 **SPICE 客戶端** 可以將聲音從客座作業系統傳送到執行 SPICE 客戶端的作業系統（Client OS）。 若要使用這種音訊傳輸情境，必須為虛擬主機選擇合適的音訊裝置類型：

- 在較舊的作業系統（例如 Windows XP）中，通常可以偵測並使用 **AC97 音效裝置**及 **Realtek 8139 乙太網控制器**。
- 在較新的作業系統（例如 Windows 7 或 Windows 10）中，AC97 通常無法正常使用，必須改採 **Intel HD Audio** 搭配 **ICH9 晶片組**。

由於實際組合情境非常多樣，我們無法在此列出所有可能的搭配與對應解法。
建議您在安裝目標作業系統之前，先評估並選定合適的模擬硬體組合。

完成上述設定並按下「Add」後，右側列表會出現對應的 Host ID。 如果目前沒有需要再調整的項目，即可直接前往「System → Host → Host Management」， 將 CD／DVD 映像檔掛載並安裝到該虛擬主機上。

Bridge Assignment (橋接分配)

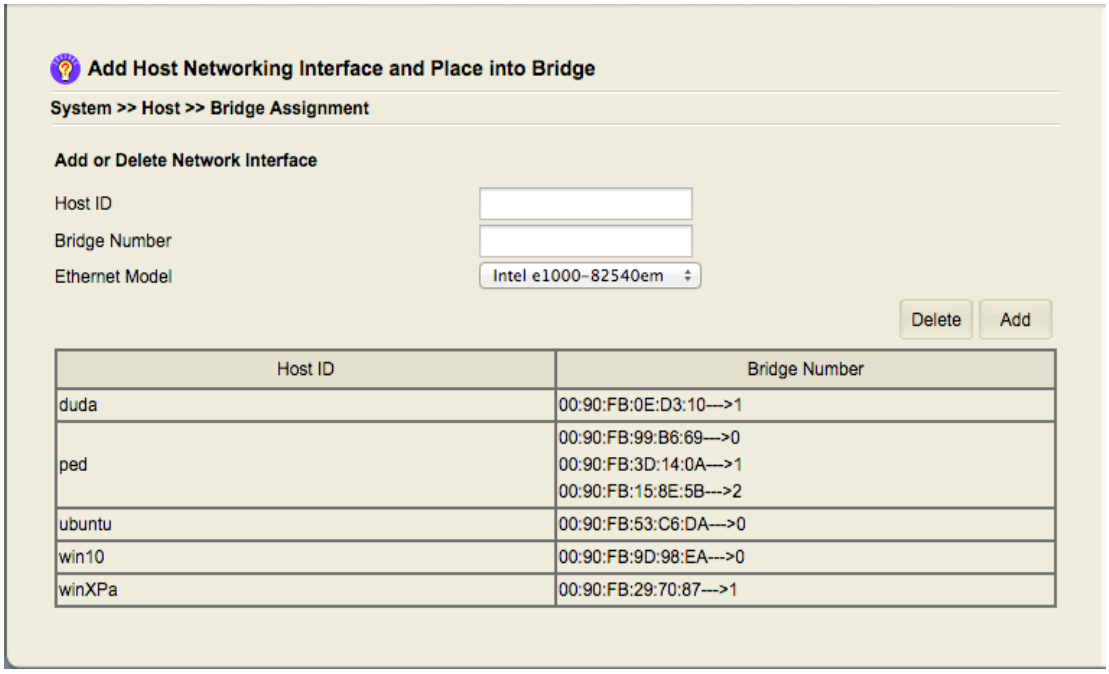


圖 20：虛擬主機之橋接器指派與乙太網卡類型

在此網頁中，管理者可以查看各虛擬主機的乙太網介面（依其 MAC 位址顯示）以及所屬的橋接器編號。
若要變更加註設定，請先在上方欄位輸入對應的 **Host ID** 與 **Bridge Number**，再按下「**Delete**」以刪除既有項目。要新增設定時，請輸入 **Host ID**、**Bridge Number** 與乙太網卡型號，然後按下「**Add**」以完成新增。

Reset Of Memory and Tablet (更改記憶體 與 USB Tablet 的設定)

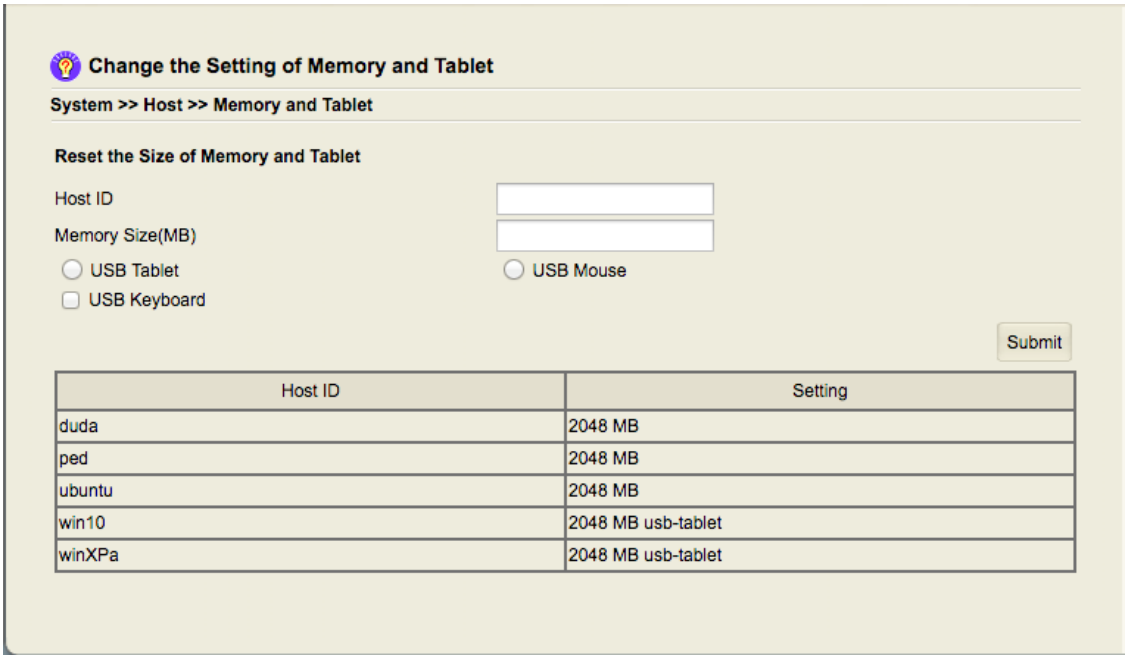


圖 21：調整記憶體容量與啟用虛擬機的 USB Tablet 裝置

若需要調整記憶體大小，可以在此畫面重新設定。
當你使用 VNC Viewer 時，如果發現滑鼠指標在「Guest OS」與「Client OS」(即你的個人電腦執行著 VNC Viewer 程式)之間不同步，可以考慮啟用 **USB Tablet 裝置**來改善這個問題。

系統預設提供的是 **PS/2 滑鼠與鍵盤**。如果你的 Guest OS 支援 PS/2 裝置，通常建議維持預設設定即可，不必更改。

然而，某些作業系統（例如 macOS *Mojave*）不支援 PS/2 滑鼠／鍵盤，也不支援 USB Tablet。在這種情況下，建議改為選擇 **USB 滑鼠與 USB 鍵盤**，並搭配 **SPICE 客戶端**（例如 `virt-viewer` 或 `remote-viewer`）來連線虛擬主機。

CPU and Chipset (調整 CPU 數量與晶片組設定)

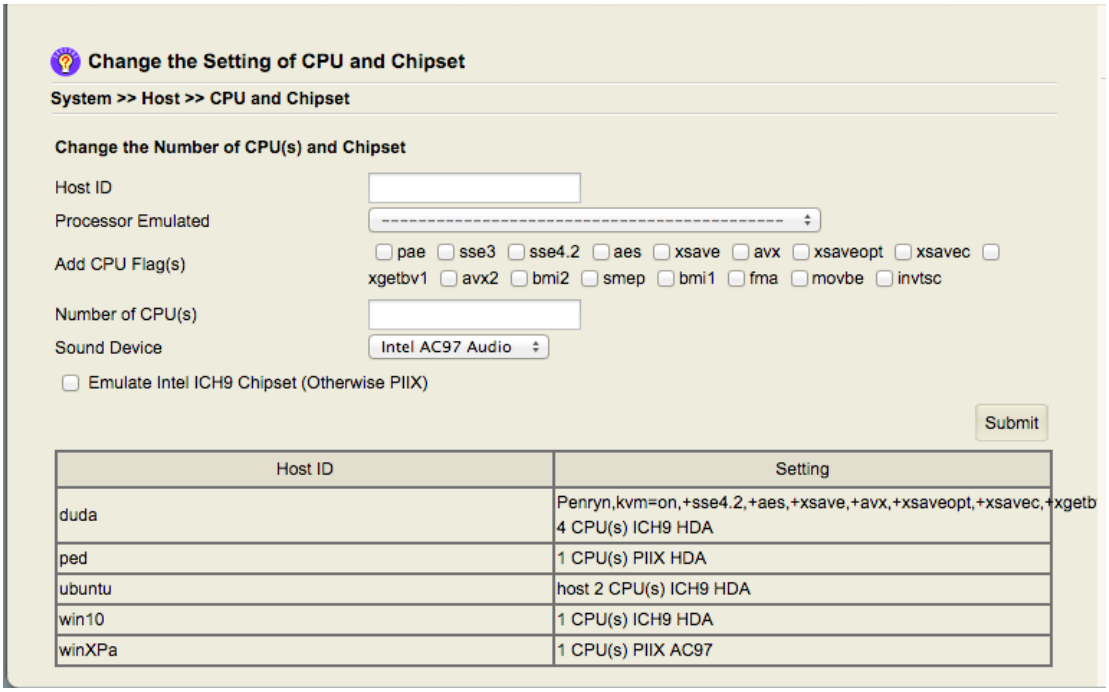


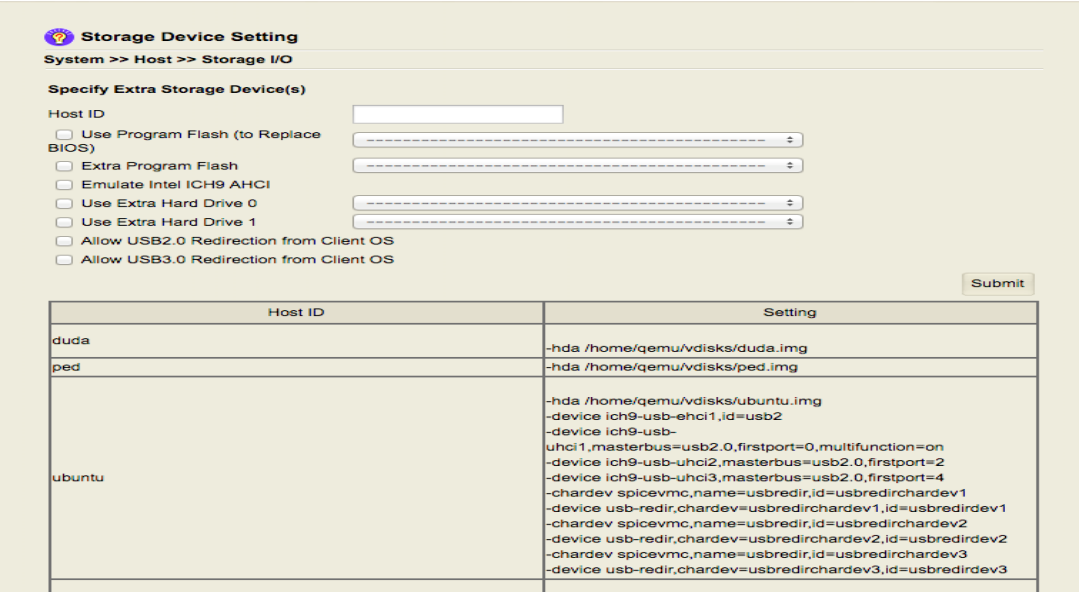
圖 22：調整 CPU 數量與晶片組設定

在建立虛擬主機實例時，系統預設使用 **Intel PIIX** 晶片組。PIIX 主要提供 PCI-to-ISA 橋接、PCI IDE 控制器，以及 AC97 等音效裝置的支援；而 **Intel ICH9** 則主要用於 **SATA（Serial ATA）** 等較新的儲存介面。

因此，若計畫在虛擬主機中模擬 **IDE 硬碟**，建議選擇 **PIIX**；若要模擬 **SATA 硬碟**，則應改用 **ICH9**。部分較舊的作業系統並不支援 SATA 磁碟，在這種情況下就必須選用 PIIX。

一般情況下，我們會選擇與 **基礎平臺實際 CPU 型號**相近的虛擬 CPU 型號，以取得較佳的相容性與效能。如果某些作業系統或應用程式需要特定的 CPU 型號或 CPU 旗標（例如 SSE4.2、AES-NI 等），也可以在此畫面中手動選擇合適的 CPU 設定。

Storage Device Setting (儲存裝置設定)



附圖 23：附加儲存裝置設定

此畫面在標準發行版本中可能不會出現；它主要用於提供「程式刷新（Program Refresh）」功能，以便為特定應用載入自訂 BIOS 與額外的儲存裝置。欲使用的映像檔必須預先放在目錄 `/home/qemu/extra` 下，才會出現在下拉選單中供選擇。

一般情況下，**基礎平臺**只會為每台虛擬主機提供一顆系統磁碟，作為 **客座作業系統（guest OS）** 的啟動與安裝磁碟。若需要額外的儲存裝置，這些額外磁碟在使用 IDE 匯流排時，會依序掛載在 hdb、hdd 等裝置名稱；若採用 **ICH9 AHCI（SATA）**，則這兩顆額外磁碟會以對應的 bus ID 前綴，出現在與原先系統磁碟相同的控制器底下。
建議磁碟映像檔使用 **qcow2** 格式；若您手上已有其他格式的磁碟檔，可在任一 Linux 系統上透過 qemu-img 工具進行格式轉換。

USB 重導向（USB Redirection） 是發生在客座作業系統與「客戶端作業系統（Client OS）」之間的功能，用於在 Client OS 上執行 SPICE 客戶端時，將 USB 裝置直接轉交給 guest OS 使用。此功能並非在所有情境下皆可啟用：

- 在 guest OS 端，必須能正確辨識由 ICH9 USB 控制器（EHCI／UHCI）所提供的 USB 2.0 裝置，或由 NEC 晶片組所提供的 USB 3.0 裝置；
- 在 Client OS 端，USB 裝置在被重導向期間，無法同時由 Client OS 自身使用；
- 某些 SPICE 客戶端還需要安裝額外元件才能啟用 USB 重導向功能——例如在 Windows 環境中使用 `remote-viewer` 時，必須先安裝 usbDK 才能支援 USB 重導向。

若已在「主機管理」畫面中指定自訂的 BIOS（或 UEFI）映像，之後即可直接在同一畫面啟動虛擬主機，無須再額外設定開機選項。

Host Management (虛擬機主機管理)

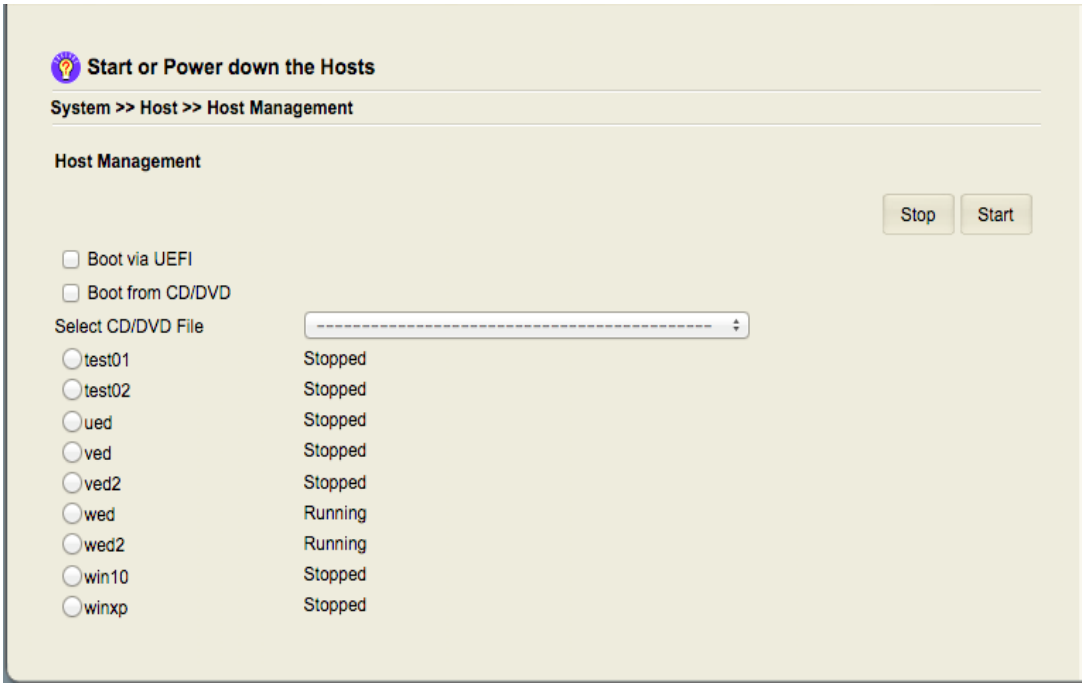


圖 24：虛擬機主機管理

首次啟動虛擬主機時，通常需要從 CD／DVD 映像檔開機，將作業系統安裝到虛擬主機的磁碟中。

在「主機管理」畫面中，請勾選「從 CD／DVD 開機」，選擇欲使用的 CD／DVD 映像檔，再指定要啟動的虛擬主機，最後按下「啟動」。若虛擬主機成功開機，在對應的「Host ID」旁邊會顯示「運行」狀態。

在某些情況下，CD／DVD 映像檔可能不是可開機光碟，而只是安裝附加軟體用的安裝片；這時您仍可掛載該 CD／DVD 映像檔，但不必勾選「從 CD／DVD 開機」。

在安裝過程中，可以透過 **VNC Viewer** 連線到虛擬主機的主控台；實際的安裝流程則依各個 guest OS 自行安排。一旦透過 CD／DVD 完成作業系統安裝後，您可以先關閉虛擬主機，再重新啟動，這次便不需要再從 CD／DVD 開機。

請特別注意：在 VNC Viewer（或 SPICE 客戶端）中連線主控台時，所使用的是**基礎平臺的 IP 位址與指定的 TCP 埠**，而不是虛擬主機本身的 IP 位址。

下圖為使用 **VNC Viewer** 連線至虛擬主機，進行 Windows 安裝時的主控台畫面示意。

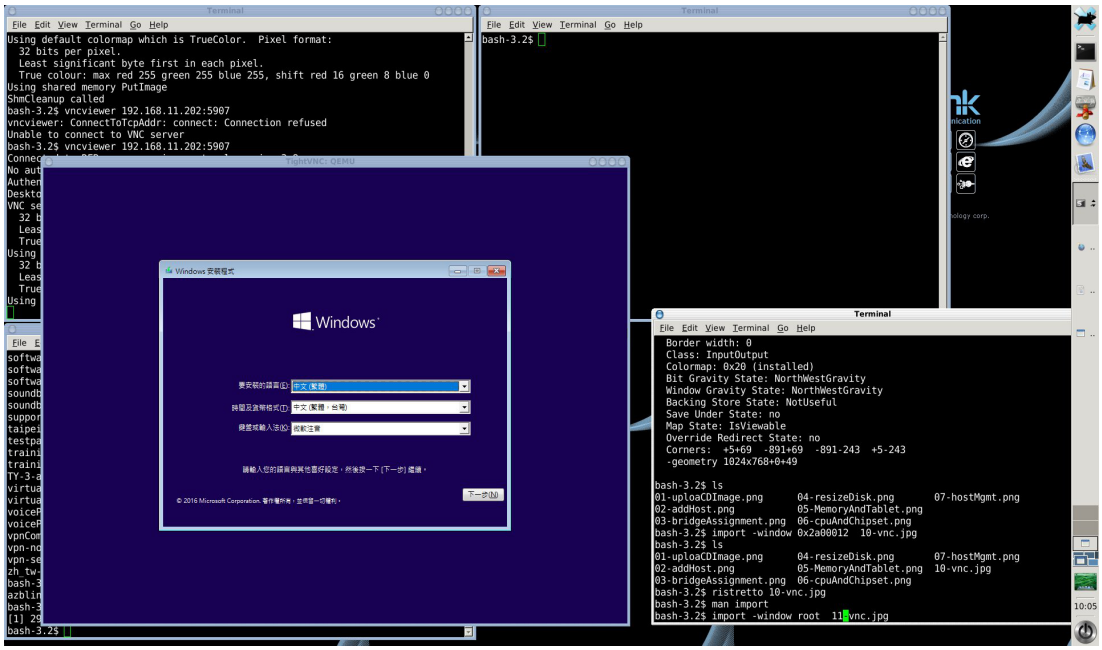


圖 25：透過 VNC Viewer 操作虛擬主機的螢幕擷圖

若需要以 UEFI 模式啟動系統，請勾選「Boot via UEFI」。請注意，本平臺所提供的 UEFI 韌體是為 **64 位元系統**設計的，因此對應的開機媒體上，UEFI 可執行檔也必須是 64 位元的二進位檔。

您也可以改用 **SPICE 客戶端**來存取虛擬機的主控台。SPICE 客戶端雖然沒有 VNC 那麼彈性，但額外支援音訊傳輸等功能。無論使用 SPICE 或 VNC 客戶端，都可以透過基礎平臺任一對外 IP 位址加上對應埠號，連線至虛擬主機的主控台。
若虛擬主機內已安裝並啟動 Microsoft Windows，且完成網路設定，之後要遠端登入該 Windows（例如透過 RDP 或其他服務）時，則應使用 **虛擬主機本身的 IP 位址**，而非基礎平臺的 IP 位址。

第三章 邊境控制 (Border Control)

Border Control mechanism on host platform (基礎平臺上的邊境控制機制)

基礎平臺所提供的邊境控制（Border Control）功能，其核心運作原則是**「以橋接器為單位」**來劃分和執行安全策略。

1. 橋接器作為安全區域劃分單位

- **定義：** 所謂「以橋接器為單位」，是指網路安全區域的劃分邊界，是根據**各個虛擬橋接器 (Virtual Bridges)** 之間的界線來定義的。
- **與實體介面的區別：** 實體乙太網介面（Physical Ethernet Interface）在這邊的角色，僅負責將基礎平臺連接到外部網路，**並不作為安全區域劃分的依據**。

2. 虛擬橋接器的邏輯模型

為了更好地理解基礎平臺上的虛擬橋接器，建議採用以下邏輯模型：

- **將橋接器視為外部交換器：** 您可以把每一個虛擬橋接器 (br0, br, …) 想像成位在**基礎平臺外部的實體乙太網交換器** （br0 , br… 每個想像成 一台台 個別的交換器）。
- **基礎平臺的連接：** 基礎平臺上的網路介面則各自連到這些「外部交換器」上。
- **IP 地址的身份：** 原本設定在虛擬橋接器上的 IP 位址，可以視為這些介面在對應「交換器」上的 **IP 位址與身份**。

總結： 這種視角強調，橋接器是邏輯上的安全區域，而基礎平臺、虛擬主機都是連接到這個區域（交換器）上的獨立節點。

3. 預設防火牆區域設定

根據邊境控制機制，基礎平臺預設將虛擬橋接器劃分歸類為三個標準防火牆區域（Zone）：

橋接器 (Bridge)	區域標籤 (Zone Label)	安全區域性質
br0	**net**	通常代表外部網路或信任度最低的區域 (WAN / Internet)。
br2	**dmz**	代表非軍事區，用於部署可被外部存取的服務。
br1, br3, br4 … br10, br11	**loc**	代表內部區域（Local），通常是信任度最高的區域 (LAN / Internal)。

4. 前置作業：虛擬主機介面配置

在配置這些防火牆規則之前，您最主要的工作就是為每一台虛擬主機決定：**它的各個乙太網介面應該連接到哪一個虛擬橋接器**。這一步驟決定了虛擬主機將被歸類到哪一個安全區域，並套用該區域的邊境控制規則。

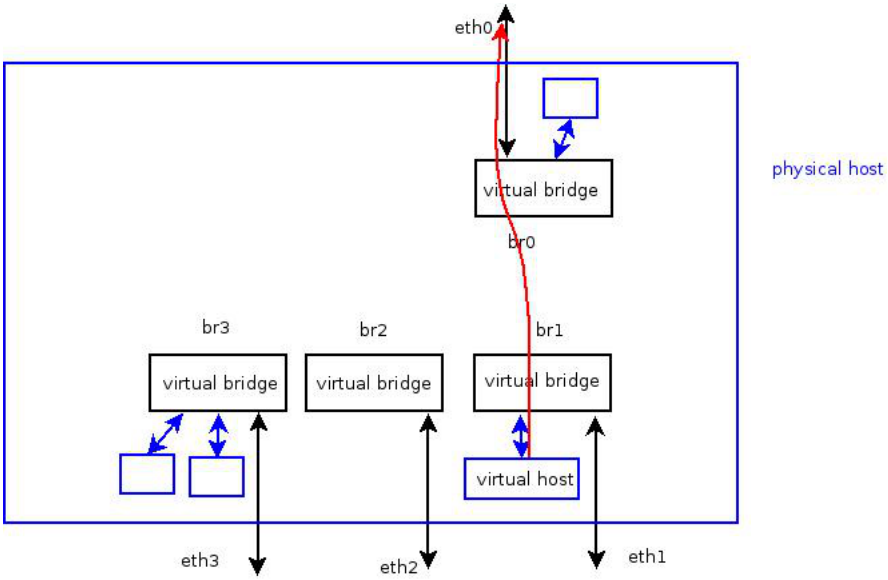


圖 26：虛擬橋接器與基礎平台

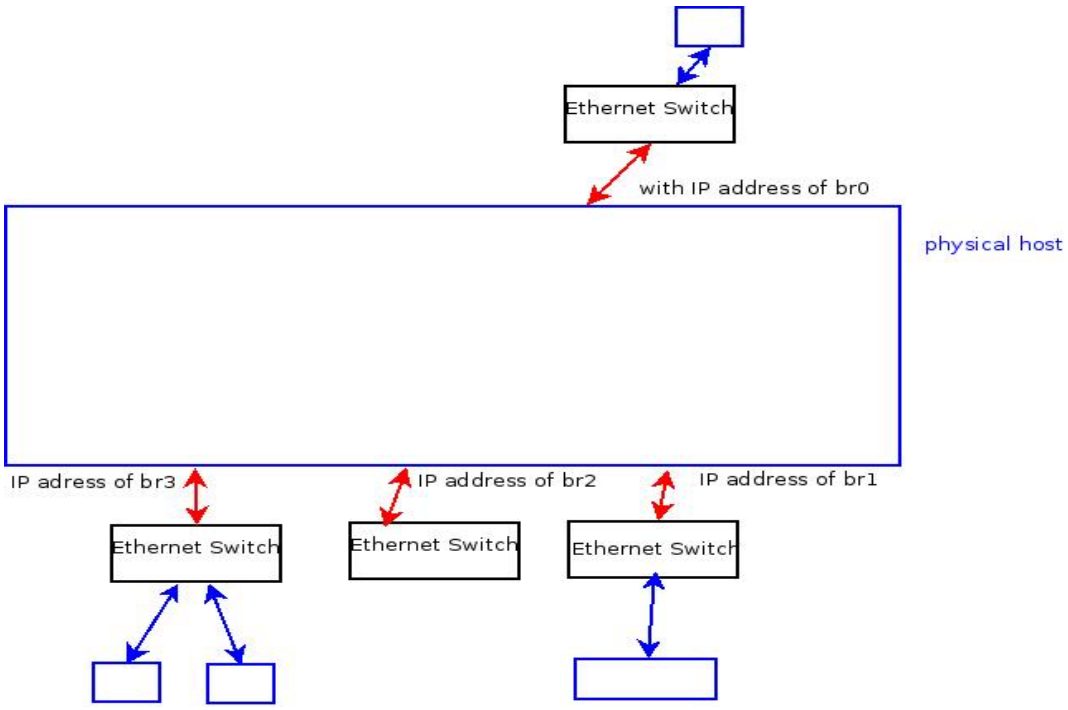


圖 27：以實體乙太網交換器建立等效拓撲模型

上方兩幅圖用來說明，應該如何理解基礎平臺上的虛擬橋接器。
你可以把每一個虛擬橋接器想像成**位在基礎平臺外部的實體乙太網交換器**，而基礎平臺上的額外乙太網介面則各自連到這些交換器上。原本設定在虛擬橋接器上的 IP 位址，可以視為這些乙太網介面在對應交換器上的 IP 位址與身分。

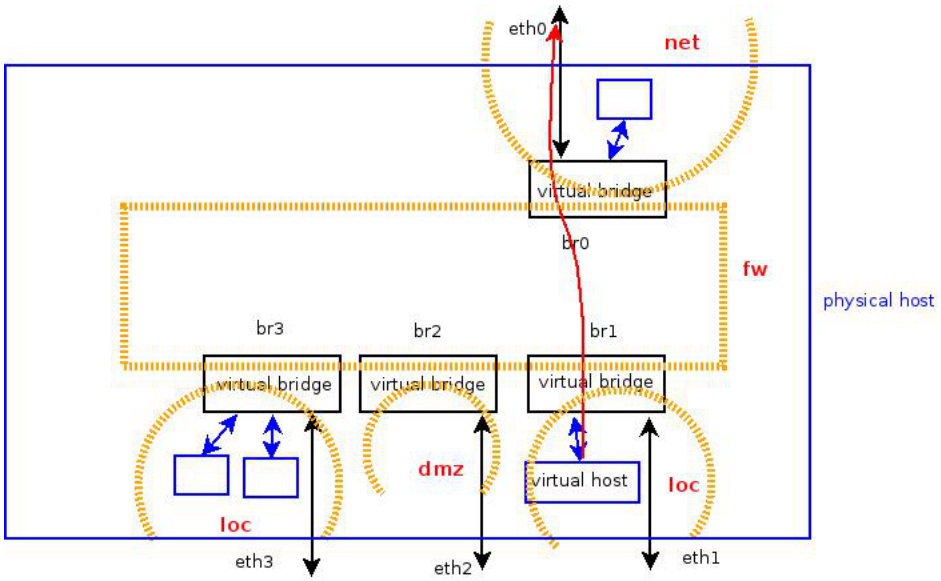


圖 28：基礎平臺上的防火牆區域劃分

虛擬橋接器與區域安全策略：預設規則總覽

當虛擬主機的乙太網介面掛載到某個虛擬橋接器上時，該介面會同時受到「所在虛擬橋接器」的規則，以及「該橋接器所屬安全區域」的規則所約束。因此，規劃部署前，必須熟悉各區域的定義及其允許的操作範圍。

1. 區域定義與用途

區域標籤 (Zone)	橋接器範例	安全性質	典型用途
net	br0	外部區域，信任度最低。	連接網際網路 (WAN)。
dmz	br2	非軍事區，對外服務區域。	承接來自 net 的對外服務連線，如訪客 Wi-Fi、對外伺服器。
loc	br1, br3…	內部區域，信任度最高。	內部本地存取的主機。
fw	N/A	代表**基礎平臺本身** (不含虛擬主機)。	基礎平臺的系統管理與控制。

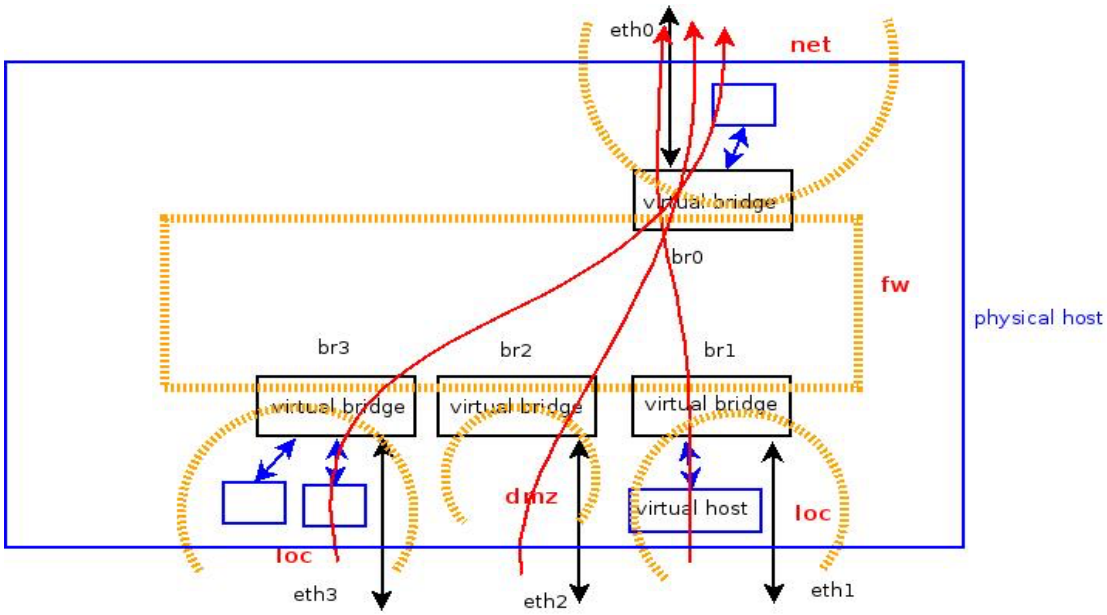


圖 29：流向「net」區域的網路流

2. 虛擬主機區域互通規則（net, dmz, loc）

以下表格總結了 **虛擬主機** 所在區域之間的**預設存取權限**：

來源區域 ↓	目的區域	net	dmz	loc
loc	→	（允許）	（允許）	（允許）
dmz	→	（允許）	（允許）	（禁止）
net	→	（禁止）	（禁止）	（禁止）

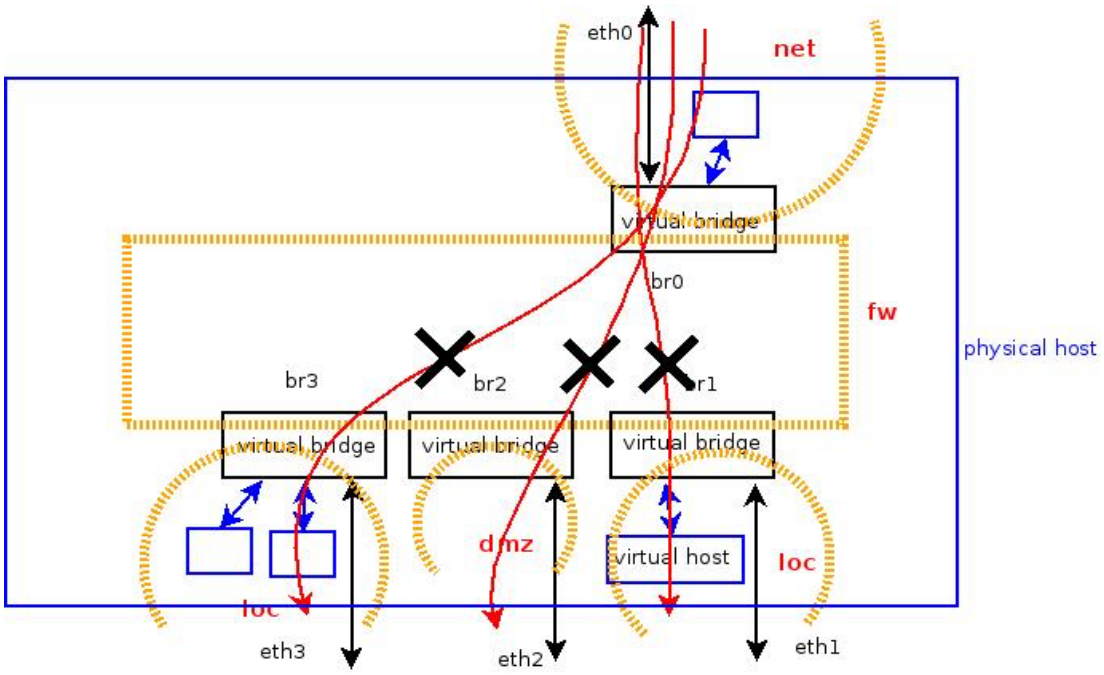


圖 30：來自「net」區域的網路流，預設不得存取「loc」或「dmz」區域。

A. net 區域（對外存取）規則

- **預設限制：** 來自 net 區域的網路流，在預設情況下**不得**存取 loc 或 dmz 區域。
- **用途：** 將 net 區域視為網際網路 (WAN)，所有內部連線均受此邊界控制。
- **例外開放：** 若需允許 net 流量到達 dmz 或 loc，**必須**設定 **埠轉發 (Port Forwarding)** 規則。
  - *埠轉發定義：* 在基礎平臺上選定一個對外開放的埠，將 net 區域抵達該埠的連線，轉送到指定的 dmz 或 loc 主機。

B. loc 區域（內部存取）規則

- **預設權限：** loc 區域中的主機，預設可以存取**任何**區域內的主機（net, dmz, loc）。
- **NAT 處理：**
  - loc ➔ net 的網路流：**會**經過 **NAT 處理**（改寫來源 IP 為 br0 的 IP）。
  - loc ➔ dmz 或 loc ➔ loc 內其他子網：**不會**套用 NAT。

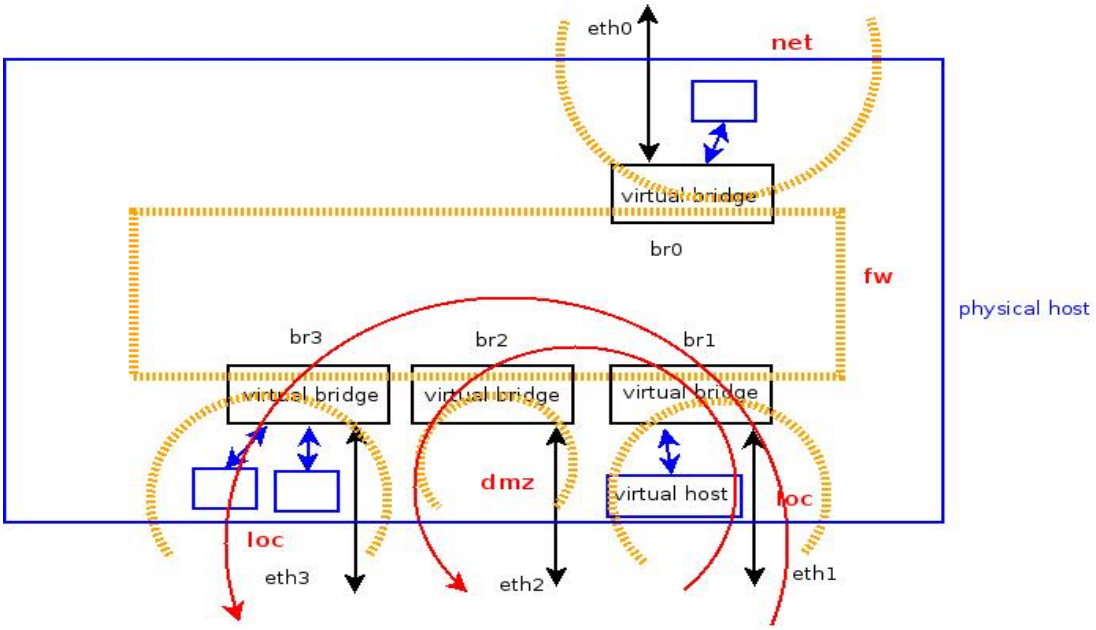


圖 31：來自「loc」區域的網路流。

C. dmz 區域（隔離存取）規則

- **預設限制：** dmz 區域中的主機，預設**不可存取** loc 區域的主機，但**可以存取** net 區域的主機。
- **用途：** 利用此特性，可將訪客 Wi-Fi AP 或處理外部連線的伺服器放在 dmz 區域，使其能連線到外部 (net)，但無法主動進入內部 (loc)。

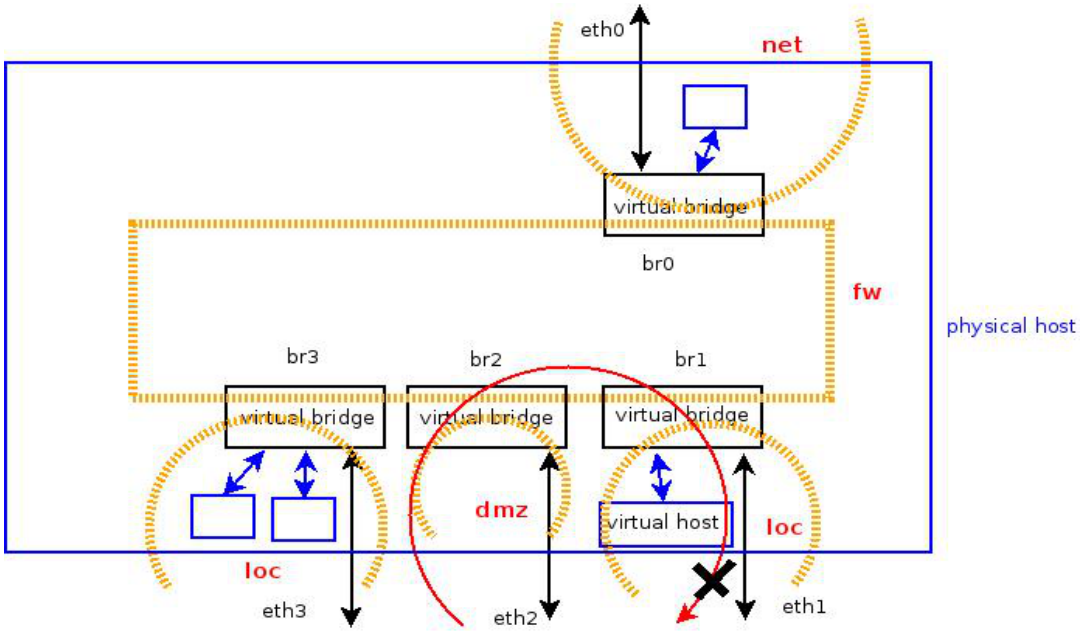


圖 32：來自「dmz」區域的網路流，在預設情況下**不得**存取「loc」區域。

3. 路由要求 (跨子網通訊)

- 只要是**跨不同 IP 子網**的通訊，都必須透過基礎平臺進行「**路由**」處理。
- 當封包進入基礎平臺所連接的橋接器後，路由判斷會自動由基礎平臺完成。
- **重要前提：** 為了讓封包正確送達其他子網上的主機，這些子網中的主機必須將「**基礎平臺在本地子網上的 IP 位址**」設定為其**預設閘道 (Default Gateway)**，或在其路由表中正確設定通往其他子網的閘道。

4. 基礎平臺本身 (fw) 的存取規則

fw 區域代表 **基礎平臺本身** 的服務（不包含其上執行的虛擬主機）。

來源區域 ↓	目的區域	net	dmz	loc
fw	→	（允許）	（允許）	（允許）
net	→	（禁止）	（禁止）	（禁止）
loc	→	（允許）	（允許）	（允許）
dmz	→	（禁止）	（禁止）	（禁止）

---

5. 規則調整與特殊區域

- 這些預先定義的規則**不會**直接顯示在網頁管理介面中，但管理者可以透過介面**新增例外規則**，或調整各區域所包含的介面與成員。
- 系統中另有一個名為 **「road」** 的區域，主要與 **VPN** 以及其他基礎平臺內部介面相關，其行為與設定將在後續章節說明。

Port Forwarding (埠轉發)

埠轉發 (Port Forwarding) 機制與設定

埠轉發是一種例外規則，用於允許來自 net 區域（通常是網際網路）的流量，能夠穿透基礎平臺防火牆，到達 dmz 或 loc 區域內的特定虛擬主機。

1. 埠轉發的運作原理與 NAT （ Network Address Translation 網路位址轉換 ）關係

埠轉發是透過 **DNAT (Destination NAT)** 動作在基礎平臺上實作的。

機制	定義與作用	相關規則
來源 NAT (SNAT)	當 dmz 或 loc **對外發起**網路流時，基礎平臺會將封包的來源 IP 改寫成 **br0 的 IP 位址**，並指派新的 TCP／UDP 埠號進行連線追蹤。回應封包藉此返回基礎平臺，再轉送回原始主機。	適用於所有 dmz ➔ net 及 loc ➔ net 的流量。
目的 NAT (DNAT)	埠轉發即是透過 **DNAT** 實作。當來自 net 的封包抵達基礎平臺時，DNAT 會**改寫封包的「目的 IP 位址」**，並依新的目的 IP 將封包轉送（即轉發）出去。	適用於 **net 主動連入**到特定埠的流量。

- **預設處理：** 來自 net 區域、目標為基礎平臺本身的流量，若無對應服務監聽或無轉送規則，預設將被**直接丟棄**。
- **埠轉發作用：** 在基礎平臺上指定一個 TCP 或 UDP 埠號，將目的地為基礎平臺且使用該埠號的流量，轉送到 dmz 或 loc 區域內的其他主機。

2. 設定步驟與介面位置

您可透過以下設定頁面新增或調整埠轉發規則： [Border] ➔ [Connection] ➔ [Port Forwarding]

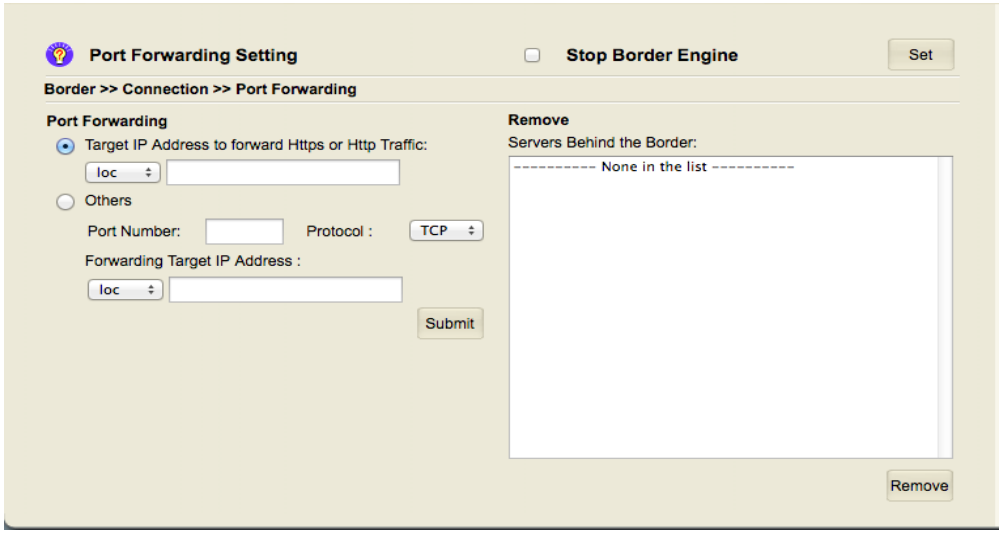


圖 33：埠轉發設定畫面

範例：轉發 HTTP (TCP 80) 與 HTTPS (TCP 443)

假設：

- 基礎平臺在 br0 上的 IP 位址為 **192.168.11.202**。
- dmz 區域內的目標主機 IP 為 **172.16.11.119**。

1. **設定規則：** 在介面中選擇目標區域為 **dmz**，並輸入目標主機 IP 位址 **172.16.11.119**。
2. **結果：** 當外部主機連線到 **192.168.11.202** 的 TCP 80 或 443 埠時，封包將依規則被轉發到 dmz 區域中的 **172.16.11.119** 主機

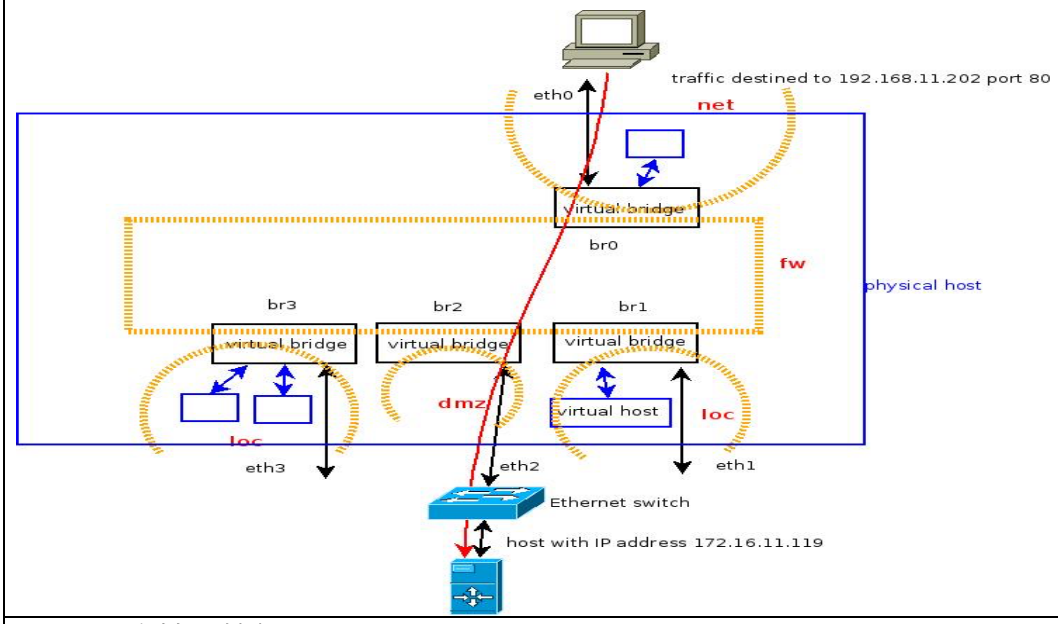


圖 34：埠轉發範例

如下圖所示：基礎平臺在「br0」上的 IP 位址為 **192.168.11.202**；「dmz」區域內有一台主機，IP 位址為 **172.16.11.119**。當外部主機連線到 **192.168.11.202 的 TCP 80 埠（HTTP 網路流）**時，這些以 192.168.11.202 為原始目標的封包，會依埠轉發規則由基礎平臺轉送到 172.16.11.119。

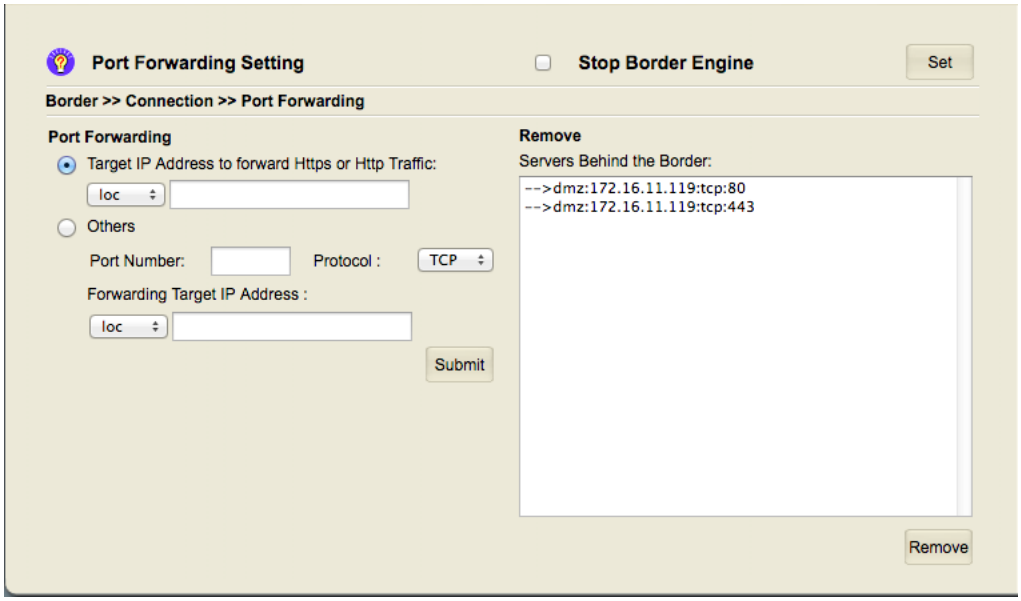


圖 35: HTTP 埠轉發範例

若要新增轉發 HTTP 與 HTTPS 的規則，只需在目標區域中選擇「**dmz**」，並輸入目標主機的 IP 位址 **172.16.11.119**，如下方螢幕擷圖所示。

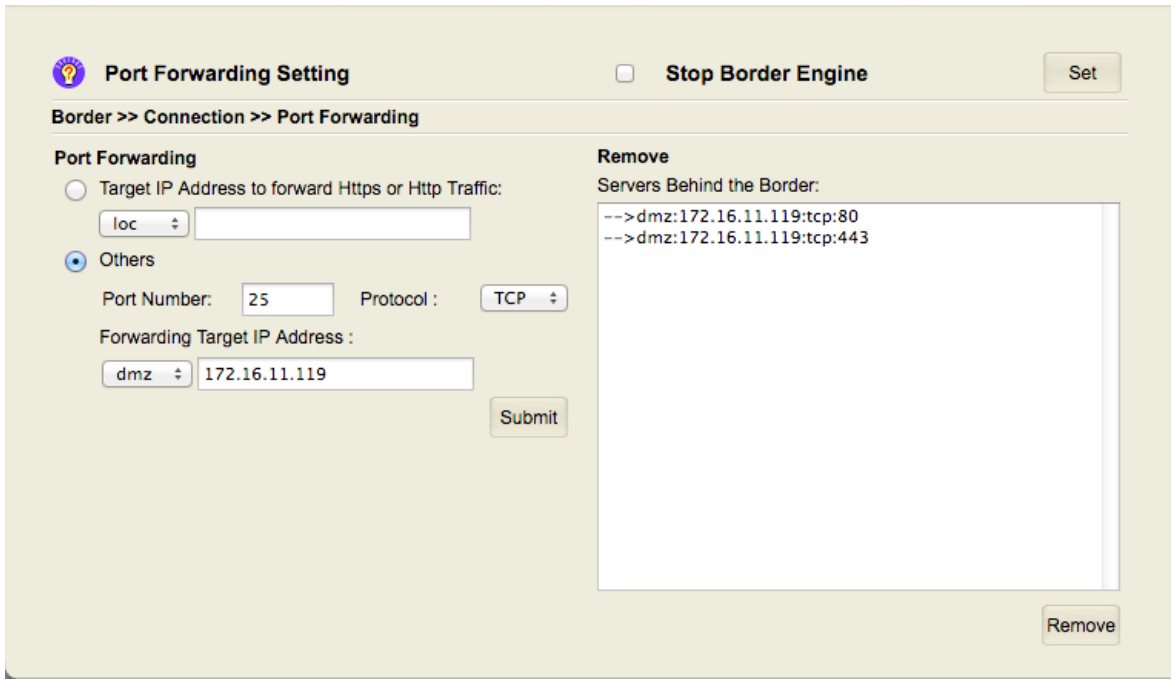


圖 36：HTTP 埠轉發設定螢幕擷圖

按下「提交」按鈕後，右側的清單會顯示一筆新規則：所有送往 TCP 埠 **80** 與 **443** 的流量，將被轉發到「dmz」區域中 IP 位址為 **172.16.11.119** 的主機。

範例：轉發 SMTP (TCP 25)

若要將 SMTP (TCP 25) 網路流轉發到主機 172.16.11.119，只需輸入目標區域與 IP 位址並提交即可。

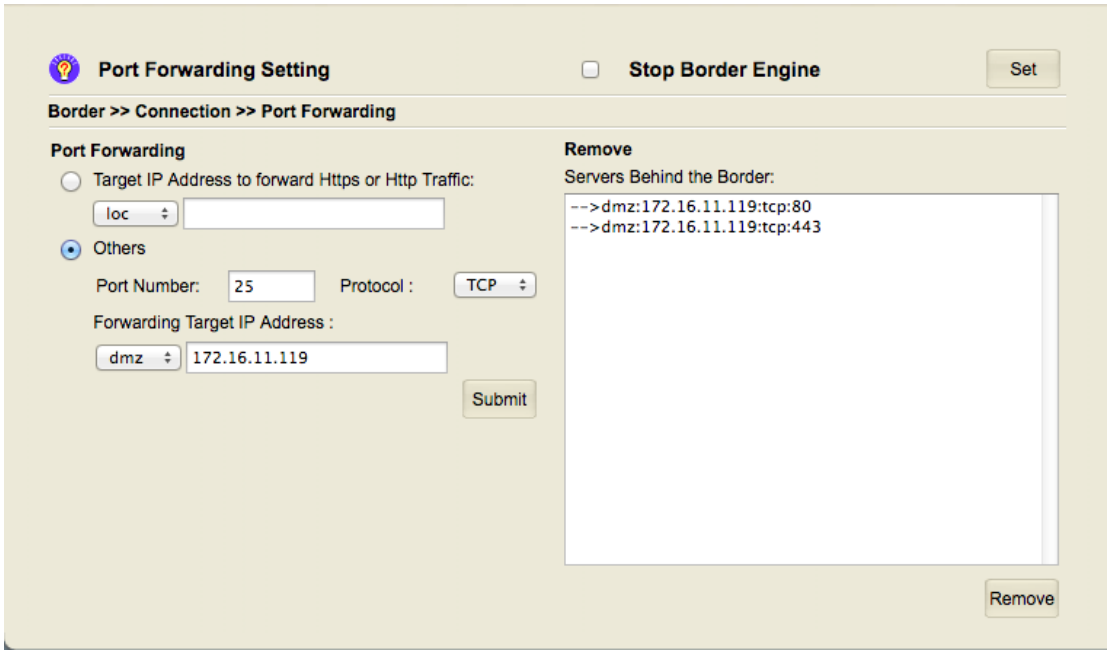


圖 37：SMTP 埠轉發範例

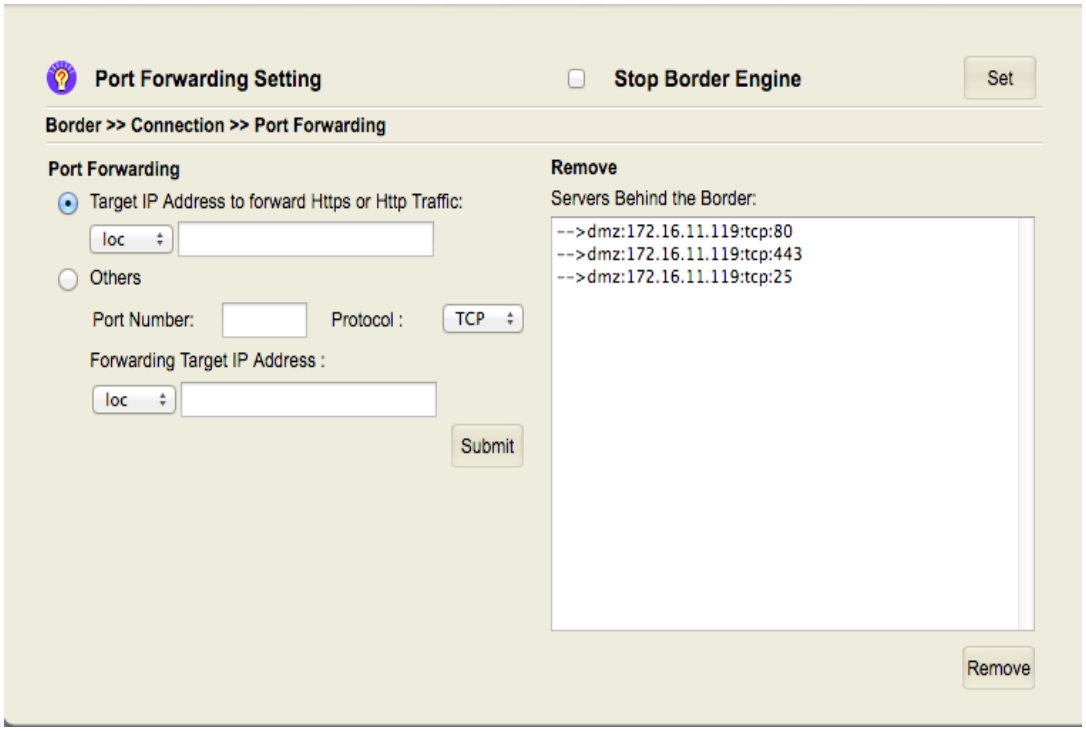


圖 38：新增 SMTP 埠轉發後的螢幕擷圖

3. 規則生效與邊境引擎重啟

- **重要：** 每次變更（新增或修改）規則後，都**必須先停用再重新啟動邊境控制引擎**，變更才會真正生效。
- **操作步驟：**
  1. 勾選上方的「**Stop Border Engine**」核取方塊並按下「Set」。
  2. 待引擎停止後，取消勾選該方塊並再次按下「Set」，即完成重新啟動。

4. 實務考量與注意事項

考量點	內容說明
目標主機閘道	接收轉發流量的主機（如 172.16.11.119）的**預設閘道**，必須設定為**基礎平臺在該子網上（本例中為 br2）的介面 IP 位址**，否則回應封包將無法正確返回原始發送端。
安全性建議	建議**優先將埠轉發設定到 dmz 區域**的主機，而非 loc 區域的主機。避免將來自網際網路的大量惡意連線直接打到內部 loc 網路，增加風險與負載。
Loopback (Hair-pin NAT)	當啟用埠轉發後，若「目標主機所在區域的用戶」也想用同一個**對外 IP 位址與埠號**來存取這台主機，則需要處理這種情境，通常稱為 **Loopback** (或 **Hair-pin NAT**)。

埠轉發的底層實作與進階考量

埠轉發機制在 Azblink 基礎平臺上是透過 目的網路位址轉換（DNAT） **動作來實現的。**

1. DNAT 實作原理與規則查看

- **DNAT 定義：** DNAT（Destination NAT）的作用是：在外部封包抵達基礎平臺時，**改寫封包的「目的 IP 位址」**，然後再依新的目的 IP 將封包轉送出去。
- **規則對應：** 當您在「Port Forwarding」頁面新增一條埠轉發規則後，可以在 **[Border] ➔ [Rule] ➔ [List / Remove Rule]** 頁面中，看到對應的規則，其動作類型即為 **DNAT**。

2. 實務安全建議：優先部署到 dmz

- **強烈建議：** 一般來說，我們建議**優先**將埠轉發設定到 **dmz 區域**的主機，而不是 **loc 區域**的主機。

風險說明： 若直接將埠轉發指向 loc 區域，可能會讓來自網際網路的大量惡意連線（例如垃圾郵件或掃描攻擊）直接進入內部網路主機，**大幅增加 loc 網路**的安全風險與主機負載。

3. Loopback (Hairpin NAT) 限制說明

- **需求情境：** 當啟用埠轉發後，常見的需求是：「目標主機所在區域的用戶，也想用同一個對外 IP 位址與埠號來存取這台主機。」
- **範例：** 某 loc 區域主機對外公布在 br0 的公共 IP 上，loc 區域內的用戶也希望用這個公共 IP 連線。要實現這種「對外 IP 在內部回頭」的存取，就需要 Loopback 功能。
- **本平台限制：**

  ⚠ **重要提醒： 本基礎平臺的這個版本中，不提供 Loopback 功能。**

- **版本建議：** 若您需要此類進階功能，請參考我們的 **「ved」版本構建**，該版本設計中包含了更完整的 Loopback 相關支援。

Connection Tracking (連接追蹤)

檢視連線狀態有助於診斷各類網路問題。
您可以透過「**Border → Connection → Connection Tracking**」，按下「**顯示**」按鈕來查看目前的連線追蹤資訊。。

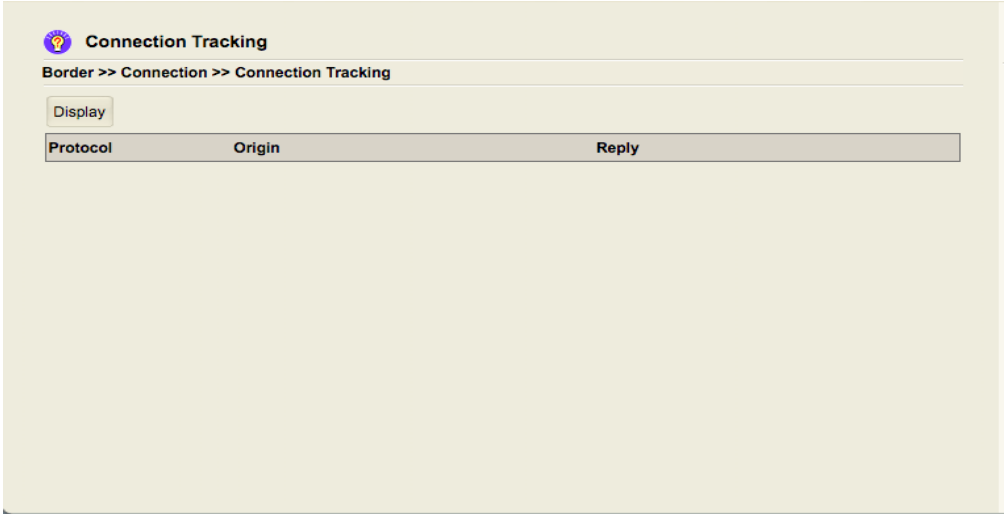


圖 39：連線追蹤螢幕擷圖

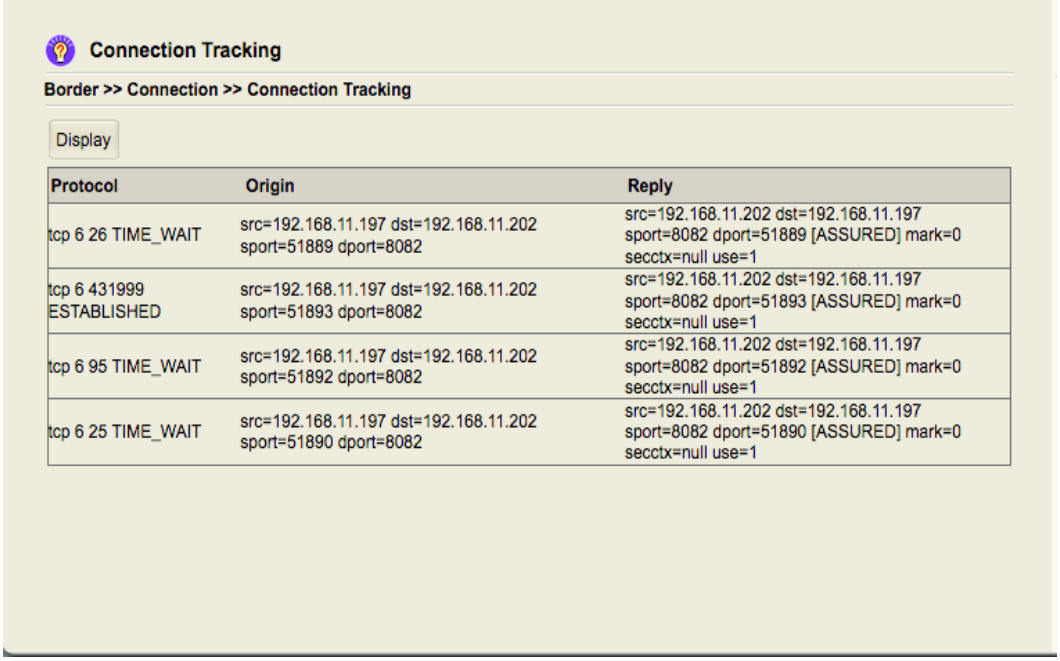


圖 40：連線狀態顯示

若發現某台個人電腦產生異常大量的連線，可能意味著該電腦已遭病毒感染，或正在執行某些點對點（P2P）軟體。您可以根據此頁面提供的連線線索，進一步追查並找出問題根源。。

Actions after Receiving Network Packets (接收網路封包後的處理動作)

當基礎平臺收到一個網路封包時，可以對該封包採取下列動作之一：ACCEPT、DROP、REJECT、DNAT、REDIRECT。

每一條以 IP 為基礎的規則，通常會根據若干封包屬性來判斷是否套用這些動作，常見屬性包括：

- 封包來源（Source）
- 封包目的地（Destination）
- 協定類型（TCP 或 UDP）
- 目的埠與來源埠
- 原始目的 IP 位址

基礎平臺會依據這些屬性是否符合規則條件，決定要對封包執行哪一種動作。
下文將先對各種動作做簡要說明，並在後續章節示範實際的使用方式。

ACCEPT 動作

「ACCEPT」表示**允許**所有符合該規則條件的網路流通過。
例如，若要允許所有從「net」區域連到基礎平臺（區域「fw」）的 Telnet 連線（TCP 埠 23），可以新增一條規則，動作選擇 ACCEPT，條件如下：

- Source：net
- Destination：fw
- 協定：TCP
- Destination Port：23

DROP 與 REJECT 動作

「DROP」表示**直接忽略**符合條件的封包，不回覆任何訊息；
「REJECT」則是在拒絕連線的同時，**主動回傳錯誤回應**（例如 TCP RST 或 ICMP unreachable），讓發送端知道連線被拒絕。由於系統已經針對各區域之間預先定義了一組預設規則，因此只有在需要覆寫或補充這些行為時，才需要另外新增「DROP」或「REJECT」規則。例如：預設情況下，從「loc」到「net」的網路流是允許的。如果你想**禁止所有從 loc 到 net 的 HTTP 存取**，可以新增一條規則，動作選擇 DROP，條件如下：

- Source：loc
- Destination：net

- 協定：TCP
- Destination Port：80

REDIRECT 動作

「REDIRECT」表示將**送往基礎平臺某個埠的網路流，轉送到基礎平臺本身的另一個埠**。
這通常用在：不想修改應用 背景常駐程式（daemon）設定，但又希望它能處理更多埠號的情境。例如，Telnet 預設使用 TCP 埠 23；若你希望送往 TCP 埠 28 的網路流，也能由同一個 Telnet 背景常駐程式（daemon）處理，就可以加上一條 REDIRECT 規則，將「目的埠 28」的流量重新導向到「目的埠 23」。

DNAT 動作

如前文所述，「DNAT」主要用於實作**埠轉發（Port Forwarding）**：
當來自「net」區域的封包進入基礎平臺時，透過 DNAT 可以將其「目的 IP 位址」改寫為位於「dmz」或「loc」區域某台主機的 IP 位址，然後再轉送出去。被轉發的目標主機不一定要是實體機器，只要該主機（包含虛擬主機）能夠處理這些被轉送的網路流即可。

Border Add Rule (添加規則)

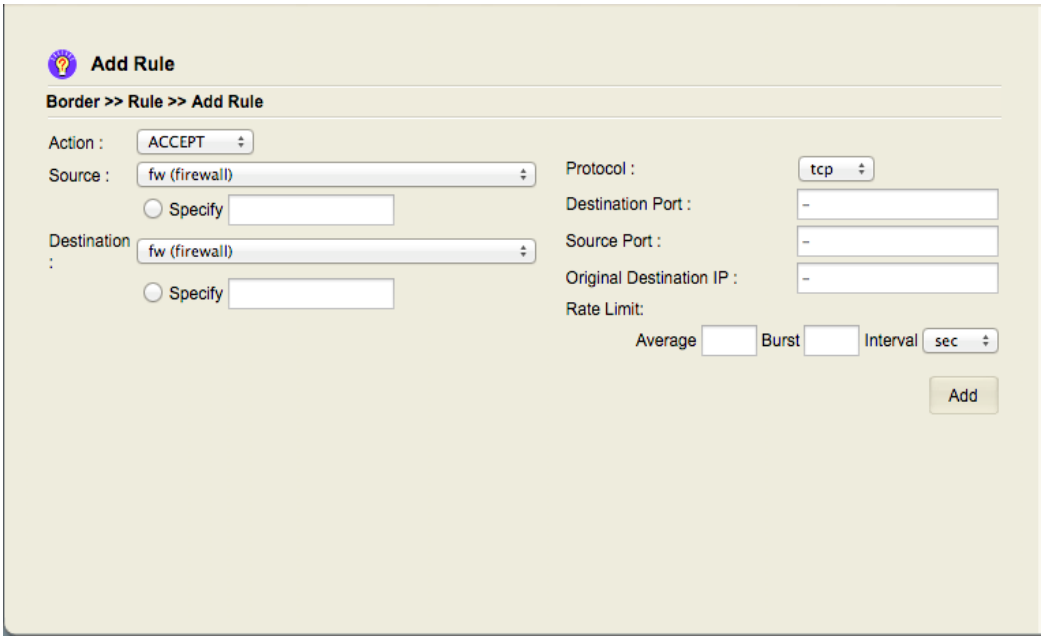


圖 41: 新增規則螢幕擷圖

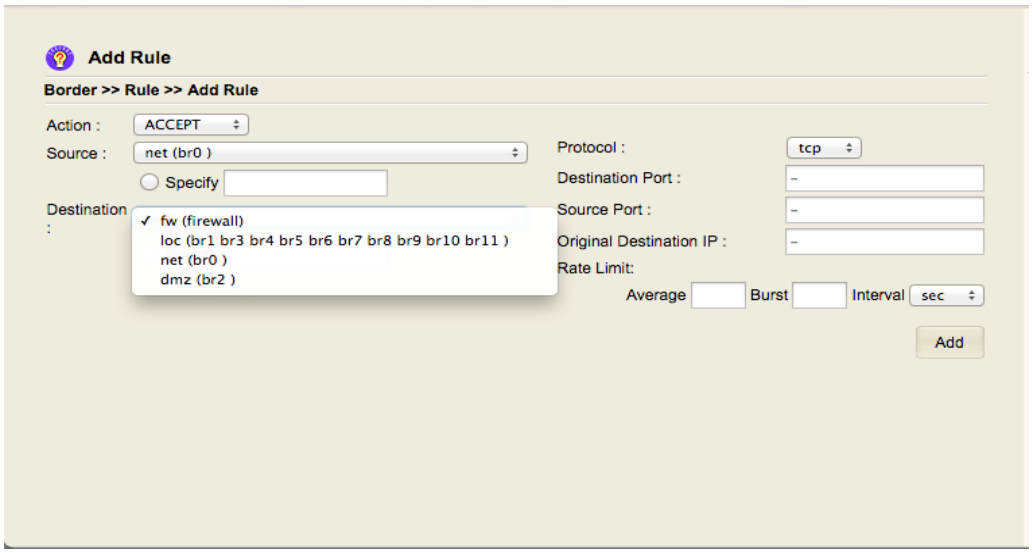


圖 42：規則的源和目的地關連

可以透過 **「Border → Rule → Add Rule」** 來新增例外規則。

在新增規則之前，請先確認目標主機所在的位置（無論是虛擬主機，或具備實體硬體的主機）。
若是虛擬主機，請確認它的虛擬網路介面已正確掛載到預期的橋接器上；若是實體主機，則請確認其實體網路線已連接到正確的介面與交換器。

對於位在 **「dmz」區域** 的主機，**無法透過 DHCP 由基礎平臺取得 IP 位址**。原因是：在基礎平臺中已預先定義了「dmz 區域不得存取 fw 區域」的規則，因此來自 dmz 的 DHCP 封包會被基礎平臺拒絕。
因此，**dmz 區域內的主機必須手動設定 IP 位址**，才能正常對外通訊。

這些預先定義的規則本身不會顯示在網頁介面中，但在新增例外規則之前，建議先熟悉它們的行為。 例如：**從「loc」區域到「dmz」區域的連線原本就被預設允許**，因此再新增一條「允許 loc → dmz」的規則意義不大； 相反地，**從「dmz」到「loc」的連線預設是被封鎖的**，若有需要，就必須透過新增例外規則來放行。

同樣地，由於「net → fw」的連線在預設情況下也是被拒絕的，所以如果希望從網際網路一側直接連到基礎平臺（fw），就需要新增相對應的例外規則。例如：

Source ：net
Destination ：fw
Protocol ：TCP
Destination Port：23

設定完成後請按下 **「Add」** 按鈕，並在之後重新啟動邊境控制引擎，規則才會正式生效。
以上只是其中一個範例；按下“Add”按鈕。它允許“telnet connections”（TCP 埠23）從“net”區域

到“fw”區域。規則將在重啟邊控引擎後生效。
在下列部分中，我們將提供一些示例供參考在接下來的章節中，我們會提供更多實際使用情境供參考。

Allowing Exceptions for TCP Connections from dmz to loc (dmz → loc 的 TCP 連線例外規則)

「dmz」區域對「loc」區域的連線在預設情況下會被封鎖。不過，在本例中我們希望新增一條規則，**允許來自「dmz」區域的 TCP 連線，存取「loc」區域中 IP 位址為 172.16.9.12 的主機**。

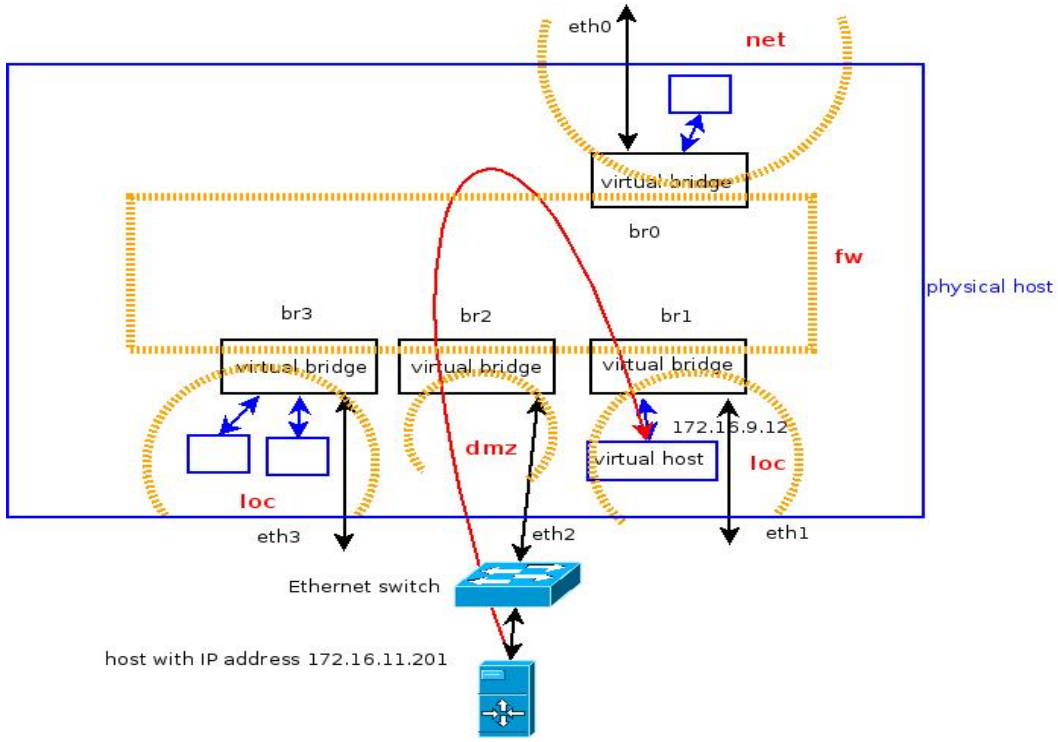


圖 43：從區域 "dmz" 到 "loc" 中新增例外規則

請參考下方螢幕擷圖：在「loc」區域中，我們將目標主機的 IP 位址明確指定為 **172.16.9.12**。

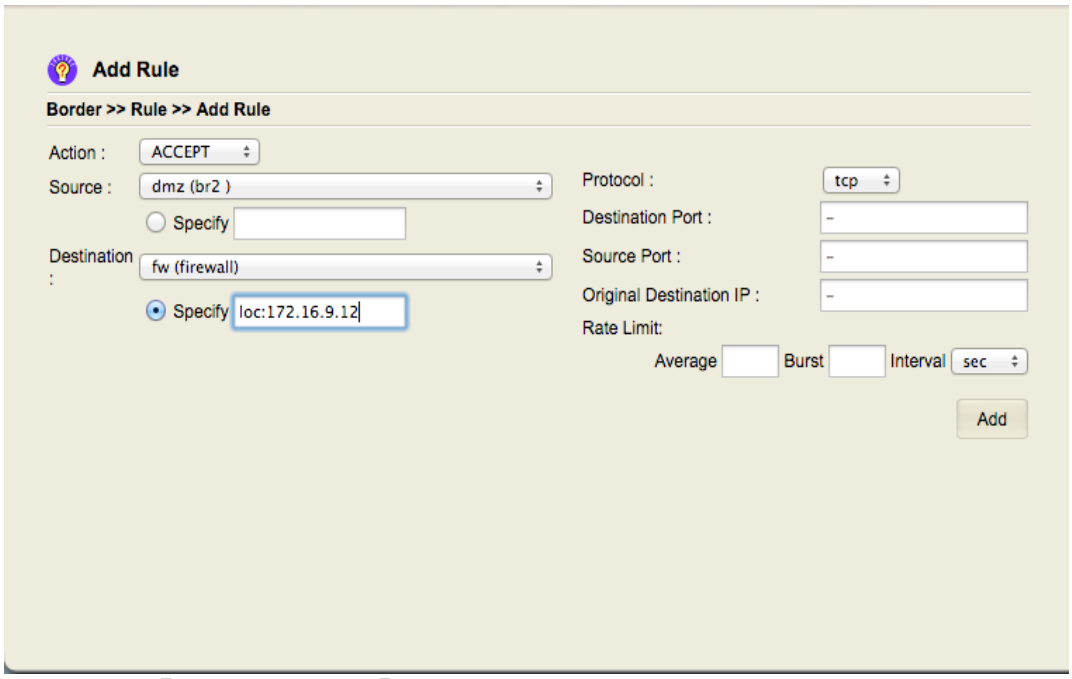


圖 44：從「dmz」連線到「loc」中目標主機的設定畫面

若要指定特定主機，請在 「Specify」 旁的輸入框中使用下列格式：

- zone:IP_ADDRESS
- 或 zone:SUBNET

以本例來說，可以填入：

- loc:172.16.9.12
- 或 loc:172.16.12.0/24

設定完成後按下 「Add」 按鈕，即可新增規則。
之後可以在 「Border → Rule → List / Remove Rule」 中檢視所建立的規則。

Border >> Rule >> List / Remove Rule

Current Rules

Action	Source	Destination	Protocol	Destination Port	Source Port	Original Destination IP	Rate Limit	Remove
?SECTION	NEW							
ACCEPT	fw	net	udp	53				
ACCEPT	loc	fw	tcp	22				
ACCEPT	fw	loc	udp	137:139				
ACCEPT	fw	loc	tcp	137,139				
ACCEPT	fw	loc	udp	1024:	137			
ACCEPT	loc	fw	udp	137:139				
ACCEPT	loc	fw	tcp	137,139				
ACCEPT	loc	fw	udp	1024:	137			
ACCEPT	loc	fw	tcp	ssh				
ACCEPT	net	fw	tcp	http				
ACCEPT	net	fw	tcp	443				
ACCEPT	net	fw	tcp	8080				
ACCEPT	net	fw	tcp	8081				
ACCEPT	net	fw	tcp	8082				
ACCEPT	net	fw	udp	1194				
ACCEPT	net	fw	udp	1195				
ACCEPT	net	fw	udp	7777				
ACCEPT	net	fw	udp	4569				
ACCEPT	net	fw	tcp	25				
ACCEPT	net	fw	tcp	23	-	-		
ACCEPT	net	fw	tcp	22	-	-		
ACCEPT	net	fw	tcp	5901-5909	-	-		
ACCEPT	dmz	loc:172.16.9.12	tcp	-	-	-		

圖 45：將規則從“dmz”顯示到“loc”上的主機。

拒絕或中斷連線

「net → loc」的連線在預設情況下已經被禁止，因此**不需要再額外新增規則**來拒絕或封鎖這類連線；同樣地，「dmz → loc」的連線預設也會被封鎖。相反地，「loc → net」的連線預設是允許的。
如果你希望**阻擋來自 loc 區域對 net 區域的 HTTP 存取**，就可以新增一條規則，條件例如：

Source ：loc
Destination ：net
Protocol ：TCP
Destination Port：80
Action ：DROP（或 REJECT）

這樣就只會封鎖 loc → net 的 HTTP（TCP 80）流量，其它協定仍維持原本的預設行為。

同理，你可對 HTTP 與 HTTPS 分別使用 TCP 埠 80 與 443 做類似的控制。請輸入這兩個埠號，做適當的設定後，按下 **「Add」** 新增規則。
這些規則會在 **重新啟動 Border Engine** 後正式生效。

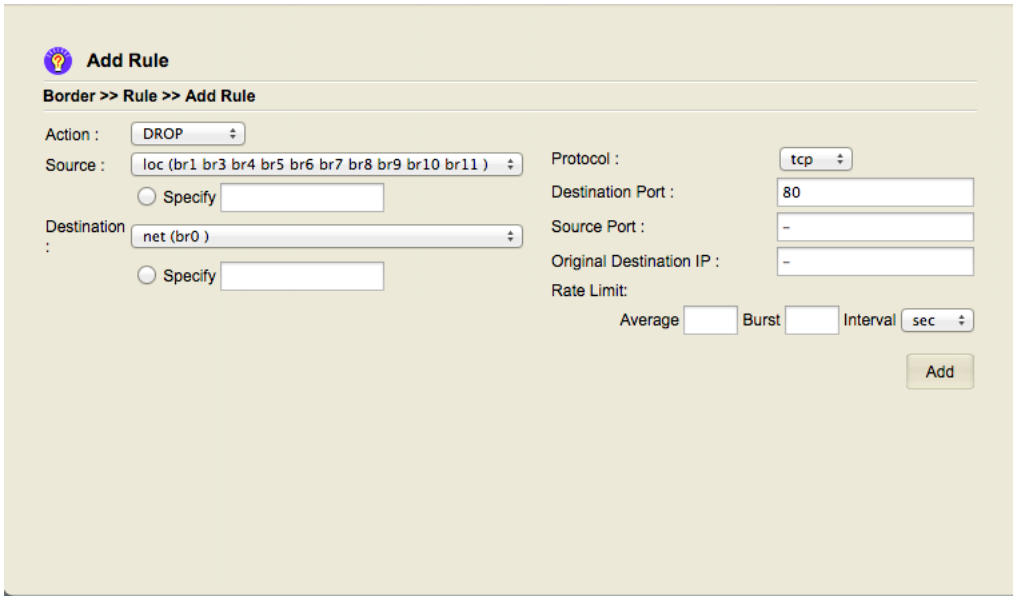


插圖 46：從“loc”到“net”的 HTTP 流量丟棄螢幕截圖

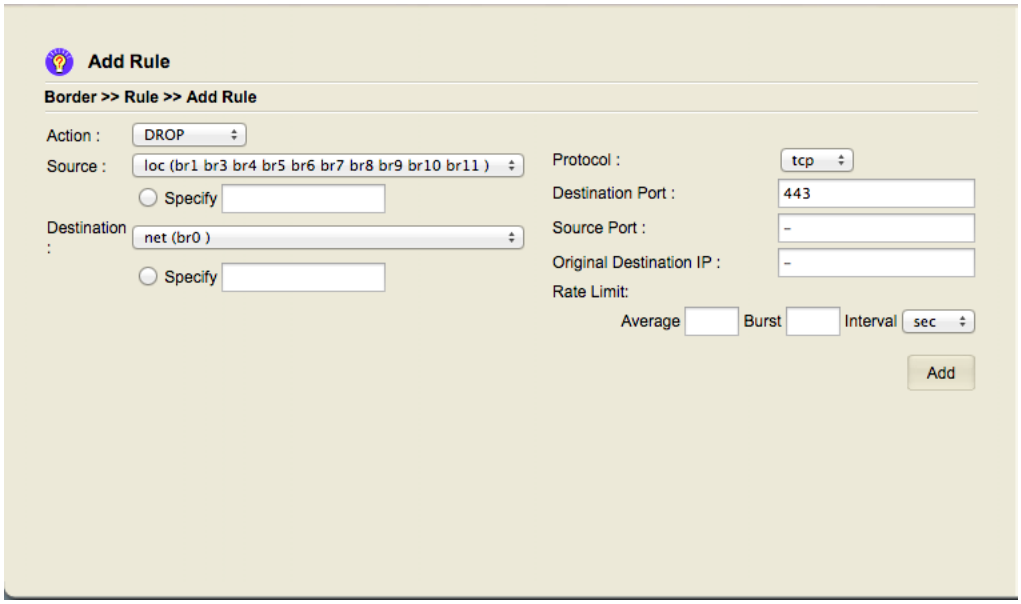


圖 47：螢幕截圖，從「loc」到「net」的 HTTPS 流量丟棄。

在做完以上設定後，以下是 設定後的結果。如下圖所示。

ACCEPT	net	fw	udp	7777				
ACCEPT	net	fw	udp	4569				
ACCEPT	net	fw	tcp	25				
ACCEPT	net	fw	tcp	23	-	-		
ACCEPT	net	fw	tcp	22	-	-		
ACCEPT	net	fw	tcp	5901-5909	-	-		
ACCEPT	dmz	loc:172.16.9.12	tcp	-	-	-		
DROP	loc	net	tcp	80	-	-		
DROP	loc	net	tcp	443	-	-		

圖 48：HTTP／HTTPS（TCP 80 與 443 埠）規則範例

Redirect Traffic to Another Port of the Base Platform (將網路流重新導向至基礎平台的其他連接埠)

當既有網路環境或舊系統服務埠號無法調整時，可使用 **REDIRECT 動作**將某一 TCP/UDP 埠的網路流量轉送到另一個埠。

實作範例：替代標準 TELNET 埠

假設標準的 **TELNET 連線 (TCP 埠 23)** 在環境中被封鎖，但允許使用其他埠號（例如 TCP 埠 29）。我們可以透過 REDIRECT 將服務導回原埠：

欄位	設定值	說明
Action	REDIRECT	執行重定向動作
Source	net	流量來源
Destination	fw	流量目的地（防火牆/基礎平台）
Protocol	TCP	使用 TCP 協定
Destination Port	29	外部連線發送的埠號
Redirect to Port	23	服務實際監聽的埠號

運作原理：

送往 TCP 埠 29 的流量，會在基礎平台上被**透明地**重新導向到 TCP 埠 23，由原本監聽 23 埠的 TELNET 服務來處理。 🚨 **注意：** 此規則需在**重新啟動 Border Engine 之後**才會生效。

應用層協定的潛在限制

雖然埠重定向能在傳輸層 (TCP/UDP) 順利轉送封包，但必須留意**上層網路協定**是否能正常運作。

- **問題點：** 許多應用層服務（如 HTTP）在其內容（例如網頁程式碼、設定檔）中，可能包含**硬指向**原始埠號的本機 URL 或參考連結。
- **後果：** 一旦服務埠改用重定向轉送，這些包含舊埠號的內部連結就可能**無法完整載入或發生錯誤**，導致服務功能受損。

這條規則會在 **重新啟動 Border Engine** 之後才會生效。

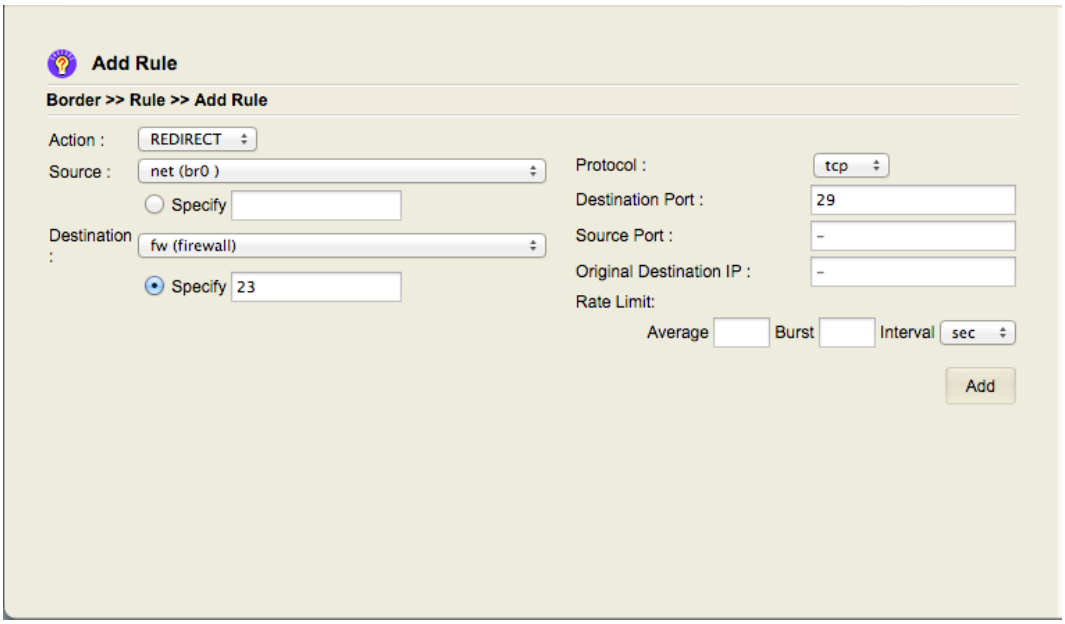
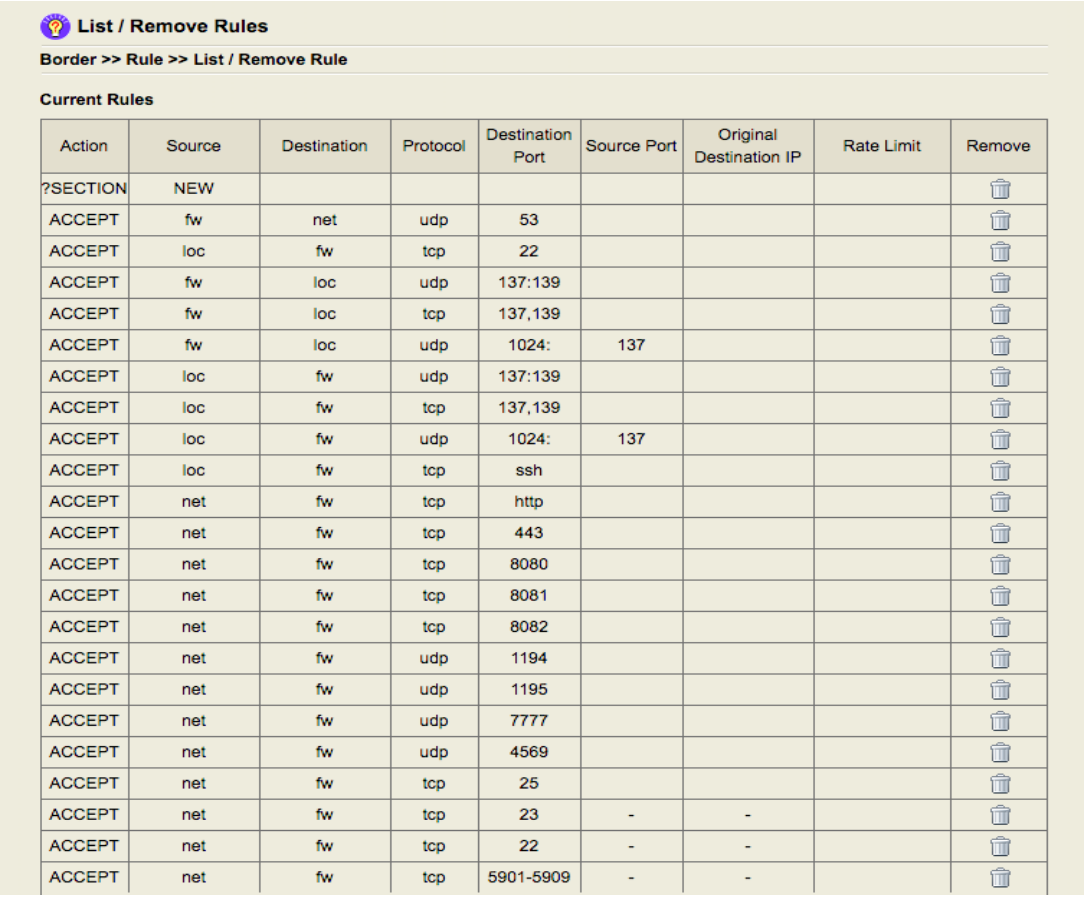


插圖 49: 將流量重定向到不同的埠

Action	Source	Destination	Protocol	Destination Port	Source Port	Original Destination IP	Rate Limit	Remove
?SECTION	NEW							
ACCEPT	fw	net	udp	53				
ACCEPT	loc	fw	tcp	22				
ACCEPT	fw	loc	udp	137:139				
ACCEPT	fw	loc	tcp	137,139				
ACCEPT	fw	loc	udp	1024:	137			
ACCEPT	loc	fw	udp	137:139				
ACCEPT	loc	fw	tcp	137,139				
ACCEPT	loc	fw	udp	1024:	137			
ACCEPT	loc	fw	tcp	ssh				
ACCEPT	net	fw	tcp	http				
ACCEPT	net	fw	tcp	443				
ACCEPT	net	fw	tcp	8080				
ACCEPT	net	fw	tcp	8081				
ACCEPT	net	fw	tcp	8082				
ACCEPT	net	fw	udp	1194				
ACCEPT	net	fw	udp	1195				
ACCEPT	net	fw	udp	7777				
ACCEPT	net	fw	udp	4569				
ACCEPT	net	fw	tcp	25				
ACCEPT	net	fw	tcp	5901-5909	-	-		
REDIRECT	net	23	tcp	29	-	-		

圖 50：顯示列表中 REDIRECT 規則

Border Rule List / Remove (列表或刪除規則)



List / Remove Rules

Border >> Rule >> List / Remove Rule

Current Rules

Action	Source	Destination	Protocol	Destination Port	Source Port	Original Destination IP	Rate Limit	Remove
?SECTION	NEW							
ACCEPT	fw	net	udp	53				
ACCEPT	loc	fw	tcp	22				
ACCEPT	fw	loc	udp	137:139				
ACCEPT	fw	loc	tcp	137,139				
ACCEPT	fw	loc	udp	1024:	137			
ACCEPT	loc	fw	udp	137:139				
ACCEPT	loc	fw	tcp	137,139				
ACCEPT	loc	fw	udp	1024:	137			
ACCEPT	loc	fw	tcp	ssh				
ACCEPT	net	fw	tcp	http				
ACCEPT	net	fw	tcp	443				
ACCEPT	net	fw	tcp	8080				
ACCEPT	net	fw	tcp	8081				
ACCEPT	net	fw	tcp	8082				
ACCEPT	net	fw	udp	1194				
ACCEPT	net	fw	udp	1195				
ACCEPT	net	fw	udp	7777				
ACCEPT	net	fw	udp	4569				
ACCEPT	net	fw	tcp	25				
ACCEPT	net	fw	tcp	23	-	-		
ACCEPT	net	fw	tcp	22	-	-		
ACCEPT	net	fw	tcp	5901-5909	-	-		

圖 51：從列表刪除規則

如同在前面幾節所示，要刪除某條規則，只需點擊該規則右側的垃圾桶圖示。系統會先要求您確認刪除。完成變更後，請記得重新啟動 Border Engine（邊界引擎），新的設定才會生效。

Using DNAT for Port Forwarding (將流量轉送到基礎平台的其他埠)

埠轉發（Port Forwarding）是透過使用 **DNAT (Destination Network Address Translation)** 動作實現的。
DNAT 的核心功能是**修改基礎平台上接收到的資料包的目的地 IP 地址**，並根據修改後的新 IP 地址和埠號來轉發資料包。

範例一：僅轉發 IP（埠號不變）

若要將外部 HTTP 流量（TCP 埠 80）轉發到 dmz 區域內 IP 地址為 172.16.11.201 的主機，同時保持埠號為 80：

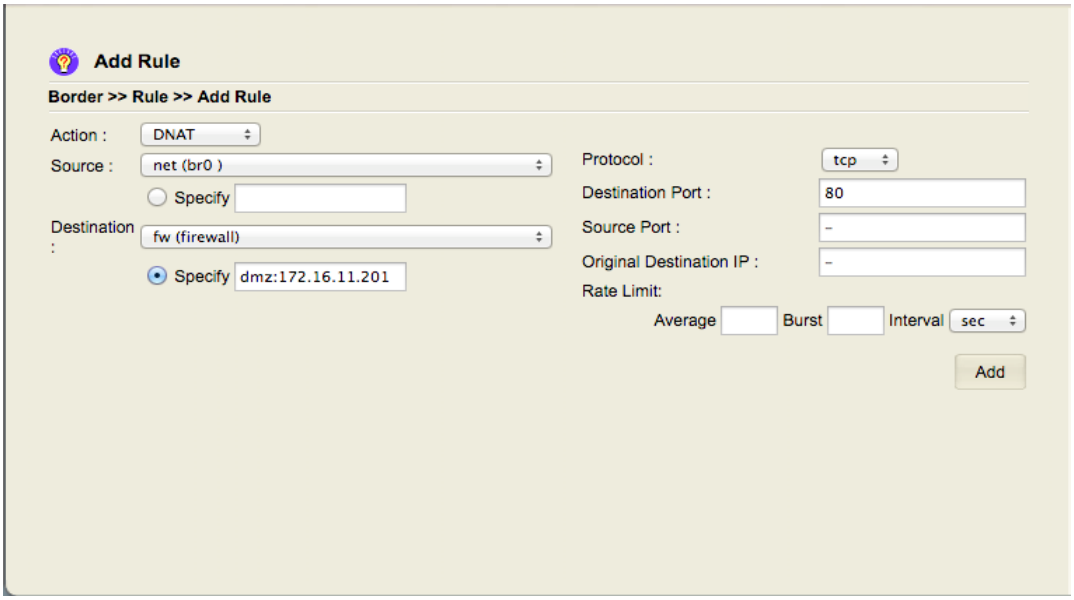


圖 52：使用 DNAT 進行埠轉發

欄位	設定值	說明
行動 (Action)	DNAT	執行目的地 IP 轉換
Source	net	來源（外部網路）
目的地 (Destination)	dmz:172.16.11.201	新的目的地 IP
Protocol	tcp	協定類型
目的地埠 (Destination Port)	80	外部連線發往的埠

效果：

基礎平台接收到的 TCP 埠 80 流量，將被轉發到內部主機 172.16.11.201 上的 **TCP 埠 80** 服務。

範例二：同時轉發 IP 與埠號

在實際應用中，我們經常需要將外部的一個非標準埠流量，轉發到內部主機的標準埠上進行處理。
例如，將外部發往 **TCP 埠 2929** 的流量，轉發到 dmz 區域主機 **172.16.11.201** 的 **TCP 埠 80**：

欄位	設定值	說明
行動 (Action)	DNAT	執行目的地 IP 與埠號轉換
Source	net	來源（外部網路）
目的地 (Destination)	dmz:172.16.11.201:80	新的目的地 IP 和埠號
Protocol	tcp	協定類型
目的地埠 (Destination Port)	2929	外部連線發往的埠

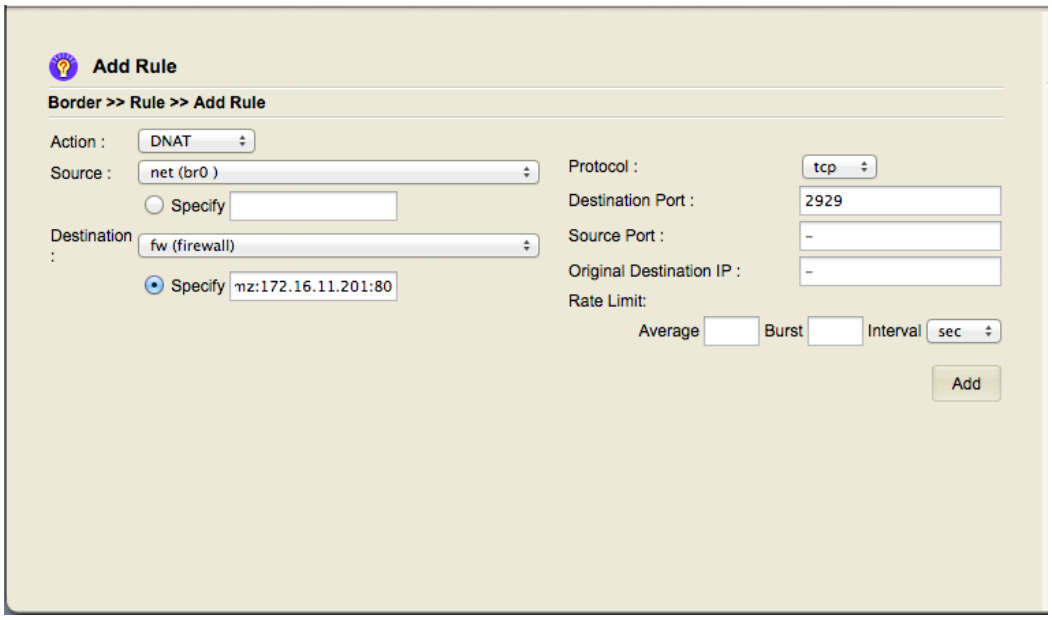


插圖 53：將埠轉發到具有不同埠的主機

按下“Add”按鈕後，規則將會顯示如下：

ACCEPT	net	fw	udp	1195				
ACCEPT	net	fw	udp	7777				
ACCEPT	net	fw	udp	4569				
ACCEPT	net	fw	tcp	25				
ACCEPT	net	fw	tcp	5901-5909	-	-		
ACCEPT	net	fw	tcp	22	-	-		
ACCEPT	net	fw	tcp	23	-	-		
DNAT	net	dmz:172.16.11.201	tcp	80	-	-		
DNAT	net	dmz:172.16.11.201:80	tcp	2929	-	-		

圖 54：埠轉發規則列表

總結存取方式：

假設基礎平台的外部 IP 地址（例如 br0 網卡）是 192.168.11.202。外部用戶即可透過存取 URL：
http://192.168.11.202:2929/

成功訪問到內部 dmz 區域主機 172.16.11.201 上的 TCP 埠 80 服務。

IP Load Balance (IP 負載平衡)

當需要將特定連線組合 (IP 位址、傳輸協定、連接埠號) 的連入流量分散給多台主機處理時，這就是 **IP 負載平衡**。這打破了傳統「**埠轉發**」(一次只能轉發到單一主機) 的限制。

在本系統中，負載平衡規則可以透過 「**Border → Rule → IP Load Balance**」 進行設定。

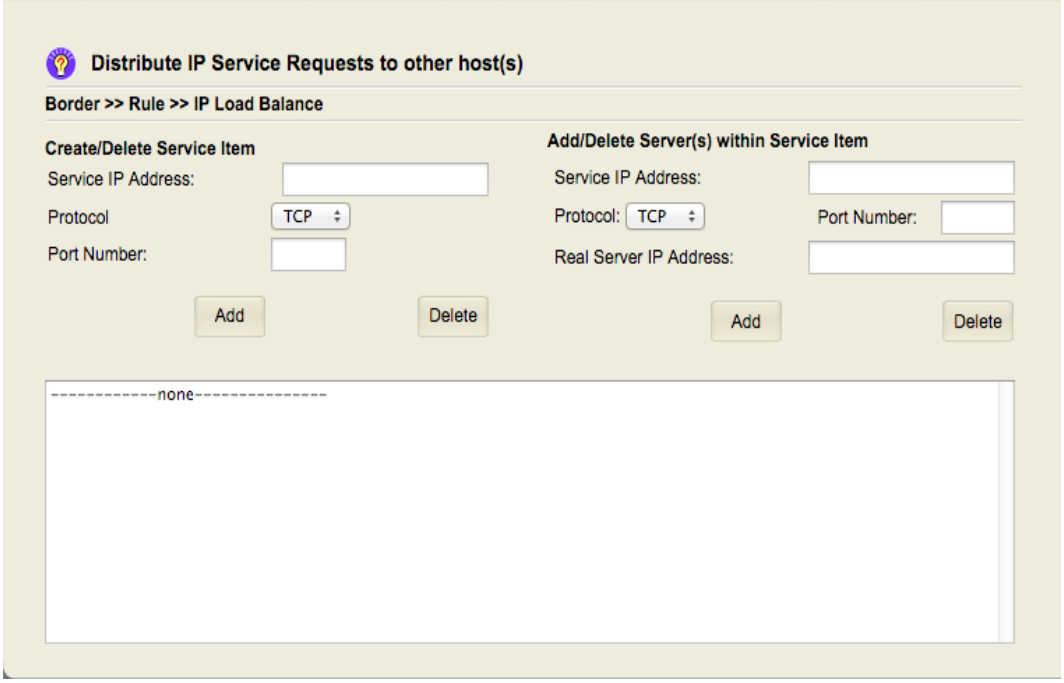


圖 55：IP 負載平衡螢幕擷圖

核心運作機制

1. 服務項 (Service Item) 定義

對於每一組特定的 (IP 位址, 傳輸協定 [TCP/UDP], 埠號)，我們將其定義為一個「**服務項**」。

2. 分派策略：輪詢 (Round-Robin)

- 對同一個服務項，可以對應**多台後端主機**。
- 基礎平台會以**輪詢** (Round-Robin) 的方式，將進入的網路請求均勻分配到這些後端主機上。

3. 連線保持：黏性連線 (Sticky Session)

- 一旦某個用戶端的請求被分派到其中一台主機，在之後的 300 **秒**內，該用戶端的**後續請求**仍會被強制送往**同一台主機**。
- **目的**： 確保單一用戶會話的連續性和穩定性。

4. 關鍵限制：應用程式資料同步

本功能僅負責將網路**連線負載**分散到多台伺服器。

！ 應用程式設計者責任： 至於應用程式本身所需的資料（如 session 狀態、檔案或資料庫內容），**必須**由系統／應用設計者自行在這些主機之間維持**同步**。基礎平台不會也無法干預主機內部的資料處理。

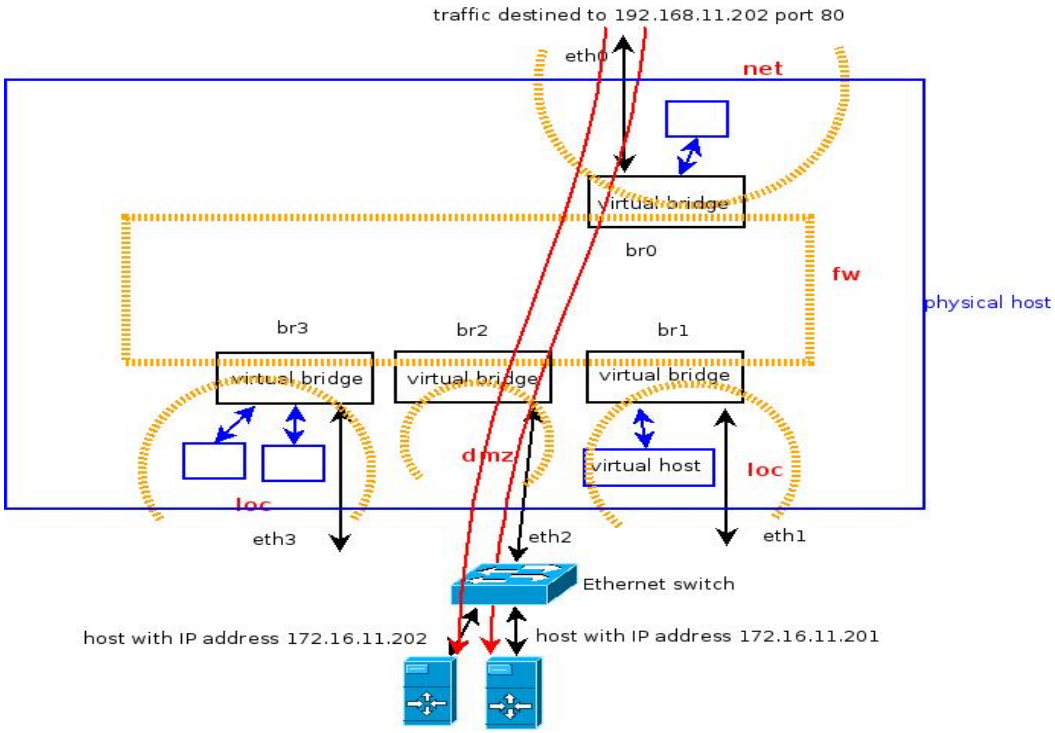


圖 56：將 HTTP 流量分發到 2 台主機

HTTP 負載平衡範例設定

目標： 將送往基礎平台 IP 192.168.11.202 的 **HTTP 流量 (TCP 埠 80)**，分散給 dmz 區域內的兩台主機 172.16.11.201 與 172.16.11.202 處理。

下步驟一：前置準備 (允許連線)

在設定負載平衡之前，必須先在「net 區域」新增一條**例外規則**，允許來自 net 到 fw 的 TCP 埠 80 流量，確保基礎平台能接受外部 HTTP 連線。

步驟二：設定服務項與後端主機

1. **創建服務項：** 使用 IP 位址 192.168.11.202 以及 TCP 埠 80 創建一個新的服務項目。
2. **新增後端主機：** 依序將主機 172.16.11.201 和 172.16.11.202 加入此服務項，作為承載負載的伺服器。

結果： 之後到達 192.168.11.202:80 的 HTTP 請求，將會被輪流分派給 172.16.11.201 和 172.16.11.202 兩台主機處理。

設置程序如下：使用“192.168.11.202”以及 TCP 埠80創建一個服務項目。

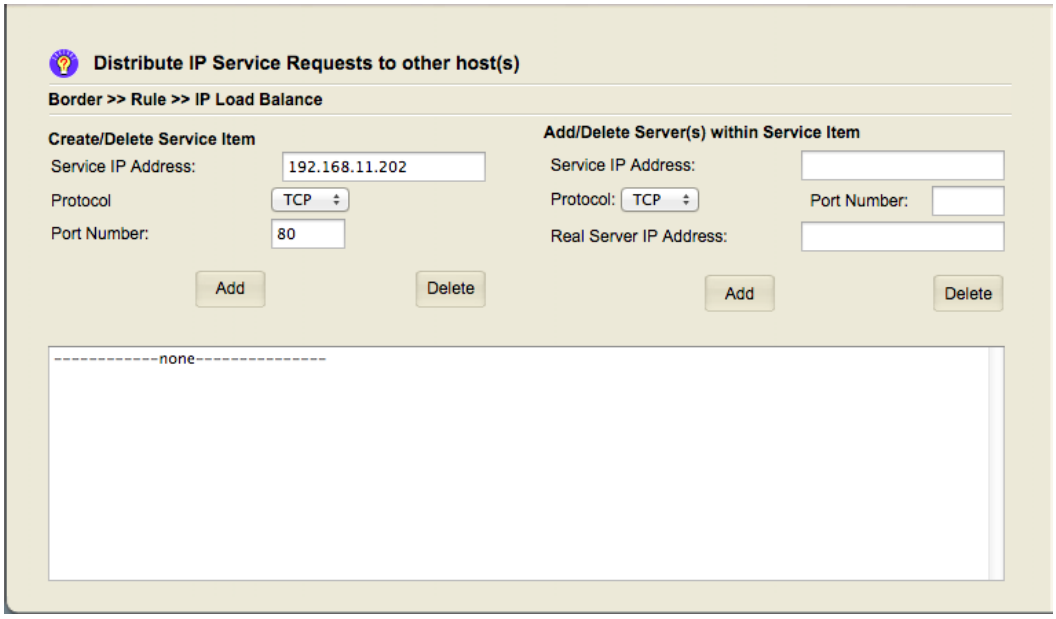


插圖 57：創建負載平衡的服務項目

然後，按下“ADD”按鈕來添加這個服務項目。

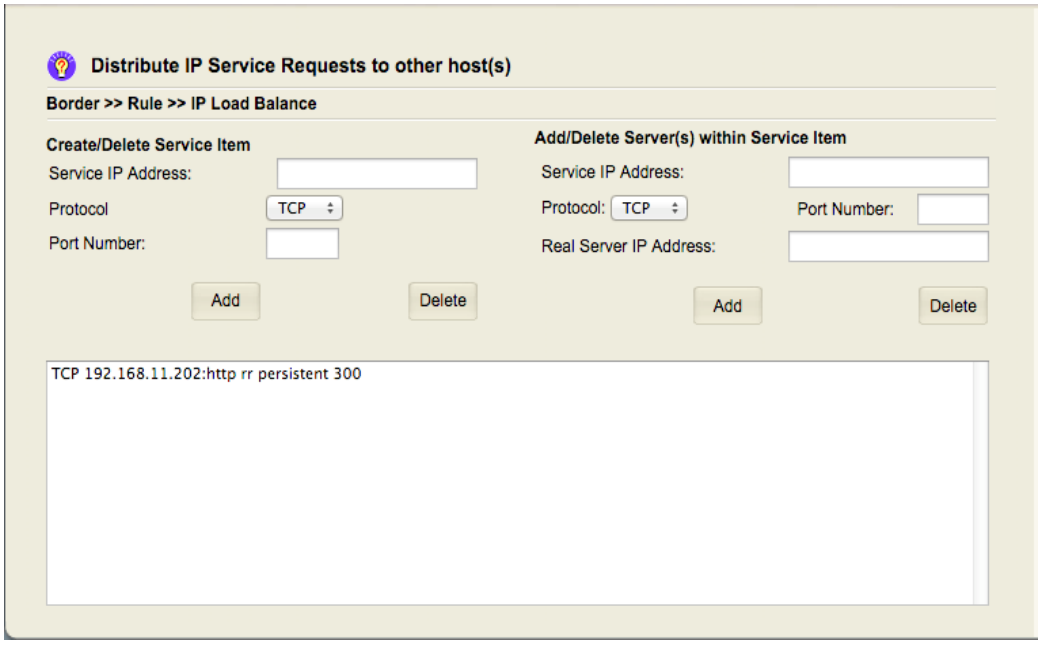


圖 58：負載平衡服務項目列表

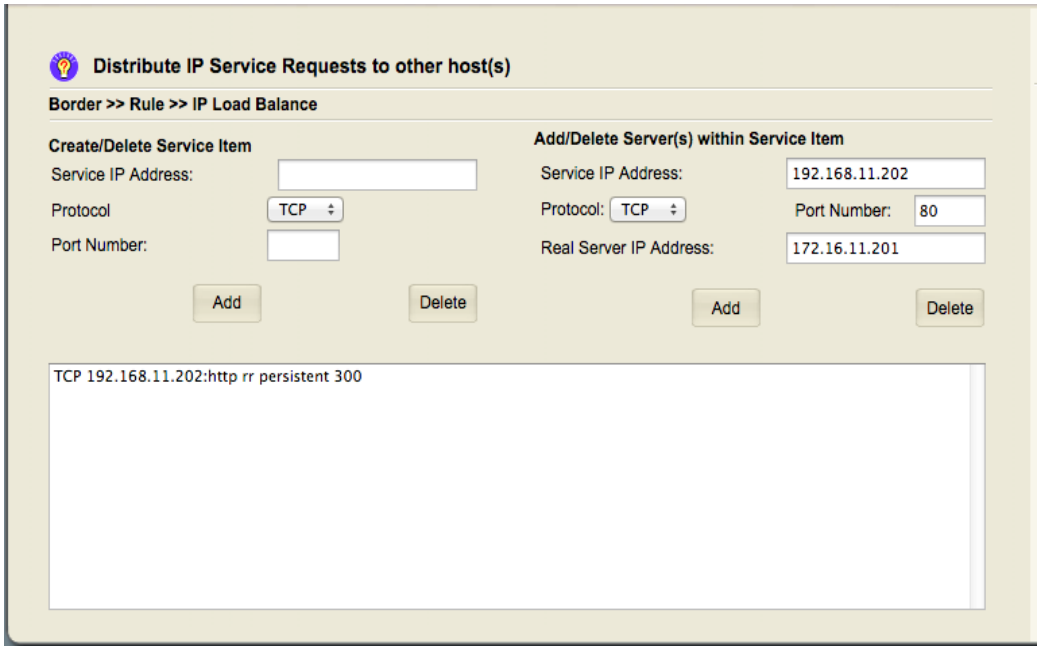


圖 59：將主機加入服務項以參與負載平衡

接著，將主機 **172.16.11.201** 加入此服務項，作為其中一台承載負載的伺服器。

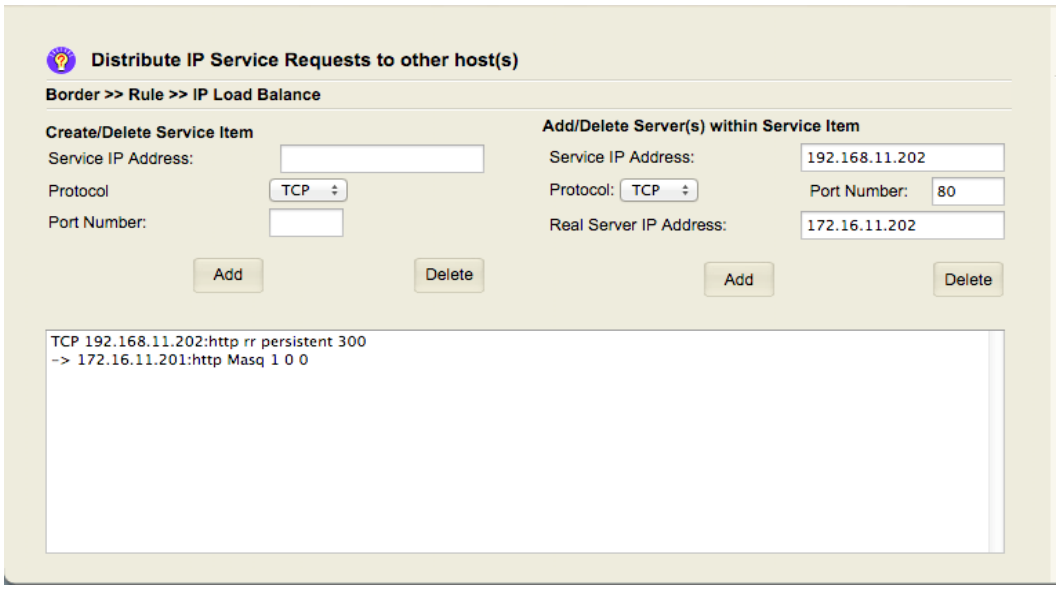


圖 60：在服務項目中新增一個主機以實現負載均衡

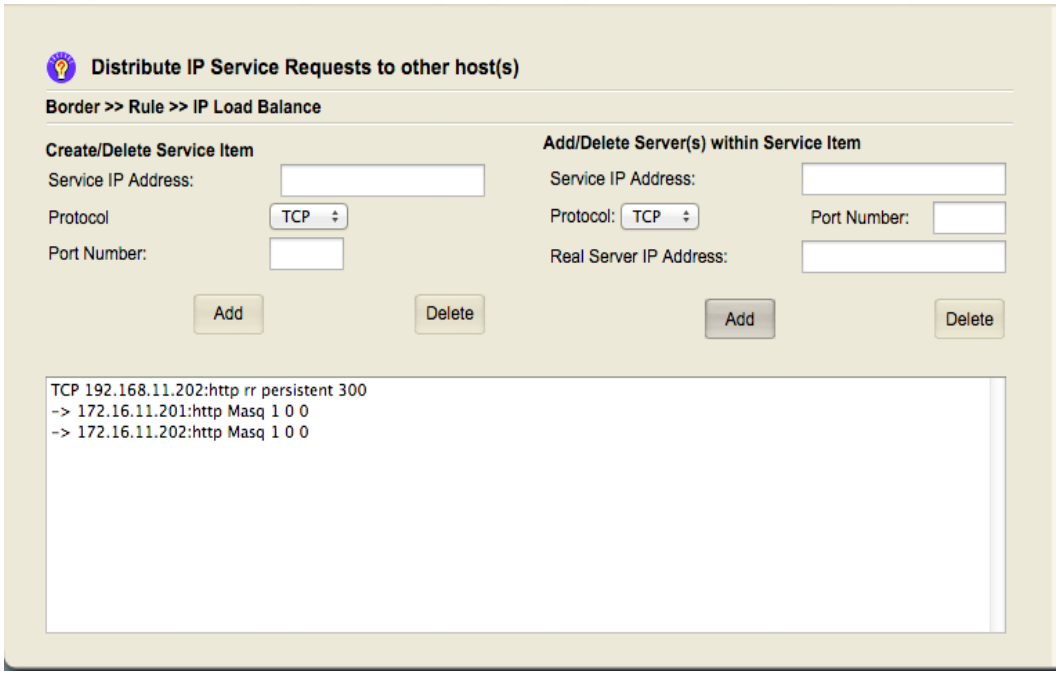


圖 61：服務項目（用於負載平衡）的服務主機列表

因此，我們能看到主機“172.16.11.201”和“172.16.11.202”負責處理到達“192.168.11.202”的 HTTP 請求。

Use Web Proxy (使用網頁代理伺服器)

Web 代理是一種**應用層**控制機制，其控制對象是 **HTTP 協定**本身。這與下列**網路層**（Network/Transport Layer）動作有根本區別：

- **網路層動作：** DNAT、ACCEPT、DROP、REJECT、REDIRECT 等，它們都是根據 **IP 位址或 TCP／UDP 埠號**來控制網路流量。

代理伺服器接收用戶端的請求，代為轉發給實際伺服器，並將伺服器回應轉回給用戶端。

一、Web 代理的核心功能

Web 代理不僅提供基本的網際網路存取服務，還具備多種控制和優化能力：

- **內容管理：** 快取先前載入過的資料，加速存取。
- **安全過濾：** 過濾特定網頁連結或內容。
- **存取限制：** 限制某些網站只能在特定時段被存取。
- **常見情境：** 通常部署在內部網路（loc），並代為存取網際網路（net）。

二、進階應用情境：跨區域存取與路由優化

Web 代理在處理**安全隔離**和**複雜路由**問題時，展現出獨特優勢：

1. 繞過區域安全隔離（net ➔ loc）

在預設的安全架構下，**net 區域**（外部使用者）**禁止**直接連入 **loc 區域**（內部 Web 主機），以確保內部安全。然而，**fw 區域**（基礎平台）對 loc 區域的連線是**允許**的。

- **架構解析：**
  - **loc 區域：** 內部 Web 主機 (172.16.9.x 等)，僅供內部同仁使用。
  - **net 區域：** 外部世界，預設規則：net ➔ loc **禁止**。
  - **fw 區域：** 基礎平台/防火牆主機 (Web 代理部署於此)，規則：fw ➔ loc **允許**。
- 解決方案：

  外部使用者 (net) ➔ 連線到 fw 上的 Web 代理 ➔ 代理 ➔ 存取 loc 區域 Web 主機。

  結論：

  「讓 net 的人看見 loc 的 Web 主機」

  實質上等同於「讓 net 的人透過 fw 上的 Web 代理進行轉接」。

2. 簡化 VPN 路由問題

在實務上，Web 代理常與 **VPN** 配合使用，以避免傳統 VPN 連線可能遇到的路由和閘道設定問題：

方式	路由流程	潛在問題	代理解決方案
僅使用 VPN	VPN 用戶 ➔ 內部 Web 主機	內部主機可能未將「VPN 子網」設定為閘道，導致連線失敗。	VPN 用戶 ➔ **fw Web 代理** ➔ 代理 ➔ 內部 Web 主機

- **優勢：** 透過代理轉接後，對於內部 Web 主機來說，所有請求的來源都是**基礎平台 (fw) 本身**的 IP 位址。路由變得單純，**避開**了因 VPN 子網路由設定缺失所導致的存取問題。

3. 滿足 IP 來源限制（舊系統相容性）

Web 代理會在所有網路介面上監聽 HTTP 請求，使得所有對內部 Web 主機的請求看起來都像是**由基礎平台自己發起**的。

- **效果：** 統一連線來源 IP。
- **應用：** 對於需要使用 Web 代理來滿足**舊式 Web 應用程式只接受來自特定 IP 位址連線**的需求尤其重要。在不修改舊系統的前提下，讓使用者先連到代理，即可用代理的「允許 IP 位址」存取應用程式。

接下來的章節，將說明如何調整基礎平台上所提供的 Web 代理設定。

。

Web Caching (Web 緩存與代理設定)

基礎平台上的網路代理預設具備 Web 緩存功能，可用來儲存先前載入的網頁資料，以提升存取速度並減少頻寬使用。

一、預設設定與調整

預設監聽埠號： 網路代理預設監聽 TCP 埠 3128。
設定修改： 如需變更預設埠號或其他相關設定，可透過「Border >> Proxy >> Web Caching」頁面進行調整。

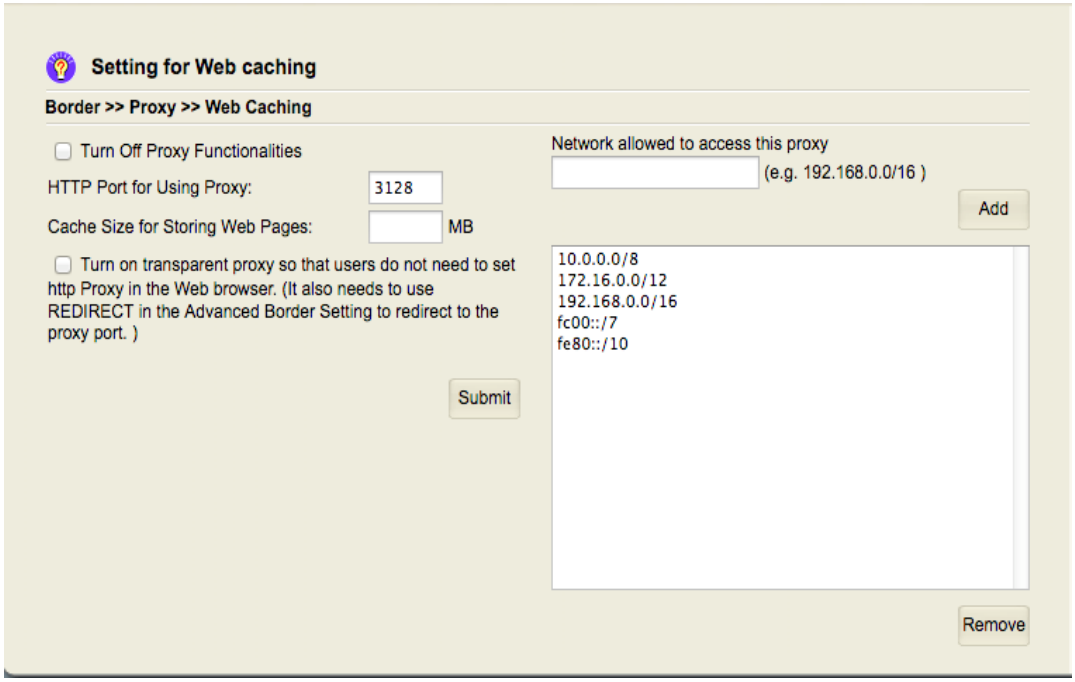


示意圖 62：網際網路代理緩存及存取畫面截圖

二、代理伺服器使用模式

使用網路代理伺服器主要有兩種模式：

1. 傳統代理 (手動設定)

- **機制：** 在一般情況下，用戶端需要在**瀏覽器**或應用程式中手動設定代理伺服器的 **IP 位址與埠號**（例如 192.168.1.1:3128）。
- **優點：** 控制精確，通常用於指定用戶或特定連線。

2. 透明代理 (Transparent Proxy)

- **機制：** 利用基礎平台上的 **REDIRECT** 動作，將特定埠（例如標準 **TCP 埠 80** 的 HTTP 連線）的流量在**不經過用戶端設定**的情況下，**自動轉送到**此代理伺服器的埠號（例如 3128）。
- **優點：** 用戶端**無需在瀏覽器中進行任何額外的代理設定**，網路流量會被基礎平台自動劫持並導向代理伺服器。

三、存取權限控制

- **設定清單：** 允許哪些網路（或 IP 範圍）可以連線到此代理伺服器，會明確列在設定清單中（位於頁面右側）。
- **調整權限：** 您可以根據需求**調整這些條目**，以**限制或放寬**對代理伺服器的存取範圍。

URL Screening (URL 篩選與 HTTPS 內容檢查限制)

啟用網路代理伺服器後，您可以透過「**Border >> Proxy >> URL Screening**」（網址篩選）功能，來**管理並新增需要封鎖的特定網址清單**。

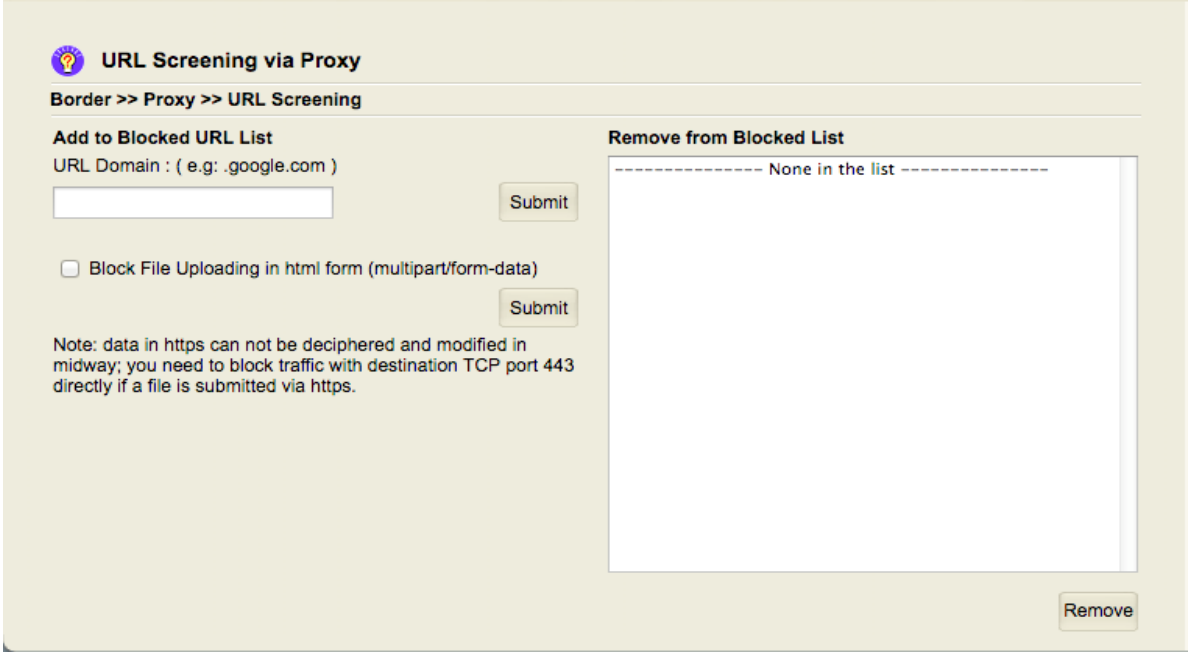


圖 63：網際網路代理伺服器中的 URL 過濾 (網址篩選)

一、HTTP 流量下的內容檢查

對於使用 **HTTP** 協定的連線，Web 代理伺服器具有**深度內容檢查**的能力：

- **檔案上傳檢查：** 代理伺服器可以**看見 HTTP 內文**，從而判斷使用者是否正在透過 HTML 表單進行檔案上傳。
- **精細控制：** 代理可以對這類上傳請求進行**檢查、拒絕或丟棄**，實現對檔案上傳動作的精細阻擋。

二、HTTPS 流量下的限制：加密的挑戰

一旦連線改用 **HTTPS** 協定，代理伺服器的內容檢查能力就會受到極大限制：

特性	HTTP	HTTPS
資料可視性	代理可見 HTTP 內文	資料被 TLS 通道加密包裹
代理解析能力	可判斷「上傳檔案」等具體動作	代理無法解密內容
阻擋精細度	可針對「上傳檔案」精細控制	只能看到「一條 HTTPS 連線」

核心問題：

如果代理沒有目標網站憑證所對應的「私鑰」，就無法解密 TLS 通道內的內容。代理只能辨識這是一個加密連線，但無法得知連線內傳輸的到底是瀏覽頁面還是上傳檔案。

三、阻擋 HTTPS 檔案上傳的對策

由於無法只針對「檔案上傳」這個動作做精細控制，若確實需要阻止使用者透過某 HTTPS 網站上傳檔案，剩下的做法只能是**粗糙但有效**的：

- **唯一對策： 直接封鎖整條 HTTPS 流量**。
- **實作方法：** 例如，直接封鎖該網站的 **TCP 埠 443** 連線，讓使用者根本無法連上該 HTTPS 網站。

Proxy Access Block Time (代理存取時間控制)

本功能允許您設定精確的時間條件，以控制使用者透過代理伺服器存取網頁的行為。
您可以依據需求設定規則，僅在特定時段允許存取，或在設定的時段內予以阻擋，並在其餘時間內一律封鎖所有透過代理的網頁瀏覽行為。

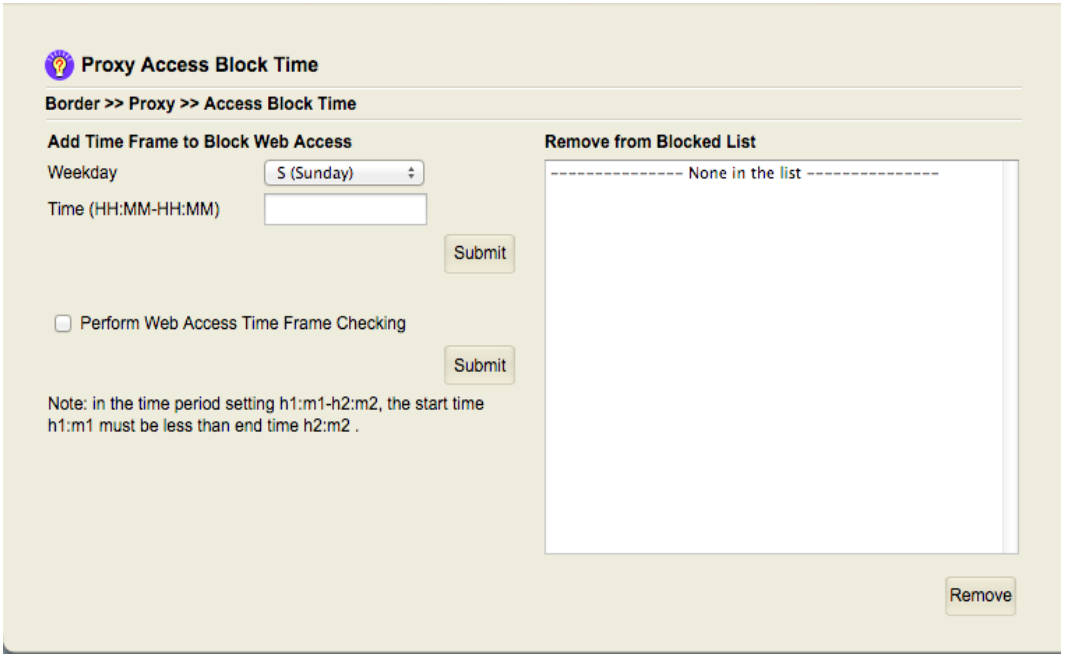


圖 64: 時間區段設定：封鎖經由 Web 代理的 HTTP 存取

設定範例：限制存取時段

若要設定**每週一上午的特定時段禁止**使用 HTTP 代理存取網頁：

設定欄位	設定值	說明
Time	08:00-12:00	定義每日的時段區間（上午 8 點至中午 12 點）。
Weekday	M (Monday)	指定生效的星期。

效果： 在每週一的 **08:00～12:00** 時段內，所有透過代理伺服器發出的 HTTP 請求將會被**禁止**。

Traffic Bandwidth Control (頻寬控管)

Azblink NFV 平台採用 **「介面總頻寬上限 + 定義類別（頻寬大小及順序）+ 依流量條件歸類」** 的三層架構來對網路流量進行頻寬控管與優化。

頻寬控管的四個步驟

以下是配置**流量頻寬控管**的完整流程：

步驟 1：設定介面總頻寬

定義： 在特定的網路介面（例如 eth0）上，定義流量的最大上行與下行頻寬上限。
範例： 上／下行頻寬設定為 100 Mbit/s。
管理介面： **Border >> Bandwidth >> Interface Limiting**

步驟 2：定義流量類別及優先順序規則 (Priority Classes)

將通過該介面的流量切分為數個**優先類別（Priority Classes）**，並為每一級指定：

設定項目	說明
最低保證頻寬 (Min Rate)	保證該類別在頻寬競爭時能獲得的最低頻寬。
最高可用頻寬 (Max Rate)	該類別在任何情況下能使用的頻寬上限。
優先順序 (Priority)	決定在頻寬不足時，哪個類別的流量優先傳輸（數字愈小愈優先）。

優先等級範例：
第 1 級：最高優先，Min 100 kbit/s，Max 180 kbit/s。
第 2 級：Min 介面總頻寬的 1/4，Max 可用整體頻寬。
管理介面： Border >> Bandwidth >> Priority Classes

步驟 3：依流量條件歸類 (Traffic Prioritizing)

透過設定歸類規則，指定哪些符合條件的封包要被分派到特定的頻寬類別：
歸類條件可包含： 來源 IP 位址或網段、目的 IP 位址或網段、TCP / UDP 埠號。
預設處理： 若封包不符合任何自訂規則，系統會自動將其歸類到**預設的「第 3 級」**類別。
管理介面： Border >> Bandwidth >> Traffic Prioritizing

步驟 4：自動分配與生效

完成以上三個步驟的設定後，介面的實際流量將會根據已定義的優先順序、保證頻寬和限制上限進行自動分配與傳輸。

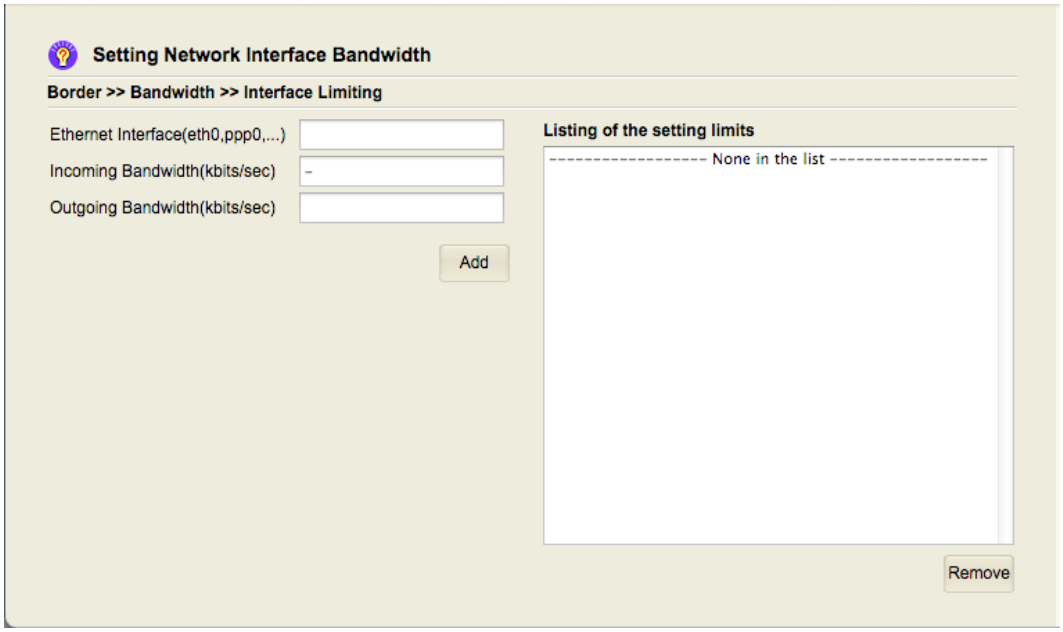


圖 65：設定網路介面頻寬

Setting Network Interface Bandwith (網路介面頻寬設定)

透過 **「Border >> Bandwidth >> Interface Limiting」**，可以為各個乙太網路介面設定「入站」與「出站」的頻寬上限。
在實體網卡的硬體規格上，連線速率通常是 **100Mb/s 1000Mb/s 2500Mb/s 5000Mb/s 或 10000Mb/s** 等固定檔位，並透您可以透過 **「Border >> Bandwidth >> Interface Limiting」**介面，為每一個乙太網路介面設定**「入站」（Ingress）與「出站」（Egress）**的頻寬上限。

頻寬設定的定義與作用

本頁面的設定是用來控制實際允許經由該介面收發的流量上限，它不會改變網卡硬體層次的實體連線速率（例如 100Mb/s, 1000Mb/s 等自動協調速率）。

入站頻寬限制的技術特性

在實施**入站頻寬（Ingress Bandwidth）**限制時，必須了解其底層限制：
處理時機： 流量實際上已經被網卡接收進來，並交由基礎平台開始處理。
限制方法： 平台唯一能採取的做法是：當入站流量超過您設定的值時，丟棄（Drop）部分已接收的封包。
潛在影響： 雖然此方法可以限制頻寬，但頻繁的丟包會導致對端必須不斷重送數據，這可能會

造成上層應用程式的有效傳輸速率明顯下降，影響使用者體驗。

實作範例與單位要求

項目	說明
管理介面	Border >> Bandwidth >> Interface Limiting
範例設定	入站與出站頻寬都設定為 100 Mbits/sec。
單位要求	介面上的數值必須以 kbits/sec 為單位輸入。

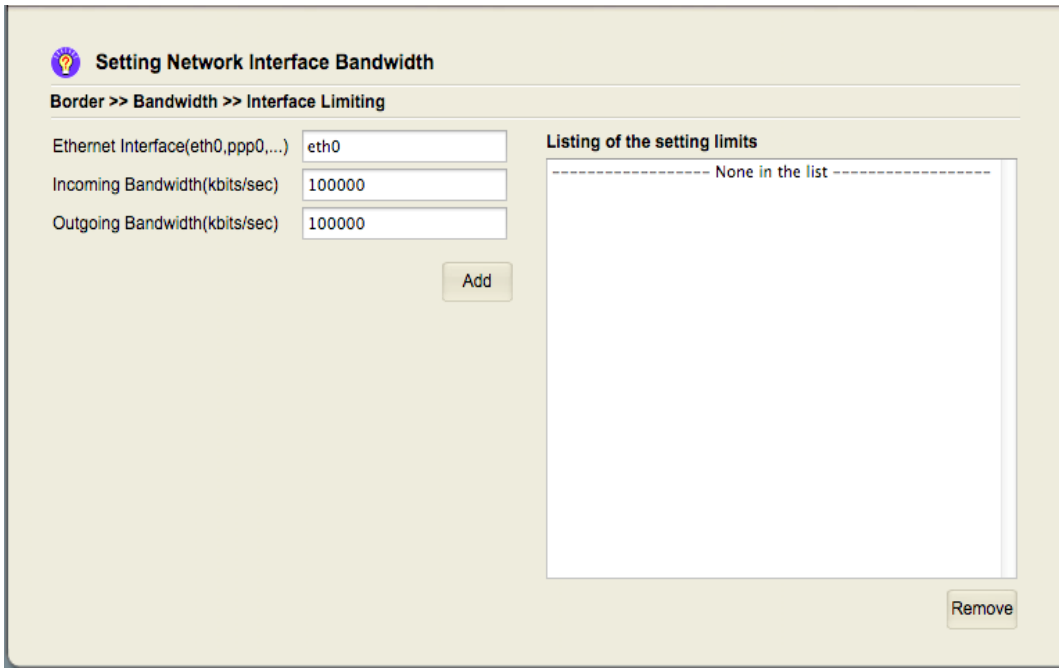


圖 66：設置 inbound 和 outbound 帶寬 圖 66：設定 inbound 和 outbound 帶寬

按下“Add”按鈕後，右側的框會顯示為

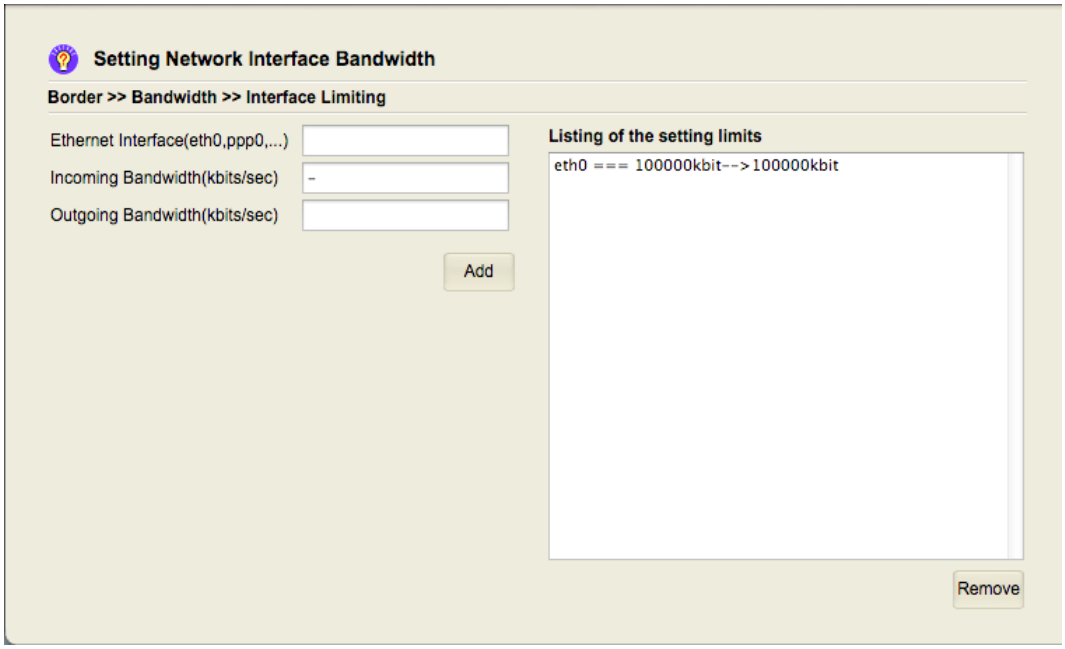


圖 67: 界面帶寬設置後屏幕快照

與此同時，4個優先級等級會自動創建。它們可以通過“Border >> Bandwidth >> Priority Classes”來查看。

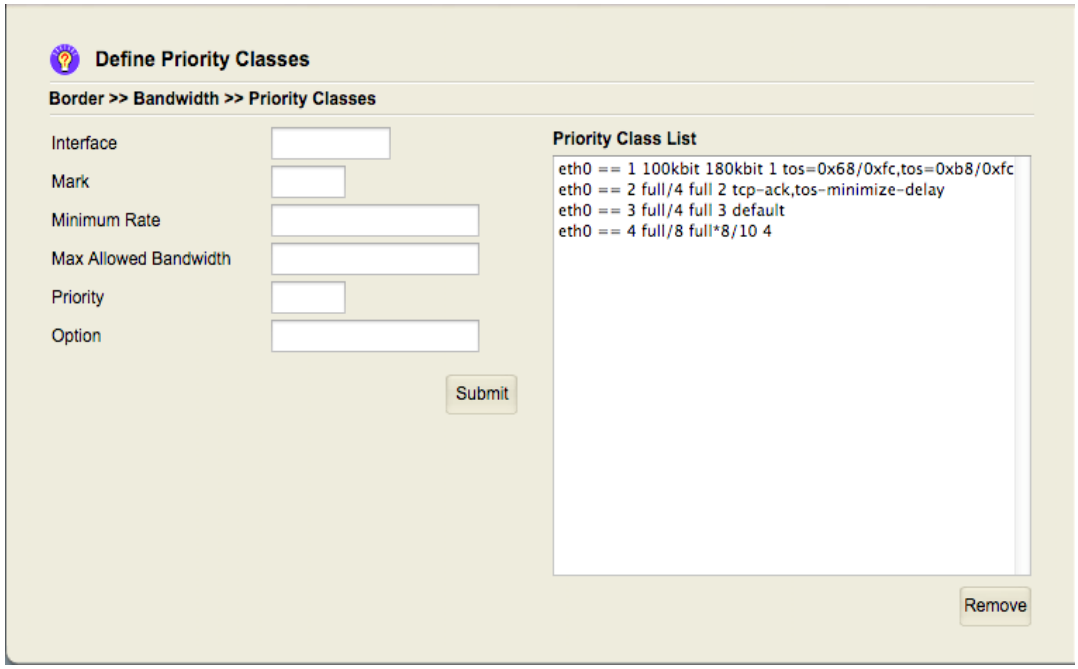


圖 68：設置接口帶寬限制後優先級類

在設定頻寬類別（Priority Classes）時，您可能需要根據您的應用需求來修改設定。然而，有一個**強制性的要求**必須遵守：

核心要求：保留默認類別

- **默認類別：** 如圖示（Mark “3” 所在位置），**預設類別**是系統中必須存在的配置。
- **必要性：** 無論您進行何種修改或調整，**您應該始終有一個默認類別用於網路流量**。

！啟動失敗警告： 如果您移除了預設類別，或沒有任何流量歸類規則來涵蓋所有未指定的流量，**邊框引擎 (Border Engine) 將無法成功啟動**。

- **目的：** 預設類別是確保不符合任何自定義規則的流量也能獲得一個處理級別（如前文所述，通常是第 3 級），避免流量在歸類時發生遺漏或錯誤。

Define Priority Classes (定義優先級類別)

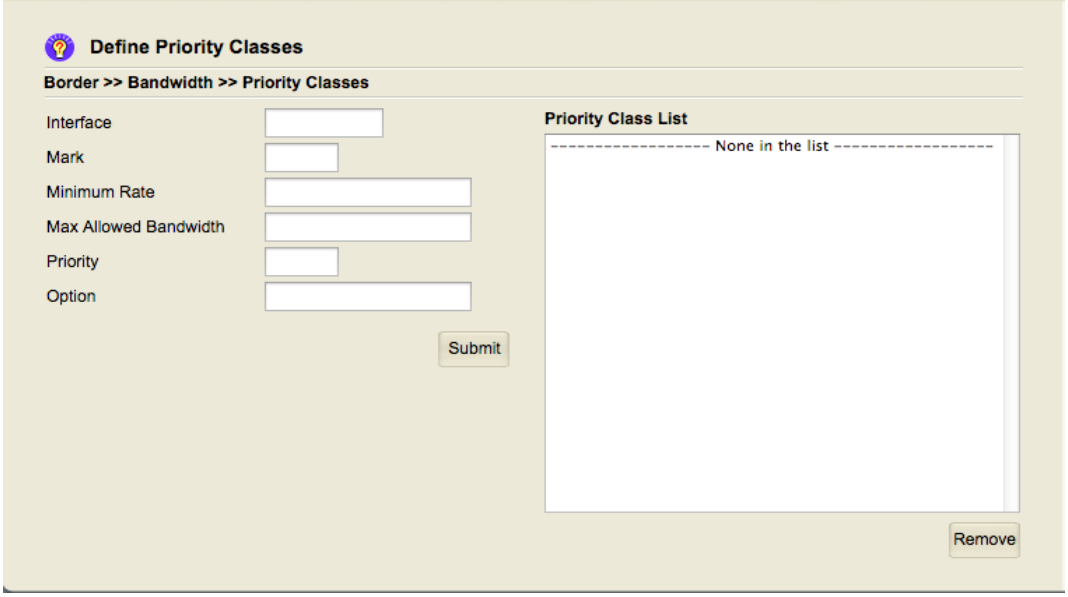


圖 69：優先級類別定義設定畫面截圖

在設定完乙太網路介面的總頻寬後，您可以透過定義優先級類別，將流量細分並給予不同的服務品質 (QoS)。系統預設會建立 4 個優先等級，這些設定可以隨時修改。

一、類別欄位與選項定義

設定每個類別時，需要配置以下欄位並可使用特定的 Option 來進行更精確的標記：

欄位名稱	數值範圍	說明
Mark	1～255 的整數	用於內部標識，供流量歸類規則 (Traffic Prioritizing) 使用。
Priority	1～65535 的整數	類別的優先順序值（數字愈小，優先級愈高）。

欄位名稱	數值範圍	說明
Option	多種型式	定義流量的歸類方式或特殊標記。

優先級類別 Option 欄位詳解

Option 欄位用於定義該頻寬類別的特殊屬性或分類條件，它允許系統根據封包的特定標記或狀態進行歸類。

Option 類型	語法/名稱	說明
預設類別	default	這是**系統的保底類別**。所有**未被其他規則明確標記（Mark）的流量，都會自動歸類到此類別。確保所有流量都有處理級別，是引擎成功啟動的必要條件。
DiffServ/TOS 標記	tos=0xvalue/0xmask	透過檢查 IP 封包標頭中的 服務類型 (Type of Service, TOS) 或 差異化服務 (DiffServ) 位元組來進行分類。 0xvalue: 期望匹配的 TOS 位元值。 0xmask: 用來決定哪些位元參與匹配的遮罩。
內建 TOS 名稱	tos-<tosname>	使用標準化的內建名稱來簡化 TOS/DiffServ 匹配，系統會自動填入對應的 value/mask 組合。
TCP 流量特殊標記	tcp-ack	專門用於匹配所有僅帶有 ACK 旗標**的 TCP 封包。這些封包通常非常小，但對傳輸效率至關重要，單獨分類可賦予它們更高的優先級。

內建 TOS 名稱及其對應值

使用內建的 tos-<tosname> 是一種更簡便的設定方式，對應的 TOS/DiffServ 值和遮罩如下：

TOS 名稱	目的	對應 TOS/Mask 值
tos-minimize-delay	**儘量減少延遲**	0x10/0x10
tos-maximize-throughput	**儘量提高吞吐量**	0x08/0x08
tos-maximize-reliability	**儘量提高可靠度**	0x04/0x04
tos-minimize-cost	**儘量降低成本**	0x02/0x02
tos-normal-service	**一般服務**	0x00/0x1e

Export to Sheets

應用提示： 通常會將 **VoIP 語音流量**設定為 tos-minimize-delay 類別（第 1 類），以確保其低延遲需求。

關於 Mark 欄位的補充

雖然 Mark 欄位不是 Option 本身，但它與 Option 有協同作用：

- **Mark (1~255):** 這個數字是流量的內部標記。在 **Traffic Prioritizing (流量歸類規則)** 中，您可以根據**封包的 IP/埠號等資訊**來設定規則，將封包標記上一個 Mark 數字。
- **關聯性：** 隨後，這些帶有特定 Mark 數字的封包會被**自動歸類**到 Priority Classes 中**具有相同 Mark 號碼**的類別，從而獲得該類別所定義的 Min/Max Rate 和 Priority。

二、實務優先級設計建議

在實際規劃時，建議根據流量對**延遲和可靠度的敏感性**進行分類。以下是一個建議的四級用途範本：

等級	類別名稱	建議用途與範例流量	建議頻寬設置
第 1 類 (最高優先)	**即時互動流量**	**VoIP、視訊會議媒體流**、遠端桌面控制通道、DNS 請求等對延遲高度敏感的流量。	**Min Rate 較小** (e.g., 100～512 kbit/s)，**最高優先權**，確保在壅塞時仍能搶佔頻寬。
第 2 類 (高優先)	**關鍵商務應用**	企業 Web/API 服務 (CRM, ERP)、電子郵件 (SMTP/IMAP)、訊息系統的控制訊號等。	**Min Rate 設為總頻寬的 1/4**，Max Rate 可用到全頻寬，Priority 略低於第 1 類。
第 3 類 (預設類別)	**一般辦公與上網流量**	**所有未特別標記的封包**，例如一般網頁瀏覽、日常雲端硬碟存取、一般軟體更新等。	**Min Rate 總頻寬的 1/4～1/2**，Priority 介於第 2 類與第 4 類之間。
第 4 類 (最低優先)	**可延後/背景流量**	P2P/BT 類流量、大量備份、同步、批次檔案傳輸、大型作業系統自動更新等。	**Min Rate 總頻寬的 1/8** 左右，**Max Rate 限制**在總頻寬的 80% 以內，避免在尖峰時段佔用過多資源。

規劃建議：

建議先從「語音／視訊」、「關鍵業務」、「一般流量」、「背景/下載」這類粗分類開始實作。系統上線一段時間後，再根據實際的流量統計數據精確調整各類別的最低與最大頻寬設定。

Packet Marking For Traffic Control (封包標記與流量控制)

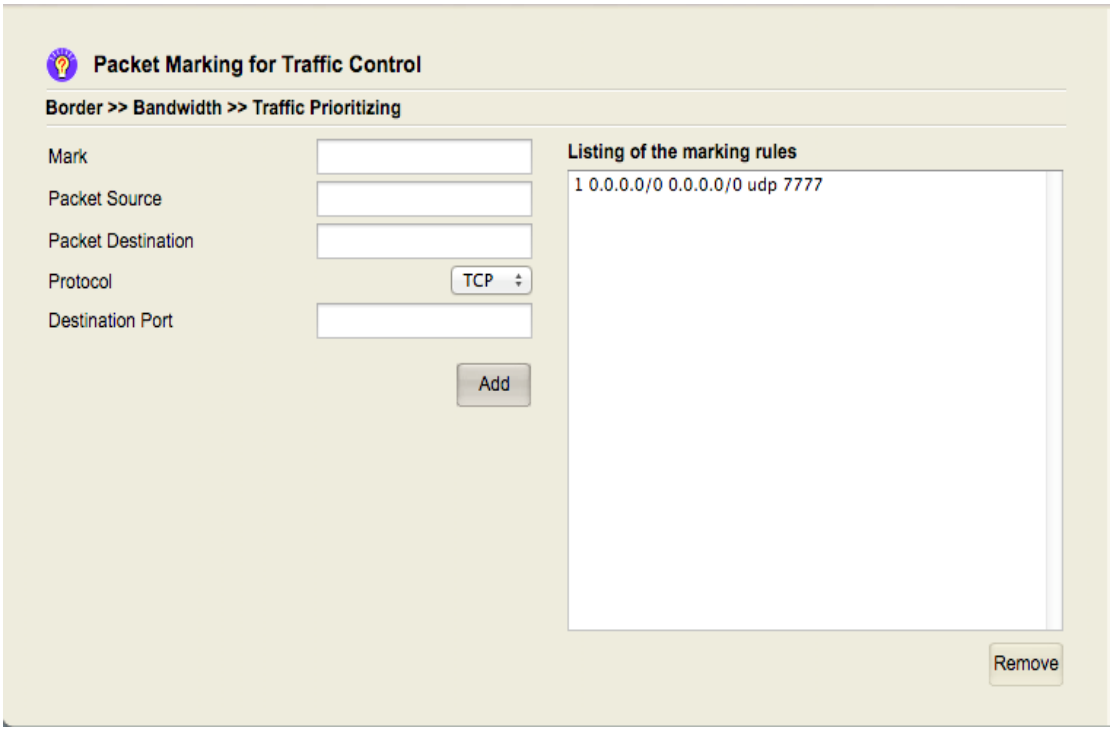


圖 70：流量優先順序設定畫面

在定義了優先級類別後，下一步就是建立**流量歸類規則 (Traffic Prioritizing)**，為每個類別指定要套用的「流量條件」，將符合條件的封包標記上對應的 **Mark** 值。

一、封包標記機制

流量歸類規則的作用是**檢查封包的屬性**，並將其**標記 (Mark)上一個數字（1 到 255）。這個 Mark 數字即是連接封包到特定優先級類別**的橋樑。

範例：站點對站點 VPN 流量設定

若要將使用 **UDP 埠 7777** 的「站點對站點 VPN」流量設定為**最高優先級**：

欄位名稱	設定值	說明
標記 (Mark)	1	將此流量標記為 Mark = 1，對應到最高優先級類別。
封包來源	0.0.0.0/0	任何來源 IP（Any Source）。
封包目的地	0.0.0.0/0	任何目的地 IP（Any Destination）。
協定 (Protocol)	UDP	協定類型。

欄位名稱	設定值	說明
目的地埠	7777	匹配 VPN 使用的特定埠號。

運作結果：

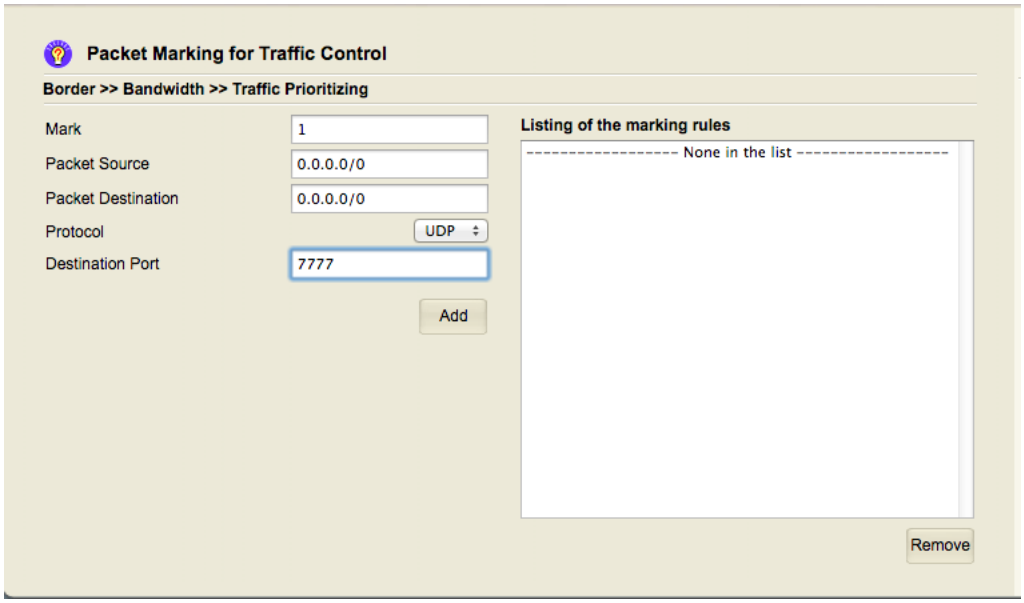


圖 71：設定流量優先標記

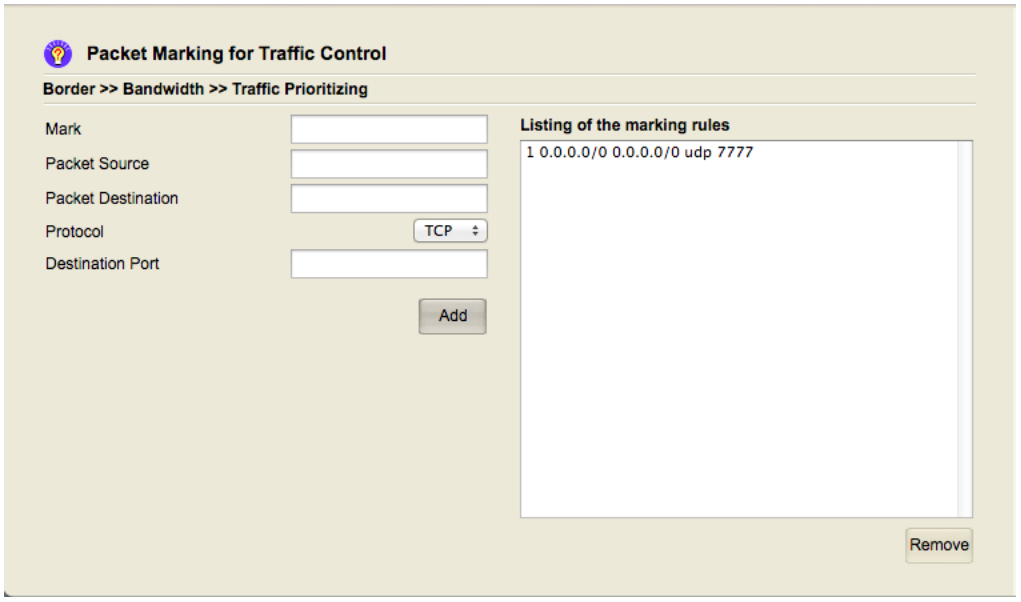


圖 72：封包標記規則列表畫面

任何符合此條件（目的地埠為 7777 的 UDP 流量）的封包，都會被標記為 Mark = 1，並依此獲得其對應類別的 Priority (例如：Priority = 1) 和 頻寬限制 (例如：Max 180 kbits/sec)。

二、實務設定考量與挑戰

雖然流量優先級主要用在**可用頻寬很有限**的情況，以確保在頻道壅塞時讓重要的網路流量先走，但要訂出合適的頻寬設定並不容易：

- **Min Rate (最小保證速率)：** 僅用於維持基本連線，實務上通常很低（例如 100 kbit/s），無法滿足實際應用需求。
- **Max Rate (最大可用頻寬)：**
  - 如果將上限設得**太高**，該類別的流量可能會佔用太多資源，導致其他應用程式「被擠到喘不過氣」。
  - 如果設得**太低**，則無法滿足關鍵應用程式的效能需求（如範例中的 VPN 應用，180 kbit/s 明顯不足）。

一般原則：

啟用頻寬限制，多半是因為系統遭到濫用（例如 P2P 或大檔案傳輸），必須對特定網段或特定類型流量「加以節流」，以保障關鍵業務的運行。因此，實際設定必須依照各自環境的使用型態來調整。

The Components of a Bridge and physical port default mapping (橋接器的元件與實體網口對應關係)

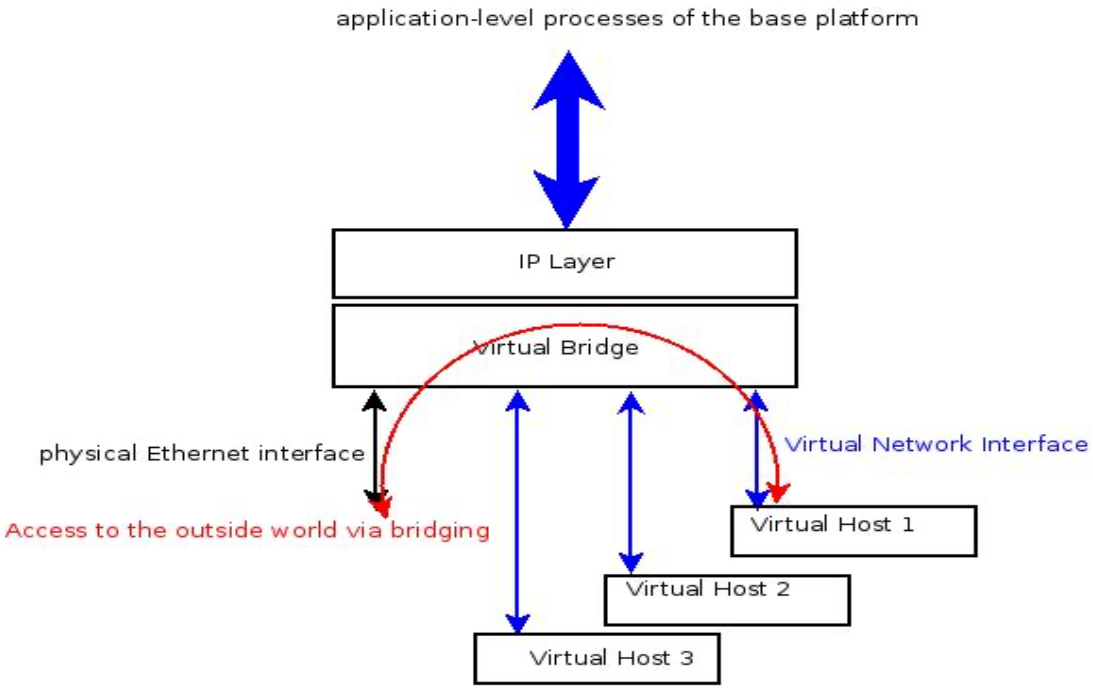


圖 73：橋接、基礎平台、物理乙太網接口和虛擬主機之間的關系

在 Azblink NFV 平台中，「橋接器 (Bridge)」是網路架構的基礎核心，用於連接虛擬介面、實體介面，並作為不同安全區域（Zone）的邊界。

一、橋接器的角色與區域劃分

- **網路中樞：** 虛擬主機的網路介面必須先**掛載**到某一個橋接器上，才能對外通訊。
- **區域邊界：** 區域劃分（如 net、loc、fw、dmz 等）正是以這些**橋接器的邊界**為基準來區分。
- **基礎平台 IP：** 基礎平台本身的 IP 位址都設定在這些橋接器介面上，作為其他主機的連入點。
- **預設數量：** 系統預先提供了 **12 個橋接器**，名稱依序為 **br0** 到 **br11**。

二、橋接器的組成元件

橋接器可以容納不同類型的網路介面：

介面類型	範例	說明
實體介面	eth0, eth1, ...	實際的乙太網路埠。
虛擬介面	tap0, tap1, ...	主要提供給**橋接模式的 VPN** 以及**虛擬主機**使用。

三、預設與建議的介面配置

1. 預設配置

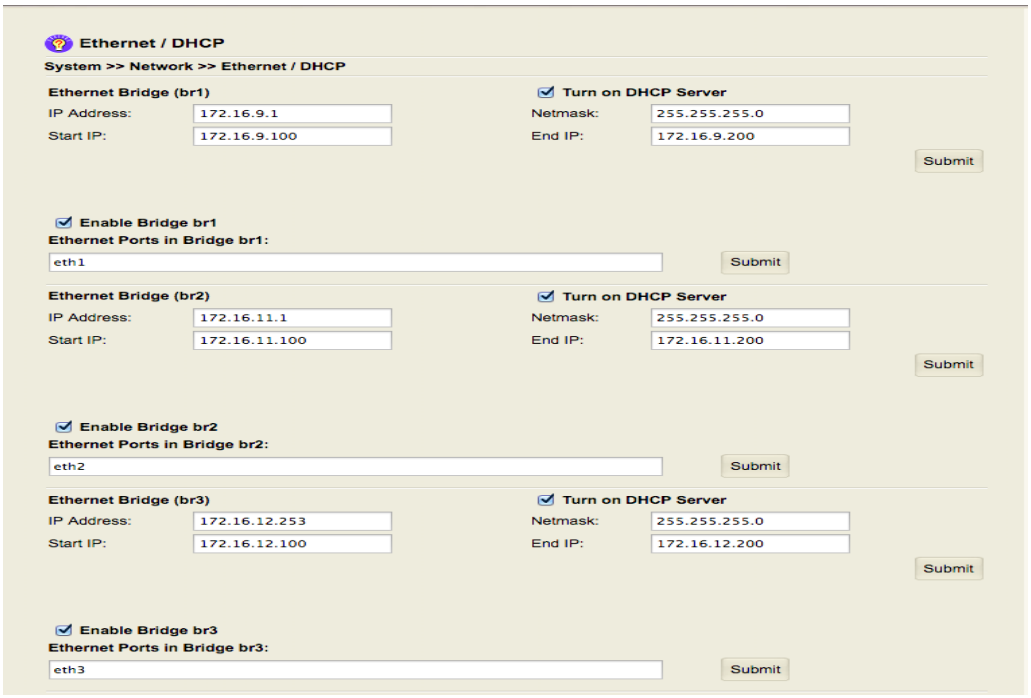
- **預設橋接：** 系統預設會將 **eth0** 加入 **br0**。
- **br0 用途：** br0 一般用作 **WAN（net 區域）**，通常承載對外的**公用 IP 位址**。**此設定通常不建議變更。**

2. 多實體介面管理建議

- **不建議橋接多個實體網口：** 雖然技術上可行，但通常**不建議**在同一個橋接器底下放入多個實體乙太網路介面。
- **實務建議：** 現今交換器硬體成本低廉，讓實體介面**各自作為「路由介面」**會比合併到同一個 bridge 更實際，除非有特定的網路設計需求。

四、相關設定與特殊情況

- **設定位置：** 相關的介面對應與橋接方式設定，可在 「**System >> Network >> Ethernet / DHCP**」 中調整。
- **DHCP 伺服器：** 每個橋接器上的 DHCP 伺服器可以單獨啟用或停用，其對應的位址池也會一併生效或停用。
- **橋接器數量限制：** 基礎平台預設提供 12 個橋接器，這與實際具備多少實體乙太網路介面**無關**。
    - **硬體超量建議：** 若您的硬體上**超過 12 個**實體乙太網路介面，建議將多出的介面**彙整到同一個橋接器之下**使用。



其他介面的對應與橋接方式，則可依實際環境，參考下列畫面進行調整。

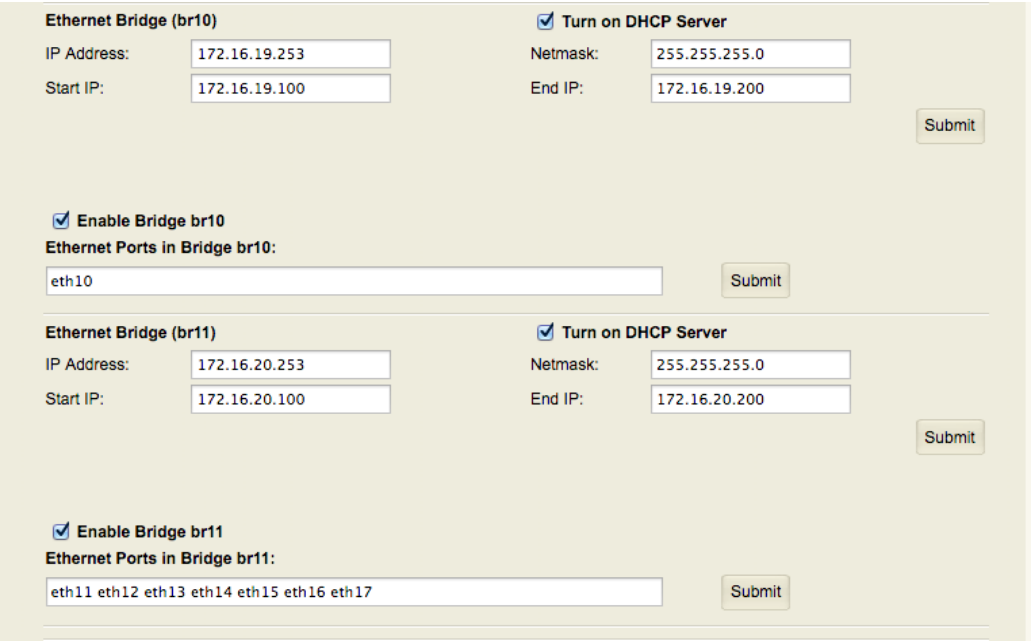


圖 74：將多個實體乙太網路介面加入同一個橋接器

「net」、「loc」與「dmz」三個區域的設定，將在後續章節說明。

Zone Definition (區域定義)

「區域 (Zone)」是本系統設計的核心，旨在**抽象化並簡化**整體防火牆與路由規則的管理，建立清晰的安全邊界。

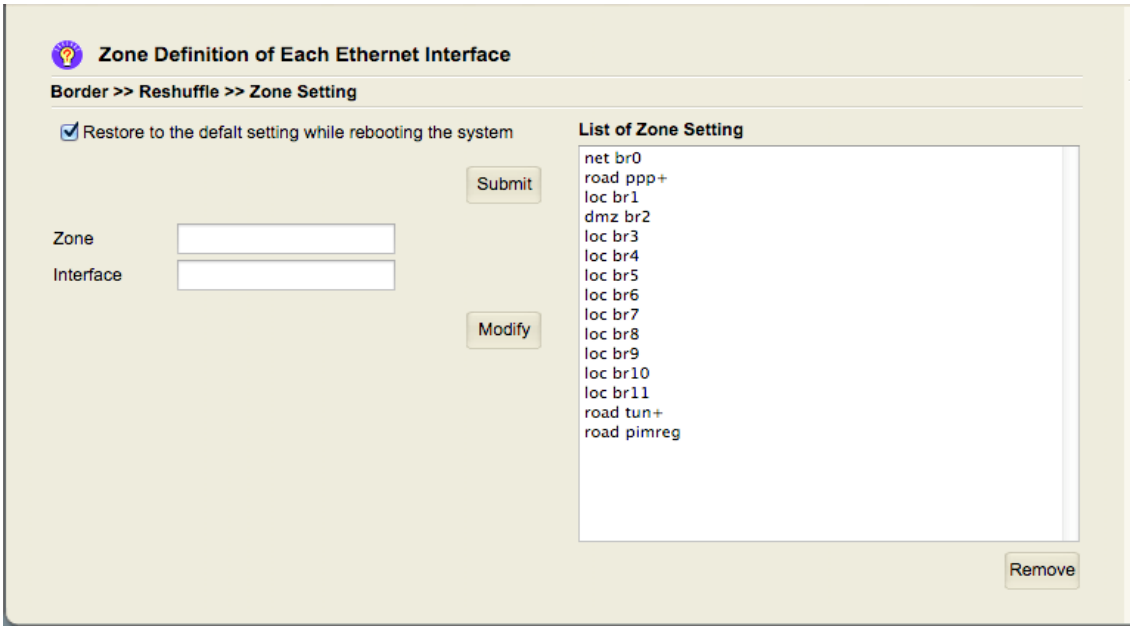


圖 75：區域設定快照

一、系統內建區域與特性

本系統將網路劃分為數個預設區域。其中，列在清單中的區域標籤為：**net**、**loc**、**dmz** 與 **road**。

區域標籤	用途	說明
net	**對外連線**	通常連接 WAN / Internet，代表不可信的外部網路。
loc	**內部可信網路**	用於內部信任的區域，如辦公室內網。
dmz	**隔離服務區**	用於對外提供服務但風險較高的主機群（非軍事區）。
road	**通道相關**	與 VPN 或其他內部通道連線相關的流量。
fw	**基礎平台本身**	**不會列在清單中**，代表基礎平台（防火牆）本身的網路介面。

二、區域設計的目的與優勢

- **簡化安全管理：** 管理者無需針對每一個 IP 位址或橋接器逐一設定安全政策。
- **核心機制：** 先將橋接器（例如 br0, br1 等）歸類到不同的區域中，再針對**區域之間**的關係訂定存取規則。

- **維護與擴充優勢：** 當實體網路或虛擬橋接器的配置變更時，只需**調整橋接器所屬的區域**，即可沿用既有的安全政策與預設行為，**不必重寫大量封包過濾規則**。
- **內建安全模型：** 每個區域之間已內建一組**預設的允許／禁止與 NAT 行為**，管理者在此基礎上再依需求新增例外規則。

總結： 這種以 Zone 為核心的設計，可以在維持清楚安全邊界的前提下，大幅降低日後維運與擴充時的複雜度與出錯風險。

⚠ **操作提示：設定儲存**

在進行區域定義或調整時，若畫面中上方的核取方塊仍然保持勾選狀態，系統在重新啟動後就會**恢復為預設設定**。

重要： 若您確定要採用目前的變更，請務必**先取消勾選該方塊**，確保設定生效。

Port Association for NAT Setting (NAT 設定的連接埠對應)

「Border >> Reshuffle >> Port Association」中的設定，決定了當封包自橋接器邊界送往外部網路時，是否需要在基礎平台上套用**網路位址轉換** (NAT)。

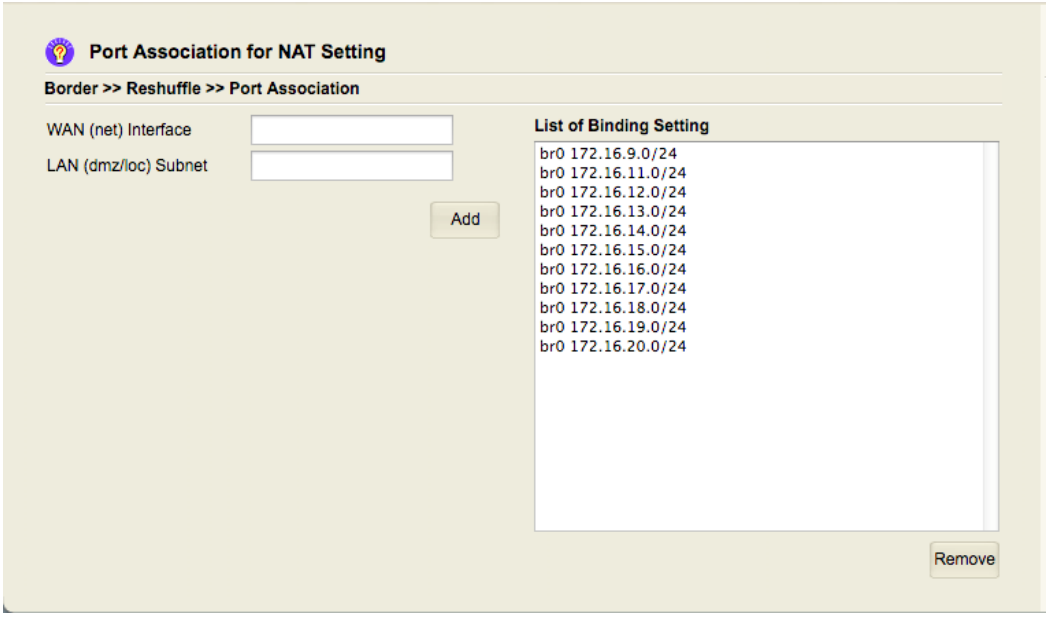


圖 76：NAT 設定

一、Port Association 的核心作用

- **決定 NAT 行為：** 管理者透過此設定，指定哪些實體埠／橋接器必須經過 NAT (通常用於對外上網的 WAN 介面)，以及哪些埠應保留原始來源位址。
- **目標：** 確保在同時兼顧**安全性**（隱藏內部 IP）與**正確的路由行為**（例如站點對站點 VPN 不需要 NAT）下的流量傳輸。

二、SNAT 實作範例與機制

以下範例展示了如何將多個內部子網的流量，透過 **SNAT (Source NAT)** 轉換為單一的外部 IP 位址：

內部橋接器	連接子網 (內部來源)	NAT 行為
br1	172.16.9.0/24	封包來源 IP 將被轉換為 br0 的 IP
br2	172.16.11.0/24	封包來源 IP 將被轉換為 br0 的 IP
br3	172.16.12.0/24	封包來源 IP 將被轉換為 br0 的 IP
br11	172.16.20.0/24	封包來源 IP 將被轉換為 br0 的 IP

運作原理：

只要封包的來源 IP 落在這些內部子網內（172.16.9.0/24、172.16.11.0/24 等），當它經由平台送出到外部世界時，其來源 IP 便會被轉換成「br0」的 IP 位址。

三、達成效果

透過這種集中的 NAT 設定，可以達成多重目的：

1. **節省公用 IP：** 多個內部私有 IP 共用一個對外公用 IP (即 br0 的 IP)。
2. **隱藏內部結構：** 外部世界只會看到單一的 br0 位址，**有效隱藏**了內部的子網劃分和私有 IP 結構，提升網路安全性。
3. **解決新增路由挑戰（VPN 整合）**：**情境：** 當需要在既有環境中新增一台 VPN 路由器時，如果應用主機（Application Host）未設定新的 VPN 網段路由路徑，VPN 客戶端將無法存取應用主機。**解決方案：** 可透過在新增的 VPN 路由器上，利用 Port Association 設定 NAT。**優勢：** 如此一來，VPN Client 發往應用主機的流量，其來源 IP 會被 NAT 轉換成 VPN 路由器（或基礎平台）的內部 IP，對於應用主機來說，請求像是來自內網的既有 IP，無需修改其 DHCP Default Gateway 或新增特定路由路徑，便能順利回覆封包。這是在擴充網路時常需考慮的簡化設定方式。

IP Policy Routing (IP 策略路由)

一般路由器路由表的設定元件

一個標準的路由器路由表通常由以下三個核心元件組成，這些元件共同決定了資料包（Packet）的轉發路徑：

1. **預設閘道 (Default Gateway) : 如圖 77 , Add Default Gateway On Non-Main Routing Table**
   - **定義：** 這是路由表中的**最後一條選擇**。當目標網路位址在路由表中找不到任何明確匹配的項目時，路由器會將資料包發送到此預設閘道。
   - **用途：** 通常指向下一跳路由器或網際網路服務供應商 (ISP)，用於處理**發往外部網路**的流量（即 0.0.0.0/0 路由）。
2. **介面 IP / 連接網路 (Interface IP / Connected Networks): 如圖 77, Add Interface into Non-Main Routing Table**
   - **定義：** 這些是路由器**直接連接的本地網路**。路由器會根據自身的**網路介面 IP 位址**和**子網路遮罩**自動生成這些路由條目。
   - **用途：** 確保路由器能夠直接將資料包發送到與其**直接相連**的網路中的設備，無需透過另一台路由器。
3. **路由條目 (Routing Entry / Static or Dynamic Routes): 如圖 77, Add Routing Entry on Non-Main Routing Table**
   - **定義：** 這是指向**遠端網路**的特定路徑。這些條目可以是**靜態路由**（由管理員手動配置）或**動態路由**（透過路由協定如 OSPF, BGP 等自動學習）。
   - **用途：** 讓路由器知道如何到達那些**不直接連接**的特定網路區段。條目通常包含**目標網路**、**子網路遮罩**和**下一跳地址**（Next-Hop Address）。

什麼是 IP 策略路由？

IP 策略路由（IP Policy Routing）是一種進階的網路功能，它打破了傳統網路流量只依賴單一路由表的限制，允許管理者為特定流量強制指定專屬的路徑。

核心機制與獨立路由表

路由的基本概念，是根據 IP 封包標頭中的來源與目的位址，查詢路由表，以決定封包應該從哪一個介面送出 。

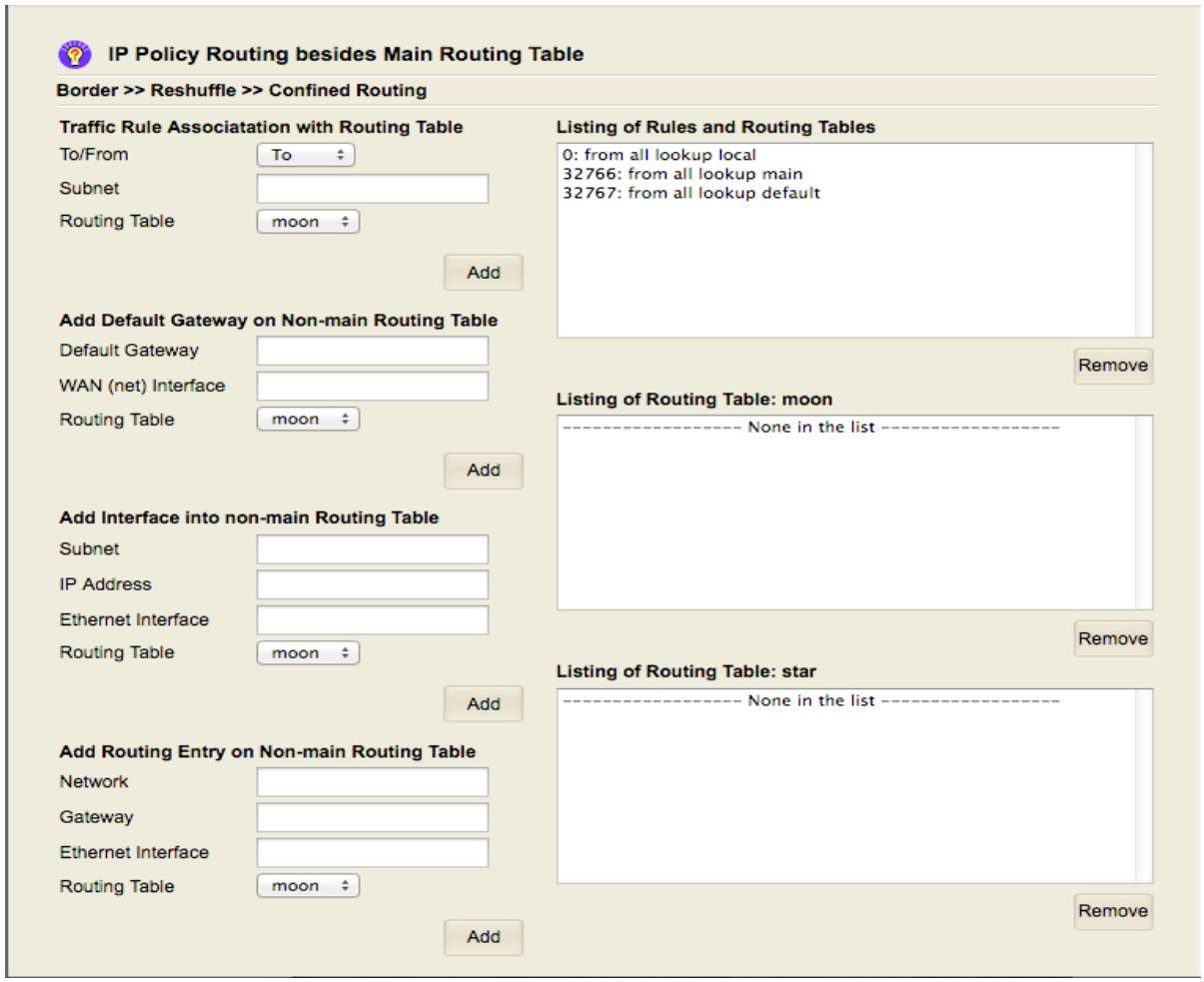


圖 77：IP 策略路由信息

- **主路由表 (Main Table)：** 在一般情況下，IP 流量都會依照「主路由表」進行轉送 。
- **獨立策略路由表：** 本基礎平臺除了主路由表之外，還額外提供兩個獨立的路由表，名稱分別為 **「moon」** 與 **「star」**。
- **策略動作：** 所謂 IP 策略路由，就是先挑選出**「符合某些條件的特定流量」**，再指定它們改用 moon 或 star 這兩張額外路由表來轉送，而不是一律走主路由表。

這些額外路由表是系統為了分流和管理進階網路行為而預先架設的基礎設施，提供隔離性和精準控制。如此一來，管理者可以在不更動主路由表一般行為的前提下，細緻地控制「哪些來源、往哪裡的流量，要走哪一條路」。

應用情境與實際價值

策略路由主要用於滿足複雜的網路需求，實現業務關鍵流量的保障、分流和強制導向。

情境	目的與價值	策略路由操作範例
多 WAN ／多 ISP 分流	讓一般上網流量和高優先級或備援流量走不同的對外線路，實現簡單分流或 QoS 需求。	設定主路由表走 br0／ISP-A。將「**某一個來源子網**」或「**某些特定服務**」（如 VoIP、VPN）的流量 指定套用 **moon** 表，並在 moon 表中設定預設匝道走 br5／ISP-B。
特定目標走 VPN	確保往總部或雲端 VPC 的流量必須經由安全的 VPN 通道，避免繞錯路或被主路由表導向公網。	在 moon 路由表中，定義到總部子網 10.20.0.0/16 應該走 VPN 介面。再用策略路由指定「來自某些內部子網」或「某幾台伺服器」的流量，到 10.20.0.0/16 時，一律套用 moon 表。

配置步驟與管理指南

策略路由的設定可以透過「**Border >> Reshuffle >> Confined Routing**」完成。

第一步：建立流量規則與路由表關聯 (定義策略)

此步驟定義哪些網路流要套用哪張路由表。

- **管理介面區塊： Traffic Rule Association with Routing Table**
- **操作：** 依據封包的 **來源或目的子網（From / To Subnet）** 建立規則，並指定要套用的路由表（moon 或 star）。
- **清單顯示：** 畫面右側最上方的清單，用來列出「哪些來自／前往特定子網的流量」要套用哪一個路由表（local, default, main、moon 或 star）。

第二步：配置策略路由表條目 (設定路徑)

此步驟旨在為 **moon 和 star 路由表 本身加入實際的轉送路徑**。

- **管理介面區塊：** Add Default Gateway on Non-main Routing Table 與 Add Interface into Non-main Routing Table
- **操作：** 為 **moon / star** 個別設定：
  - 出口介面與預設匝道（例如為 第二條 WAN 或 VPN 建置網流的介面）。
  - 該路由表需要辨識的內部子網與對應介面。
- **清單顯示：** 畫面右側中間與下方兩個清單，則分別顯示 moon 與 star 這兩張路由表的實際路由條目。

額外資訊：主路由表檢視

系統預設的 **main** 路由表內容，可於「**System >> Network >> Static Routing**」中查閱與設定。

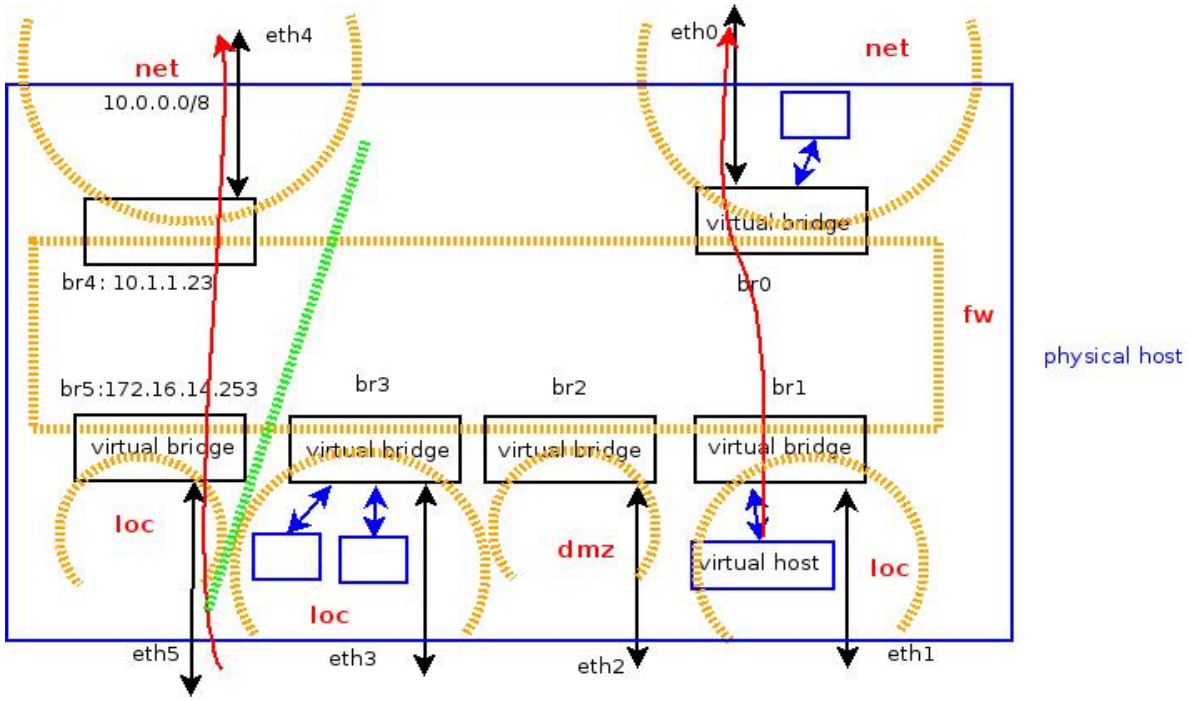


圖 78：兩個 WAN 埠的例子

策略路由應用案例：安全存取 ISP 私有網路

本案例旨在說明如何利用 IP 策略路由與網路位址轉換（NAT），讓特定內部子網的設備能夠安全且透明地存取外部 ISP 所提供的私有網路資源。

網路環境概述

基礎平台位於「net」區域，網路配置如下：

介面	網路類型	網段/用途	基礎平台 IP
br0	WAN	連接至公開 Internet	(未提供，預設出口)
br4	私有 WAN	連接至 ISP 私有子網 10.0.0.0/8	待配置為 10.1.1.23/8
br5	內部 LAN	連接至內部子網 172.16.14.0/24	172.16.14.253/24

ISP 存取 10.0.0.0/8 網段的連接資訊為：

- **預設閘道：** 10.1.1.1
- **IP 位址：** 10.1.1.23
- **子網遮罩：** 255.0.0.0

我們的目標

我們的目標是讓位於內部子網 172.16.14.0/24 中的設備，能夠透過基礎平台**順利連線並存取** ISP 的私有網路 10.0.0.0/8 資源。

面臨的技術挑戰：來源 IP 轉送問題

由於 172.16.14.0/24 是一個我們內部使用的私有網段，ISP 不會幫助轉送（路由）「**來源 IP 為 172.16.14.x**」的封包。這將導致流量無法正確返回，通訊失敗。

解決方案：IP 策略路由與 NAT 結合

為了解決這個源位址不匹配的問題，我們必須強制來自 172.16.14.0/24 且目標為 10.0.0.0/8 的流量，採取以下步驟：

1. **實施 NAT (位址轉換)：**
    - 流量到達基礎平台時，必須在發送前進行 **網路位址轉換（Source NAT/SNAT）**。
    - 將封包的來源位址從 172.16.14.x 轉換成 **br4 介面上的 IP 位址 10.1.1.23**。
2. **實施 IP 策略路由 (指定路徑)：**
    - 由於 10.0.0.0/8 網段的流量必須通過 10.1.1.1 這個特定閘道，我們不能使用主路由表。
    - 必須使用 IP 策略路由，將這類流量 **強制指定到額外路由表（例如 moon）**。
    - 在 **moon** 路由表中，配置目標 10.0.0.0/8 的路由條目，並指定出口走 10.1.1.1。

結果：

透過將策略路由與 NAT 結合配置，我們成功地維持了內部網段 172.16.14.0/24 的獨立性，同時讓該子網內的設備能夠以 10.1.1.23 的身份安全且有效地存取 ISP 所提供的私有網路資源。

配置實務：br4 介面的網路區域劃分與 IP 配置

本節將說明如何將 br4 介面納入 **net** 區域，並配置其 IP 位址，以滿足與 ISP 私有網路的安全隔離與連接需求。

問題：為什麼要將 br4 放入 net 區域？

儘管 br4 連接至 ISP 的私有子網，但該子網上可能存在其他客戶。基於安全考量，我們的目標是：

- **流量阻擋：** 默認情況下，阻止來自該子網（即 br4）的外部流量。
- **介面隔離：** 確保 br4 和 br5 子網與系統的其餘部分隔離，以維持系統其他部分的正常運行。

將 br4 劃歸至 **net** 區域（通常用於對外、不可信的網路）能更好地實現這種預設的隔離和限制。

設定步驟

要完成這些設置，您需導航存取以下設定頁面：
1.「**System >> Network >> Ethernet / DHCP**」
2.「**Border >> Reshuffle >> Zone Setting**」
3.「**Border >> Reshuffle >> Port Association**」
4.「**Border >> Reshuffle >> Confined Routing**」

請依照以下步驟配置 br4 介面的 IP 位址與網路區域：

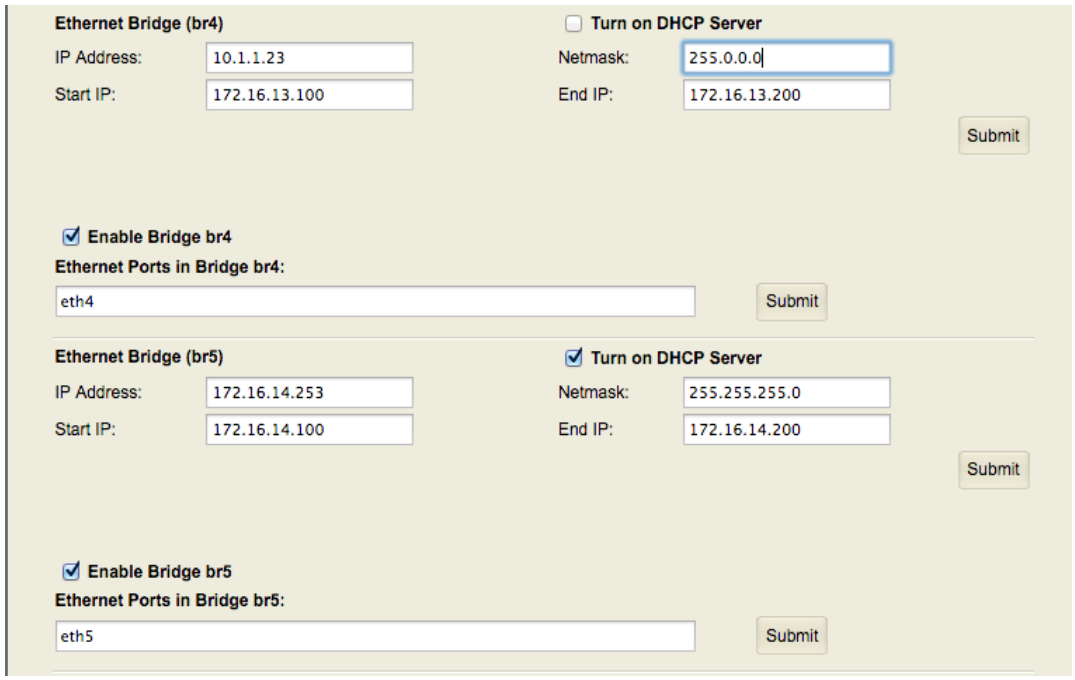


圖 79：更改橋的 IP 位址

步驟一：配置 br4 的 IP 位址
首先，我們需要將 br4 介面的 IP 位址變更為 ISP 要求的配置。

1. 導航至：「**System >> Network >> Ethernet / DHCP**」。
2. 找到 br4 介面，並將其 IP 位址設定為 **10.1.1.23**，子網遮罩設定為 **255.0.0.0**。
3. **注意：** 此設定生效後需要**重啟系統**。

步驟二：修改 br4 的區域定義
接著，將 br4 從預設的 **loc** 區域移除，並添加到 **net** 區域。

1. 導航至：「**Border >> Reshuffle >> Zone Setting**」。
2. 修改 br4 的區域定義：
   - 從「**loc**」區域中移除 br4。
   - 將 br4 添加到「**net**」區域。
3. **防止重啟還原：** 為了避免在重啟後設定返回默認值，請**取消勾選頂部的複選框**。

完成以上設定後，br4 將具備正確的 IP 配置，並被納入 **net** 區域，滿足預設安全隔離的需求。

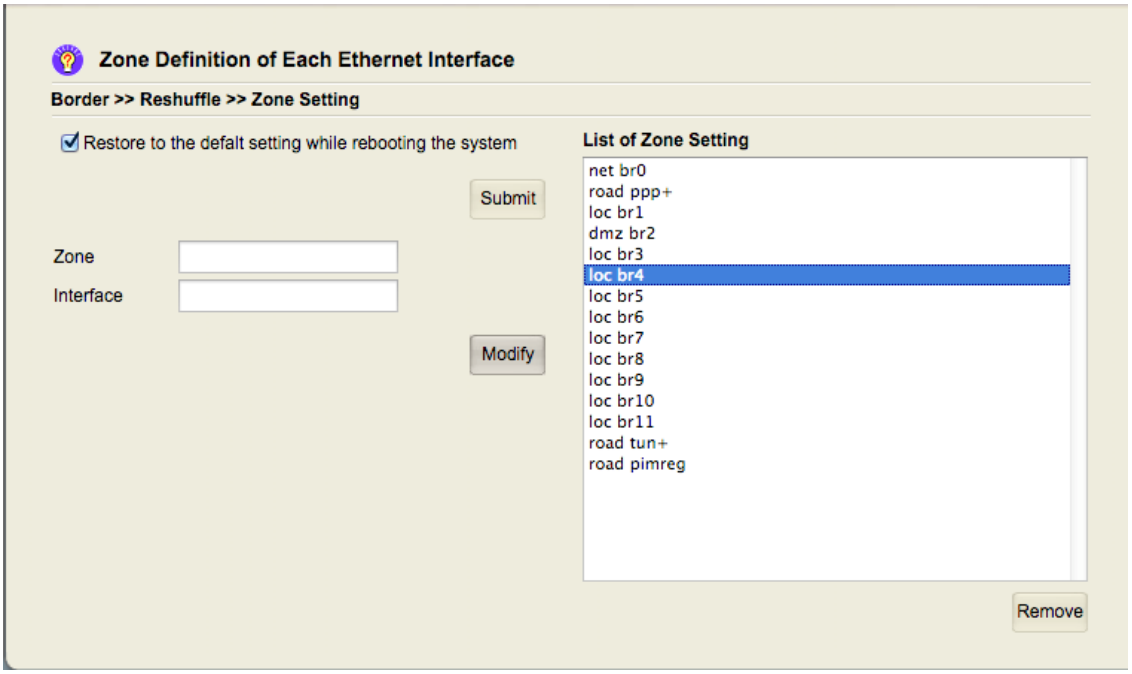


圖 80: 從“loc”區域移除“br4”

下面是從「loc」區域移除「br4」後的的截圖 ：

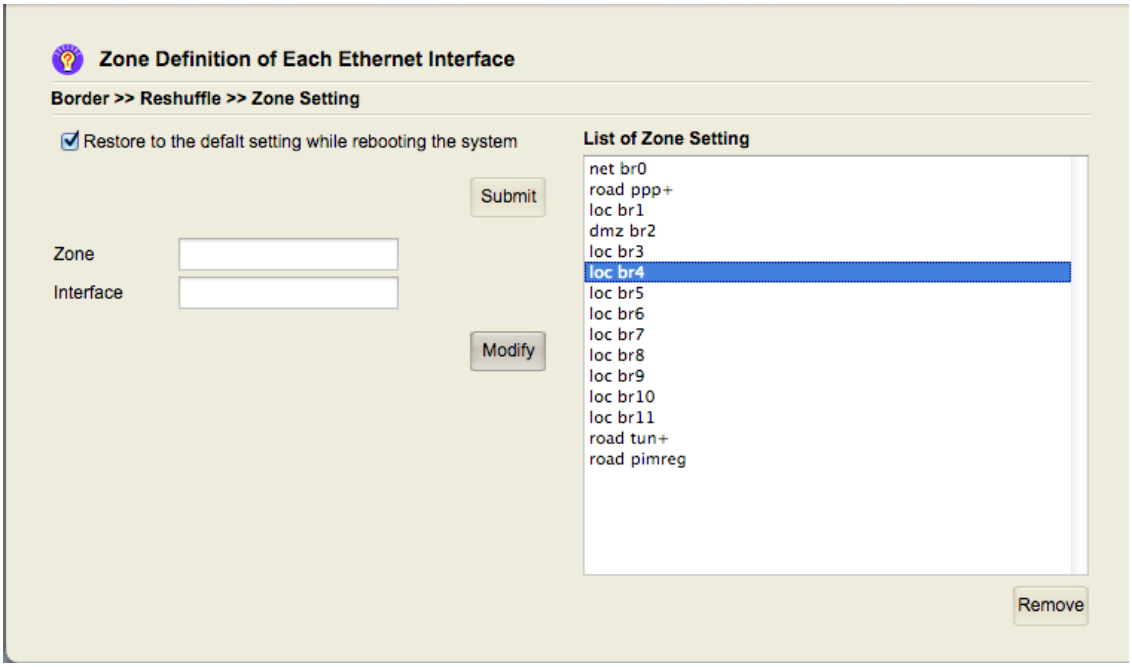


圖 81：從「loc」區域移除「br4」後的列表

以下屏幕截圖用於將“br4”添加到“net”區域：

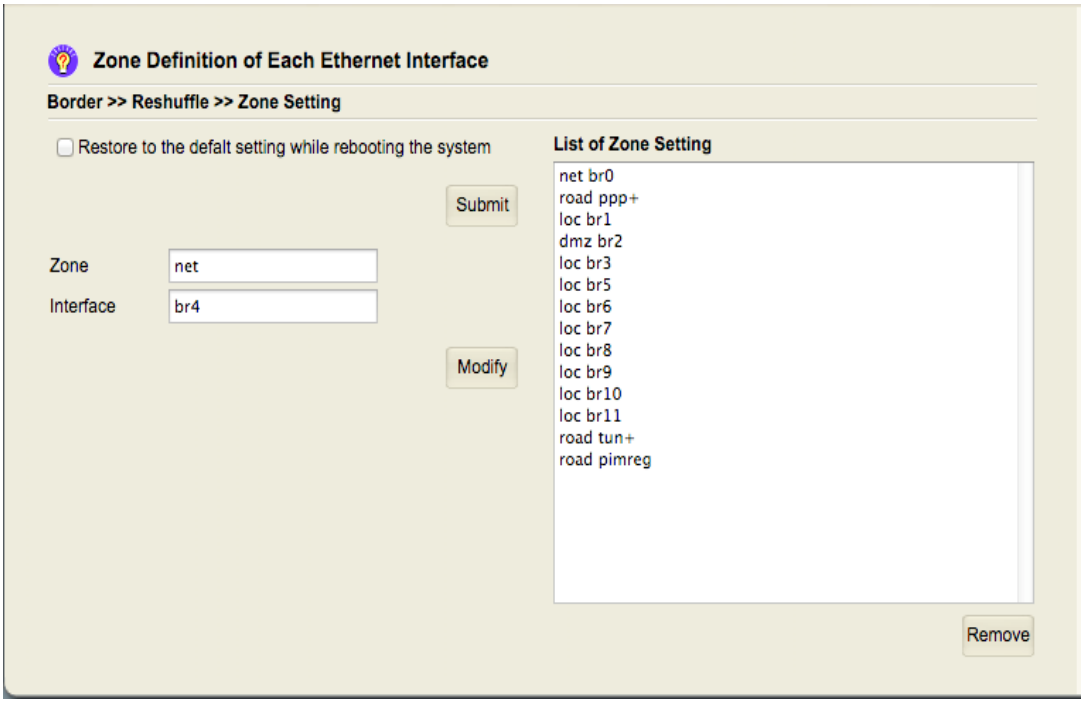


圖 82：將“br4”添加到“net”區域。

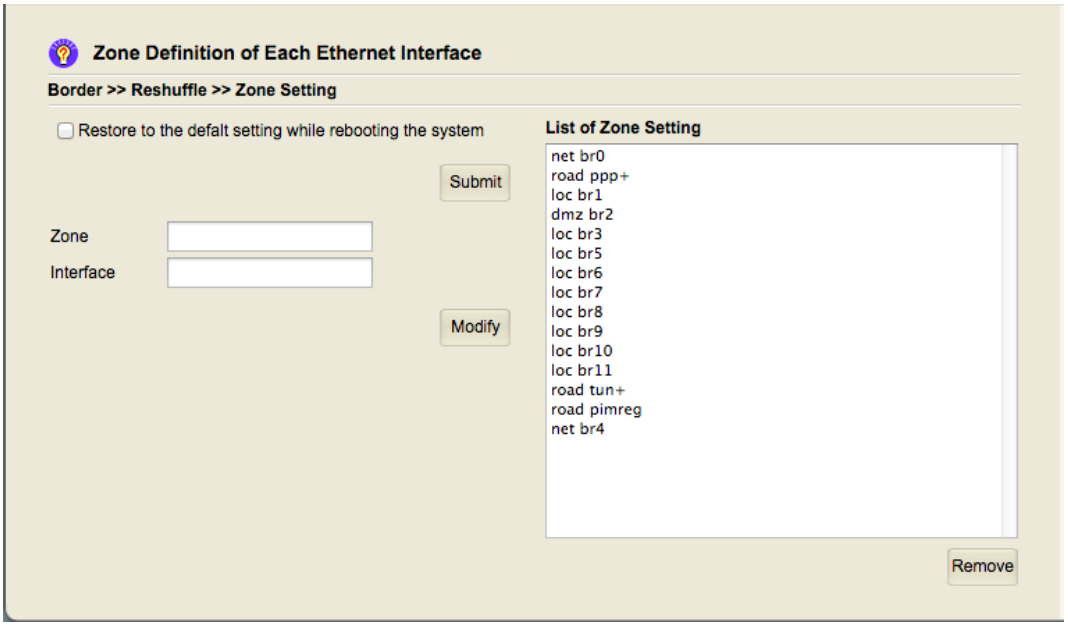


圖 83: 列表“br4”在區域“net”中

配置實務：步驟三 – 實施 NAT 轉換（位址偽裝）

完成 IP 位址和區域的配置後，接下來的關鍵步驟是設置 **網路位址轉換（NAT）** 規則。此設定是為了確保來自 172.16.14.0/24 內部子網的流量，在進入 ISP 的私有網路 10.0.0.0/8 之前，能將來源 IP 位址偽裝成 br4 介面的合法 IP。

為什麼需要 NAT？

如同案例分析所述，ISP 不會轉送（路由）來源為內部私有網段 $172.16.14.x 的封包。因此，所有從 br5（源）發往 br4（目的）的流量，必須執行 NAT 轉換。

設定步驟

步驟一：移除預設設置

為了避免配置衝突，首先需移除 br4 和 br5 介面可能存在的預設或原始設置。

1. 導航至：「**Border >> Reshuffle >> Port Association**」。
2. 移除清單中所有與 **br4** 和 **br5** 相關的原始設置。

步驟二：配置 NAT 轉換規則

接著，設定規則以允許從 br5 到 br4 的流量進行源位址轉換。

1. 在「**Port Association**」介面中，新增一條規則：
    - **源介面 (Source Port)：** 選擇 br5。
    - **目的介面 (Destination Port)：** 選擇 br4。

- **動作 (Action)：** 設置為 **NAT 轉換（或位址偽裝）**(即：源 IP 地址將被替換為 br4 介面的 IP 地址 $10.1.1.23)。

結果：

透過此配置，任何來自 $172.16.14.0/24 且目標是 10.0.0.0/8 的封包，在離開 br4 介面時，其源 IP 位址將被轉換成 10.1.1.23。

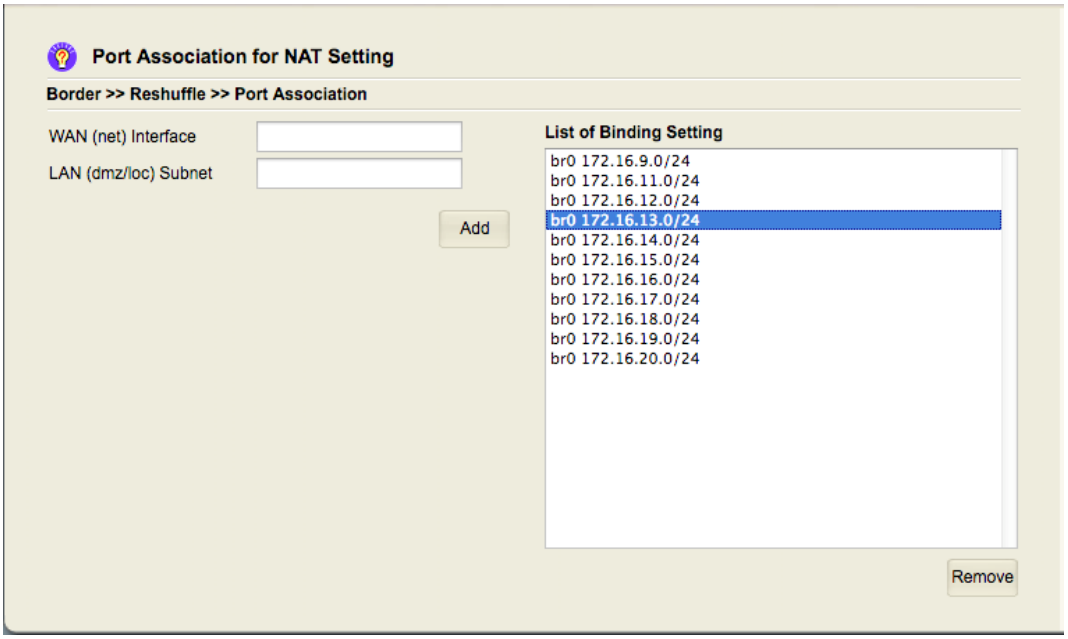


插圖 84: 移除“br4”的原本子網路，以便在“br0”下使用 NAT。

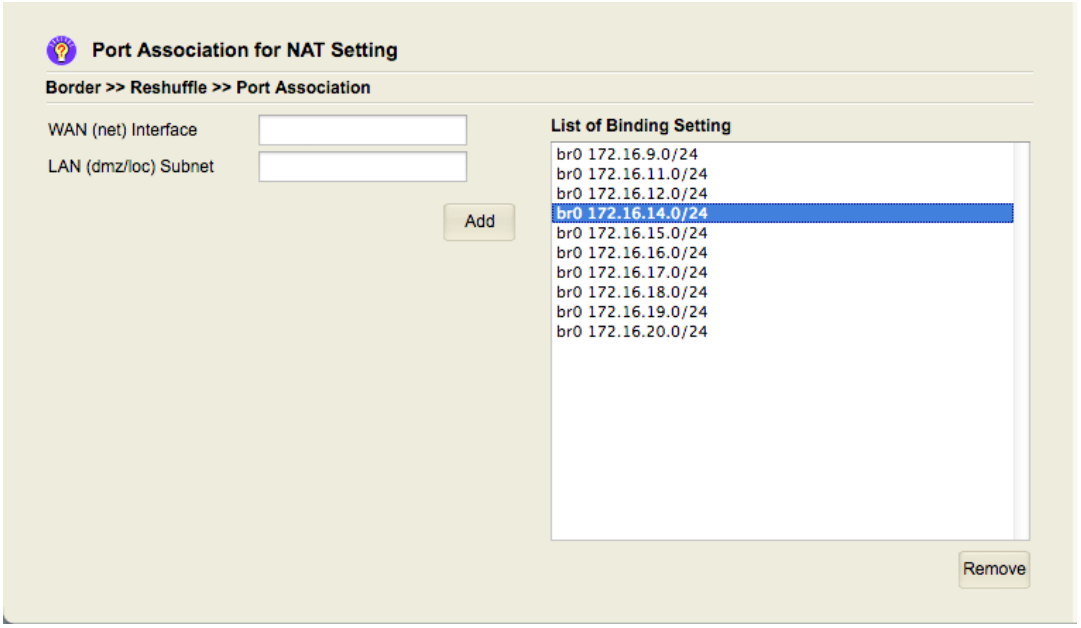


圖 85：使用“br0”下的 NAT 刪除子網“br4”

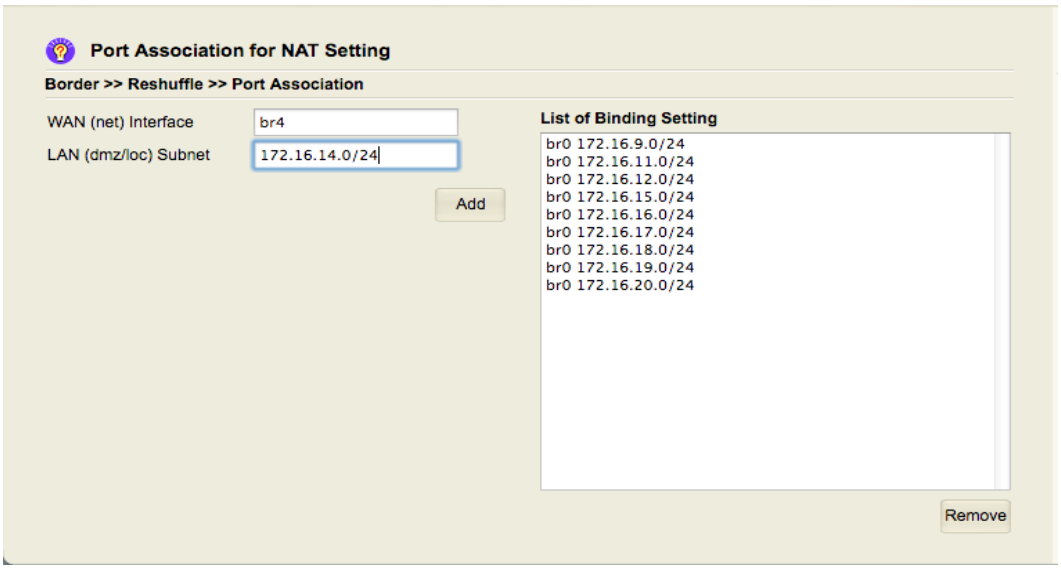


插圖 86：使用 NAT 下，新增子網路 "br5" 位於 "br4" 下。

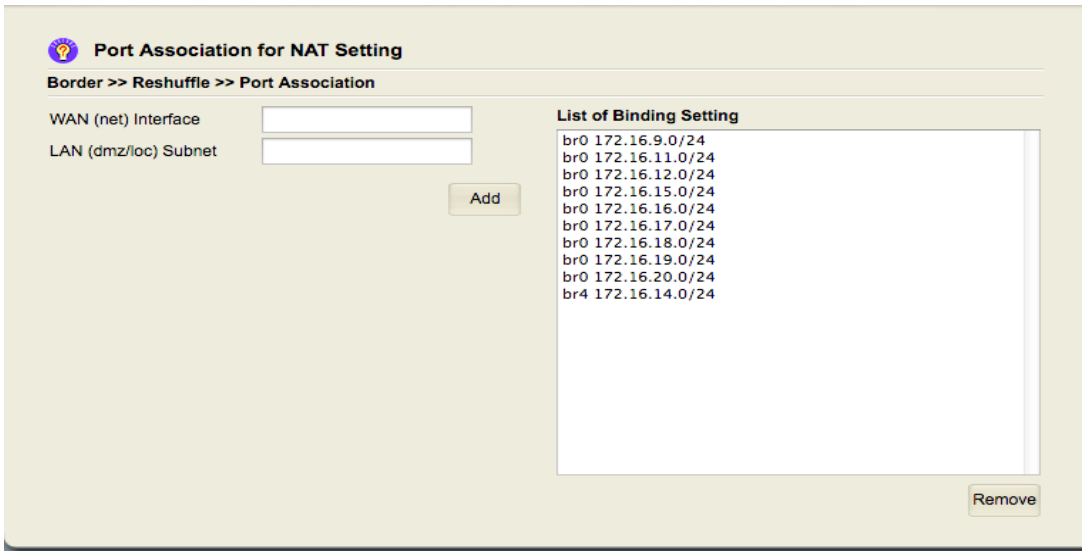


圖 87：使用 NAT 的“br5”子網列表（在“br5”中）

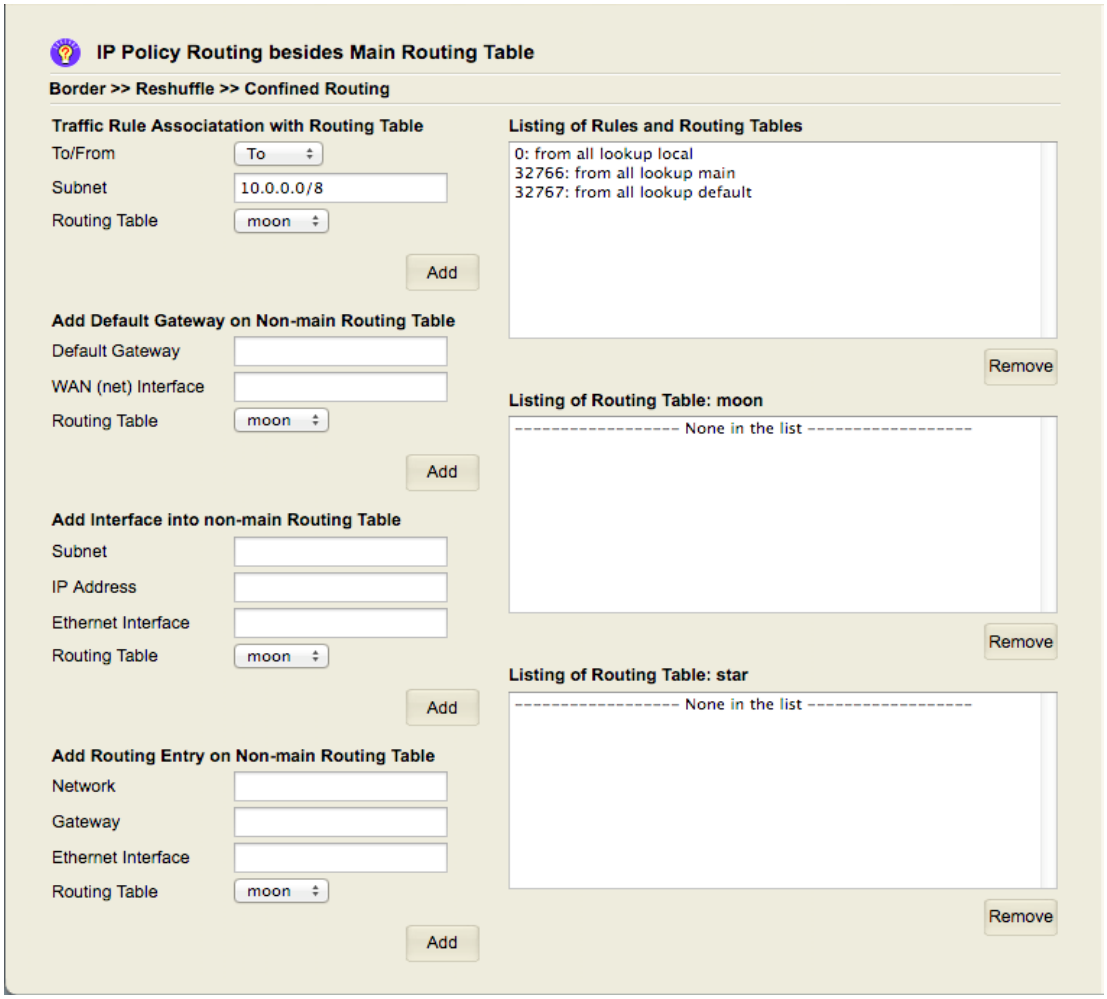


圖 88：捕獲指向子網路 "10.0.0.0/8" 的流量

配置實務：步驟四 – 建立策略路由規則（強制導向 moon 表）

在完成 IP、區域和 NAT 的基礎配置後，我們現在將專注於 **IP 策略路由** 的核心：建立規則來決定特定流量的路徑。

策略規則的執行機制

在「**Border >> Reshuffle >> Confined Routing**」介面中，最上方的清單（Traffic Rule Association with Routing Table）顯示了策略路由規則，其執行機制如下：

1. **逐層檢查：** 對於通過基礎平台傳輸的每一個網路封包，系統會從上到下逐條檢查策略規則。
2. **規則匹配：** 如果一條規則沒有匹配，則會檢查下一條規則。
3. **預設行為：** 清單最底部的規則通常涵蓋所有未匹配的流量，並將其導向「**main**」或「**default**」路由表。

設定目標：捕獲特定流量導向 moon 表

我們的目標很明確：**捕獲 br5 和 br4 子網之間的流量，並強制這些流量使用「moon」路由表**來決定最終的去向。

配置步驟：新增策略規則

1. 導航至：「**Border >> Reshuffle >> Confined Routing**」。
2. 在 **Traffic Rule Association with Routing Table** 區塊，新增一條規則：
    - **流量條件：** 設定規則以涵蓋 br5 子網（172.16.14.0/24）和 br4 子網（10.0.0.0/8）之間的流量[2]。
    - **目標路由表：** 指定該流量應套用「**moon**」路由表 。

重要性： 這條規則必須放置在清單中較高的位置（在預設的 main 規則之上），以確保來自 172.16.14.0/24 且目標為 10.0.0.0/8 的封包能被優先捕獲並導向 **moon** 表。

以下屏幕截圖表明，我們希望捕獲“10.0.0.0/8”子網的流量，並强制它們查找路由表“moon”。

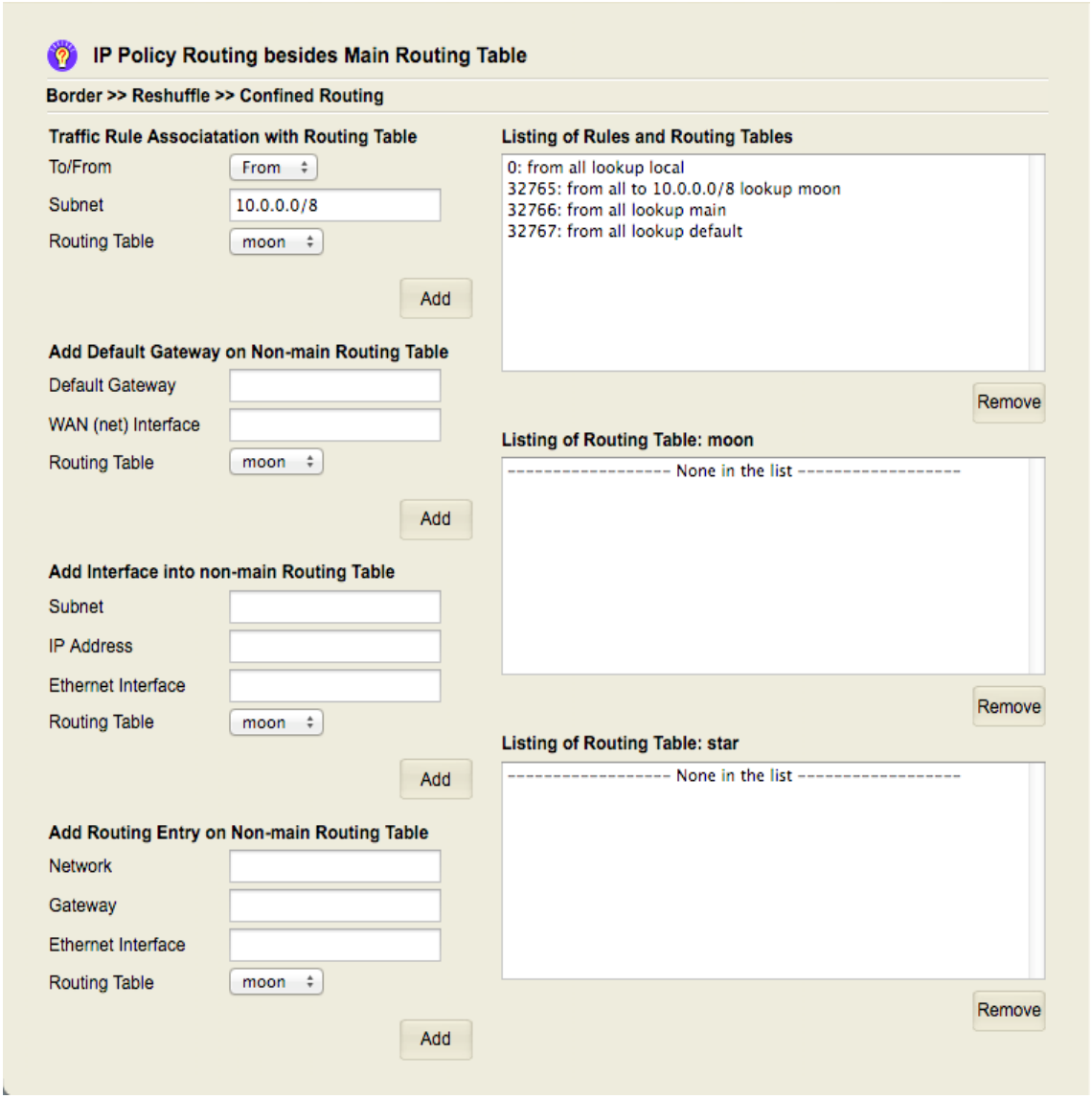


圖 89：捕獲來自子網“10.0.0.0/8”的流量

配置實務：步驟四優化 – 策略路由規則的精簡原則

延續上一步驟，我們將精進策略路由規則的設定方式，以提高效率和彈性。

策略規則優化原則：集中捕獲來源子網

我們的目標是讓來自 br5 子網（172.16.14.0/24）的流量能夠進入 ISP 的 10.0.0.0/8 網路。

1. **問題分析：** br4 連接到 10.0.0.0/8 子網。然而，在這個 ISP 的 10.0.0.0/8 私有網路後面，**有可能存在大量可通過 ISP 閘道訪問的其他遠程子網**。
2. **效率考量：** 如果我們在策略規則中一一列出所有可能的目的子網，將會耗費大量的配置資源。相比之下，**源子網的數量相對較少**。

3. **優化原則：** 更好的方法是**集中捕獲「源」子網**，而不是列出所有目的子網。

設定步驟：配置精簡規則與路由表分層

根據上述優化原則，我們在 **「Border >> Reshuffle >> Confined Routing」** 中配置規則時，應專注於控制來源：

1. **捕獲源子網：**
   - 在 **Traffic Rule Association with Routing Table** 中，我們應設定規則，精確捕獲來自 br5 子網（即 172.16.14.0/24）的流量。
   - 即使目的子網是 10.0.0.0/8，我們仍應使用 **moon** 路由表來處理。
2. **目的子網處理：**
   - **鄰近子網（To Subnet）：** 在策略規則的 **「To」** 子網欄位，我們只需指定 **相鄰子網**（例如 10.0.0.0/8）即可。
   - **遠程子網（Remote Subnet）：** 對於任何位於 10.0.0.0/8 之後的 **遠程子網**，我們則讓 **moon 路由表中的預設閘道** 來處理它們。

簡而言之，策略規則（第一步）負責將 172.16.14.0/24 的流量導向 **moon** 表；而 **moon** 路由表（第二步）則負責透過其預設閘道 10.1.1.1 處理所有 10.0.0.0/8 以外的目的地。

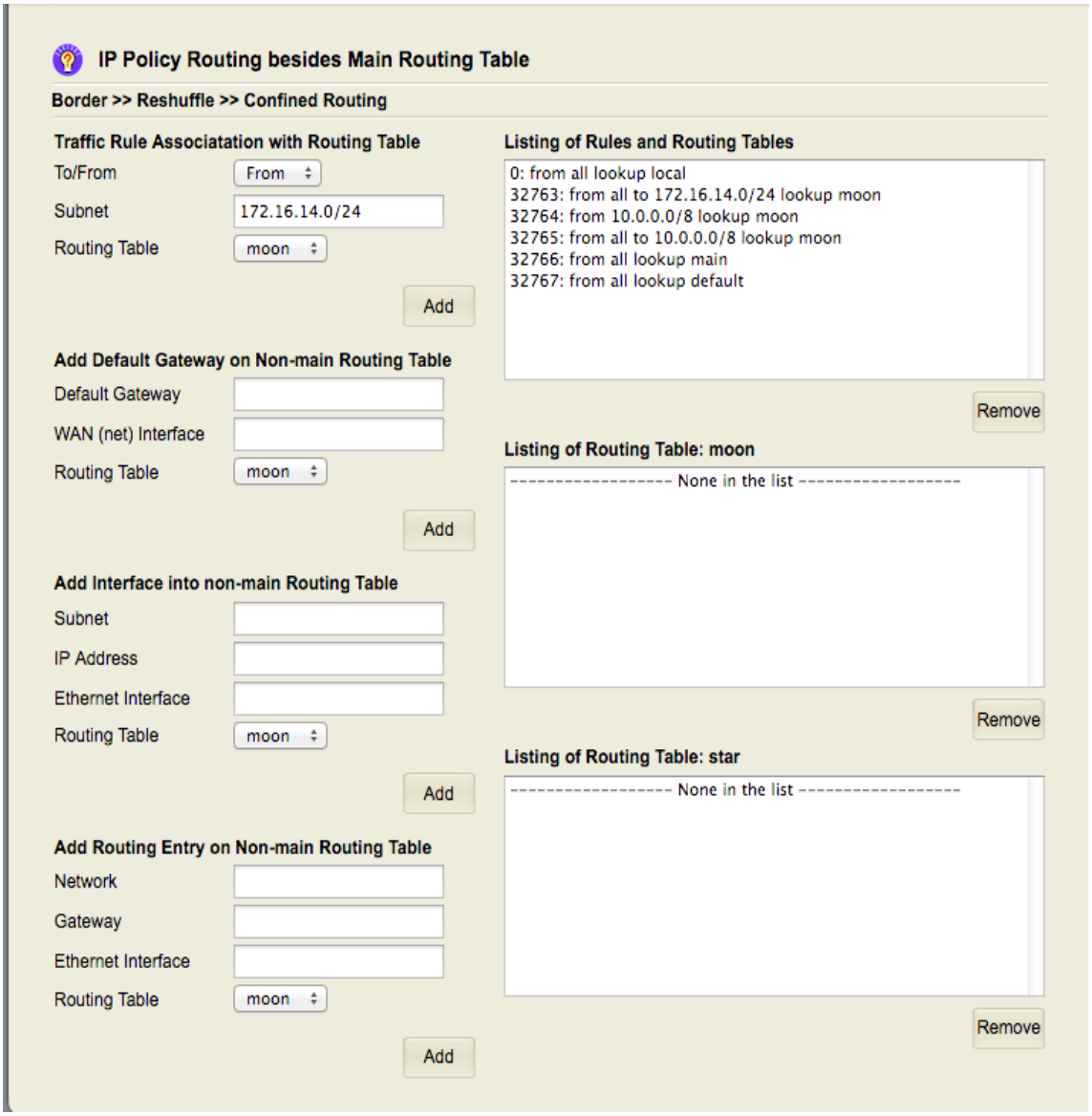


圖 90：抓取來自 “172.16.14.0/24” 的流量

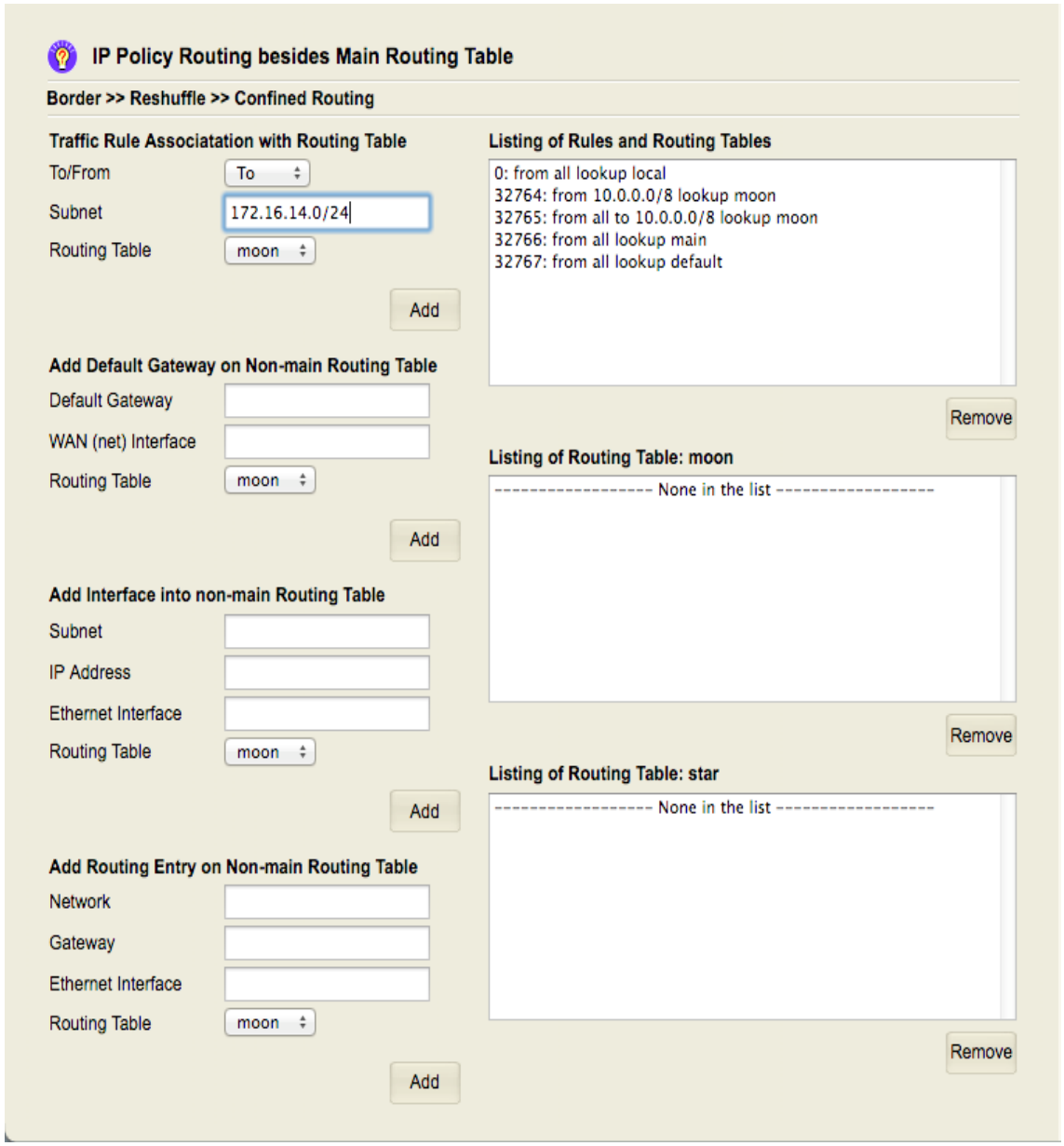


圖 91：捕獲前往“172.16.14.0/24”的流量

請記憶體住，“172.16.14.0/24”是“br5”連接到的子網。

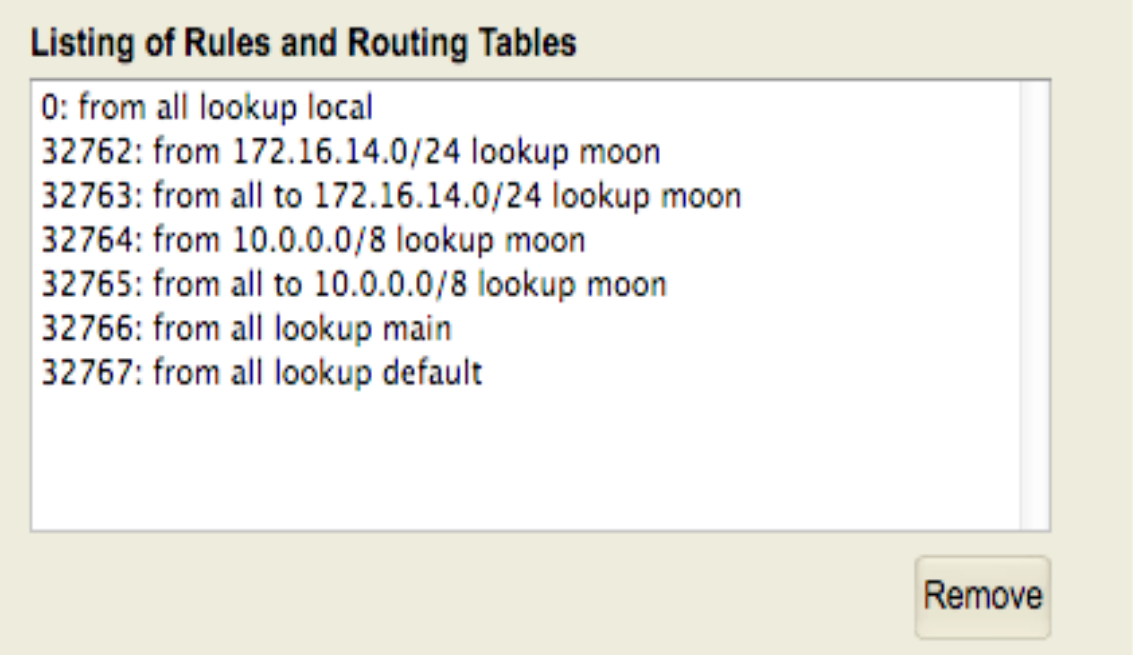


圖 92：查找路由表規則列表

配置實務：步驟五 – 建立 Moon 路由表條目（設定路徑）

在策略規則已經定義完成後，我們必須為「**moon**」路由表手動添加具體的轉送條目，確保被捕獲的特定流量能夠正確地從 ISP 的閘道送出。

設置目標與路由表內容情景

我們已透過策略規則，要求 172.16.14.0/24 和 10.0.0.0/8 子網之間的流量查閱「**moon**」路由表。 現在的目標是手動創建「**moon**」路由表的內容，使其具備類似於直接連接的接口路由，並包含一個預設出口。

一個完整的路由表主要包含以下幾類條目：

1. **直接連接子網路由：** 這些是與本機以太網接口（如 br4 和 br5）直接連接的子網路由。這些條目在設定接口 IP 位址時會**自動**在「**main**」路由表中創建 ，但在「**moon**」等其他策略路由表中，我們需要**手動添加**它們 。
2. **預設閘道 (Default Gateway)：** 如果没有任何路由條目匹配數據包的目的地，則數據包將被發送到預設閘道 。
3. **其他本地子網閘道：** 可能會有一些機會為其他本地子網添加一些閘道 。
4. **配置步驟：將條目新增至 Moon 路由表**

步驟一：添加預設閘道 (Default Gateway)

我們將 ISP 提供的 10.1.1.1 作為 **moon** 路由表的預設出口。

1. 導航至：「**Border >> Reshuffle >> Confined Routing**」介面 。
2. 操作區塊：**Add Default Gateway on Non-main Routing Table** 。
3. 輸入 ISP 提供的預設閘道 **10.1.1.1**。
4. 按下「**Add**」按鈕。對應的路由條目將在右側的「**moon**」路由表中顯示 。

步驟二：添加直接連接子網路由

為了讓 **moon** 表能正確識別並轉發來自 br5 的流量，我們需要將這些直接連接的路由手動加入 **moon** 表。

1. 操作區塊：**Add Interface into Non-main Routing Table** [8]。
2. **操作：** 類似於設置以太網接口的 IP 地址和子網掩碼，為 br4 (10.0.0.0/8) 和 br5 (172.16.14.0/24) **手動** 添加對應的路由條目 [9]。

關鍵配置檢查與系統限制

故障排除：預設閘道未顯示

- **現象：** 如果添加預設閘道後，「**moon**」路由表中没有顯示任何內容 [10]。
- **潛在原因：** 設置可能不正確 。最可能的原因是 **br4 的 IP 地址未生效**。
- **解決方案：** 必須返回檢查 br4 的 IP 地址是否設置正確，並**重新啟動系統使其生效**。
- **明顯錯誤：** 必須確保閘道 IP（10.1.1.1）和接口 IP（10.1.1.23$）在同一子網中，否則會是一個明顯的錯誤 。

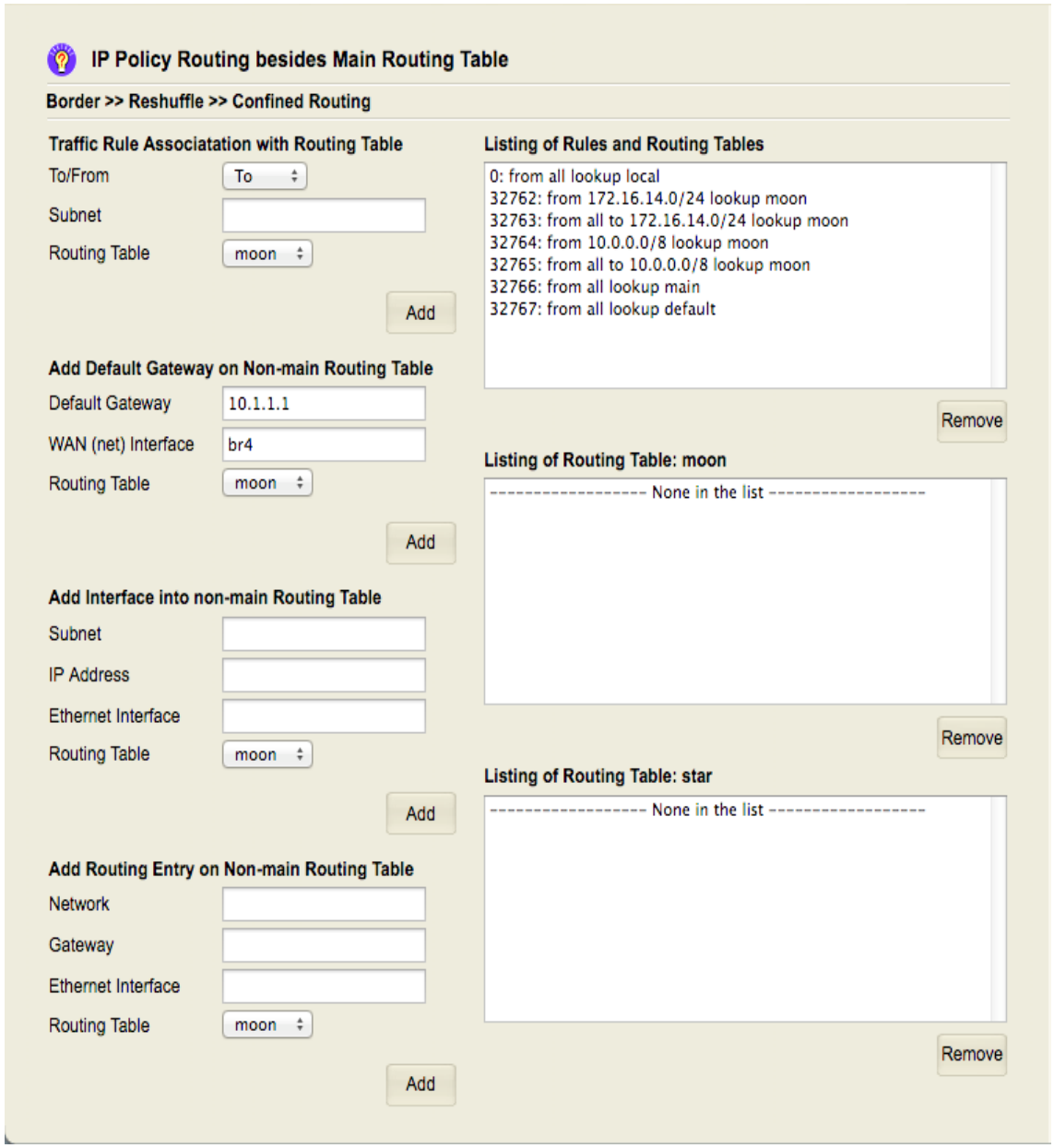


圖 93：在“moon”路由表添加默認閘道

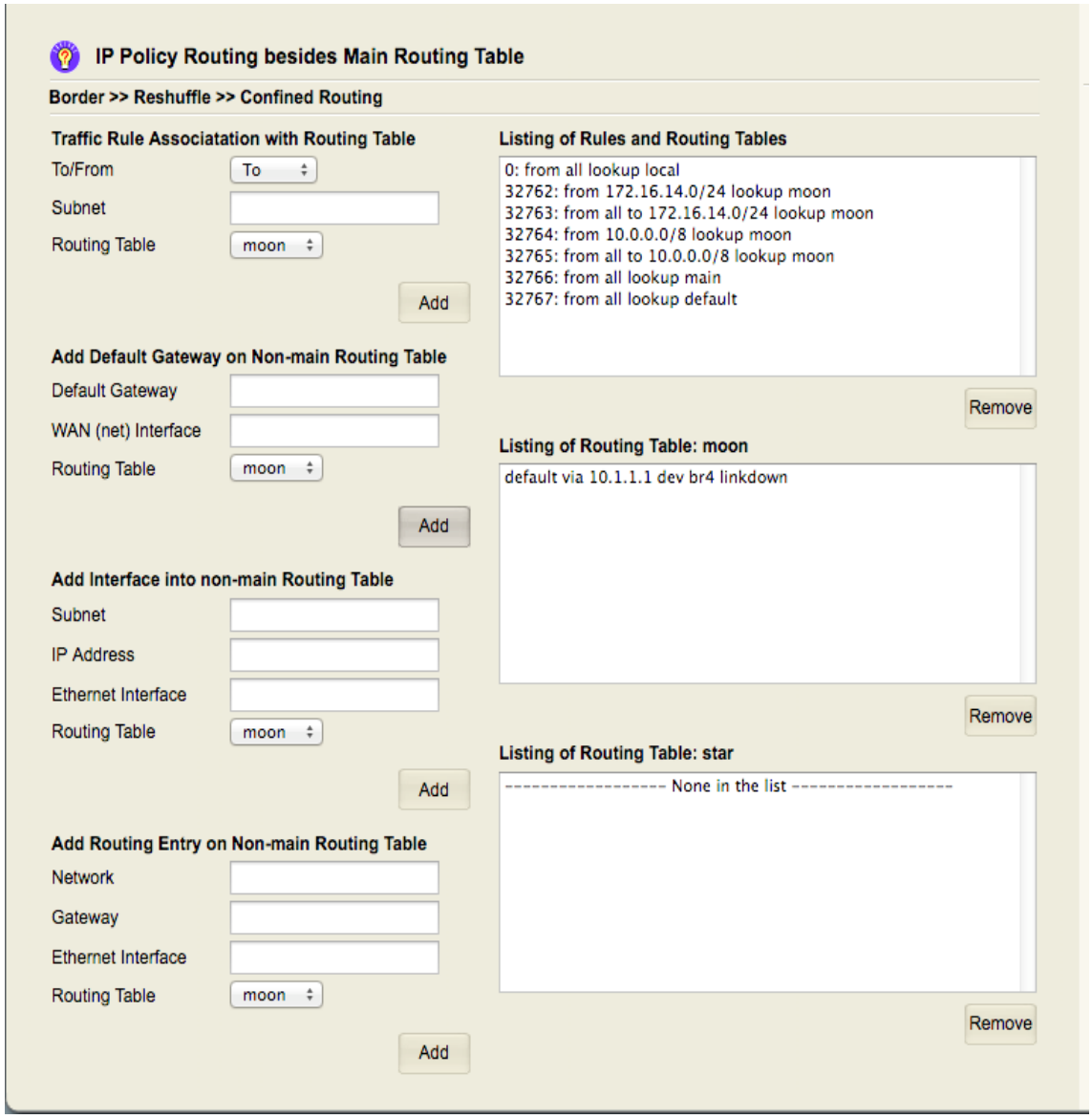


圖 94："moon" 路由表中默認閘道

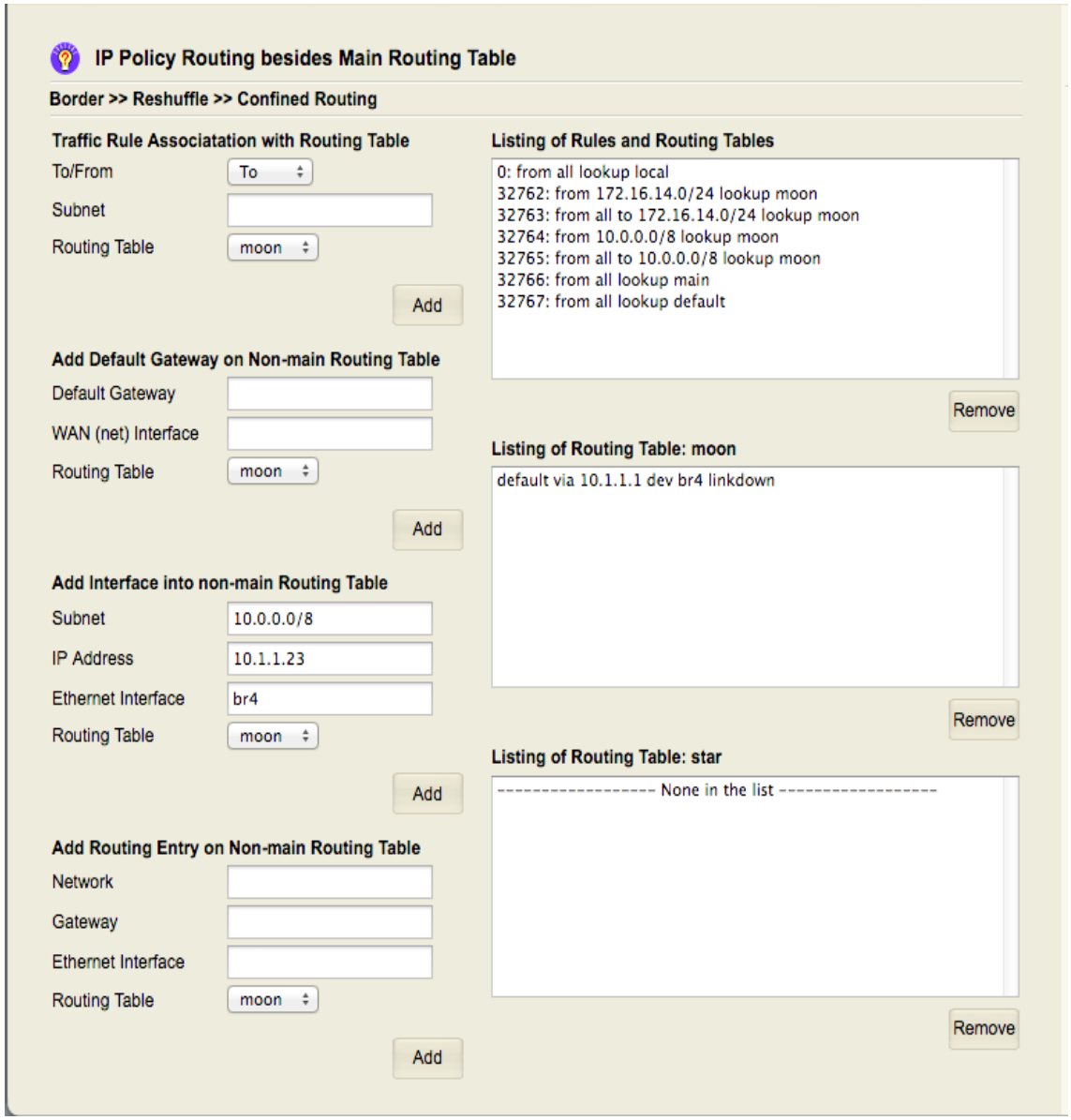


圖 95：為連接到“br4”的“moon”子網添加路由條目。

以下屏幕快照用於添加針對“br4”直接連接的子網的路由條目。

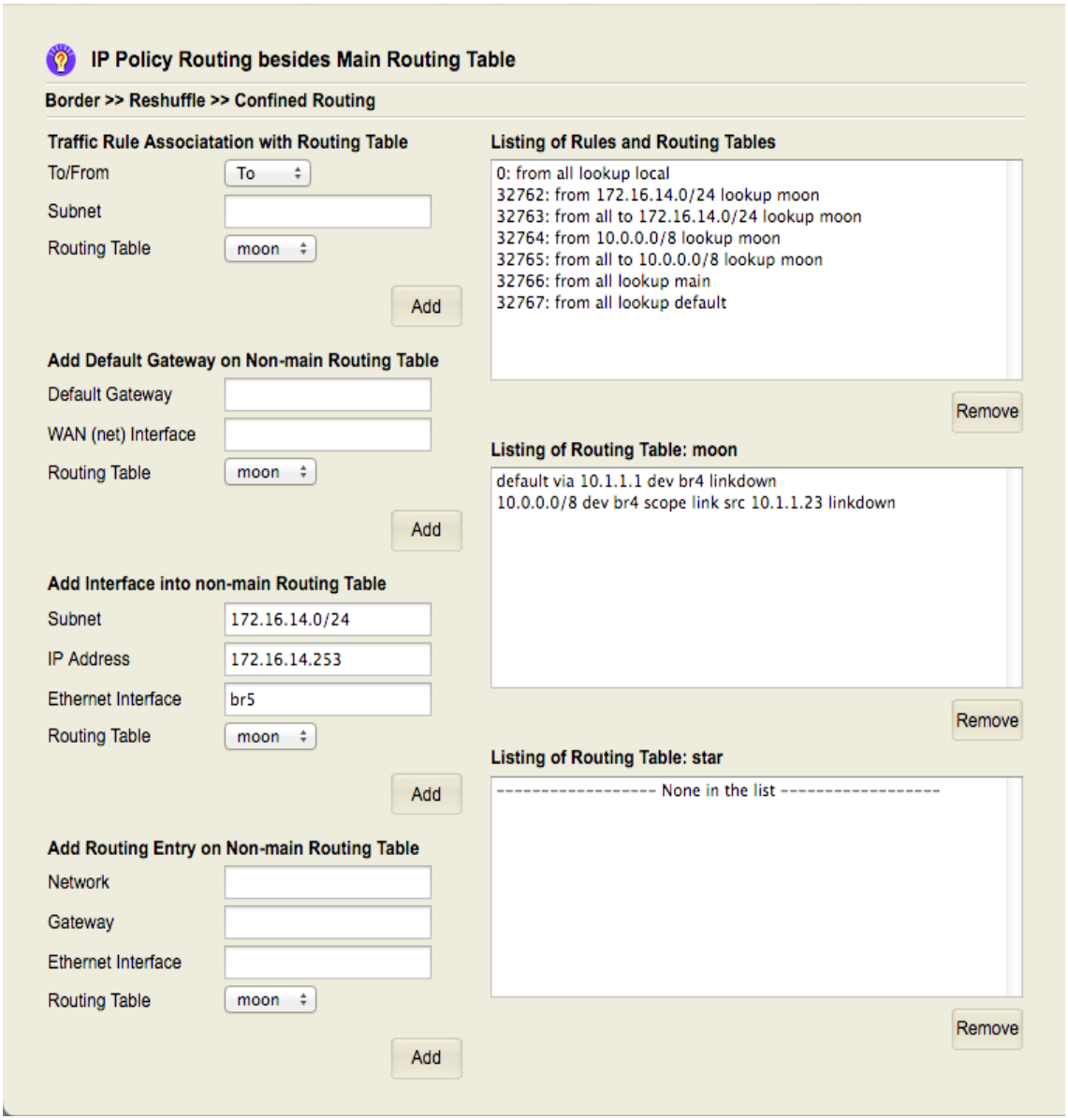


圖 96：添加將連接到“br5”的子網的路由條目，在“moon”中。

上方截圖是用於添加路由條目，用於“br5”連接到的子網。

“moon”路由表最终結果如下：

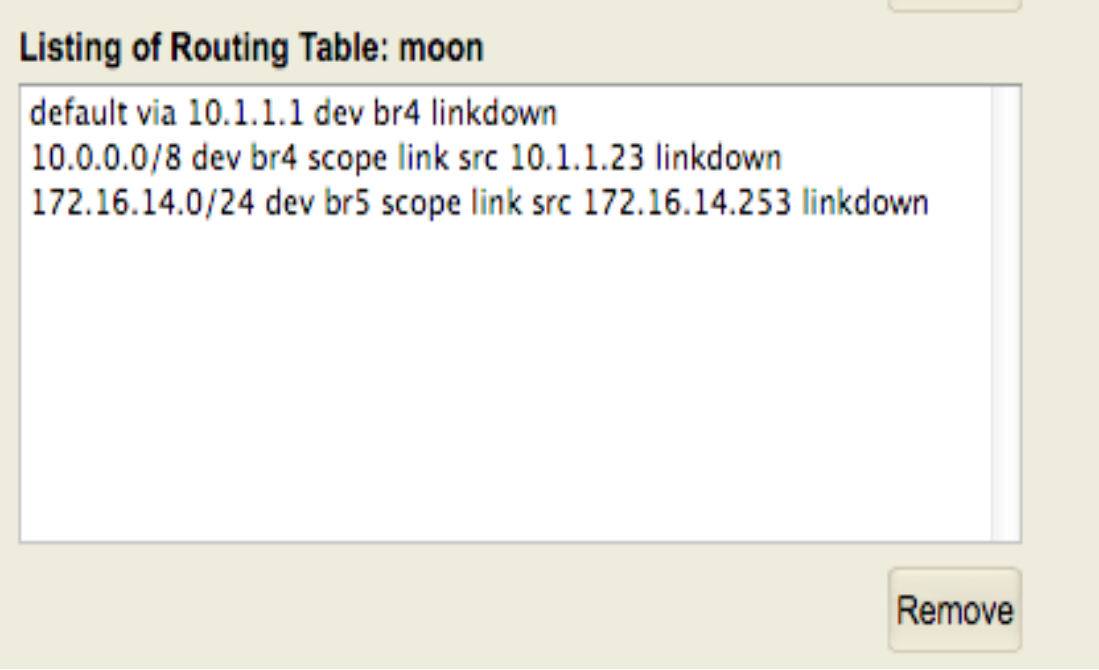


圖 97： “moon” 路由表内容

系統限制與擴展：獨立策略路由表 (Moon & Star)

特性	說明
「star」路由表的來源	**star** 路由表是基礎平臺額外提供的獨立路由表之一 。它與 **moon** 表一起，在系統啟動時就已經存在於底層網路堆疊中。
「star」路由表的用途	它的目的是提供**第二個獨立的策略分流空間** 。如果您的有另外一對橋接器用於類似用途，可以將那些特定流量的路由條目放在該路由表中 。 **star** 路由表就是您的第二個獨立策略路由工具箱，用來實現更複雜或多層次的網路流量分流需求。
配置「star」的位置	與 **moon** 表一樣，您需要透過 「**Border >> Reshuffle >> Confined Routing**」來：1. 建立流量規則，指定流量套用 **star** 表。2. 在 **Add Default Gateway on Non-main Routing Table** 等區塊，為 **star** 表手動新增路由條目。
動態路由限制	在此處介紹的網路路由僅限於 **靜態路由**。動態路由只發生在 **main** 路由表中，相關主題將在稍後介紹。

Http Reverse Proxy for Request Filtering (透過 HTTP 反向代理過濾進站 HTTP 請求)

HTTP 反向代理：實現進站請求的過濾與分流

核心問題：埠轉發與負載均衡的限制

在先前的章節中，我們介紹了「埠轉發」與「IP 負載均衡」兩種機制，它們都能將外部的 HTTP 要求（TCP 埠 80）轉送到內部私有網路中的伺服器。 然而，這兩種機制存在一個關鍵限制：

- **判斷依據單一：** 它們都是根據 **IP 標頭** 來做判斷與轉送——也就是只能依據「目的 IP 位址」與「目的埠號」來決定要送到哪一台主機。

當出現 **多個網域名稱**（例如 www.gaga.z 與 www.dada.z）**同時指向同一個公開 IP 位址**時，單純依靠 IP 標頭就 **無法分辨**「這一個 HTTP 要求原本是給哪一個網域」。因此，也就無法將不同網域導向不同的內部伺服器。

解決方案： Azblink NFV 平台的 HTTP 反向代理

為了因應這種情境，Azblink NFV 平台提供 **HTTP 反向代理** 功能。

反向代理的運作機制

HTTP 反向代理與傳統的 IP 轉發不同，它會深入讀取 HTTP 封包的內容：

1. **讀取 HTTP 標頭：** 反向代理會讀取 HTTP 要求中的 **Host 標頭** 與 **URL**。
2. **依網址分流：** 依照「網址中的主機名稱或路徑」來決定轉發目標。

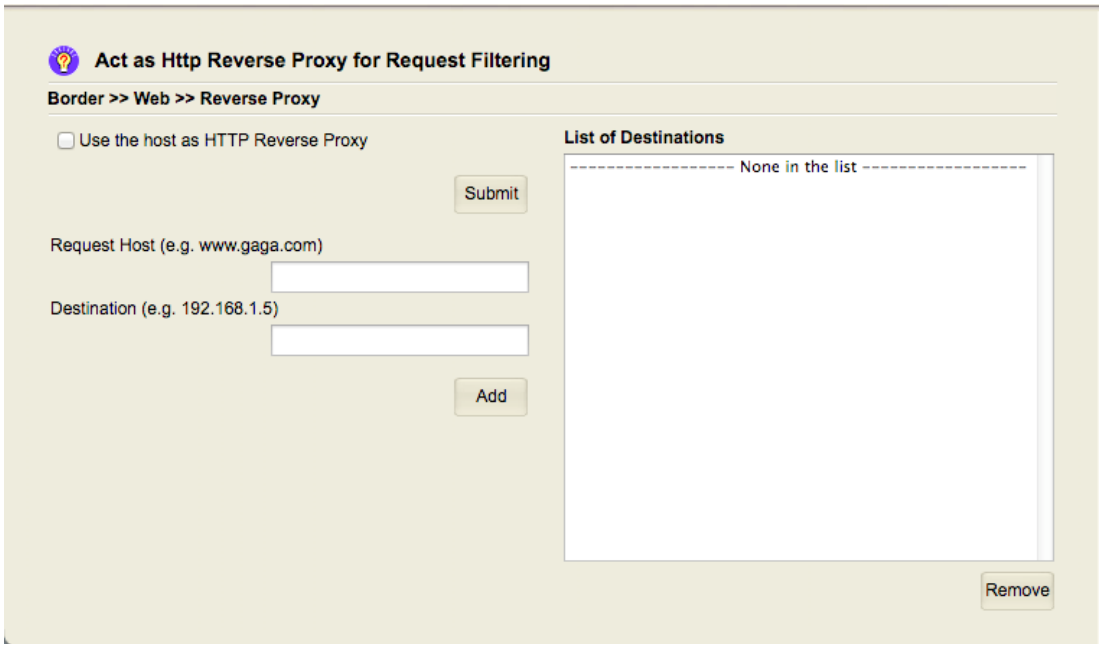


圖 98: HTTP 反向代理設定畫面

應用範例

透過反向代理，即使多個網域共用同一個公開 IP 位址，也能在單一 Azblink NFV 平台前端，彈性地將不同網域或服務，分流到對應的內部 HTTP 伺服器上。

網域名稱	判斷依據	轉發目標
www.gaga.z 請求	Host 標頭：www.gaga.z	轉發到內部伺服器 A
www.dada.z 請求	Host 標頭：www.dada.z	轉發到內部伺服器 B

總結： HTTP 反向代理提供了更細緻的控制能力，使得多個網域能夠共用單一公開 IP，同時確保每個網域或服務的請求都能被精確地導向其專屬的內部伺服器。

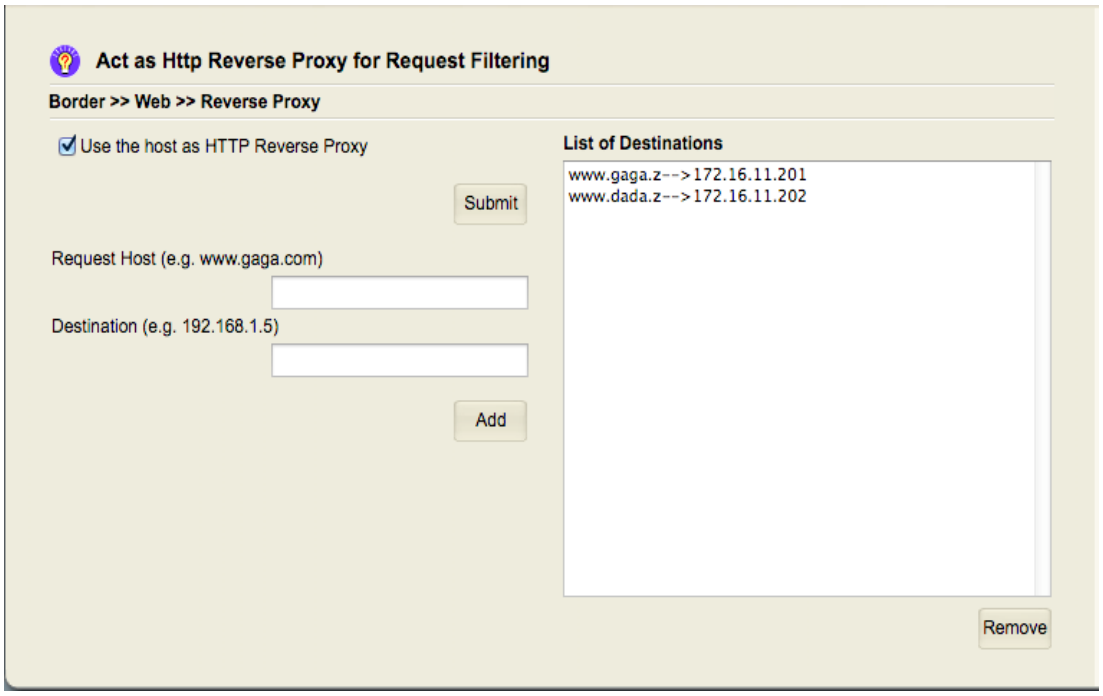


圖 99：HTTP 反向代理用於兩個不同主機名稱

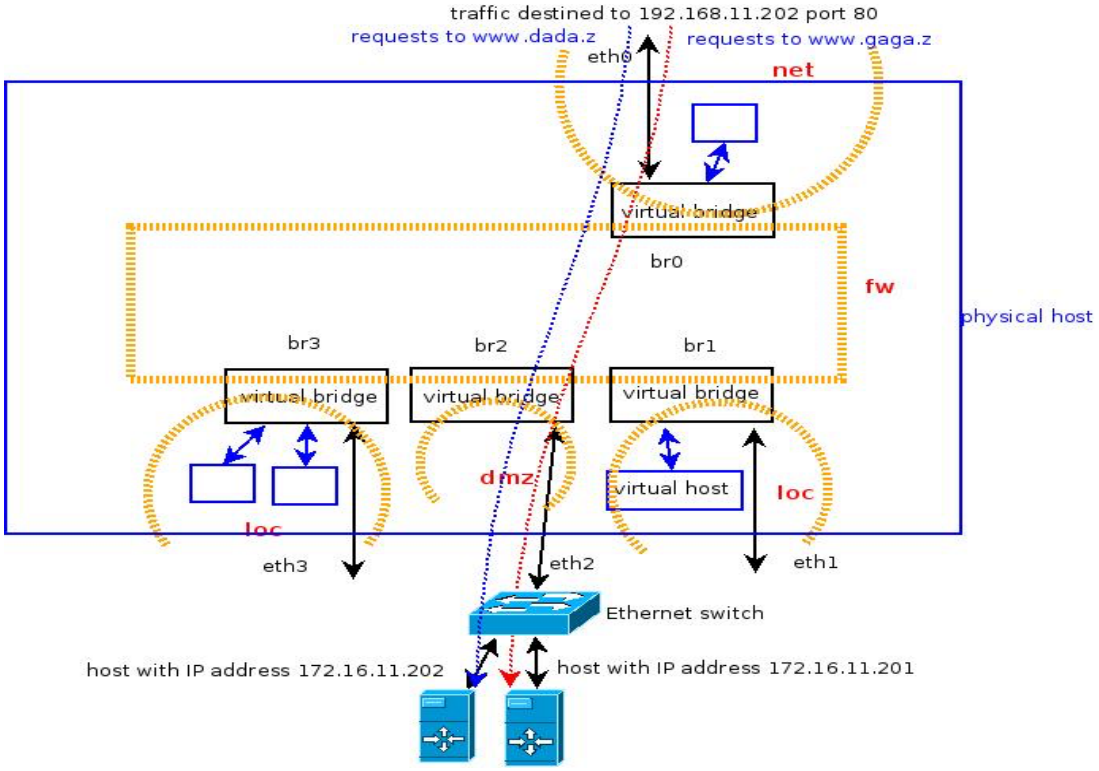


圖100: HTTP 反向代理適用於兩個主機

第四章 虛擬私人網路（VPN Virtual Private Nework)

虛擬私人網路 (VPN)：NFV 平台中的安全延伸

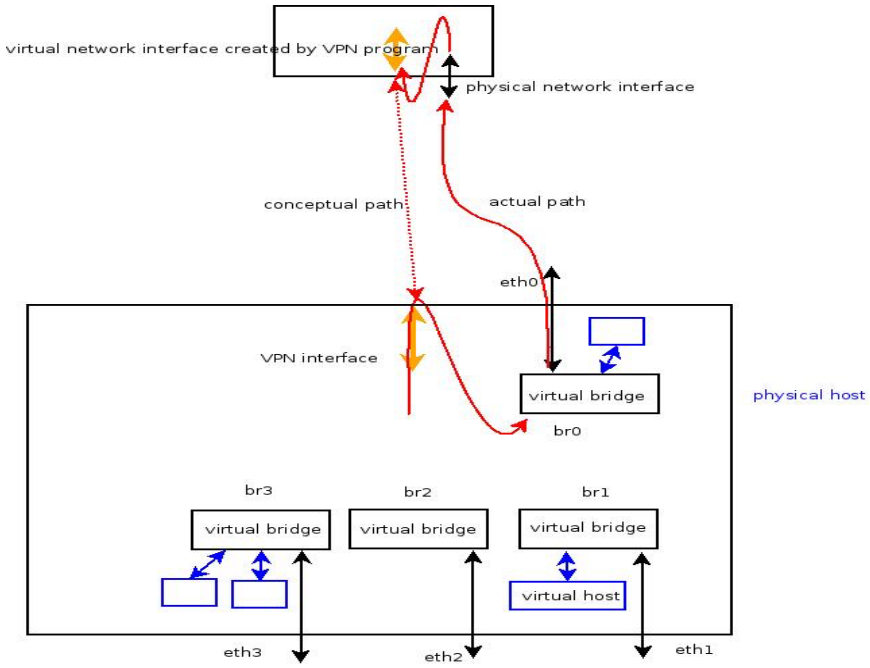
在 Azblink NFV 平台的架構中，我們已探討了如何透過橋接器、區域定義（Zone Definition）、策略路由 和 頻寬控管等機制來建立一個**結構嚴謹且安全隔離的內部網路環境**。然而，企業的運營需要**安全地延伸其網路邊界**，以支援遠端工作者、分支機構連線或雲端資源存取。

虛擬私人網路 (VPN) 在此扮演了不可或缺的角色。VPN 負責在不可信的公共網路（通常是網際網路）上建立**加密、安全的通道**，使得遠端的用戶或網路可以**如同身處內部區域（如 loc 區域）一般**，安全地存取企業資源。在本 NFV 平台中，VPN 不僅提供了傳統的**遠端存取 (Remote Access)** 功能，同時也透過 **站點對站點 (Site-to-Site)** 連線，將分散的地理位置納入統一的安全策略管理之下。本章節將詳細說明如何在平台上配置不同類型的 VPN 服務，確保您的資料在傳輸過程中的**機密性與完整性**。

在本手冊中，VPN（Virtual Private Network，虛擬私人網路）指的是：在 VPN 伺服器與 VPN 用戶端之間，各自建立一個「虛擬網路介面」，並將送往這些虛擬介面的網路封包，包裝成一般 IP 封包，透過實體網路介面傳送到對端。到達另一端後，VPN 伺服器／用戶端會將這些封包解開，再交給本機的虛擬網路介面，由應用程式像使用一般網卡一樣收發資料。

從應用程式的觀點來看，VPN 就是一個額外出現的網路介面，可用來收送只有 VPN 兩端才能看見的「私人」網路流量，而不需要關心中間經過的是什麼實體網路或公共網際網路。

圖例 101：VPN 運作原理



在本文件中，**VPN（Virtual Private Network）** 指的是：在 VPN 伺服器與 VPN 用戶端（或另一端的 VPN 裝置）之間，建立一組虛擬網路介面。傳送到這些虛擬介面的封包，會先被封裝成一般 IP 封包，經由實體網路介面送到對端；對端的 VPN 程式再將封包解封，交給本機上的虛擬網路介面。
從應用程式的角度來看，VPN 就像是系統多了一張網路卡，可以直接用來收送封包。

實務上還有許多不同形式的 VPN 實作，但在本手冊中，我們只討論**透過虛擬網路介面與其它系統組件互動**的這一類型。

在拓樸上，常見有兩種型態：

- **Client-to-Site（用戶端對站點）VPN**
- **Site-to-Site（站點對站點）VPN**

在封包處理層級上，又可以分成兩種模式：

1. **路由模式（Routed VPN）**
    - VPN 擁有獨立的 IP 子網。
    - VPN 行程會建立一個虛擬網路介面，其 IP 位址配置在該子網內。
    - 要使用這條 VPN，只需要在路由表中加入規則，讓指定的流量「走向」這個 VPN 子網即可。

2. **橋接模式（Bridged VPN）**
    - 封包是以乙太網框（Ethernet frame）形式被封裝與傳送。
    - 通常用於小規模網路，因為它等於把兩端當成同一個 L2 廣播網段。
    - 若在大規模網路上採用橋接模式，等於把非常多主機塞在同一個 IP 子網內，會使廣播與雜訊變多，也不容易把流量隔離到較小的區段。

Client-to-Site VPN 的典型用途，是讓行動用戶或遠端員工連回公司內網：
用戶端先啟動 VPN 程式，建立虛擬網路介面，取得一個屬於企業內部 VPN 子網的 IP 位址；之後，任何送往企業內部主機的流量，都會透過這個虛擬介面進入 VPN 通道，由 VPN 伺服器再轉送到企業私有網路中的各個子網（如圖中的 subnet A/B/C）。

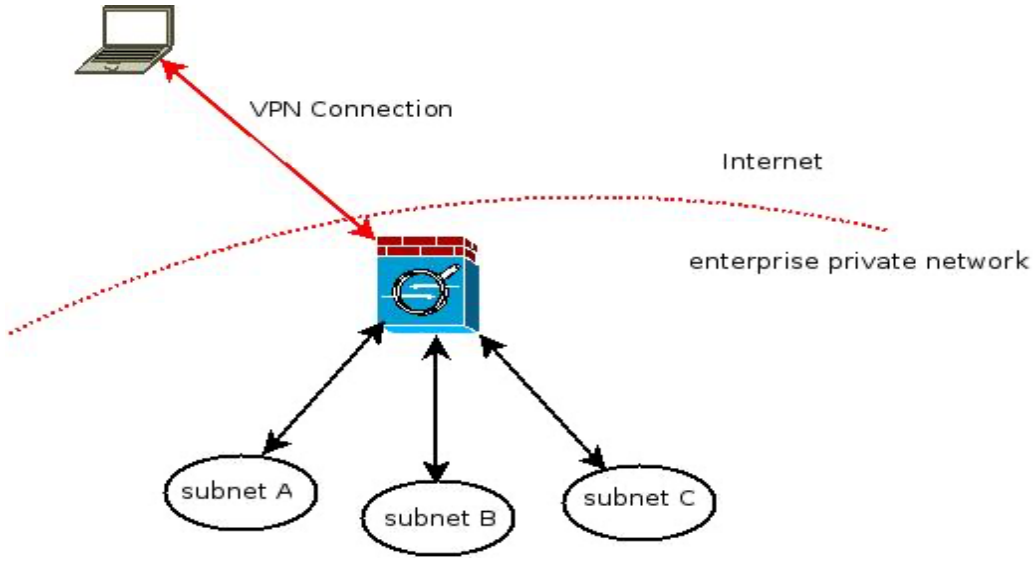


圖 102：客户端至站點 VPN

站點對站點 VPN（Site-to-Site VPN）是指在兩端設備之間建立一條持續的 VPN 通道，讓各自背後的子網可以彼此存取，就像位於同一個延伸網路中一樣。

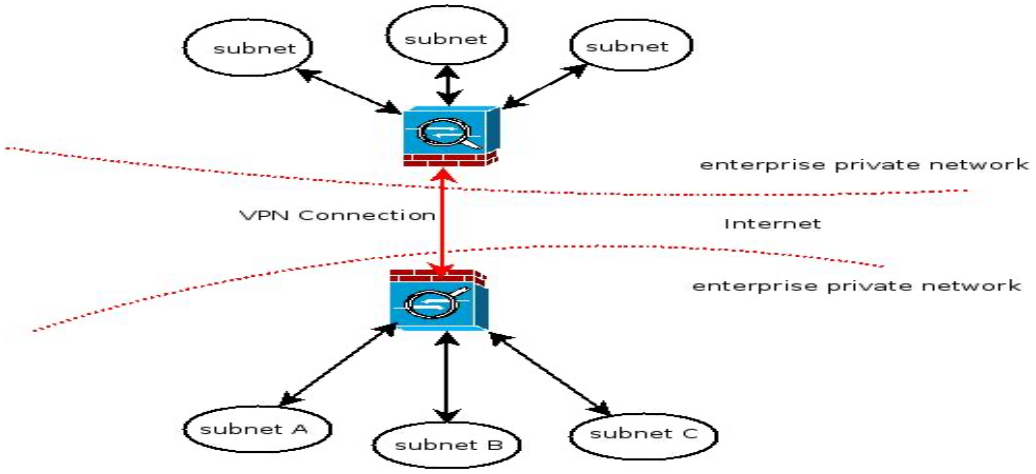


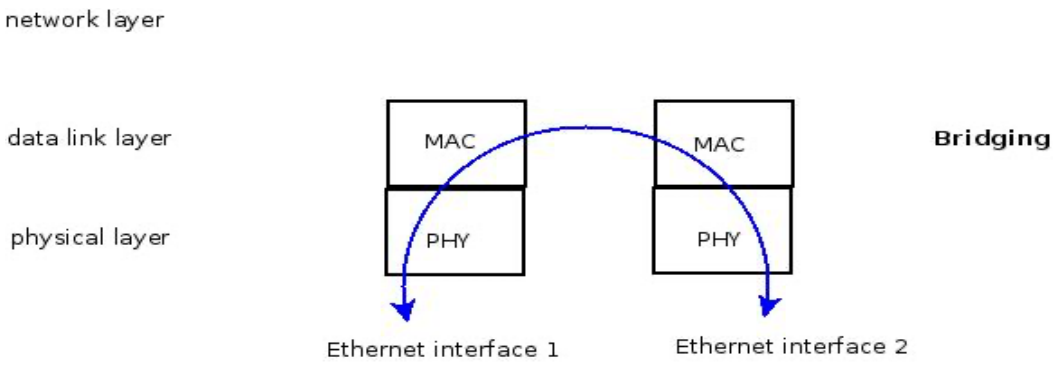
圖 103：點對點 VPN

若 **Client-to-Site VPN** 以「橋接模式」運作，VPN 用戶端的 IP 位址必須落在企業內部某一個私有子網中；同樣地，**Site-to-Site VPN** 若採用橋接模式，則代表同一個 IP 子網會橫跨兩個實體站點。由於這樣的設計在大型環境中較難控管與隔離廣播流量，本產品 **不支援 Client-to-Site VPN 的橋接模式**，僅支援在 **路由模式** 下運作。

請再次回想本手冊前面提到的差異：

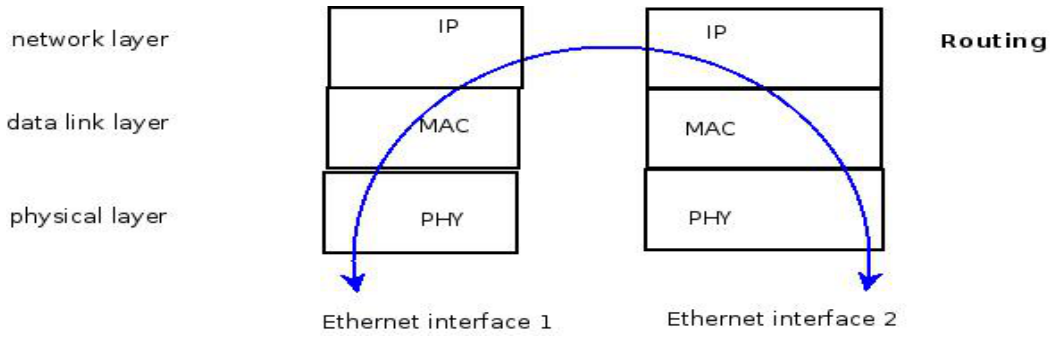
- **橋接（Bridge）**：工作在資料鏈結層，根據乙太網路框中的 **MAC 位址** 來決定封包應從哪一個實體介面送出。
- **路由（Routing）**：工作在網路層，依據 IP 標頭中的 **IP 位址** 來判斷封包的下一個路由節點。

對於 **採橋接模式的 VPN**：
由 VPN 程式建立的虛擬網路介面 **不配置 IP 位址**，而是被加入某一個橋接器，與其他實體或虛擬介面直接交換乙太網路框。



圖例 104: 橋接

對於 **採路由模式的 VPN**：
由 VPN 程式建立的虛擬網路介面會 **配置 IP 位址**，並由系統的路由表決定封包應轉送到哪一個下一跳網路。



圖例 105: 路由

Azblink 基礎平台內建一套自己的 **CA（Certificate Authority，憑證簽發單位）**，可自行簽發數位憑證。這些憑證會用於：

- 驗證 VPN 使用者與設備身分
- 在 VPN 控制通道上進行加密與保護交握資訊

後續各種 VPN 應用情境，皆會沿用相同的「金鑰與憑證生成流程」。因此本章僅做概念性說明，不刻意深入加密演算法與實作細節。

VPN CA & Key Management (VPN 憑證與金鑰管理)

在本文件中所提到的 VPN（Virtual Private Network），是指：
在 VPN 伺服器與 VPN 客戶端設備上建立一對虛擬網路介面，將送往這些虛擬介面的流量，封裝為一般 IP 封包，經由實體網路傳送到另一端，再由 VPN 軟體將其解封，交給上層應用程式使用。
對應用程式而言，VPN 就像是「多了一張可用來傳送／接收封包的網卡」，而不需要了解背後封裝與加密的細節。

要讓這條「虛擬專線」既安全又可信，就必須使用**憑證與金鑰**來做身分驗證與通道加密。
Azblink NFV 平台內建一套完整的 CA（Certificate Authority）機制，可以在本機上自行簽發 VPN 所需的伺服器憑證與用戶端憑證，**完全不依賴外部公開 CA 機構**。系統管理員也可以在後台調整 CA 的組織名稱、單位、國家等欄位，建立出客戶自己專屬的企業 CA。

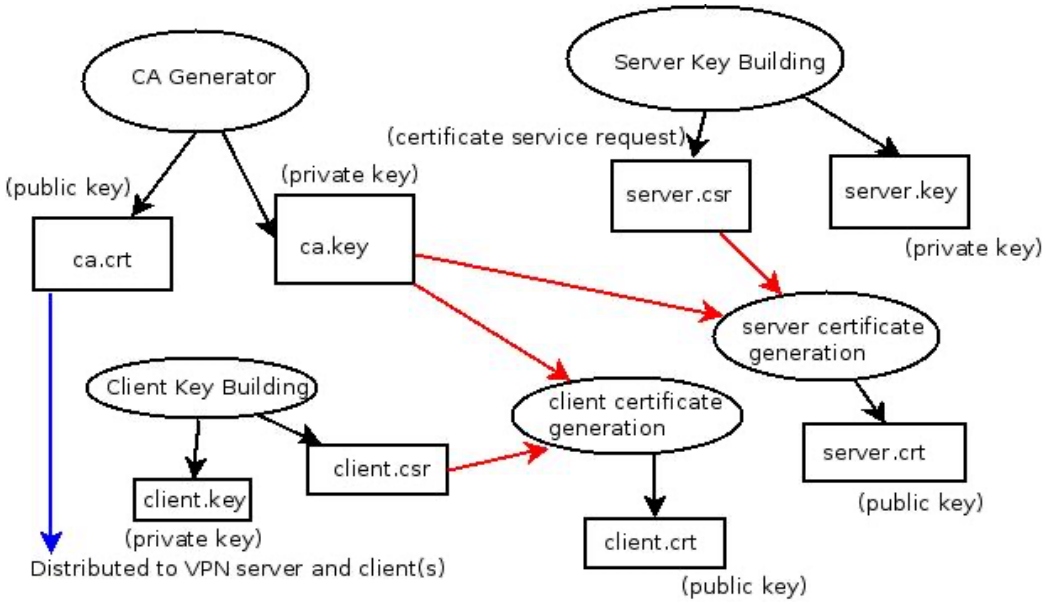


圖 106：VPN 鍵生成過程

以下說明整體流程中，各個角色與步驟。

---

1. 建立 Azblink NFV 內建 CA（根憑證與金鑰）

在首次啟用 Azblink NFV 的 VPN 功能時，平台會先初始化一個「內建 CA」：

1. **產生 CA 金鑰對**
    - **CA 私鑰**：只儲存在這台 NFV 基礎平台主機上，必須嚴格保護，不對外散佈。

- **CA 公鑰／CA 根憑證**：會提供給本機的 VPN 伺服器，以及後續匯出的用戶端設定套件，用來驗證之後所有由此 CA 簽發的憑證是否可信。

2. **自訂 CA 資訊**
   系統管理員可以在平台中修改 CA 的主體資訊（如公司名稱、組織單位、Country code 等），讓這台 NFV 成為客戶自己網域內的「專屬憑證中心」。

自此之後，所有 VPN 相關憑證（伺服器憑證、用戶端憑證）都會由這個**內建 CA** 簽發，組成一個完全封閉、由客戶自行掌控的信任鏈。

---

2. 在 NFV 上產生 VPN 伺服器憑證

接著，需要替 Azblink NFV 平台上的 VPN 伺服器建立專屬的憑證，用來讓用戶端辨識「我連上的確實是正牌的 VPN 伺服器」。

在 Azblink NFV 上，這整個過程都在同一台主機內自動完成，管理員只需要在管理介面發出建立指令：

1. VPN 伺服器元件會在本機上產生一把 **伺服器私鑰**。
2. 系統同時在本機產生對應的 CSR（憑證簽署請求），並直接送交**同一台 NFV 主機上的內建 CA 模組**。
3. 內建 CA 使用自己的 **CA 私鑰** 為該 CSR 簽章，產生 **伺服器憑證**：
   - 憑證內包含伺服器的 **公鑰**。
   - 同時帶有 CA 使用 CA 私鑰所產生的數位簽章，代表此憑證是由該 CA 核發。
4. 伺服器私鑰與伺服器憑證會直接儲存在 NFV 平台的本地檔案系統中，並由 VPN 伺服器程序載入使用，**不需要任何「匯出再安裝」的手動動作**。

只要 VPN 客戶端預先匯入這台 NFV 的 **CA 根憑證**，就能使用 CA 公鑰驗證該伺服器憑證的真偽，確保不會連到假冒的 VPN 伺服器。

---

3. 在 NFV 上集中產生與發放 VPN 用戶端憑證

Azblink NFV 平台的 VPN 解決方案建構於 **OpenVPN** 技術之上，採用**伺服器集中產生用戶端憑證套件（Client Certificate Bundle）**的方式來管理存取權限，確保高標準的安全與可追溯性。

核心實作流程：

1. 用戶端項目建立： 系統管理員透過 NFV 管理介面，為每一位 VPN 使用者（或每一台終端設備）建立一個專屬的 VPN 用戶端項目。
2. 伺服器端憑證生成： 對於每個用戶端，NFV 平台會自動在伺服器端產生一組完整的非對稱金鑰基礎設施（PKI）元件：

- 用戶端私鑰 (Private Key)
- 用戶端憑證 (Client Certificate)：此憑證由系統內建的 CA（憑證授權機構） 簽發。

3. 設定檔打包與匯出： 平台隨後會將「用戶端私鑰、用戶端憑證、CA 根憑證」安全地打包為用戶端可匯入的設定檔格式（例如 PKCS#12 或常見的 OpenVPN 客戶端配置格式），供管理員或使用者下載。
4. 端點設備部署： 使用者只需將此設定檔匯入到自己的端點裝置（PC、筆電或手機）上，即可完成 VPN 憑證佈署，並準備連線。

優勢： 這種方法消除了在客戶端手動產生金鑰的複雜性，並確保了所有憑證的來源統一且受控，是 OpenVPN 環境下建議的安全且高效的佈署模式。

每一個 VPN 客戶端都會擁有一組獨一無二的用戶端憑證與對應私鑰，這組憑證就是該裝置的「VPN 身分」。 完成匯入後，該裝置在建立 VPN 連線時，**不再需要額外輸入帳號與密碼**；驗證動作完全依賴憑證與金鑰。

這種設計有幾個實務上的關鍵優點：

- **連線體驗更直覺**：
  一旦裝置安裝好憑證，日後只要啟動 VPN 客戶端就能直接建立連線，無需反覆輸入帳號／密碼。
  對一般辦公使用者來說，操作更接近「開／關 VPN 開關」的體驗，而不是每次都要記帳號與複雜密碼。
- **特別適合「自動啟動」的通話情境**：
  在結合 Azblink UC / VoIP 或行動電話應用時，常見情境是：

  > 手機收到來電 → 系統需要先在背景建立 VPN → 再把呼叫導入內部總機或應用伺服器。 在這種情況下，如果還要使用者臨時輸入 VPN 帳號密碼，整個流程會被中斷、體驗非常不自然。透過憑證式驗證，VPN 可以在背景自動升起，對使用者來說就像「直接接電話」一樣直覺，卻仍然維持足夠的安全性。

- **權限控管依然集中在伺服器端**：
  若某台裝置遺失或不再信任，只要在 NFV 管理介面撤銷該用戶端憑證，該憑證對應的裝置立刻失去 VPN 存取權限，無需依賴帳號密碼更改。

總結來說：
在 Azblink NFV 上，不需要額外的 CSR 上傳流程，也不會要求由用戶端自行產生金鑰再送回簽署；所有用戶端憑證的生成與發放，都集中由 NFV 伺服器端自動完成。
每一個要連線的用戶端，都必須擁有一份由 NFV 內建 CA 簽發的有效用戶端憑證；只有持有合法憑證的設備，才有資格與 VPN 伺服器建立受信任的加密通道。

4. VPN 連線建立時實際發生的事

當用戶端要連線到 Azblink NFV 的 VPN 伺服器時，大致會經歷以下步驟：

1. **握手與憑證交換**
    - 用戶端連線至 VPN 伺服器，伺服器送出自己的「伺服器憑證」。
    - 用戶端使用事先匯入的 **CA 根憑證（CA 公鑰）** 驗證：
        - 這張伺服器憑證是否由「這台 NFV 的 CA」簽發；
        - 憑證簽章是否正確；
        - 憑證是否在有效期限內。
    - 同時，用戶端會將自己的「用戶端憑證」送給伺服器，伺服器亦會使用同一個 CA 根憑證進行驗證。
2. **協商加密方式與工作金鑰**
    - 雙方透過金鑰交換協定，協商要使用的加密演算法（Cipher Suite）與一組對稱加密的「工作階段金鑰」（Session Key）。
    - 從這一刻起，兩端之間的所有 VPN 流量（含 IP 封包負載）都會使用這組 Session Key 加密傳輸。
3. **建立虛擬網路介面**
    - VPN 伺服器與用戶端各自建立一個虛擬網路介面，分配對應的虛擬 IP 位址。
    - 用戶端上的應用程式只需要把封包送到這個「虛擬網卡」，VPN 程式會自動負責封裝、加密、送出，並在接收端解封與還原。

整個過程對終端使用者而言，只要「開啟 VPN」，無須再額外輸入帳號與密碼；但在底層，Azblink NFV 仍然透過憑證與金鑰，完整地做到雙向驗證與通道加密。

---

5. 使用內建 CA 的實際好處

將 CA 直接內建在 Azblink NFV 平台，有幾個實務上的重要好處，可以在手冊中讓讀者一目了然：

- **不依賴外部公開 CA**
  在完全封閉或與網際網路隔離的環境中（例如產線網路、實驗室、內部試算中心），仍然可以自行簽發與管理憑證。
- **每個客戶擁有自己的「信任網域」**
  每一台 Azblink NFV 都可以建立獨立的 CA 主體資訊，不會與其他組織共享同一套憑證體系，符合企業對信任邊界的需求。
- **憑證簽發與註銷流程集中管理**
  由於 CA、VPN 伺服器與管理介面都在同一台平台上，新增／撤銷用戶端憑證、調整憑證有效期限或層級，都可以在同一個管理介面一次完成。

- **搭配 NFV 架構，簡化多區域安全部署**
  配合 NFV 上的分區（net / dmz / loc 等）與虛擬機部署，管理者可以很輕易地為不同安全層級、不同子網的虛擬主機建立對應的 VPN 存取政策。
- **對自動化與通話場景特別友善**
  例如行動端來電、UC 通話或自動化腳本觸發 VPN 的場景，都可以在**無人工輸入帳號密碼**的前提下，自動建立安全通道，大幅提升體驗的一致性與即時性。

Client to Site VPN Connection (客户端至站點 VPN)

正如前文所述，本平台**不支援「橋接模式」的 Client-to-Site VPN**，只在**「路由模式」**運作。因此，必須先為 VPN 規劃一個專用 IP 子網。 在本平台中，VPN 連線分為**控制通道**與**資料通道**兩個層次：

- 控制通道使用 UDP 連線（預設埠號 **1194**），並採用 TLS/SSL 所提供的加密與鑑別演算法，用來**協商資料通道所使用的對稱加密演算法與金鑰**。
- 資料通道則通常使用 **AES-CBC** 或 **AES-GCM** 等安全性較佳的套件進行加解密。

註：某些版本仍提供 SEED、CAST、Blowfish（BF）等 64-bit block cipher。因為 64-bit 區塊在高流量情境下有被攻擊的風險，實作上會在傳送一段資料量（例如 64 MB）後強制重新協商金鑰，以降低風險。若硬體運算能力較弱，管理員可以在了解風險後，視需要選擇這類密碼套件。有研究指出，對於 64 位元分組加密套件（例如 SEED、CAST、Blowfish 等），如果在同一把金鑰下累積加解密的資料量接近 700GB，就可能出現被破解的風險。為了降低這個風險，系統會在每傳送約 64MB 資料後，自動重新協商加密套件與金鑰。
因此，即使在硬體效能較弱的環境中仍選用這類密碼套件，依然可以維持實務上足夠的安全性。

如果 VPN 使用的子網是“172.16.38.0/24”，那麼 VPN 服務器在子網中的 IP 地址是“172.16.38.1”。你可以使用這個 IP 地址來訪問 VPN 服務器。例如，如果在基礎平台中安裝了一個虛擬主機，並為通過 VNC 訪問控制台設置 TCP 埠 5904，那麼你可以使用 VPN 通過 VNC 觀察者，並設置以下參數來訪問虛擬主機的控制台:

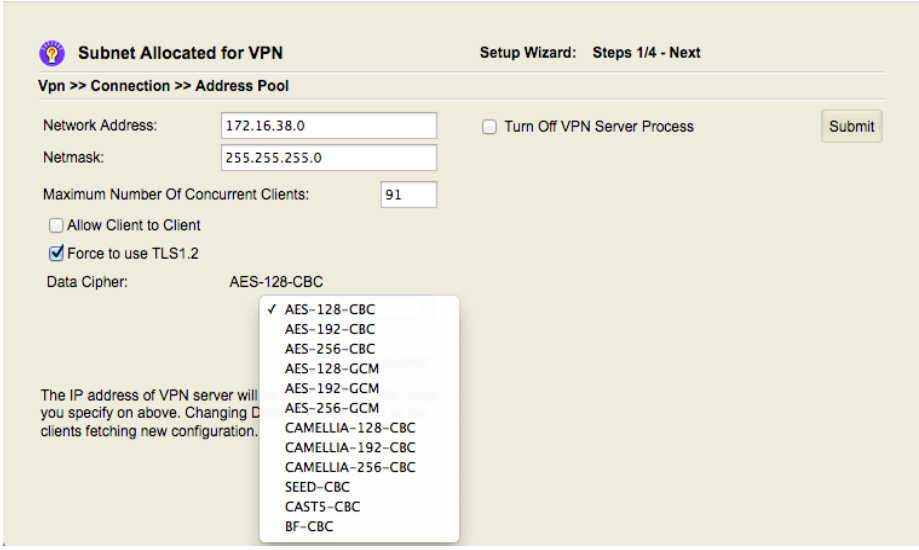


圖 107：客戶端至網站 VPN 位址池

圖 108: 數據密碼器選擇

VPN 子網與伺服器位址 (路由設定與推送 (Pushed Setting)

目的：確保路由可達性

VPN 客戶端連線成功後，其系統預設只知道如何到達 VPN 伺服器所在的虛擬子網。然而，它並不知道企業內部其他私有子網的存在與路徑，這就形成了路由盲點。

Pushed Setting 的目的正是為了解決這個問題：透過推送路由，可以明確告訴客戶端如何將流量導向企業內網的特定子網，並將 VPN 伺服器設定為這些內部子網的下一跳 (Next-Hop) 閘道，確保流量能安全地轉發到目的地。

實際運作機制

假設 VPN 使用的子網為 **172.16.38.0/24**：

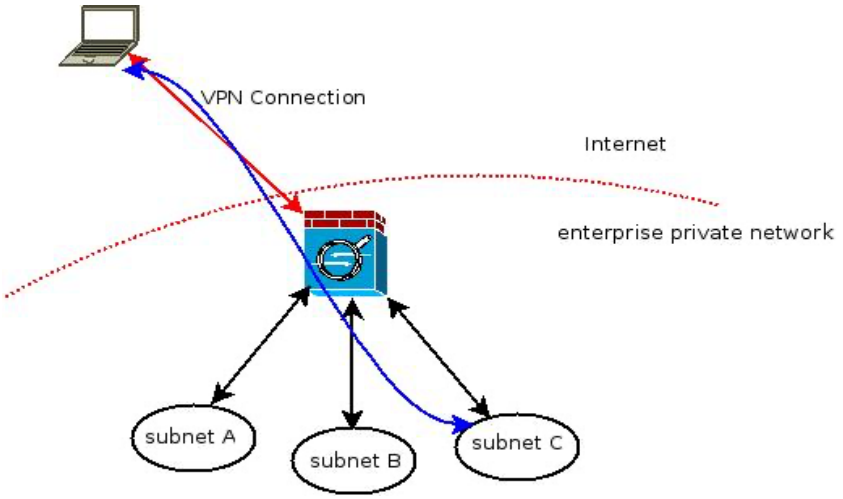
- VPN 伺服器在此子網中的位址為 **172.16.38.1**。
- 所有連線成功的 VPN 客戶端，會從這個子網中取得各自的虛擬 IP。

管理員可以透過「**Vpn >> Connection >> Pushed Setting**」，將特定子網的路由資訊推送給 VPN 客戶端。 例如：若希望 VPN 客戶端能存取「子網 C」，就需在 Pushed Setting 中加入該子網的路由項目，如此客戶端就會自動將 **VPN 伺服器 IP**（172.16.38.1） 視為該子網的**閘道 (Gate-way)**。

同一個畫面也可以指定：

- 供 VPN 客戶端使用的 **DNS 伺服器**
- 以及（在需要時）**WINS 伺服器**
- 當客戶端套用這些設定後，便會透過 VPN 通道查詢內部 DNS / WINS，進一步存取企業內部資源。

圖 109：VPN 客戶端訪問子網



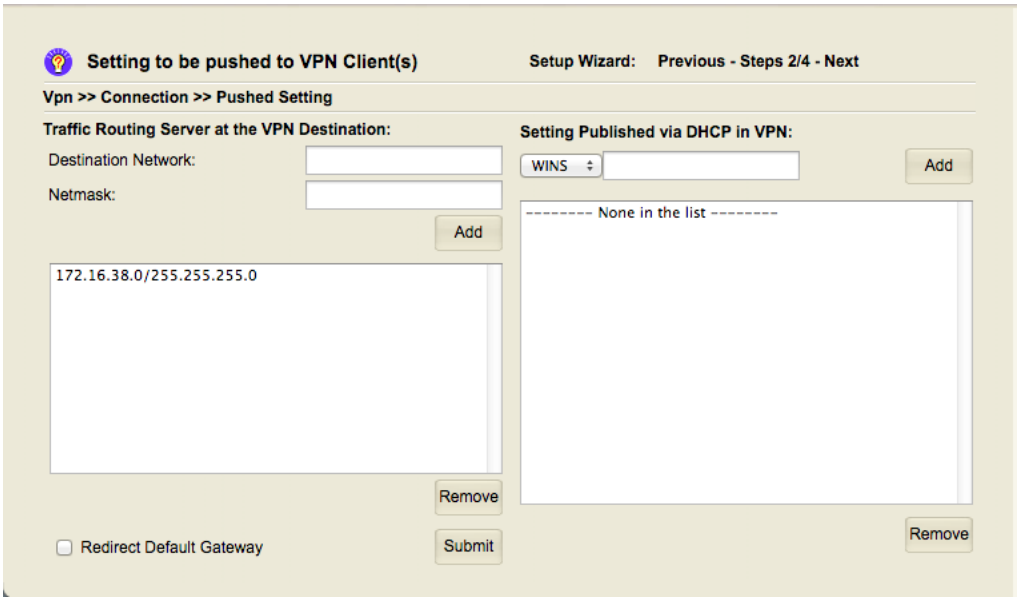


圖 110：客戶端至網站 VPN 的推動設定

這樣一來，VPN 客戶端將使用 VPN 服務器的 IP 地址作爲該子網的閘道。此外，它還可以指示 VPN 客戶端使用上圖右側所示的指定 WINS 服務器或 DNS 服務器。

透過 VPN 存取虛擬主機範例

若在 NFV 平台上已建立一台虛擬主機，並將其 VNC 控制台設定在 TCP 埠 **5904**，則遠端使用者可依下列方式存取該虛擬主機：

1. 在端點裝置啟動 VPN，用平台發放的憑證與金鑰連線至 VPN 伺服器。
2. 連線成功後，啟動 VNC Viewer，連線目標設定為：

   主機位址：**172.16.38.1**
   連線埠：**5904**

請注意：

- VNC (Virtual Network Computing) 客戶端連線的對象是 **NFV 平台本身的 IP（172.16.38.1）**，而不是虛擬主機的內部 IP。
- 只要 VPN 通道建立妥當，VNC 便能透過平台轉接到對應的虛擬主機控制台。

這樣的設計非常適合需要「先 VPN、再開桌面」的情境，例如遠端辦公、維運人員登入管理桌面，或是搭配未來 Azblink 電話/通話應用：

- 因為採用**憑證式驗證**，裝置在取得並安裝好客戶端憑證後，
- **建立 VPN 連線時不需再輸入帳號與密碼**，可大幅簡化實際操作流程，
- 對於來電彈跳、通話自動接通等互動流程，體驗會更直覺，不會被額外的輸入步驟打斷。

NFV 內建 CA (金鑰與憑證的建立流程)

Azblink NFV 平台**內建專屬 CA（憑證授權中心）**，不依賴外部公開 CA。管理者可以直接在平台上：

1. **清除預設憑證**（若要建立自己專屬的信任體系）
   - 於「**Vpn >> Connection >> Key Generation**」介面按下「**Purge**」，
   - 系統會刪除所有預載的 CA / 伺服器 / 客戶端憑證與金鑰。
2. **建立新的 CA（Certificate Authority）**
   - 在同一畫面中輸入 CA 的 **Common Name**（例：Azblink-Corp-RootCA），
   - 按下「**Generate**」，平台會產生：
     - CA 私鑰（僅儲存在 NFV 上，不外流）
     - CA 憑證（日後用於驗證伺服器與客戶端憑證是否由此 CA 簽發）
3. **建立 VPN 伺服器憑證與金鑰**
   - 繼續在同一畫面以新的 **Common Name**（例：Azblink-VPN-Server）建立伺服器金鑰與憑證。
   - 生成過程可能需要數十秒到數分鐘，視硬體性能而定。
4. **為每一個 VPN 客戶端產生專屬憑證**

- 為每個使用者或裝置指定一個 **唯一的 Common Name**（例如：alice-phone、bob-laptop），並設定憑證有效天數（Valid days）。
- 平台會自動產生對應的「客戶端私鑰 + 客戶端憑證」，並以內建 CA 進行簽章。

5. **重啟 VPN 服務**
   - 當新的 CA 及伺服器憑證建立完成後，建議**重新啟動 VPN 伺服器**，使其開始使用最新的 CA / 伺服器金鑰與憑證組合。
   - 客戶端憑證則可在 CA 與伺服器憑證就緒後隨時新增，不影響既有連線。
6. **下載客戶端設定檔**
   - 於「**Vpn >> Connection >> Client File Download**」頁面，選擇先前建立的客戶端 Common Name（例如 earth），
   - 下載已打包好的客戶端設定檔，其中已內含：
     - 該客戶端的私鑰
     - 該客戶端憑證
     - 以及 NFV CA 憑證
   - 使用者只需在終端設備上匯入此設定檔，即可建立與 NFV 的加密 VPN 連線。

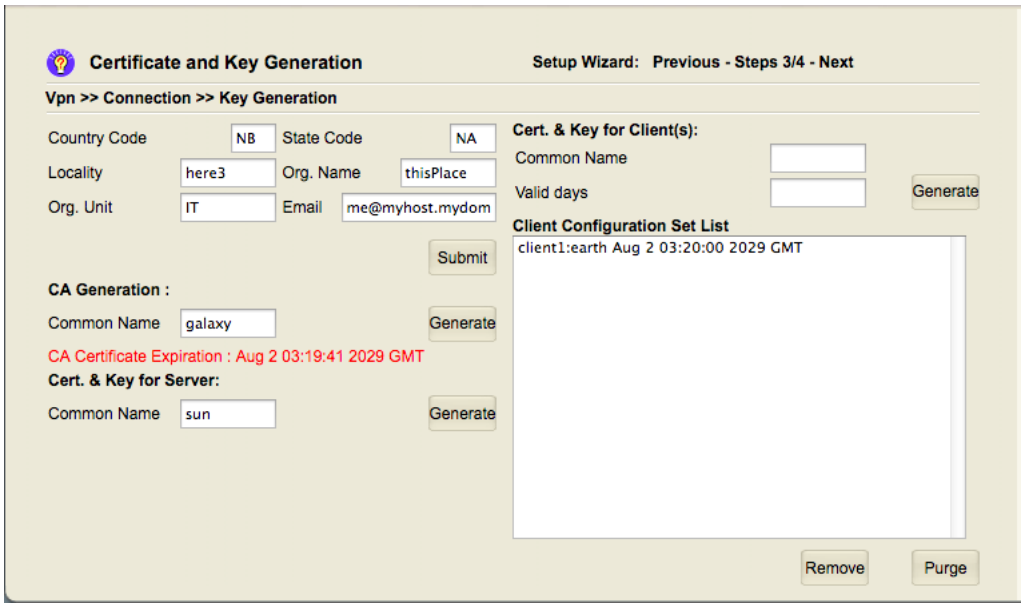


圖 111：客戶端至站點 VPN 證書和密鑰生成

憑證式 VPN 的實際優點

由於 Azblink NFV 採用 「**每一個客戶端一把私鑰、一張憑證**」的設計：

- 裝置一旦完成憑證安裝與設定，之後建立 VPN 連線時**通常不需再輸入帳號與密碼**；
- 若憑證遺失或裝置遭盜用，只需在 NFV 上**撤銷（revoke）該憑證**即可立即失效，不必修改其他使用者帳號；

- 對於需要「自動撥號、自動接通、背景連線」的場景（例如與 Azblink UC 系統 ucChat 整合），端點裝置可以在使用者幾乎無感的情況下完成 VPN 建立，再安全地存取內網應用及 統合通訊服務。

整體而言，這套機制讓 Azblink NFV 同時扮演：在一台設備上，即可完成網路加密、身分驗證、與應用系統的安全串接。

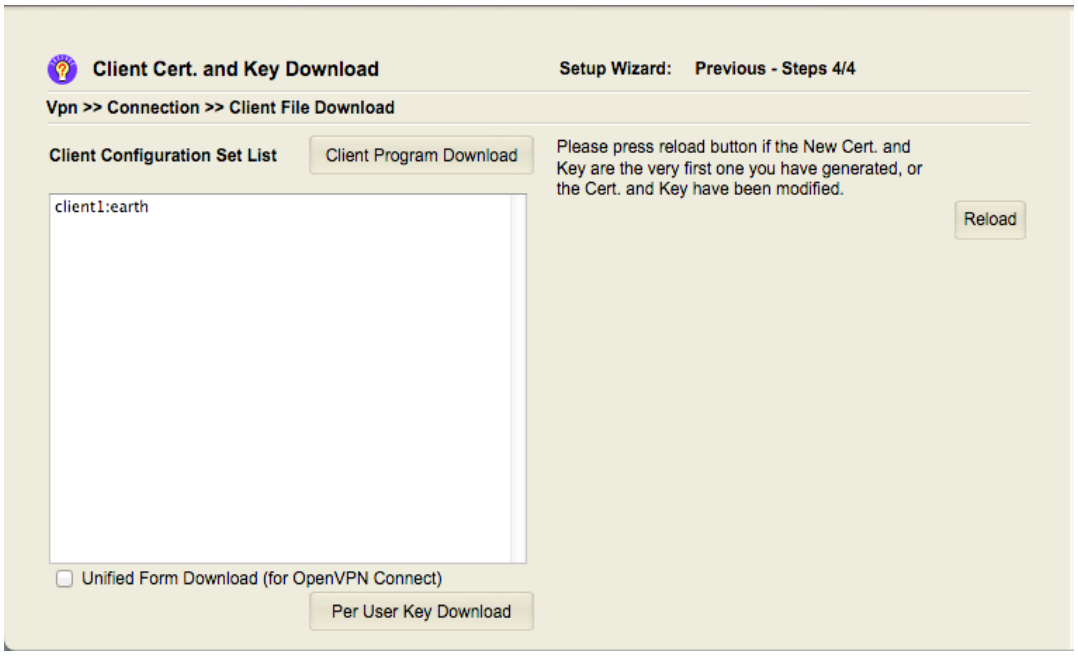


圖 112：客戶憑證下載用於客戶端到網站 VPN。

- **企業內部私有 CA 的中心，**
- **VPN 伺服器，**
- **虛擬主機與應用服務的入口。**

在上面的畫面中選取欲下載的「檔案集」（Client File Set），再按下下方的 **Download** 按鈕，即可取得對應用戶端的密鑰與憑證封裝檔。系統會為每一個用戶端產生一份 **專屬且唯一** 的檔案集，其中包含該裝置所需的設定檔、用戶端私鑰與用戶端憑證。系統管理員只需將這個檔案安全地傳送給使用者（例如透過離線媒體或受保護的傳輸管道），並指示使用者將其匯入到 VPN 用戶端程式中即可完成設定。

一旦用戶端成功匯入這組檔案，以後在與 Azblink NFV VPN 伺服器建立連線時，就**不必再輸入帳號與密碼——**憑證本身就是身份驗證憑據。這種設計特別適合「來電即自動建立 VPN」這類情境：例如我們一直提到的：手機或軟體電話在撥號／接聽時，由應用程式在背景自動啟動 VPN 連線，不需要使用者額外操作登入步驟，即可在加密通道內完成語音與訊號交換，兼顧使用體驗與安全性。

Site to Site VPN Connection Routing Mode (站對站 VPN 連線 路由模式)

站點對站點 VPN 適用於「需要讓遠端據點的部分主機，透過網際網路安全連回總部或另一個據點」的情境。與客户端對站點 VPN 不同的是，站點對站點 VPN **不需要**在遠端每一台電腦或行動裝置上安裝 VPN 客戶端程式；只要在兩端各部署一台充當 VPN 閘道的主機，並在這兩台主機之間建立 VPN 通道，即可讓兩側子網彼此互通。在本節中，我們僅討論「以路由模式運作」的站點對站點 VPN。
站點對站點 VPN 會使用獨立的 CA（憑證授權中心），**與客户端對站點 VPN 使用的 CA 不同，必須另外建立**。**Azblink 基礎平台預設使用 UDP 埠 7777 作為站點對站點 VPN 的通訊埠**，兩端 VPN 閘道會互相記錄並鎖定對方的「公開 IP 位址」，以確保僅在可信任的兩點間建立加密通道。

如前文多次提到，本平台提供的 VPN 功能，其核心是建立一個「虛擬網路接口」（又稱 *本地通道介面*）。

- 在 **路由模式** 下，這個虛擬介面會被指派一組 IP 位址，並參與該系統的 IP 路由。
- 在 **客户端對站點 VPN** 中，伺服器與客户端所使用的虛擬 IP 位址，都是從同一個地址池中自動分配。
- 在 **站點對站點 VPN** 的情境，則只需要在兩端各配置一個本地隧道介面的 IP 位址，形成一對點對點的虛擬連線即可。

請務必注意：
這兩個本地通道介面的 IP 位址 **不得與企業內任一現有子網的 IP 區段衝突**，以避免路由混淆與管理困難。只要妥善規劃通道 IP 與兩端子網路由，站點對站點 VPN 便能在不打擾使用者終端設定的前提下，為兩個據點建立長期、穩定且加密的跨站連線。

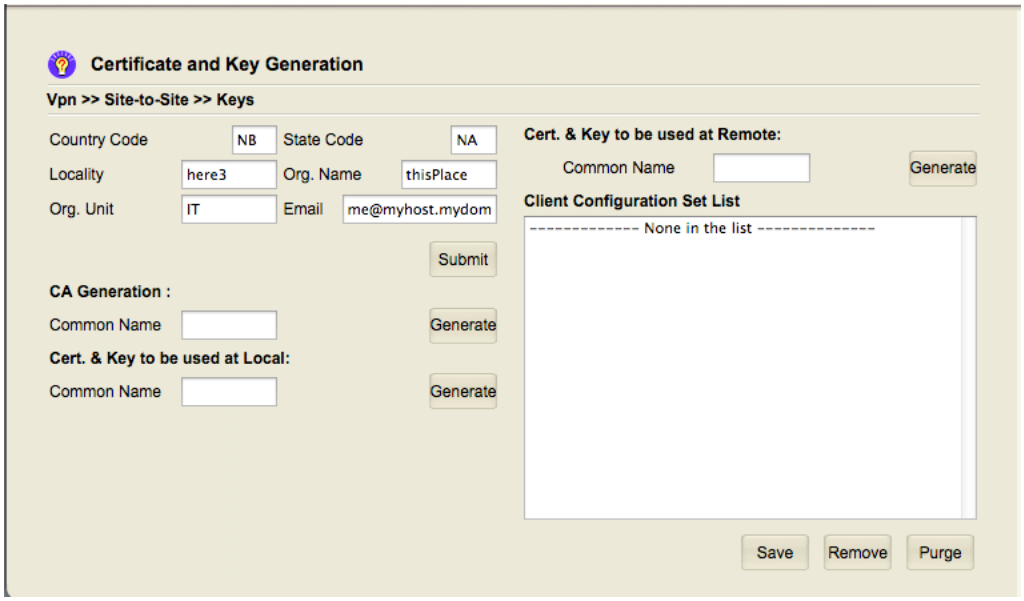


圖 113： 站對站 VPN 鍵生成

以下是密鑰生成過程中的屏幕截圖。
Site-to-site VPN'的 CA 是由以下方式生成的：

圖 114：站對站 VPN CA 生成

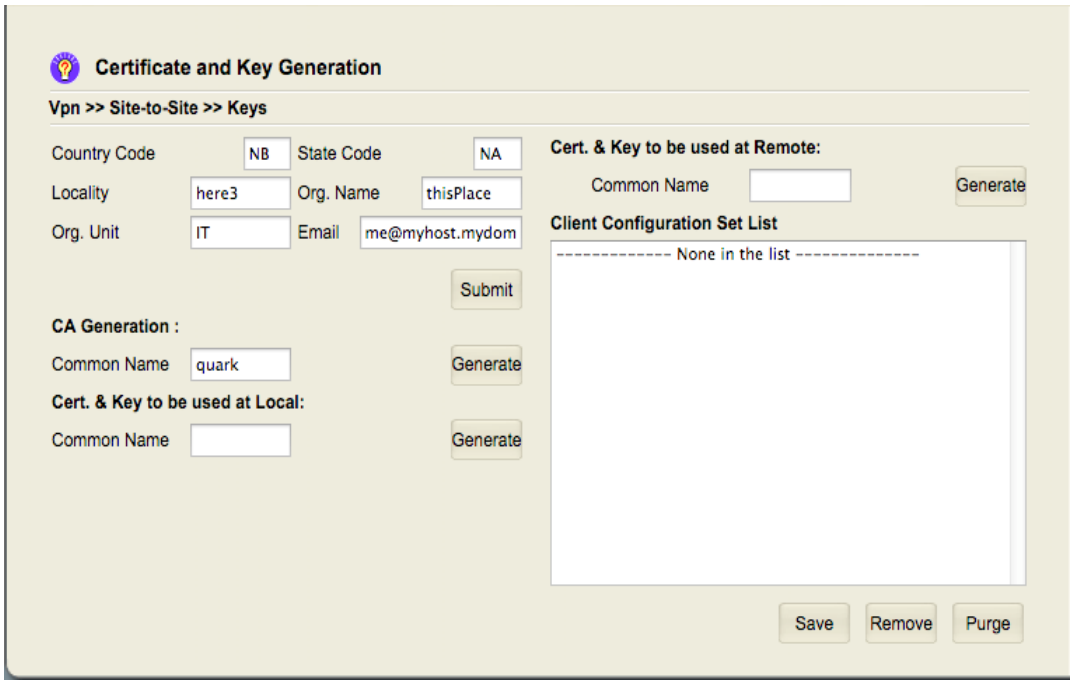
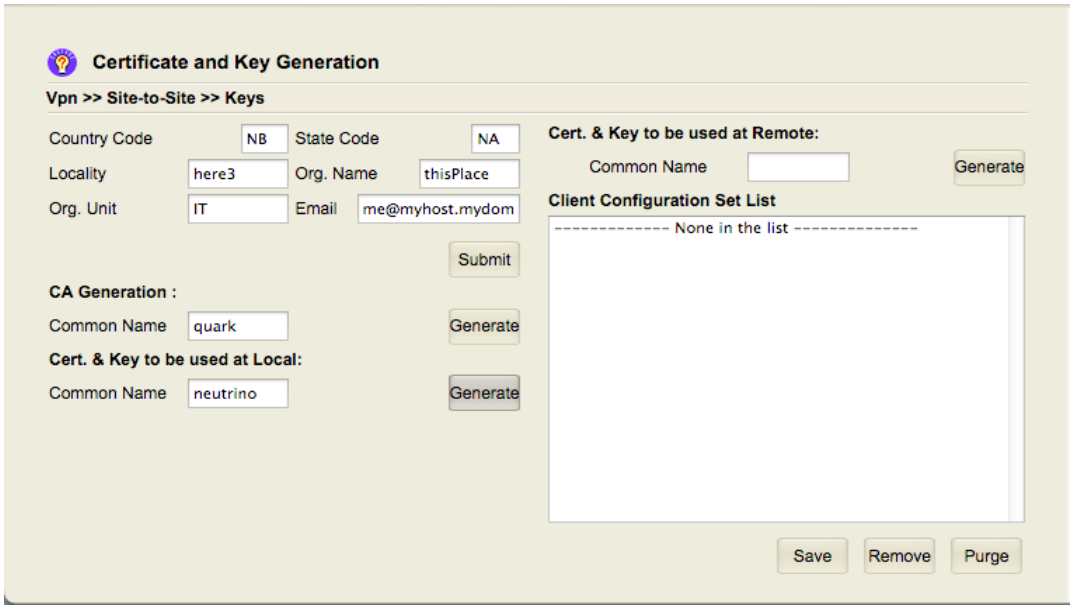


圖 115：站點至站點 VPN：服務器密鑰和證書生成



服務器密鑰和證書是通過輸入"Common Name"並按下"Generate"按鈕生成的。
為了生成客戶金鑰和證書，請填寫"Common Name"欄位，然後點擊"Generate"按鈕：

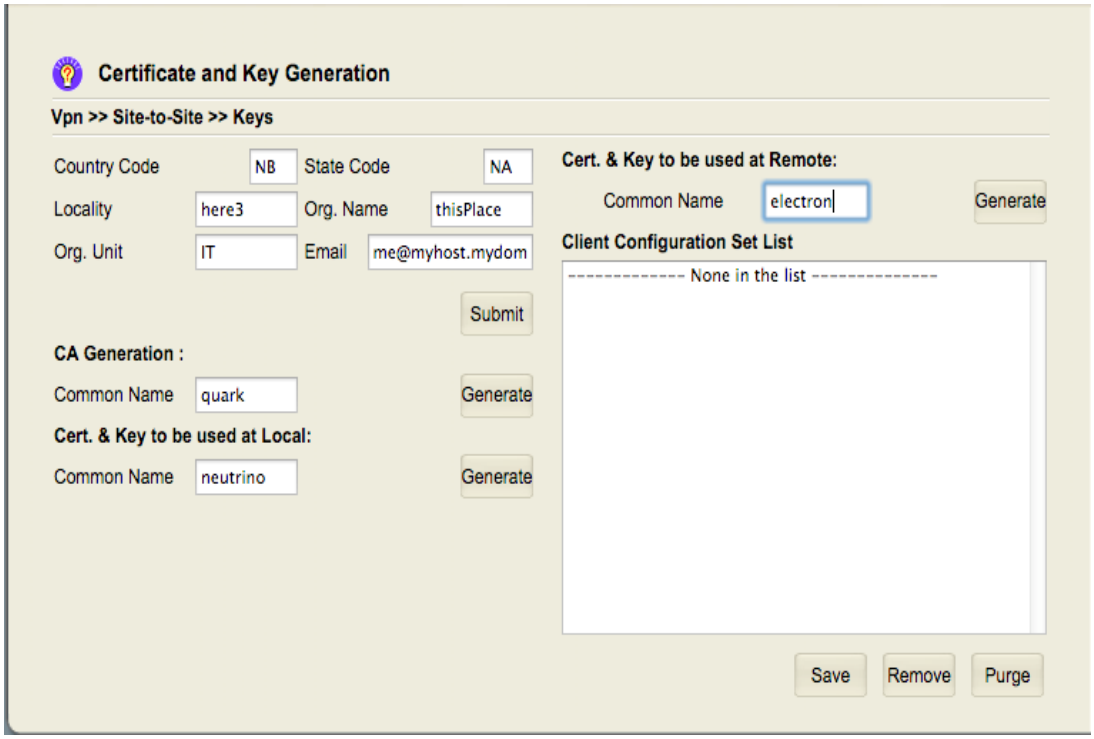


圖 116：站點至站點 VPN：客户端密鑰和證書生成

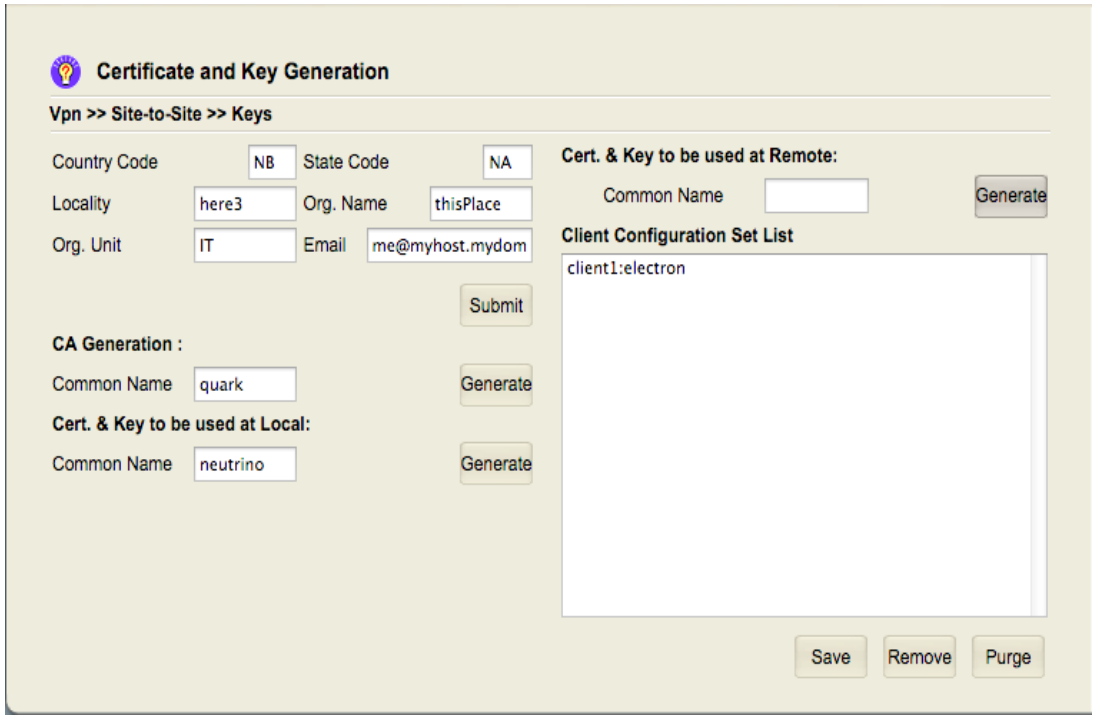


圖 117：點對點 VPN： 鍵生成範例

此側生成的客户端密鑰和證書應提交至另一側。可以通過選擇檔並點擊“保存”按鈕進行下載。

然後將其上傳至另一台機器的右側"**Vpn >> Site-to-Site >> Gateway Network Setting**"選單下。

以下兩個屏幕截圖是兩個 VPN 閘道設置的例子。

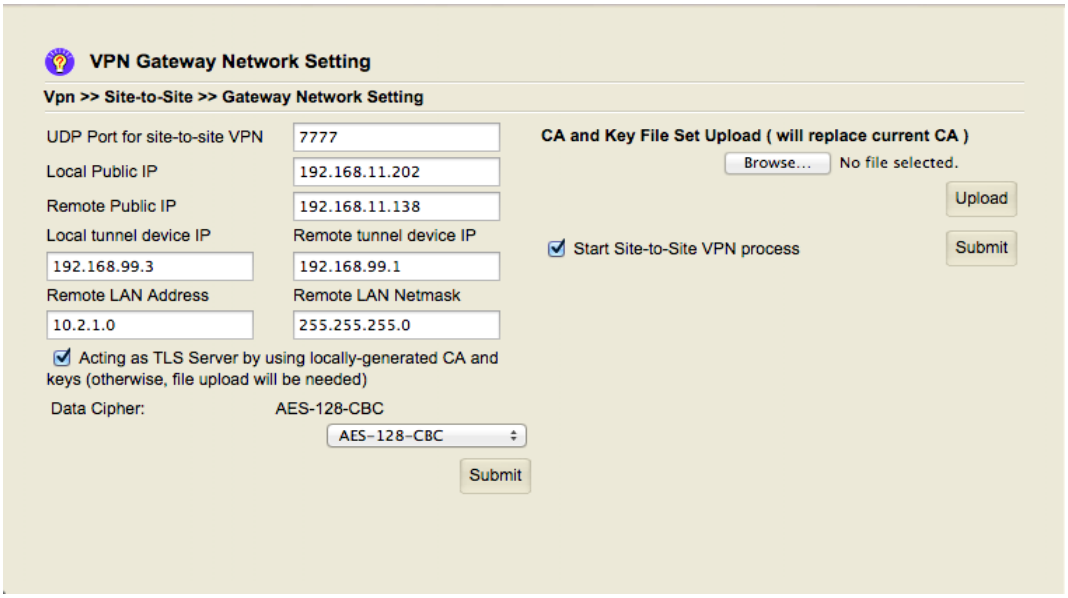


圖 118：VPN 閘道器作為 TLS 伺服器

本地主機的 IP 位址為 **192.168.11.202**，遠端主機的 IP 位址為 **192.168.11.138**。
在這條站點對站點 VPN 連線中：

- 本地通道介面（local tunnel interface）的 IP 為 **192.168.99.3**
- 遠端通道介面的 IP 為 **192.168.99.1**

在遠端網路中尚有一個子網 **10.2.1.0/255.255.255.0**。凡是送往 **10.2.1.0/24** 這個子網的封包，都應透過此 VPN 通道、由對應的 VPN 閘道轉送。

以下為遠端主機的設定範例（見圖 119）：

- 遠端主機的實體 IP 位址為 **192.168.11.138**
- 與之建立站點對站點 VPN 連線的對端 IP 為 **192.168.11.202**
- 在這一端，本地通道介面 IP 為 **192.168.99.1**，遠端通道介面 IP 為 **192.168.99.3**
- 另一端的私有子網為 **172.16.9.0/24**，送往該子網的封包應透過此 VPN 閘道轉送

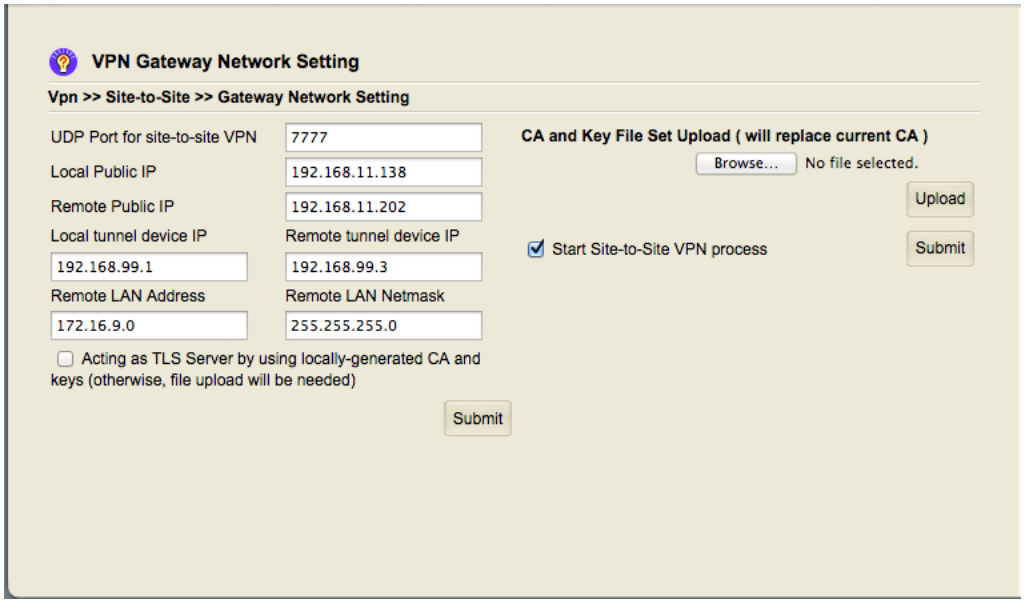


圖 119：遠端站點上的 VPN 閘道設定示意

以上範例使用的是實驗室環境的私有 IP。若要在互聯網上實際跨站點使用，必須將 **192.168.11.202** 與 **192.168.11.138** 這兩個位址改為各站對外的 **公眾 IP 位址**。

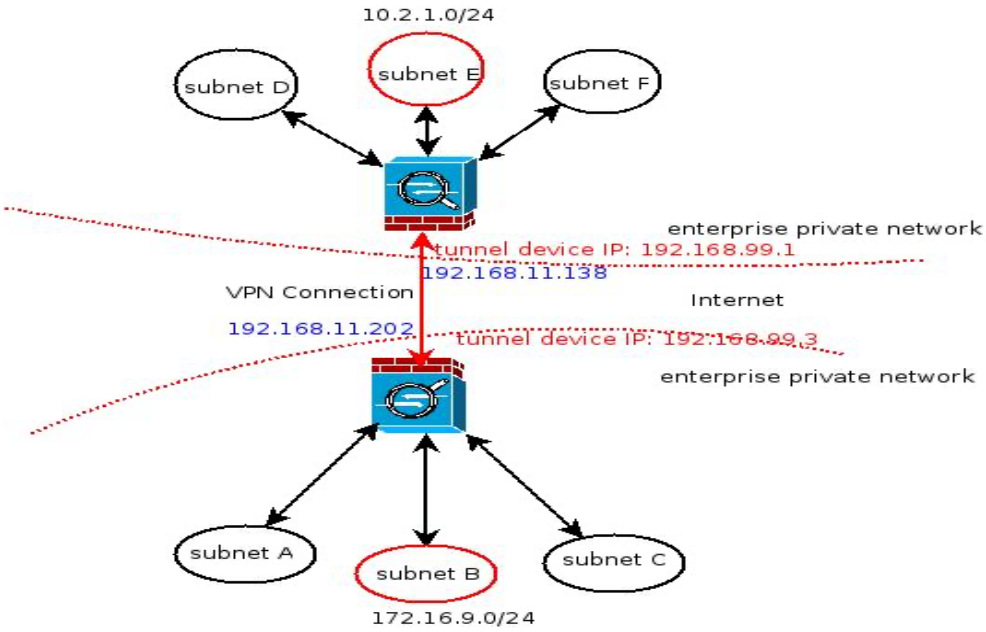
為了讓整體路由正常工作，位於這兩個子網 **172.16.9.0/24** 與 **10.2.1.0/24** 之中的主機，其預設閘道（或指向對方子網的更精確路由）都必須設成 **各自所在子網對應的 VPN 閘道 IP 位址**。

- 例如：圖中的 VPN 閘道在下方連到子網 **172.16.9.0/24**，其介面 IP 為 **172.16.9.1**。
  那麼此子網中的主機，只要想存取 **10.2.1.0/24**，就應將 **172.16.9.1** 設為預設閘道，或至少設為通往 10.2.1.0/24 的路由閘道。

至於其它子網的跨站連線需求，則需在兩端 VPN 閘道的**主路由表**中，額外加入指向各遠端子網的靜態路由，讓封包能正確選擇走 VPN 通道而不是一般 Internet 閘道。

多條站點對站點 VPN 實例

圖 120：站對站 VPN 範例設定



在某些部署情境下，一台 NFV 主機上可能需要同時建立多條站點對站點 VPN 連線（例如連往多個分點）。這時可以為每一條 VPN 連線建立一個獨立的「站點對站點 VPN 實例」，其設定畫面如圖 121 所示。

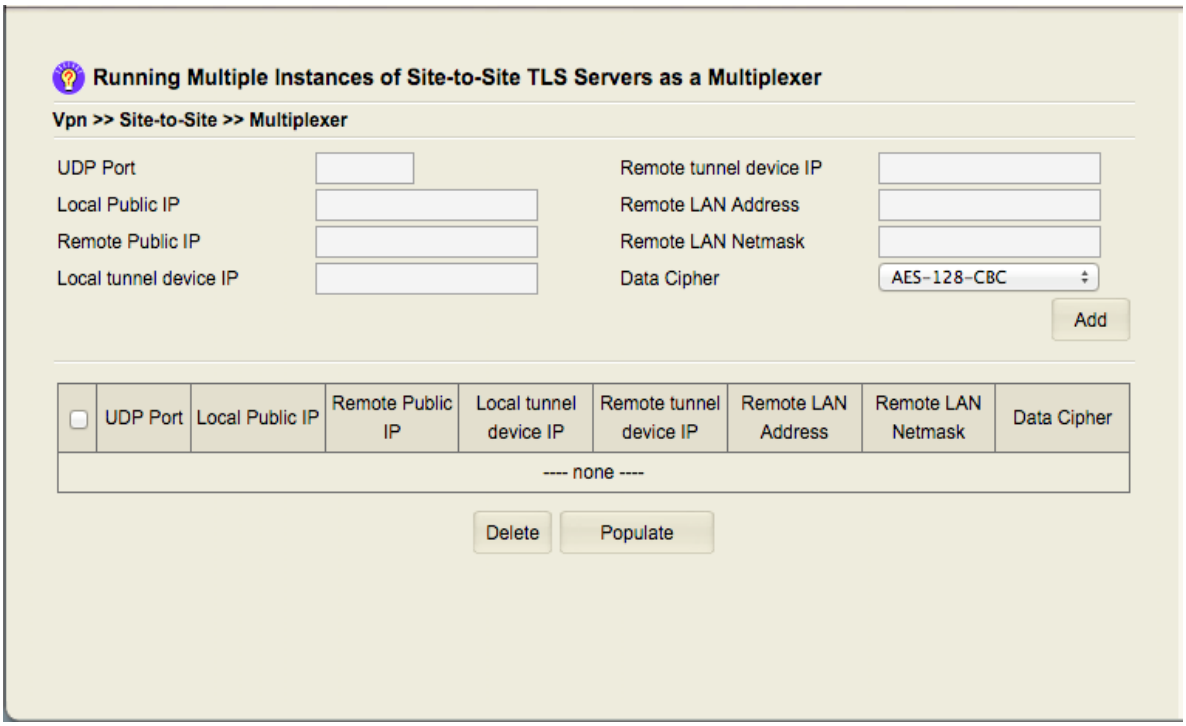


圖 121：站點對站點 VPN 多實例設定畫面

重點說明如下：

- 每一個實例都要設定成 **TLS Server**，但 **使用不同的 UDP 埠號** 進行監聽。
- 這些實例會共用同一組「伺服器私鑰與伺服器憑證」，其管理介面在 **Vpn >> Site-to-Site >> Keys**。
- 因為伺服器憑證是共用的，**同一台主機上建立多個站點對站點 VPN 實例時，不可混用 TLS Client 與 TLS Server 身分**。
    - 若在同一台機器上同時扮演 TLS Server 與 TLS Client，將導致 CA 與憑證管理混亂，甚至破壞原本的憑證信任鏈。

依照上述原則，只要妥善規劃好每一條 VPN 通道的：

1. 對外公眾 IP 與 UDP 埠號
2. 各自的本地／遠端通道介面 IP
3. 兩端內部子網的靜態路由設定

就能在一台 Azblink NFV 平台上，同時穩定地維護多條站點對站點 VPN 連線。

Site-to-site VPN in Bridging Mode (站對站 VPN 橋接模式)

站點間 VPN 若採「橋接模式」運作，會遇到一個常見問題：原本只會留在本地子網內的廣播或查詢流量，會被透過 VPN 通道帶到遠端站點。
例如：DHCP 用戶端尋找 DHCP 伺服器的封包、尋找 uPnP 裝置的廣播，本來都應該只存在於各自的 IP 子網之內；但在站點間 VPN 採用橋接模式後，這些封包會被視為同一個二層網路的一部分，自然就被轉送到對端。
這並不是網路協定本身的錯誤，而是因為「橋接模式的站點間 VPN，等同於把兩端網路併成一個單一 IP 子網」。採用這種模式時，管理者往往必須重新檢視整體網路規劃：

- 一個 IP 子網內應只保留一台 DHCP 伺服器；
- 必須事先檢查兩邊網路合併後，是否會產生 IP 衝突。

儘管如此，站點間 VPN 的橋接模式仍有其優點：

- IP 多播封包可以跨越 VPN 通道到達遠端站點。由於多數 ISP 並不在骨幹網路上轉送 IP 多播封包，一般情況下多播流量無法直接跨越 Internet；
- 一些路由協定（例如 OSPF）本身就依賴 IP 多播進行 Hello／發現與鄰居建立程序。
  在這類情境中，利用「站點間 VPN（橋接模式）」把兩端網路視為同一個二層網路，反而可以簡化多播部署與動態路由協定的實作。

在橋接模式下建立站點間 VPN 的概念

下面的示意圖用來說明，在橋接模式下建立站點間 VPN 的基本概念。
為了方便在實驗室中模擬，我們使用兩台主機，IP 分別為 192.168.11.202 與 192.168.11.138；若要實際跨 Internet 使用，這兩個位址應改為各自的公網 IP。
目標是建立一條橋接模式的站點間 VPN 通道，讓子網 172.16.9.0/24 可以同時「出現在」192.168.11.202 與 192.168.11.138 這兩端的背後。

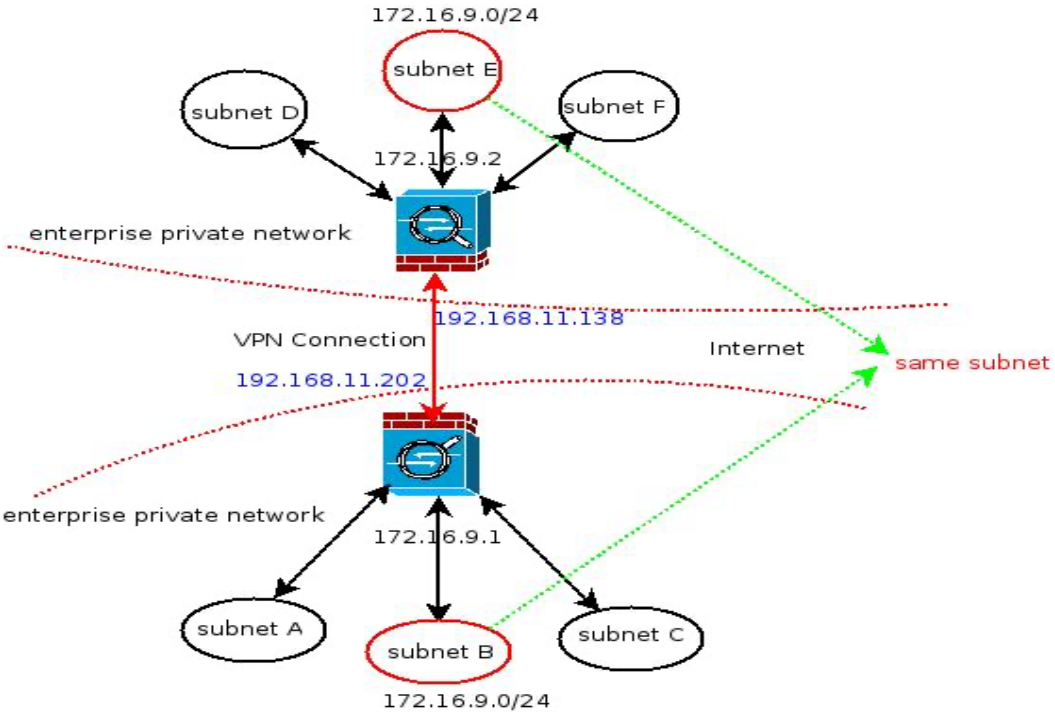


圖 122：橋接模式下的 Site-to-Site VPN 範例

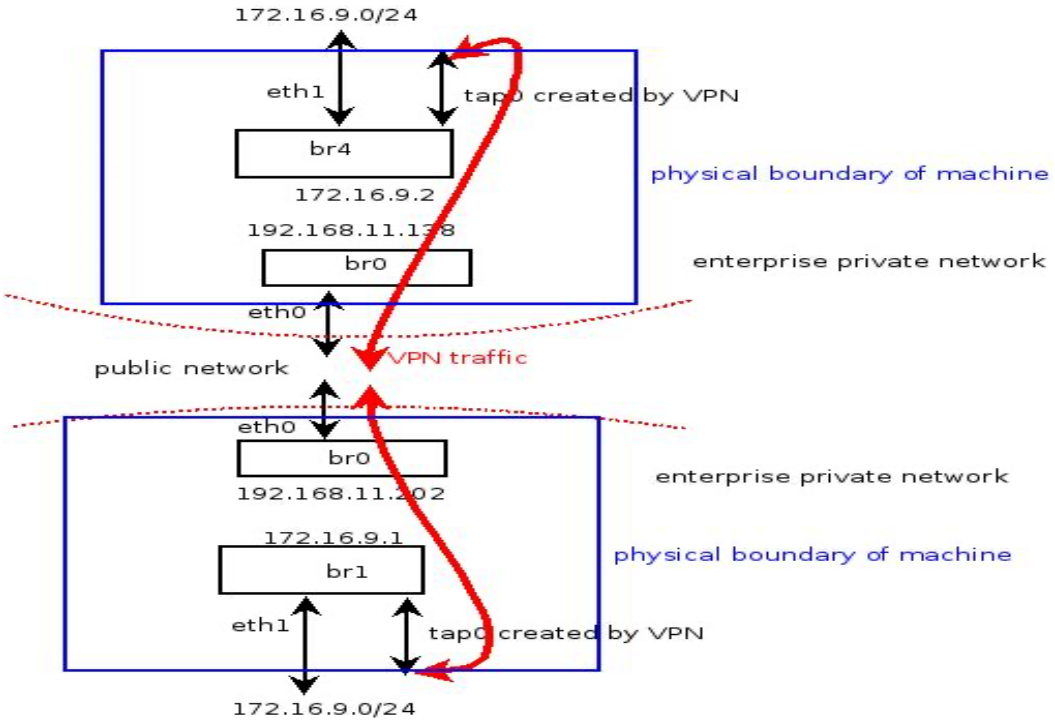


圖 123：橋接模式下 Site-to-Site VPN 的詳細操作說明

如前文所述，VPN 在橋接模式下，由 VPN 行程建立的虛擬網路介面本身不配置 IP 位址，它只負責收送以太網封包。若要與其他網路介面互通，這個由 VPN 建立的虛擬介面必須「加入某一個橋接器」，和其他介面一起交換二層封包。

在 Azblink NFV 基礎平台中，系統預先建立了多個虛擬橋接器，以及對應的實體乙太網介面。設定站點間 VPN 的橋接模式時，**不需要再新增額外的橋接器**；我們只要讓 VPN 行程所使用的虛擬介面（預設命名為 tap0）在每一端都加入既有的橋接器即可。
換句話說：

- 每一端的 tap0 都掛到本地的某個橋接器（例如 br1）上；
- 兩端的 VPN 行程再透過加密通道把這兩個橋接器「接在一起」，形成一個橫跨兩地的單一 IP 子網。

由於每一個橋接器本身在系統上都配置了 IP 位址，當兩個橋接器被視為同一子網時，**這兩個橋接器在同一子網內的 IP 位址不可重複**，必須分別設定為不同的位址。

憑證與金鑰的建立流程

站點間 VPN（橋接模式）所使用的 CA、伺服器金鑰與憑證、以及用於對端的用戶端金鑰與憑證，與「用戶端到站點 VPN」與「站點間 VPN（路由模式）」是獨立的一組，需要另外產生。這些憑證與金鑰可以透過 **「Vpn >> Bridge >> Server/Client」** 進行建立。

假設我們先在 IP 位址為 192.168.11.202 的這一端進行操作：

1. 在此 NFV 平台上建立或重設專屬的 VPN CA、伺服器金鑰與伺服器憑證。
2. 在同一畫面中產生「用於遠端站點」的用戶端金鑰與憑證。
3. 將這一組用戶端金鑰／憑證套件下載下來，安全地傳送到另一台主機（例如 192.168.11.138），讓對端匯入。

在這個架構中：

- 192.168.11.202 這台機器同時持有 CA 與伺服器金鑰／憑證，因此會扮演「VPN 橋接伺服器」；
- 192.168.11.138 則匯入先前下載的用戶端金鑰與憑證，扮演「VPN 橋接用戶端」。

將 tap0 加入橋接器

在兩端各自完成憑證設定後，接著必須讓 VPN 虛擬介面 tap0 加入對應的橋接器：

- 在 192.168.11.202 這一端，將 tap0 加入橋接器（例如 br1），與實體介面（如 eth1）同時掛在 br1 上。
- 在 192.168.11.138 那一端，也同樣讓 tap0 加入當地的橋接器。

這樣一來，當站點間 VPN（橋接模式）啟動後，兩端的 br1 就會藉由 tap0 與 VPN 通道被視同一個二層廣播域。

圖 124：CA、鍵和憑證用於橋接模式的點對點 VPN。

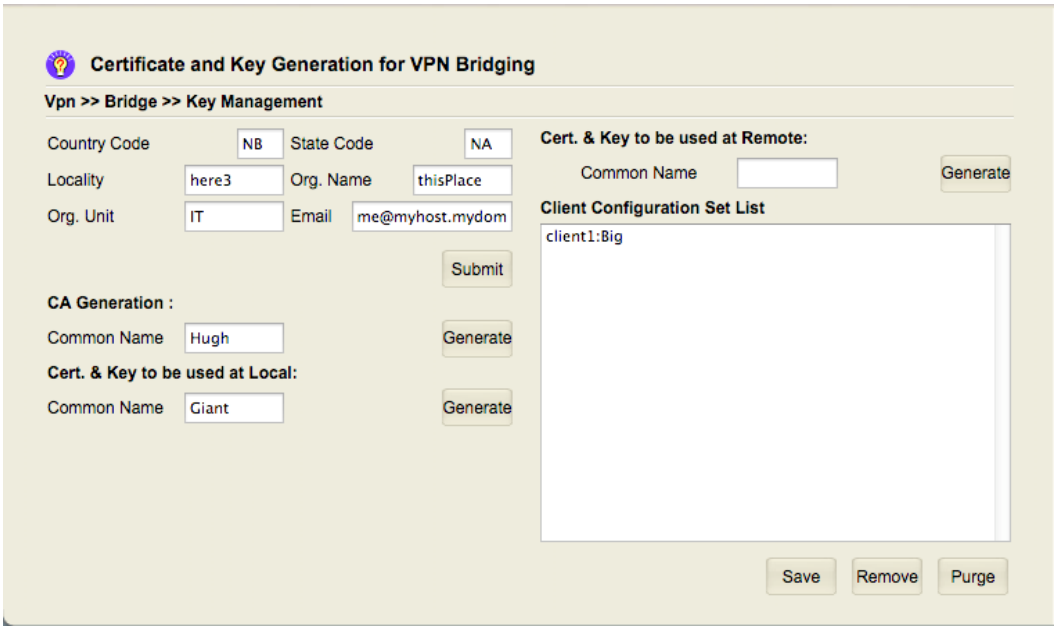


插圖 125：VPN 橋接伺服器示範設定

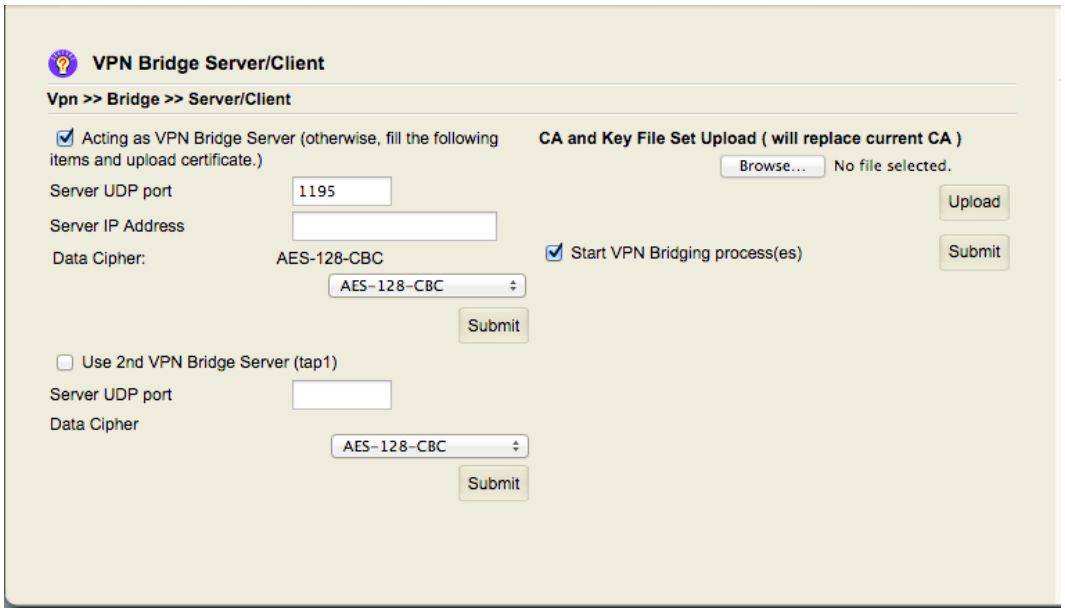


圖 126：VPN 網路設備連接橋接器（伺服器端）範例

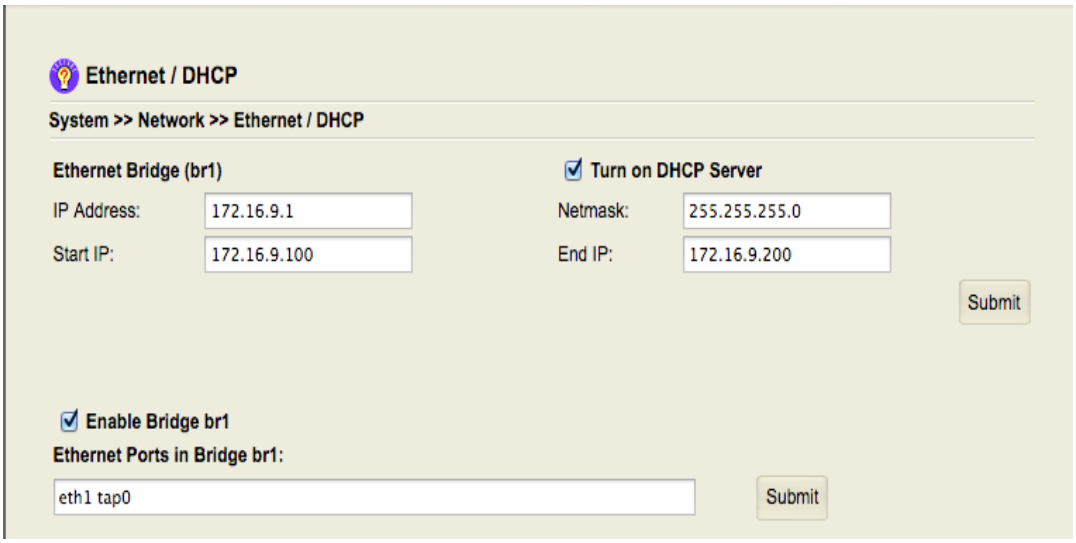
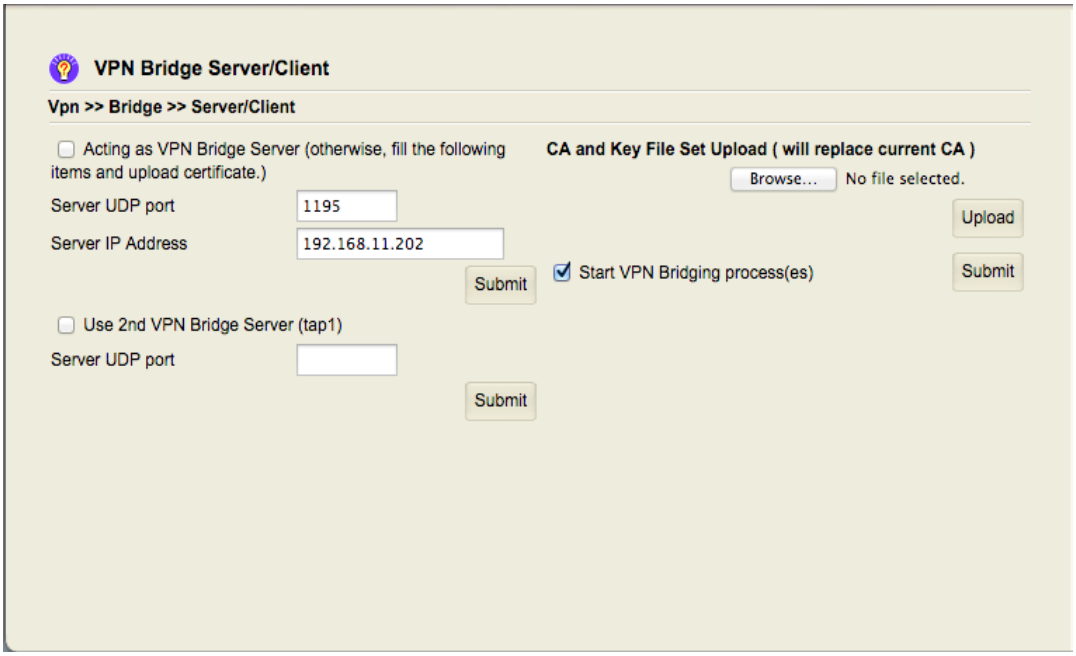
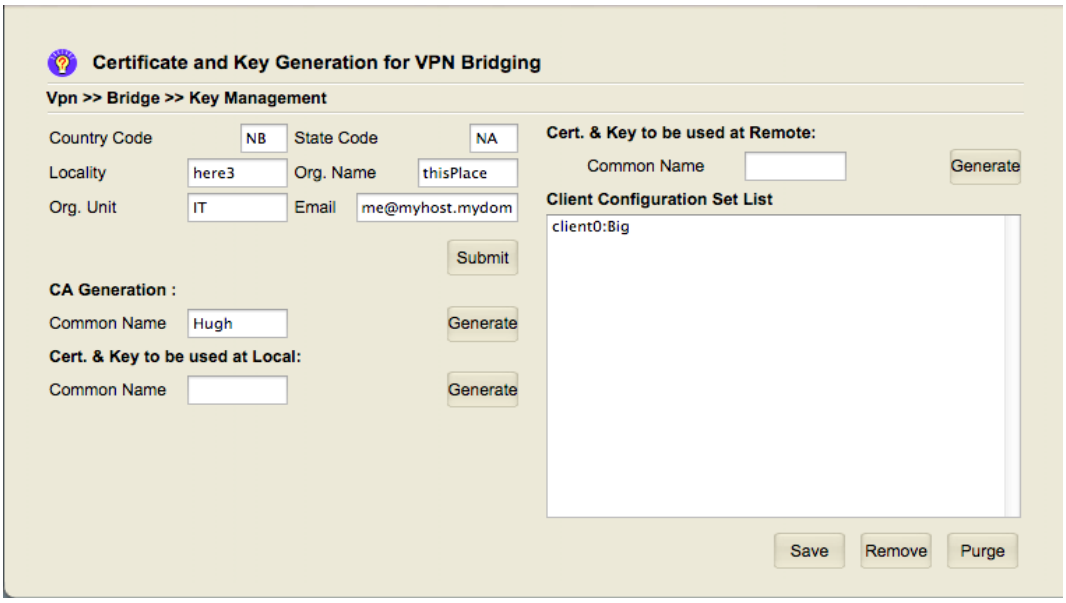


圖 127：站對站 VPN 在橋接模式下的客戶端範例設定



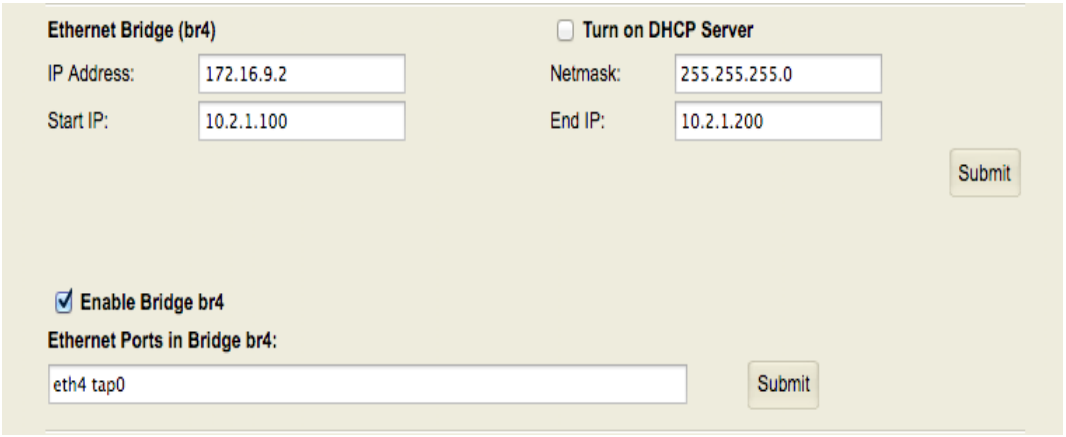
在另一側（機器“192.168.11.132”），我們上傳客户端密鑰和證書，在“Vpn >> Bridge >> Server/Client”選單中。提交客戶端密鑰和證書後，“**Vpn >> Bridge >> Key Management**”屏幕將顯示如下：

圖 128: 客户端證書顯示



請別忘記讓“tap0”加入橋接器。

圖 129：VPN 網路設備連接橋接 (客戶端部分) 的範例



在上述範例中，站點間 VPN 使用的是 UDP 埠 **1195**。請務必在「邊界控制」中允許對應的 UDP 埠通過，否則 VPN 無法建立連線。
VPN 設定變更後，通常需要重新啟動兩端的 VPN 服務，或重新開機，使設定完整生效。

在基礎平台上，我們已经創建了虛擬橋接器以及物理網路接口。對於此操作，不需要創建額外的橋接器。我們使用“tap0”用於由 VPN 進程在橋接模式下創建的網路設備。因此，想法只是讓“tap0”在每個站點加入橋接器，並且两端的 VPN 進程會將這两個橋接器在同一 IP 子網下連接起

來。由於我們為每個橋接器設置了 IP 地址，這兩個在同一子網下的橋接器應該設置為不同的 IP 地址。

這大致是關於站點至站點 VPN 在橋接模式下設置的想法。我們首先創建 CA、服務器密鑰和證書，以及客户端密鑰和證書。 它們同樣獨立於客户端至站點 VPN 和站點至站點 VPN 在路由模式下設置。 它們可以通過“Vpn >> Bridge >> Server/Client” 創建。

關於 tapN 介面的命名與啟動順序

需要特別注意的是，NFV 平台在為虛擬主機建立網路介面時，也會使用 tapN 這類名稱（例如 tap0、tap1、tap2…）。由於虛擬主機的 tap 介面是在啟動虛擬機時動態產生，因此在「尚未啟用站點間 VPN（橋接模式）」之前，有時 tap0 這個名稱可能已被某台虛擬主機占用。

為避免名稱衝突造成設定錯誤，建議遵循以下順序：

1. 先啟用「站點間 VPN（橋接模式）」設定，並讓系統啟動對應的 VPN 虛擬介面（例如 tap0），再將它加入橋接器。
2. 確認 VPN 連線正常後，再逐一啟動其他虛擬主機，讓它們使用後續的 tapN 名稱。

只要遵守這個順序，就能避免 tap0 被虛擬主機搶先使用而導致 VPN 無法正常掛入橋接器的情況，整體站點間橋接 VPN 也會更穩定、容易維護。

第 五 章 動態路由 (Dynamic Routing)

深入淺出：理解 IP 路由 (IP Routing)

在這份文件中，「路由」專指 **IP 封包的傳輸路徑選擇**。簡單來說，路由器就像是網路世界的交通指揮中心，決定資料要走哪條路才能到達目的地。

我們將路由方式分為兩大類：**靜態 (人工手動)** 與 **動態 (自動溝通)**。

1. 秒懂概念：地圖與導航的比喻

為了快速理解兩者的差異，我們可以想像你在開車：

- 靜態路由 (Static Routing) 就像是「紙本地圖」

  出發前你已經規劃好路線。如果路上遇到修路或塞車，地圖不會變，你必須自己手動重新查路，否則就會卡在路上。

- 動態路由 (Dynamic Routing) 就像是「Google Maps 導航」

  導航系統會隨時接收路況資訊。如果前方修路，它會自動幫你計算並切換到另一條順暢的路，你不需要操心。

---

2. 靜態路由 (Static Routing)

- **定義：** 由網路管理者**手動 (Manual)** 一筆一筆輸入到路由器中的路徑規則。
- **運作方式：**
  - **直接連線：** 插上線、設好 IP，路由器就知道怎麼去隔壁。
  - **跨網段：** 若要去較遠的網路，你必須明確告訴路由器：「要去 A 網段，請走 B 閘道」。
  - **預設閘道 (Default Gateway)：** 這是一種特殊的靜態路由。當路由器不知道信要把去哪裡時，就會全部丟給預設閘道處理（類似「不知道丟哪裡的信，全部丟總局」的概念）。
- **優點：** 行為固定好預測、容易除錯 (Debug)。
- **缺點：** 維護很累。在大公司裡，如果改了一個網段，可能要手動去更新 10 台路由器的設定。

3. 動態路由 (Dynamic Routing)

- **定義：** 路由器之間會使用協定互相**交換資訊**，自動學習路徑。
- **運作方式：** 路由器會告訴鄰居：「我連接了哪些網路」。鄰居收到後會更新自己的地圖。如果某條線路斷了，它們會立刻通知彼此：「這條路不通，請改走另一條」。
- **優點：** 省力、自動化。網路擴充或故障時，路徑會自動修復。
- **缺點：** 初始設定的觀念較複雜，且路由器負載會稍微高一點（因為要一直溝通）。

---

4. 總結比較表

特徵	靜態路由 (Static)	動態路由 (Dynamic)
控制權	手動 (管理者全權決定)	自動 (路由協定決定)
維護難度	高 (網路變更需逐台修改)	低 (路由器會自動更新)
適用場景	小型、架構簡單的網路	大型、架構複雜或常變動的網路
遇到故障	無法自動應變，需人工介入	自動尋找替代路徑
生活比喻	紙本地圖	GPS 導航

---

5. 實務建議：該選哪一種？

根據文件中的建議：

- **情境 A (簡單)：** 如果您的環境只有約 **12 個子網段**，且沒有串接複雜的外部路由器，使用**靜態路由** 最簡單省事。
- **情境 B (擴充)：** 如果網路會跨越多台路由器，或者未來有擴充需求，建議採用或搭配 **動態路由**，維護起來會輕鬆很多。

---

6. 特殊情況：靜態路由做不到的事 (IP 多播)

有一種情況是靜態路由完全無法處理的，那就是 **IP 多播 (IP Multicast)**。

- **單播 (Unicast)：** 就像講電話 (1 對 1)。靜態路由可以處理。
- **多播 (Multicast)：** 就像訂閱雜誌或加入群組 (1 對多)。

為什麼靜態不行？

因為使用者會隨時動態加入或退出某個多播群組。靜態路由是寫死的，它無法知道「現在這一秒，誰加入群組了？誰又退出了？」。只有動態路由能即時追蹤這些成員的變化，把資料準確送到有加入的人手上。

Azblink *NFV 平台支援的動態路由協定*

在 Azblink NFV 平台上，我們針對 **IPv4** 環境提供了完整的路由解決方案，涵蓋了「單播 (Unicast)」與「多播 (Multicast)」兩大類需求。

1. 單播路由協定 (Unicast Protocols)

這是最常見的傳輸方式（點對點傳輸）。我們提供兩種協定，您可以根據網路規模來選擇：

協定名稱	全名	適用場景	特色說明
RIPv2	Routing Information Protocol, v2	中小型環境	**設定簡單、門檻低。** 適合結構單純、路由器數量較少的網路。
OSPF	Open Shortest Path First	中大型環境 或 **複雜拓樸**	**智慧型、擴充性強。** 屬於「鏈路狀態 (Link-state)」協定，路由器能掌握完整的網路地圖，算出最短路徑，適合企業級網路。

2. 多播路由協定 (Multicast Protocols)

如果您有「一對多」的串流傳輸需求（例如視訊會議、網路電視），則需要使用此協定。

- **PIM (Protocol Independent Multicast)**
  - **用途：** 用來在不同的子網段之間轉送多播流量。
  - **運作關鍵：** PIM 自己不負責找路，它必須**依賴**底層的單播協定（如上面的 OSPF）來運作。
  - *白話：OSPF 負責把路鋪好，PIM 負責在這條路上面運送多播的資料。*

---

3. 設計建議：打造最佳路由架構

「靜態」與「動態」並非二選一，而是可以互補的。

作為管理者，您可以根據實際需求靈活搭配：

- 在核心或複雜區域使用 **動態路由 (如 OSPF)** 以確保彈性與備援。
- 在邊緣或簡單區域使用 **靜態路由** 以保持穩定與簡單。

透過這種混合搭配，您就能為 Azblink NFV 平台建立一個既**穩定 (Stable)** 又**容易維護 (Maintainable)** 的網路架構。

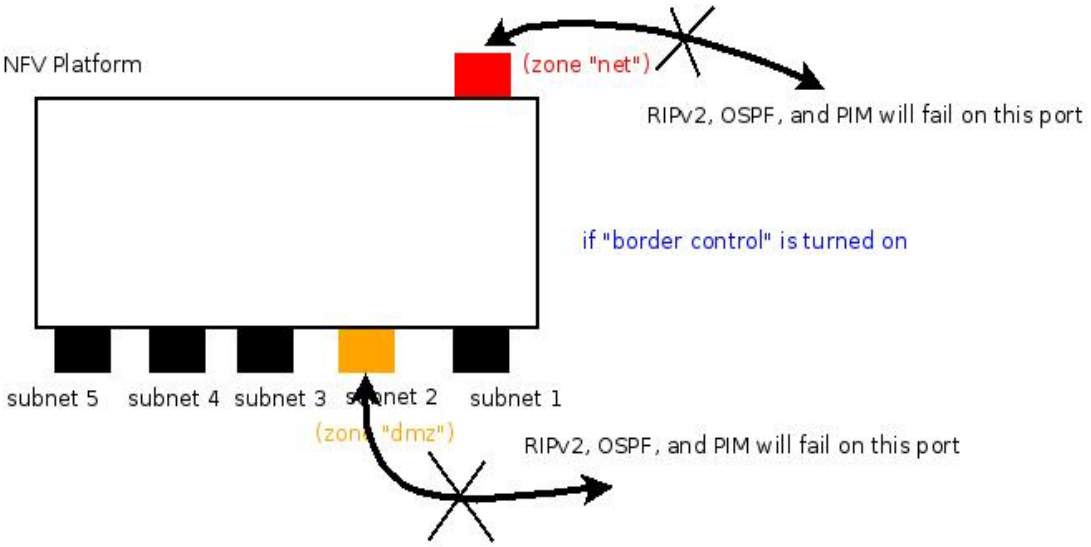


圖 130：邊境控制與動態路由

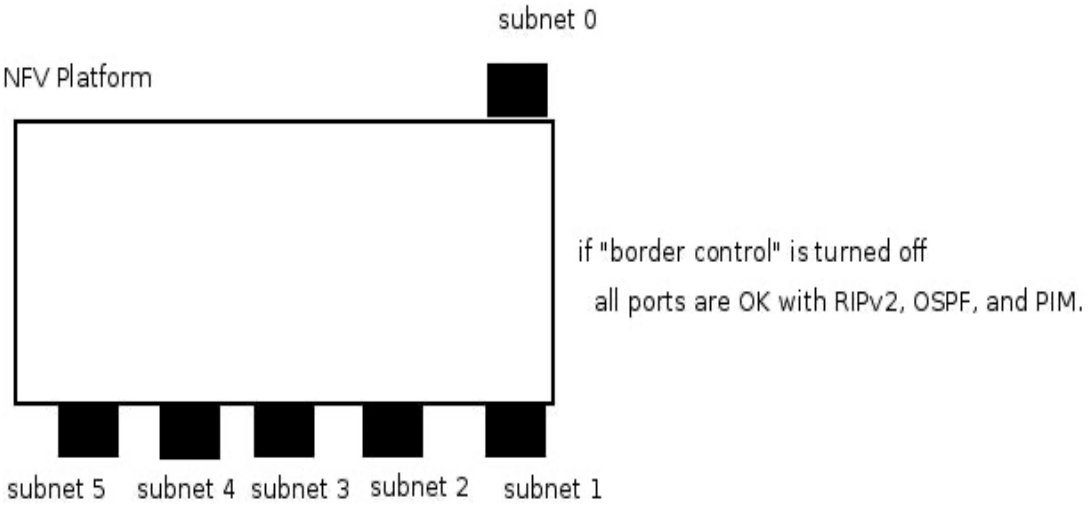


圖 131：關閉邊界控制

邊界控制與動態路由的注意事項

本章節說明如何在基礎平台上管理與邊界控制 (Border Control) 相關的網路操作。請務必了解防火牆規則與動態路由之間的交互影響，以免設定錯誤導致網路不通。

1. 啟用/停用邊界控制

您可以透過以下路徑來啟用或停用邊界控制功能：

路徑： Border >> Connection >> Port Forwarding （位於頁面頂部的設定）

2. 重要警告：防火牆規則對路由的影響

在啟用邊界控制的預設高安全性情境下，防火牆區域 (Zone) 間的流量規則如下：

- **Net (外部網路) 到 fw (防火牆本機)：** 流量會被丟棄 (Drop)。
- **DMZ (非軍事區) 到 fw (防火牆本機)：** 流量也會被丟棄。

這將導致動態路由無法運作！ RIPv2、OSPF 與 PIM 等協定，都需要與防火牆本機進行通訊以交換路由資訊或群組成員狀態。如果防火牆阻擋了流入 fw 區域的流量，這些協定將無法建立鄰居關係，導致路由失敗。 因此，若您打算使用動態路由，請務必調整邊界控制策略以允許相關流量通過。

---

3. 技術細節：協定運作機制與硬體需求

為了確保路由協定正常運作，請注意以下底層通訊機制：

- **多播 (Multicast) 的依賴：**
  - **RIPv2**、**OSPFv2** (IPv4) 與 **OSPFv3** (IPv6) 皆使用 **IP 多播** 來搜尋並發現網路上的其他路由器。
  - **RIPv1** 則使用 **廣播 (Broadcast)**。
- **IGMP 支援：** 由於上述協定依賴多播在子網段內運作，請確保該網段內的網路設備（如交換機 Switch）皆支援並正確設定 **IGMP (Internet Group Management Protocol)**。

---

4. OSPF 區域 (Area) 概念補充

若您選擇使用 OSPF 協定，請參考以下配置概念：

- **角色分工：**

  - **OSPF / RIP：** 用於路由器之間的溝通（交換路徑表）。
  - **PIM：** 用於跨 IP 子網段傳送多播封包（依賴 OSPF/RIP 建立的路徑）。
- **OSPF 區域 ID (Area ID)：** OSPF 採用分層架構，核心概念為「區域 ID」。
  - **Area 0 (骨幹區域)：** 這是 OSPF 的核心網路，所有其他區域都必須與 Area 0 相連。
  - **數值格式：** ID 範圍從 0 到 2^32-1，也可以寫成 IP 格式 (例如 0.0.0.0 代表 Area 0，1.1.1.1 代表 Area 1)。
  - **ABR (區域邊界路由器)：** 當一台路由器同時連接 Area 0 和另一個區域時，這台路由器就被稱為 **ABR (Area Border Router)**，負責在區域間傳遞訊息。

OSPF 和 RIP 在路由器之間使用；PIM 用於跨 IP 子網傳輸多播包。

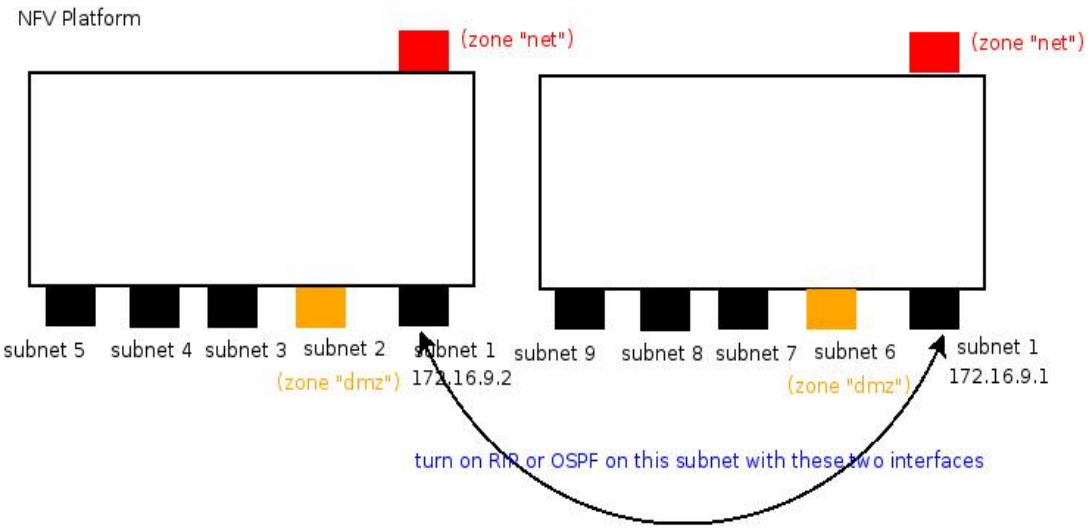


圖 132：將兩台機器連用

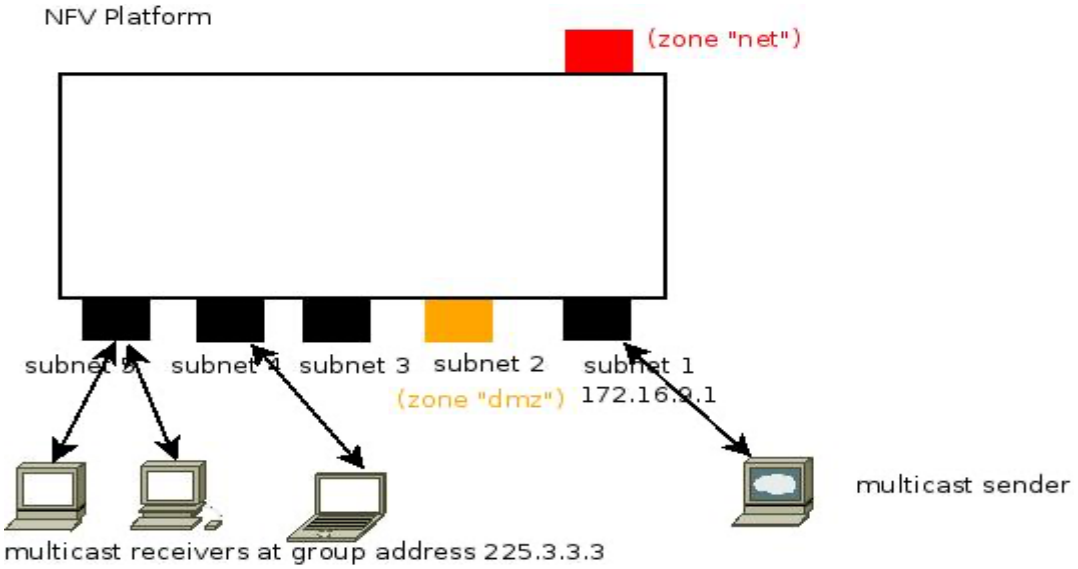


圖 133：使用 PIM 的情景

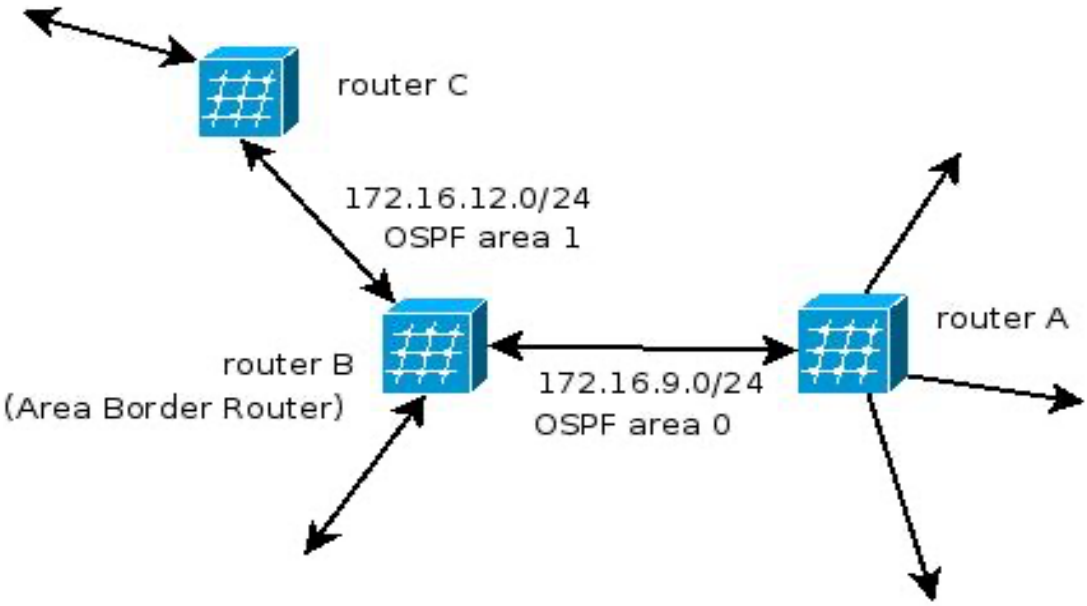


圖 134：OSPF ABR (區域邊界路由器)

RIPv2 (Router Information Protocol, version 2)

RIPv2 設定實戰範例

本範例將示範如何透過 **RIPv2** 協定連接兩台機器（機器 A 與 機器 B），使其自動交換路由表，省去手動逐一新增路由條目的麻煩。

1. 網路環境架構 (Topology)

首先，請確認兩台機器之間的實體連接與 IP 設定如下圖所示：

- **機器 A (Machine A)**
  - 介面：**br0**
  - IP 位址：172.16.9.1
- **機器 B (Machine B)**
  - 介面：**br4**
  - IP 位址：172.16.9.2

說明： 兩台機器的介面（br0 與 br4）必須連接在同一個子網段內，才能建立溝通。

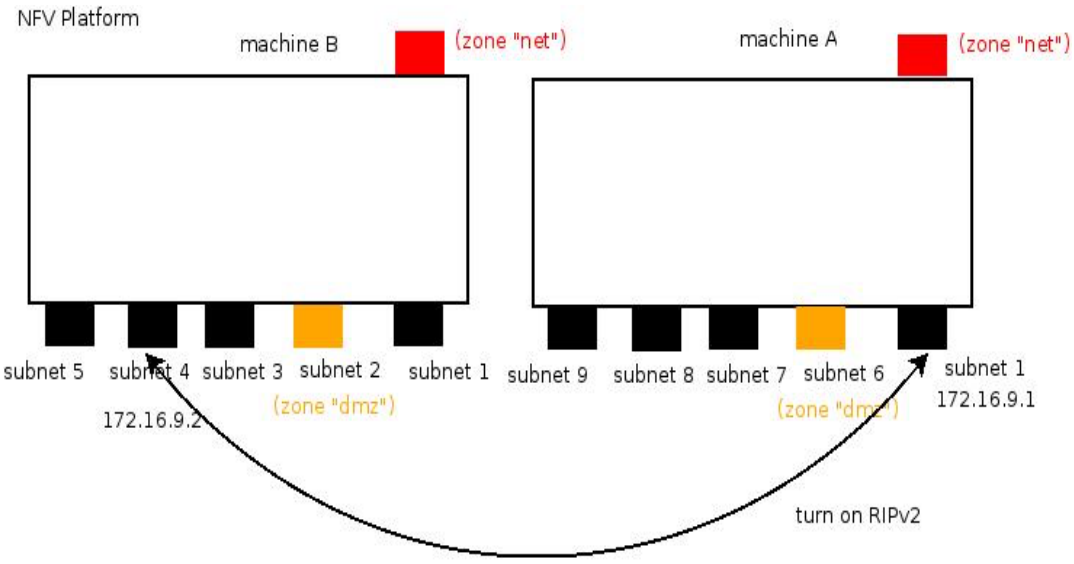


圖 135：RIPv2 範例

2. 設定步驟 (Configuration)

請依序在兩台機器上完成以下設定：

步驟一：指定運作網段 我們需要告訴 RIP 協定要在哪個網段發送更新資訊。

- **路徑：** System >> Network >> RIP
- **操作：**在此頁面指定上述的連接子網段。

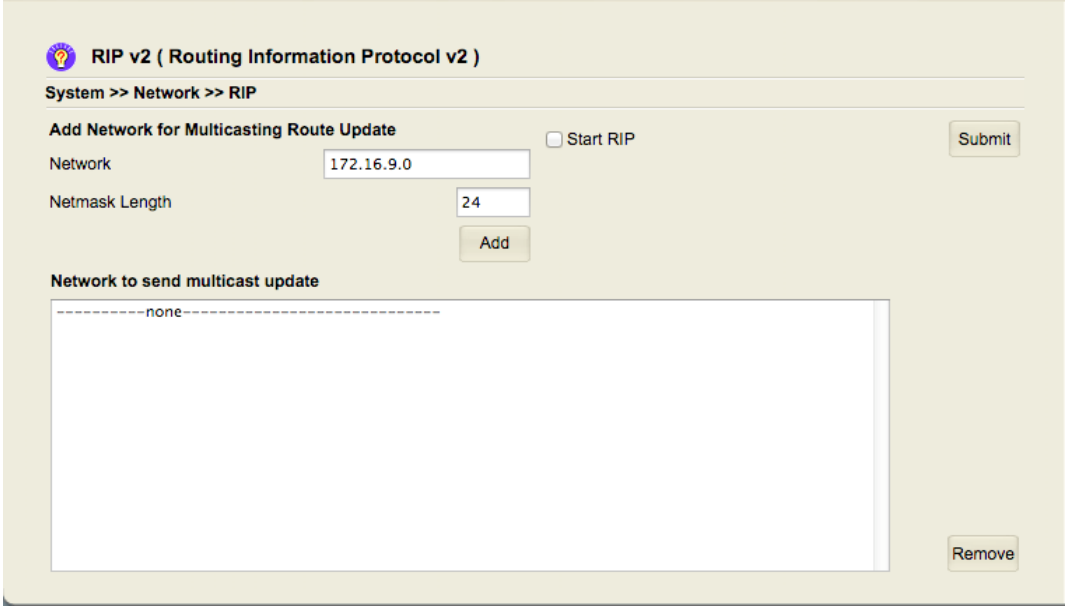


圖 136：RIPv2 發送多播更新的子網

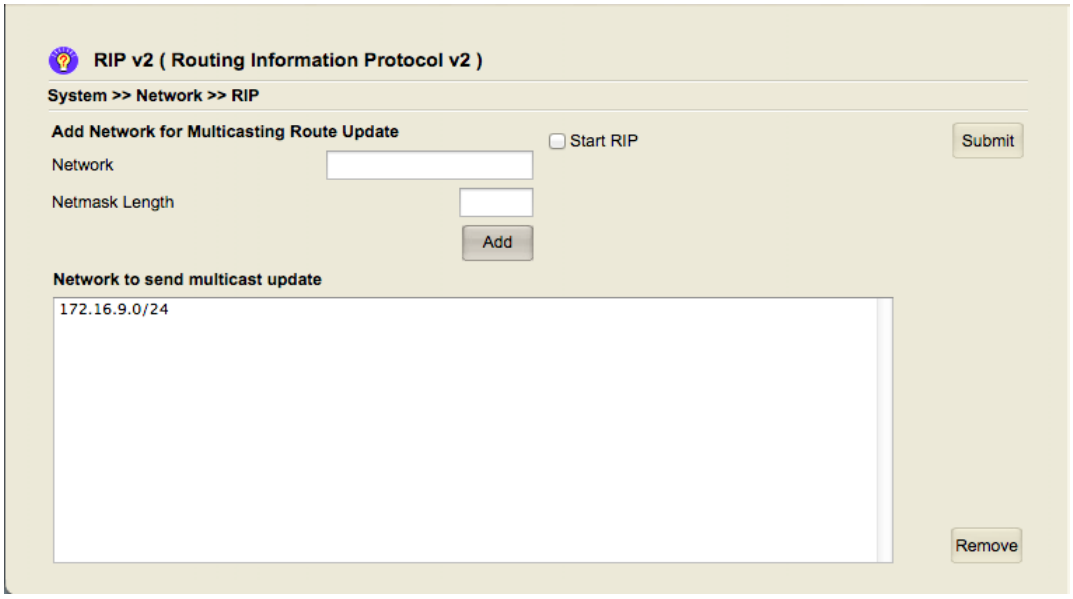


圖 137：發送多播更新的子網列表

步驟二：設定認證密鑰 (Authentication) 為了安全起見並允許雙方交換資訊，必須設定一組共用的認證密鑰。

- **路徑：** System >> Network >> RIP Auth
- **操作：** 輸入認證密鑰（Key/Password）。
  - *注意：機器 A 與機器 B 的密鑰必須完全相同。*

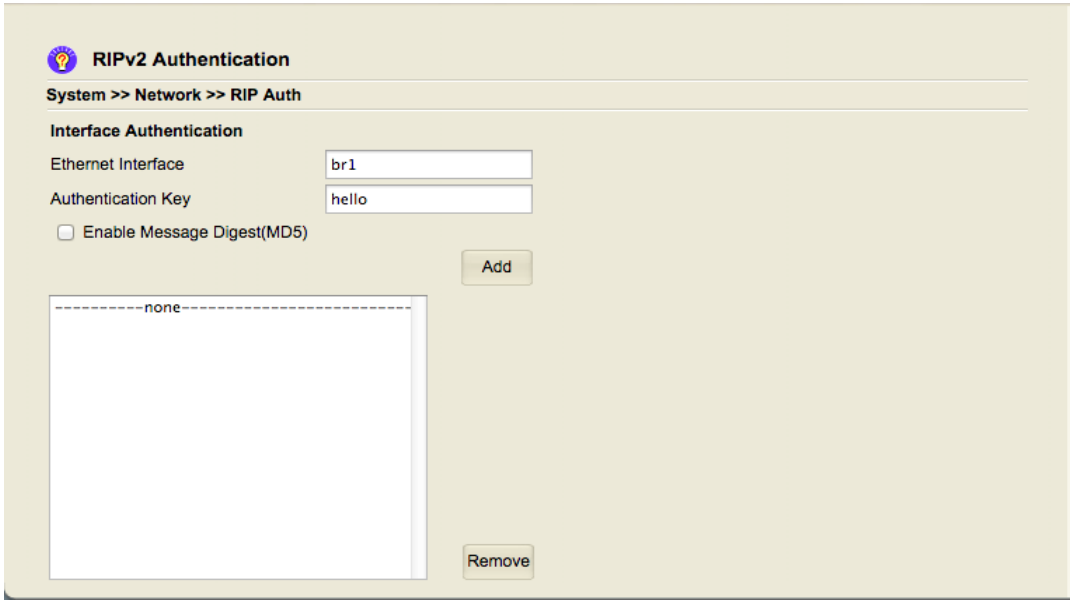


圖 138：設置 RIPv2 身份驗證書密鑰

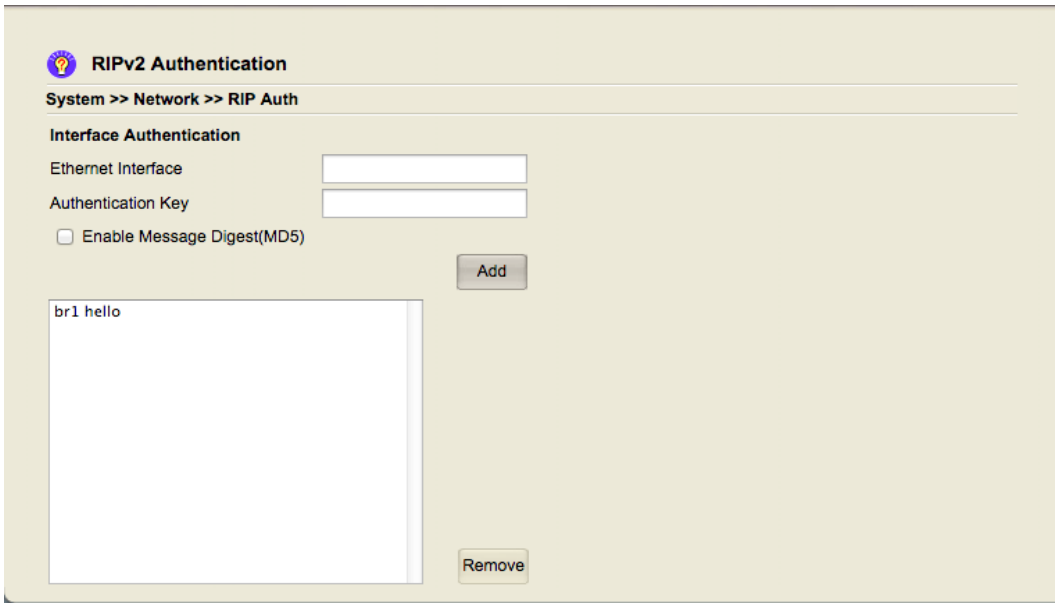


圖 139：身份驗證書密鑰列表

在另一側（機器 B），我們也同樣有類似的配置。 我們按如下方式設置了子網和認證密鑰：

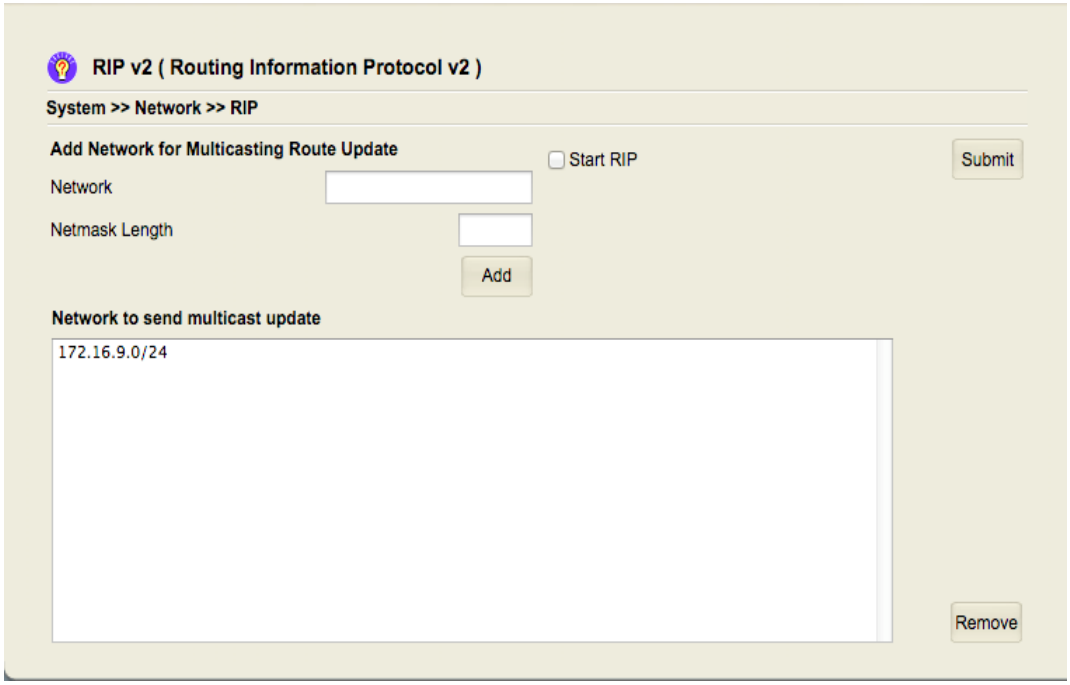


圖 140：在另一台機器上進行多播更新的子網

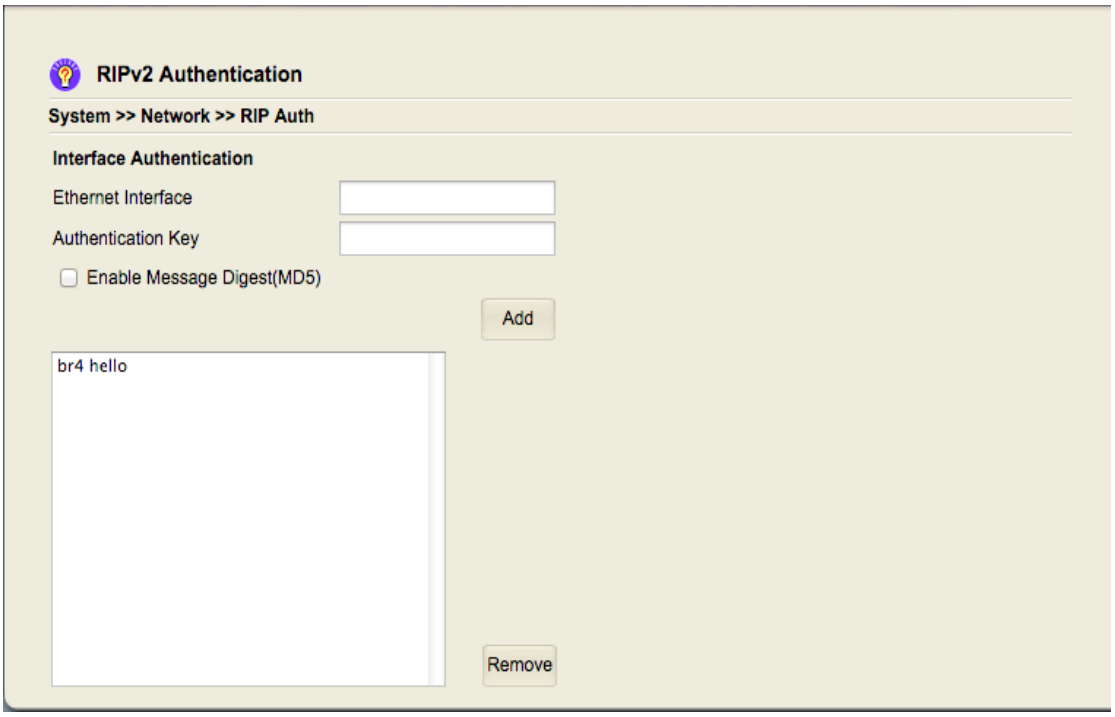


圖 141：認證書密鑰列表

步驟三：啟用路由程序 完成網段與認證設定後，必須正式啟用服務。

- **路徑：** System >> Network >> RIP
- **操作：** 勾選啟用 (Enable) 的複選框。

---

3. 驗證結果 (Verification)

當雙方（機器 A 與 機器 B）都完成上述配置（包含在介面 br0 與 br4 上正確套用密鑰）後，系統即開始自動交換路由資訊。

您可以透過以下方式檢查是否成功：

- **路徑：** System >> Network >> Static Routing (或路由表頁面)
- **預期結果：** 您應該會看到對方的子網段已經自動出現在本機的路由表中。

以下展示**機器 A** 成功學習到路由後的畫面：

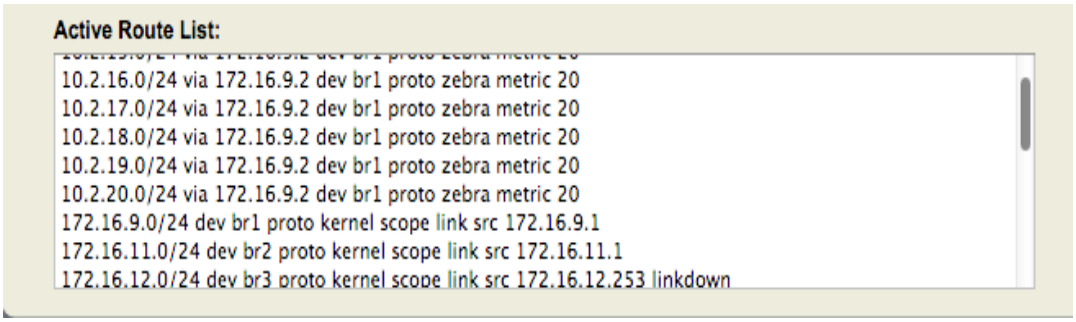


圖 142：啟動 RIPv2 後路由表內容

OSPF (Open Shortest Path First)

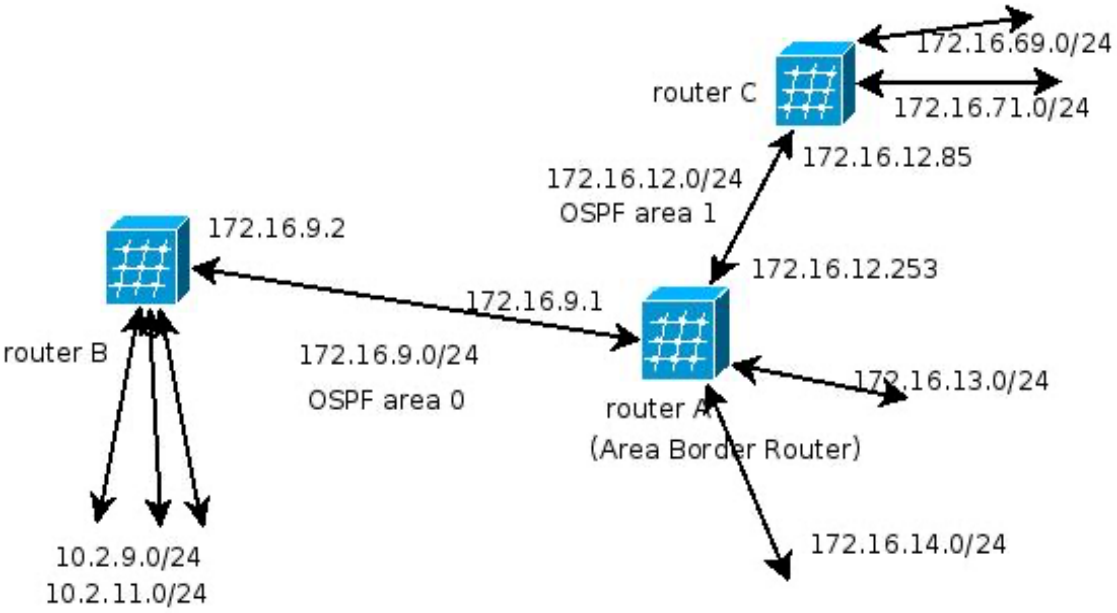


圖 143：OSPF 設置示例

本指南描述了如何使用三個基礎平台實例（路由器 A、B、和 C）來設置 **OSPF (開放最短路徑優先)** 路由。

路由器和子網配置

我們將使用三個路由器（A、B、C），每個路由器配置了以下 IP 子網：

- **路由器 A 的子網：**
  - **172.16.9.0/24,** 172.16.11.0/24, **172.16.12.0/24,** 172.16.13.0/24
  - 172.16.14.0/24, 172.16.15.0/24, 172.16.16.0/24, 172.16.17.0/24
  - 172.16.18.0/24, 172.16.19.0/24, 172.16.20.0/24
- **路由器 B 的子網：**
  - **172.16.9.0/24,** 10.2.9.0/24, 10.2.11.0/24, 10.2.12.0/24,
  - 10.2.14.0/24, 10.2.15.0/24, 10.2.16.0/24, 10.2.17.0/24
  - 10.2.18.0/24, 10.2.19.0/24, 10.2.20.0/24 [6]
- **路由器 C 的子網：**
  - **172.16.12.0/24,** 172.16.17.0/24, 172.16.69.0/24, 172.16.72.0/24
  - 172.16.73.0/24, 172.16.74.0/24, 172.16.75.0/24, 172.16.76.0/24
  - 172.16.77.0/24, 172.16.78.0/24, 172.16.79.0/24, 172.16.80.0/24

OSPF 區域劃分

OSPF 區域 ID 範圍是 0 到 2^{32}-1，其中 **0** 代表核心網路（骨幹網）。

- **區域 0 (核心網)：** 路由器 A 和 B 彼此相鄰，通過 **172.16.9.0/24** 子網連接。
- **區域 1：** 路由器 A 和 C 彼此相鄰，通過 **172.16.12.0/24** 子網連接。

! 重要提示： 請不要對 **net** 區域和 **dmz** 區域的子網進行 OSPF 更新，因為 OSPF 更新將在這些區域中無法工作。

圖 144：路由器 A 的子網和區域 ID 設置

路由器 A 的 OSPF 配置

在路由器 A 上，您需要啟用以下兩個相鄰子網的 OSPF 更新：

1. **172.16.9.0/24** (分配給 **區域 0**)
2. **172.16.12.0/24** (分配給 **區域 1**)

您可以在 **System >> Network >> OSPF** 介面中指定這些子網並輸入其對應的區域 ID 。

。

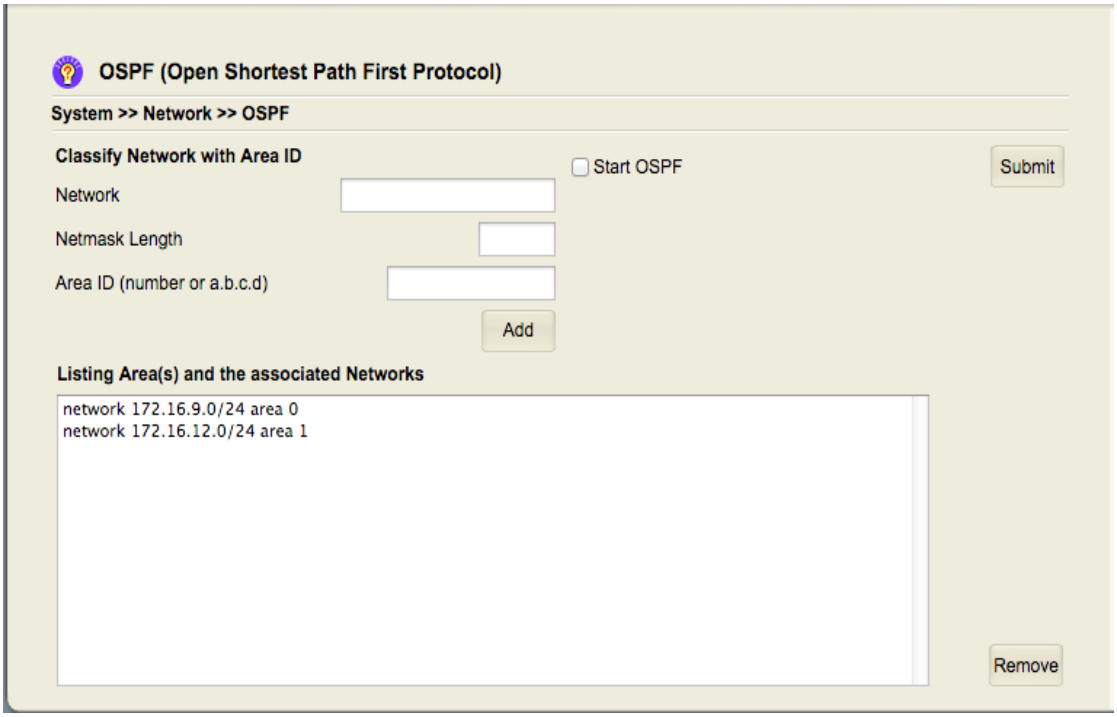


圖 145：路由器 A 的子網列表

身份驗證與密鑰

OSPF 路由器之間需要進行身份驗證，以確保路由信息的安全交換。

- 每個介面都需要一個身份驗證密鑰 (Authentication Key) 。
- 此密鑰必須與相鄰路由器上的設置相匹配。
- 如果需要對整個區域進行身份驗證，只需輸入相關的區域 ID 並點擊添加。
- **注意：** 設置路由器 B 和 C 時，其身份驗證密鑰必須與路由器 A 上設置的密鑰匹配。

。

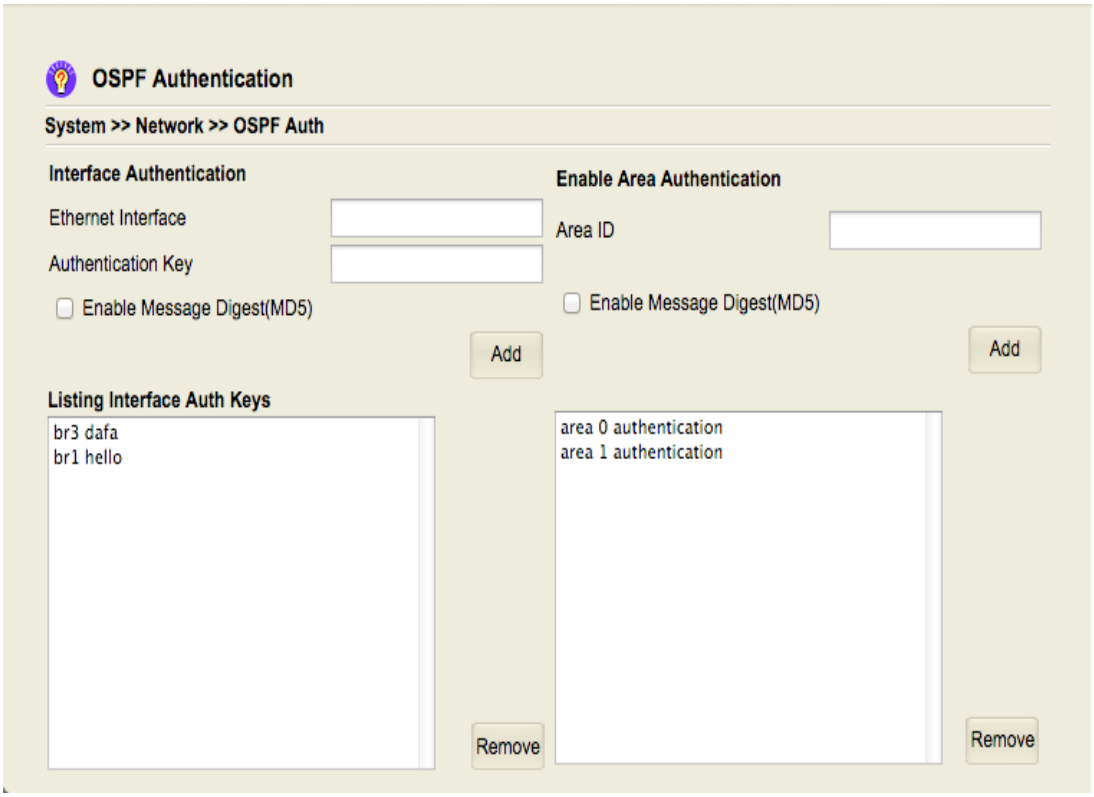


圖 146：OSPF 身份驗證書設置（路由器 A）

網路介面和路由器 ID

- IP 地址是在基礎平台上的橋接設備（如 **br0, br1, br11** 等）上設置的，這些橋接設備在 OSPF 配置中被用作乙太網介面。
- 這些介面的 IP 地址在路由器中必須是唯一的。
- 其中一些 IP 地址將被用作每個路由器的 **路由器 ID (Router ID)**，它用於計算路由路徑 。
- 即使您認為某些子網未被使用，您仍應確保每個網路介面都具有唯一的 IP 地址。

啟動 OSPF 流程

完成所有介面和區域設置後，返回 **System >> Network >> OSPF** 介面以啟動 OSPF 流程。

以下圖表是“路由器 B”的設置：

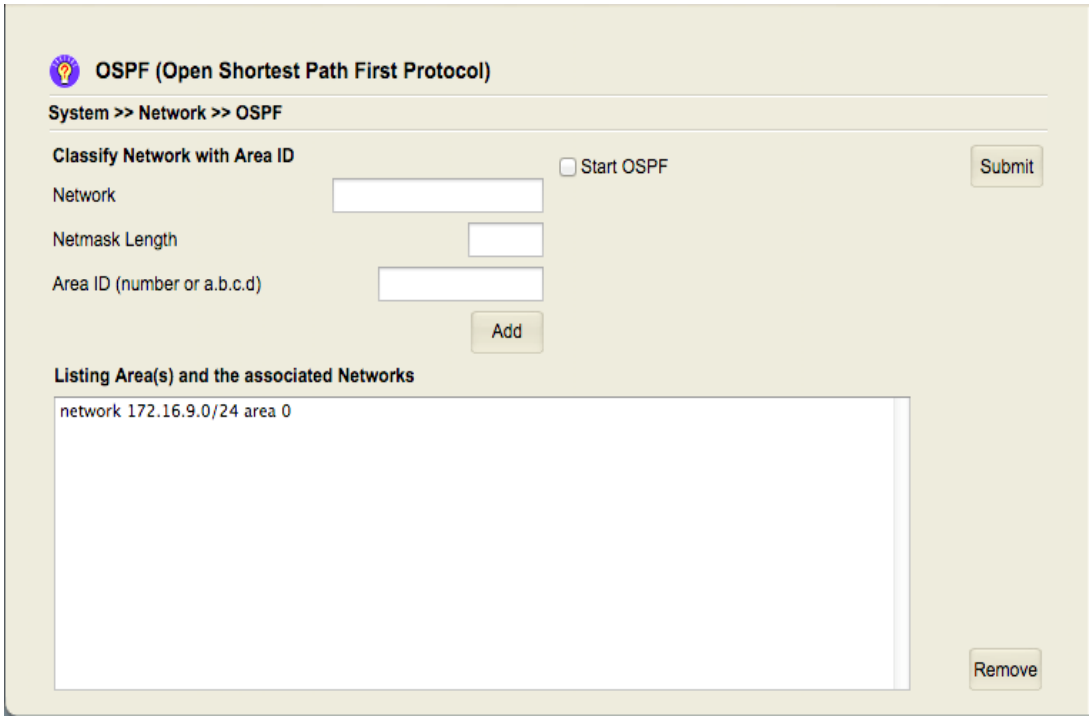


圖 147: 虛撥網路設定 (路由器 B)

請注意，身份驗證密鑰應與“路由器 A”上設置的密鑰匹配。

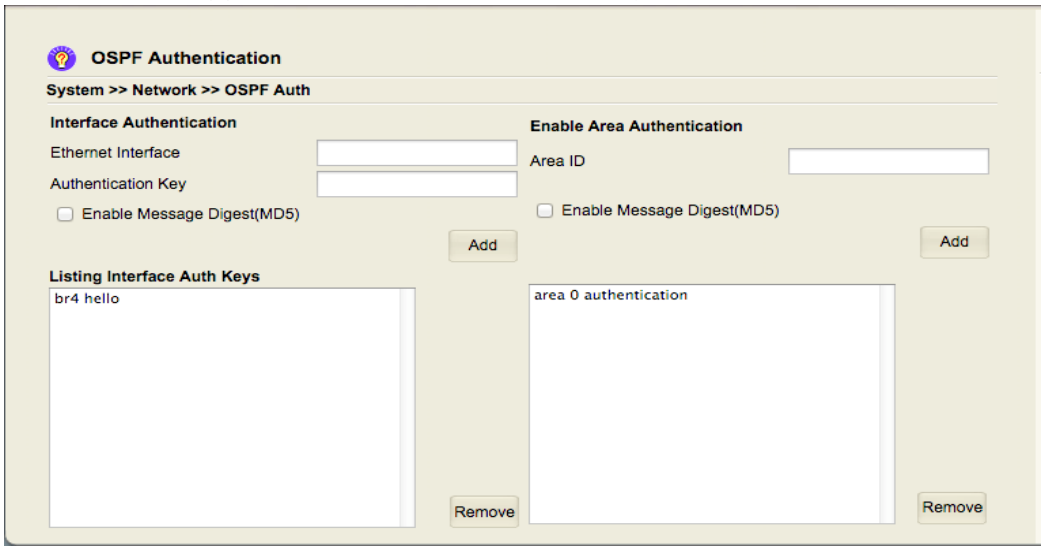


圖 148：路由器 B 的認證書設置

驗證設置

1. 在開啟 OSPF 和 路由器 A 和 路由器 B 後，我們檢查路由器 A 的路由表，確認路由器 B 的子網已成功 **發佈**（或 **傳播**）到路由器 A。。
2. **路由器 C：** 在路由器 C 上啟動 OSPF 後，檢查其路由表。

附註：如果子網的鏈接中斷，則對應的路由表條目不會被 **傳播** 到其他路由器。 。在某些情況下，您可能需要檢查相關鏈接是否中斷 。

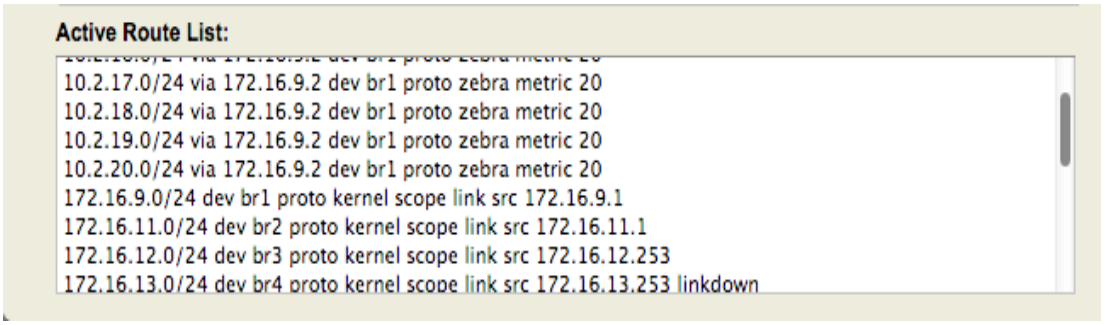


圖 149：路由器 A 目前的路由表 (OSPF 開啟後)

然後我們開始設置“路由器 C”：

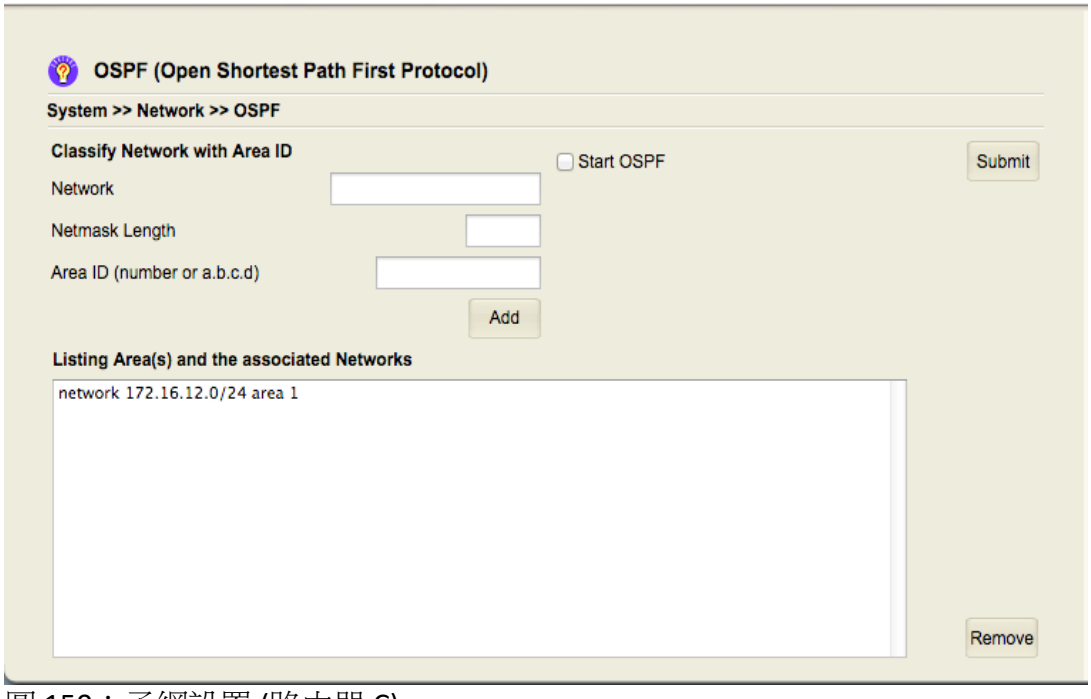


圖 150：子網設置 (路由器 C)

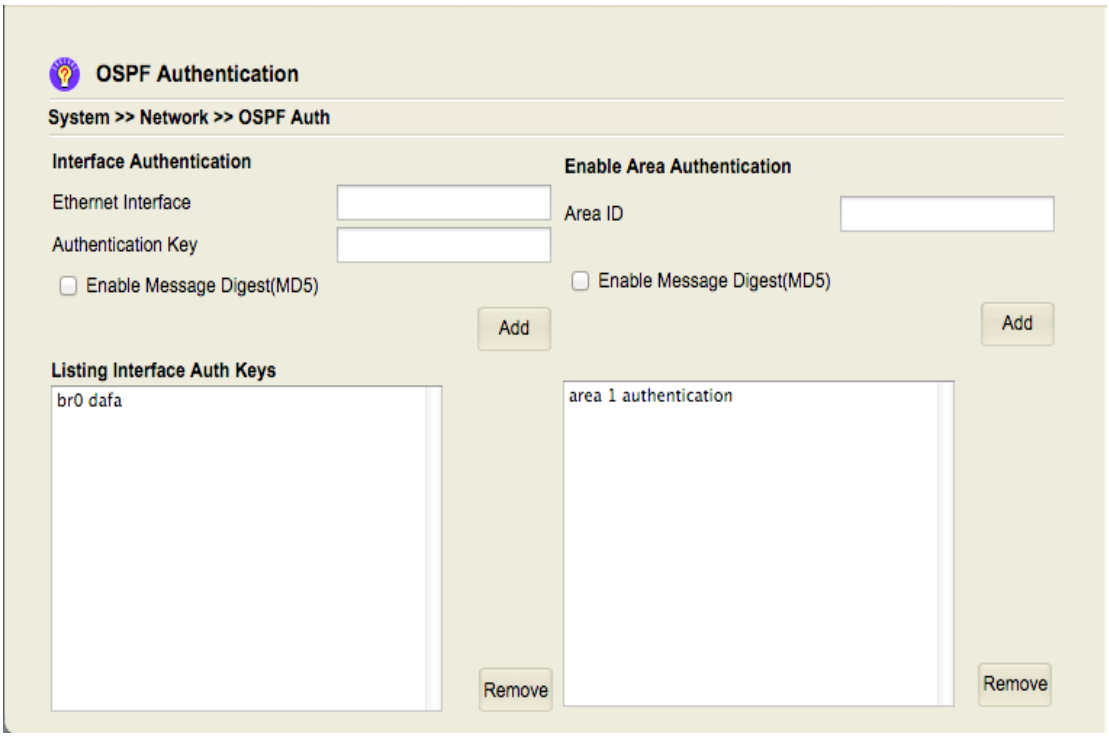


圖 151：認證設置（路由器 C）

同樣，這裡的認證書密鑰應該與“路由器 A”上設置的密鑰相匹配。在“路由器 C”上啟動 OSPF 後，“路由器 C”的路由表如下所示：

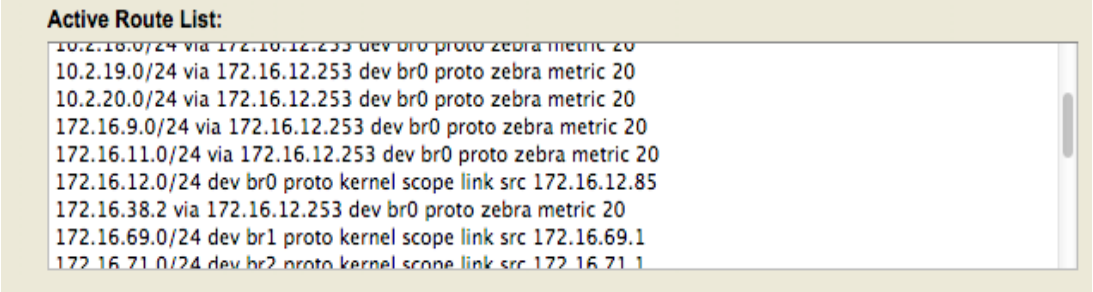


圖 152：路由器表（路由器 C）

如果子網的鏈接中斷，則對應的路由表條目不會被復制到其他路由器。在某些情況下，你可能會檢查相關的鏈接是否中斷。

PIM (Protocol Independent Multicast)

什麼是多播？

多播允許發送方只發送一次 IP 數據包，但該數據包可以被多個接收方接收 。這項技術通常基於 **UDP** 協議，因此它不依賴於可靠傳輸（與 TCP 不同） 。跨子網多播的需求

- **同一子網內：** 只需要所有網路設備支持 **IGMP (網際網路組管理協議)** 即可。
- **跨不同子網：** 路由器需要支持多播路由協議，例如 DVMRP、MOSPF 或 **PIM**。在我們的基礎平台上，我們使用 **PIM** 來支持多播路由。

啟用 PIM 的基本步驟

PIM 用於處理發送方發送多播包，而其他子網上的接收方能夠接收這些包的場景 。

1. **開啟 PIM 功能：**
    - 導航至：**System >> Network >> Multicast Routing** 。
    - 在此處開啟 PIM 功能 。

附註： 如果發送方和接收方位於**同一子網**，則**不需要** PIM，只需 IGMP 即可 。

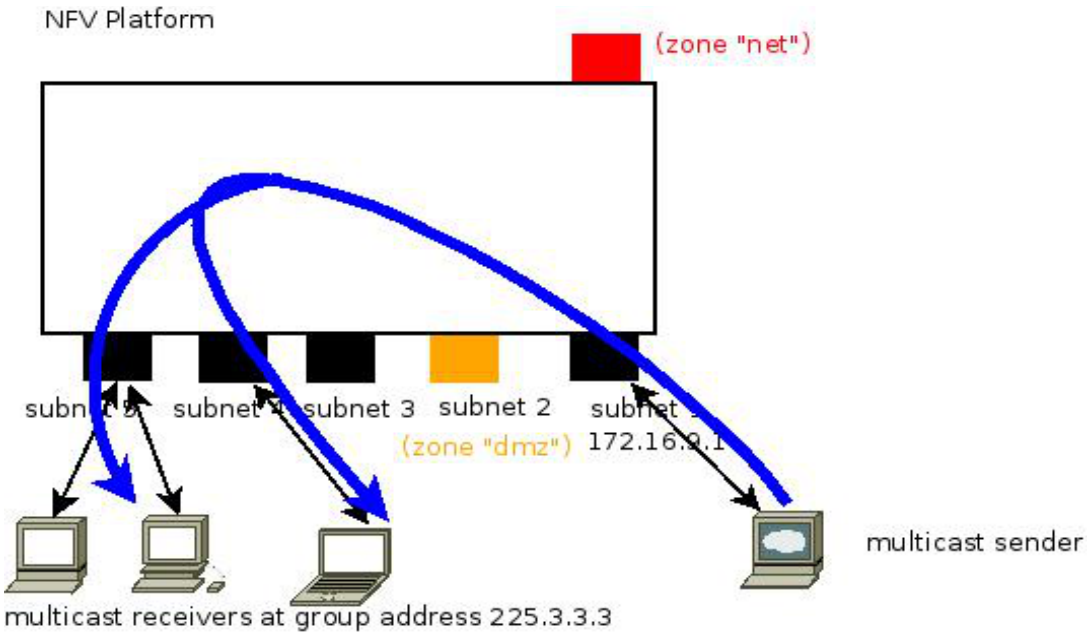


圖 153：使用 IGMP 和 PIM 的場景

⚠ 策略注意事項

請記住，來自 **net** 區域和 **dmz** 區域的數據包在默認情況下將會被阻止（drop）。在配置多播時，您可能需要考慮不同區域之間的策略影響。

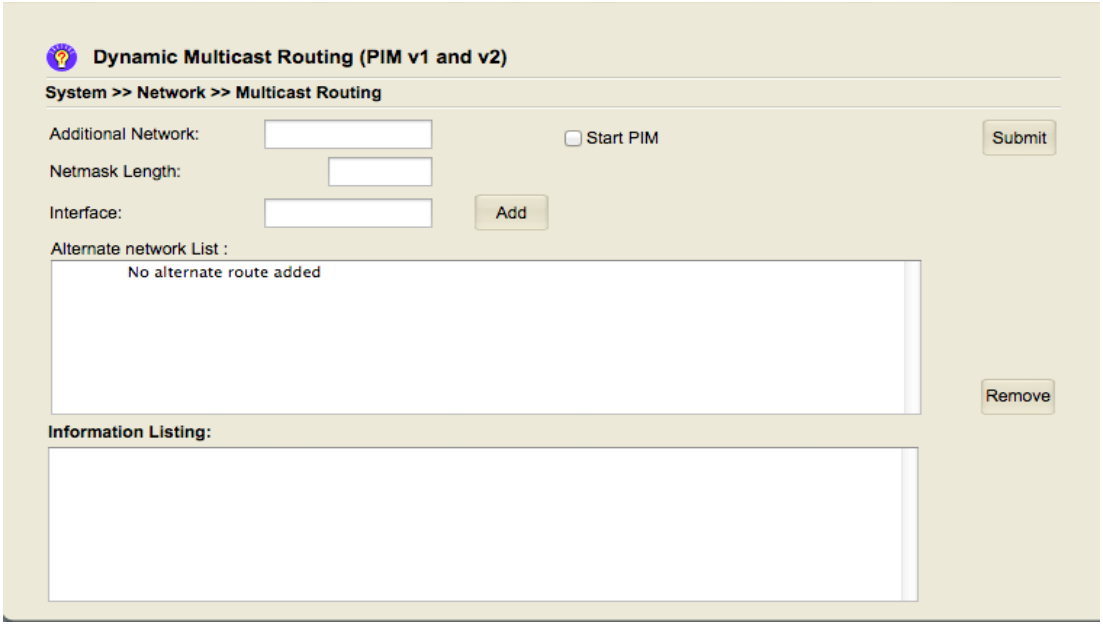


圖 154：多播路由設置

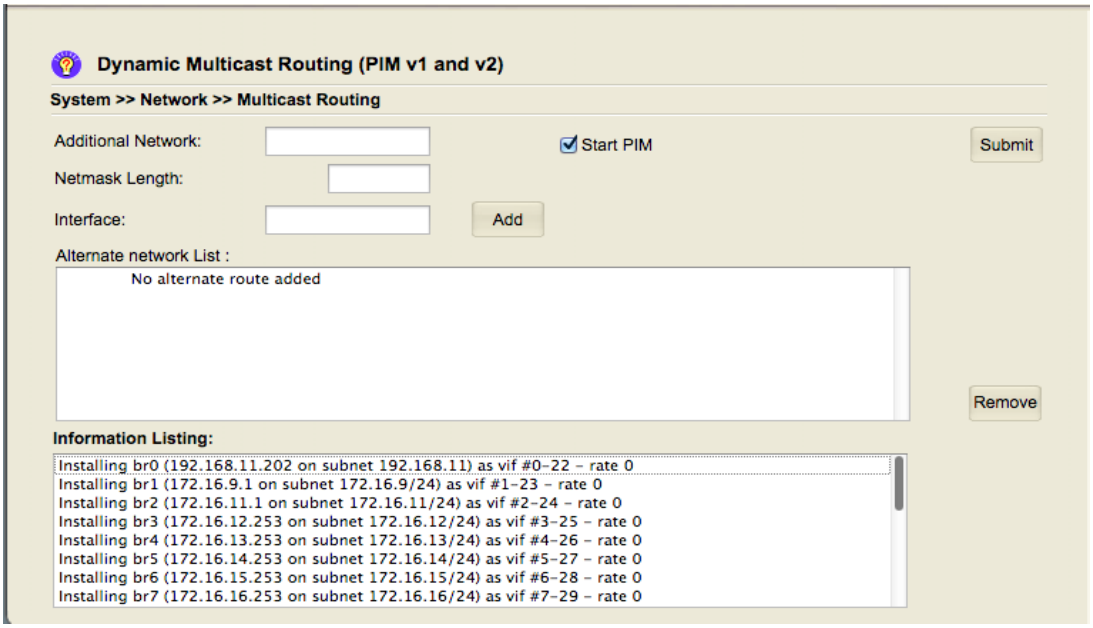


圖 155：PIM 開機後

進階配置 (可選)

在大多數情況下，只需啟用 PIM 即可，它會自動設置特定多播組的 **Rendez-vous 點 (RP)** 。只有在特定情況下才需要額外的配置：

情境	所需操作	介面
發送方位於未直接連接到路由器的子網	您需要指定該子網以及相關的介面，以確保多播包能夠到達 。	**System >> Network >> Multicast Routing**
限制多播傳播範圍	為了防止 IP 多播傳播得太遠，您可以要求 PIM 不要將多播包發送到特定的網路介面 。	**System >> Network >> Multicast Control**
靜態設置 Rendez-vous 點 (RP)	雖然可以靜態設置特定組地址的 RP，但在只有一台機器（基礎平台）的情況下，這是不需要的 。	N/A

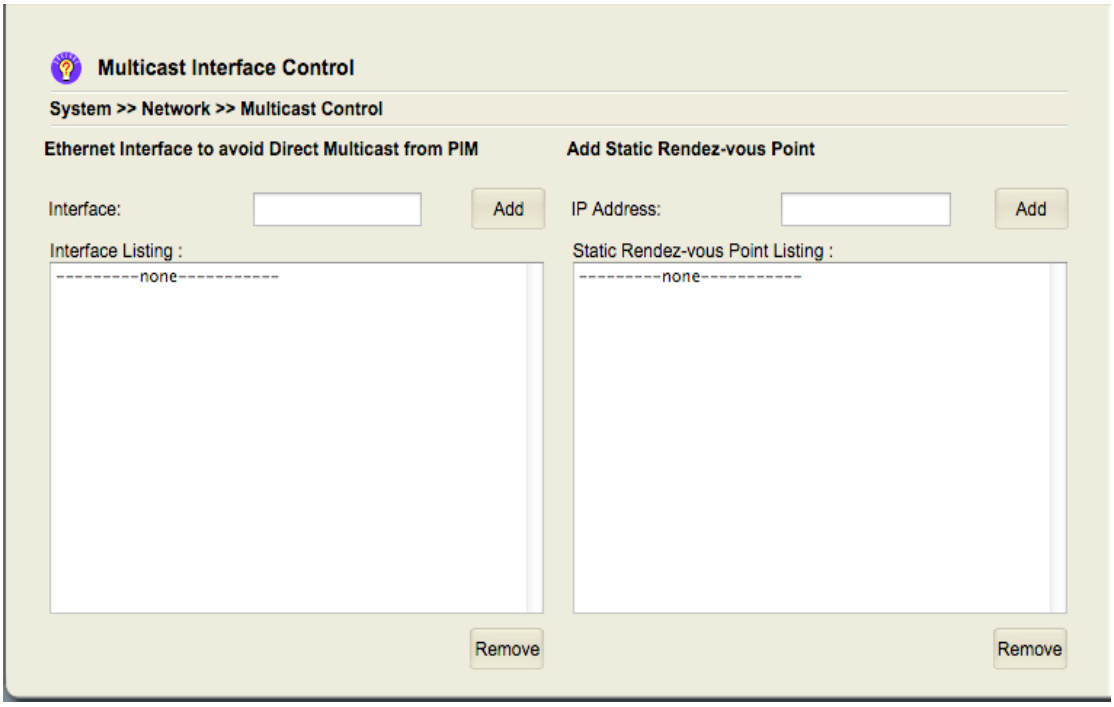


圖 156：多播控制

如何驗證 PIM 功能

多播基於 UDP，因此驗證需要檢查數據包是否成功到達 。

- **驗證方法：** 檢查多播包是否到達預期的子網，以及 UDP 包是否因網路擁塞而丟失 。

Using Site To Site VPN Bridge Mode Support OSPF RIPv2 (SD-WAN 常用情境)

站點間 VPN 路由配置指南

本指南說明如何在**站點至站點 VPN (Site-to-Site VPN)** 環境中**配置**路由條目，以便在缺乏物理連接的情況下實現動態路由協議（如 OSPF 或 RIPv2）的運行 。

路由交換挑戰

在配置 VPN 時，目標是確保路由器能夠交換路由條目，以便它們可以訪問相同的 IP 子網 。

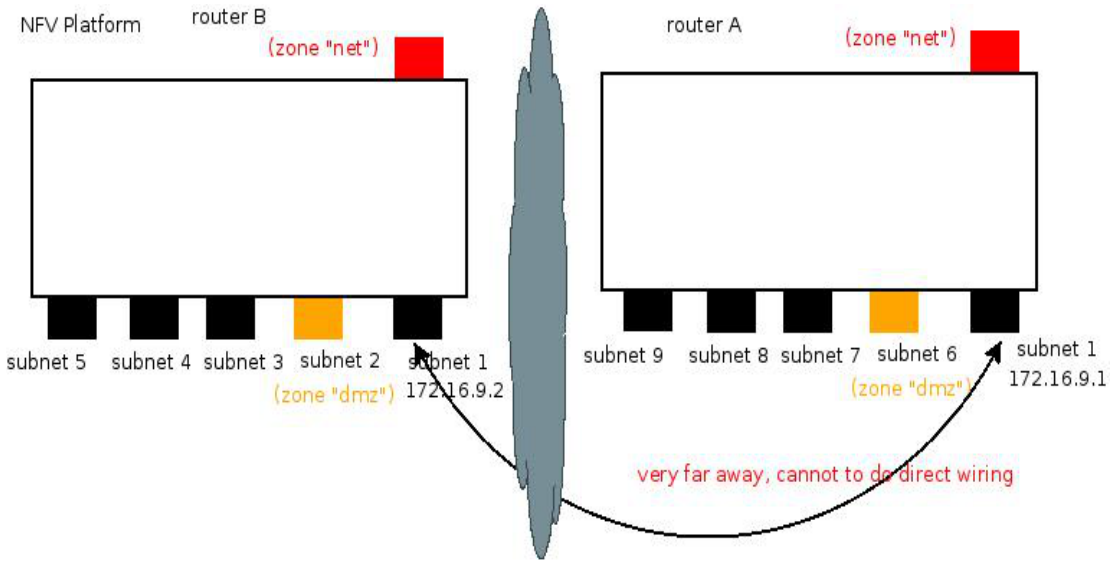


插圖 157：兩台路由器跨越互聯網

挑戰	說明	解決方案
地理限制	由於路由器位於網際網路的不同站點，無法在各埠之間建立直接的物理連接（例如，通過乙太網電纜或交換機）。	建立虛擬鏈路並橋接埠。
動態路由受限	在沒有直接物理連接的情況下，**OSPF** 或 **RIPv2** 等動態路由協議無法直接實現。	在 VPN 連接上啟用 OSPF 或 RIPv2。

實現動態路由的步驟

當物理鏈路不可行時，我們需要使用虛擬化和橋接技術來模擬單一網路段，從而啟用動態路由。

1. **建立虛擬鏈路：** 建立虛擬鏈路來跨越那些缺乏物理連接的埠[8]。
2. **橋接埠：** 將路由器的埠**橋接**起來。一旦埠被橋接，**OSPF** 或 **RIPv2** 就可以在同一個子網中運行。
3. **啟用 VPN 路由交換：** 對於**站點至站點 VPN** 在橋接模式下，我們啟用 **OSPF** 或 **RIPv2** 來交換 VPN 連接之間的路由信息。

透過建立虛擬鏈路並橋接埠，我們能夠在 VPN 連接的兩端成功運行動態路由協議，實現路由信息的自動交換。

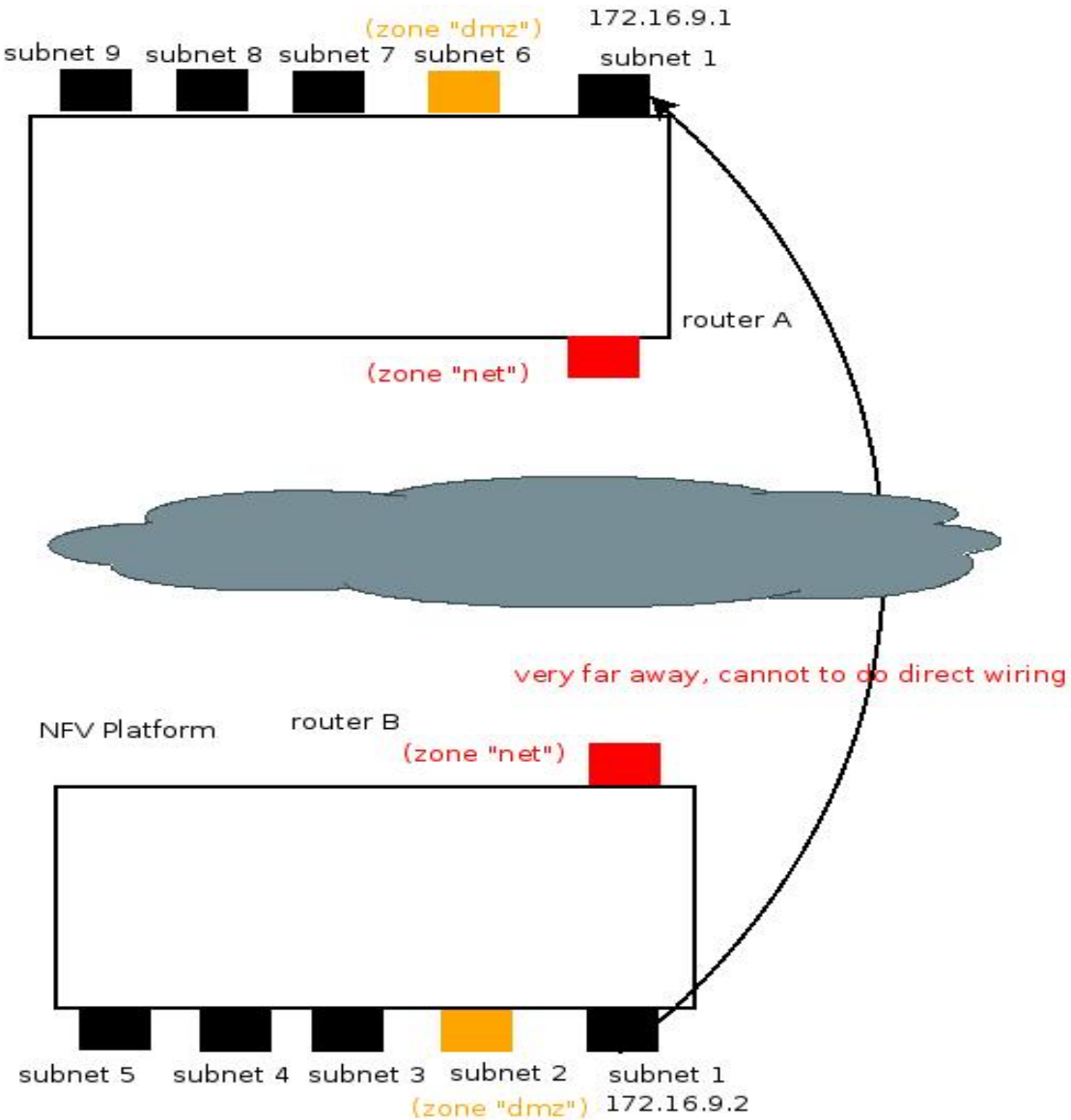


圖 158：兩個私人路由器嘗試通過互聯網連接。因此，問題最終確定為：在橋接模式下建立網站到網站 VPN。當然，VPN 連接必須通過“net”區域才能到達另一側。

網路連接場景描述

本案例描述了兩個位於不同地點的**私人區域網路 (LAN)**，嘗試透過網際網路（Internet）建立一個安全的**站點到站點 (Site-to-Site) 虛擬私人網路 (VPN)** 連線。

核心問題：

如何在兩個私人路由器之間建立一個**站點到站點 (Site-to-Site) VPN** 連線，特別是在涉及以下條件時：

- **路由器操作模式：** 路由器配置為**橋接模式 (Bridge Mode)**。
- **路由/防火牆要求：** VPN 連線（即加密通道）必須經由路由器的 “net”或廣域網 (WAN) 區域才能成功穿越網際網路並到達對端網路。

簡化重點：

需要配置兩個處於橋接模式的私人路由器，以建立一個穿過網際網路（ “net” 區域）的站點到站點 VPN 通道。

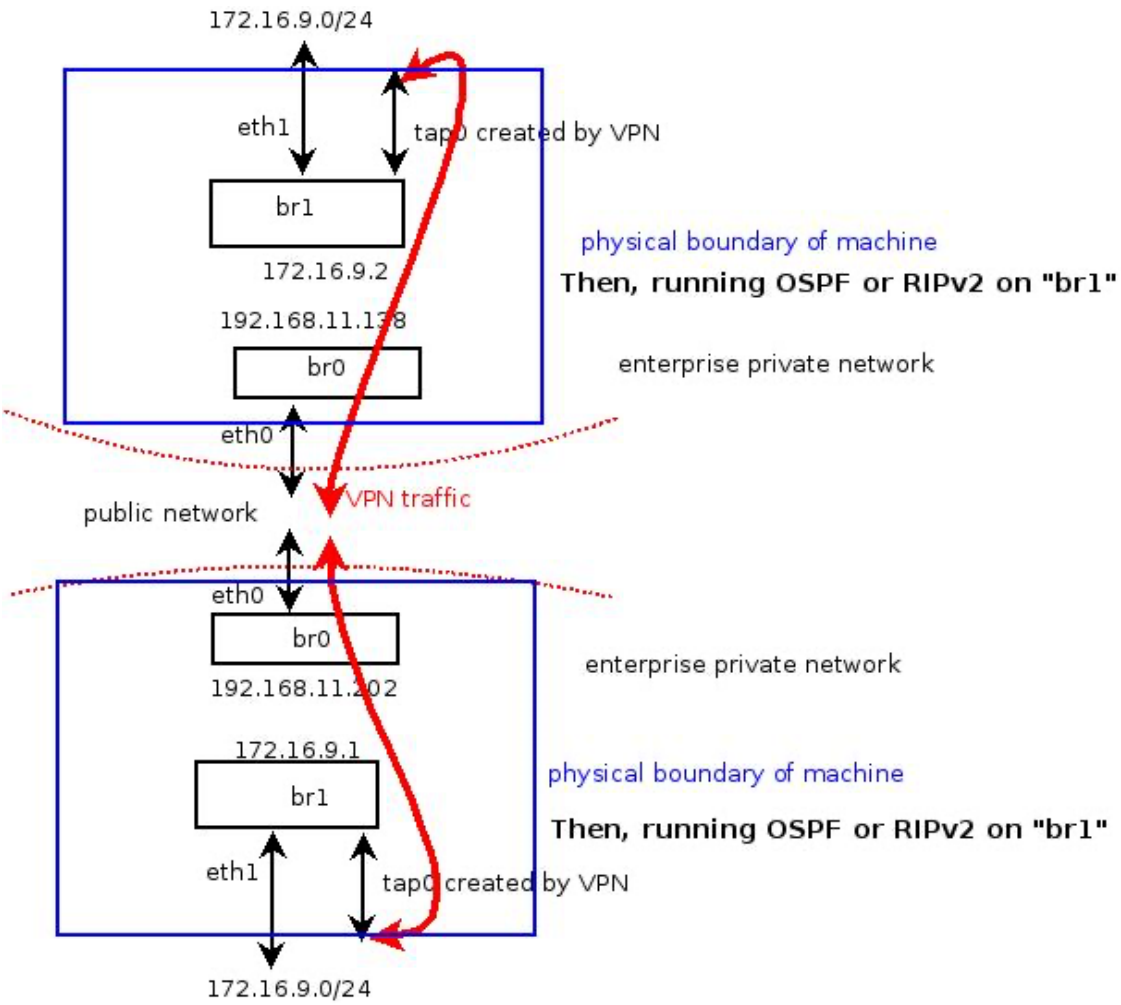


圖 159：VPN（橋接模式）中 OSPF 或 RIPv2

SD-WAN (WAN 邊緣服務) 常用情境的關鍵要素

這種組合在 SD-WAN 和複雜網路環境中極為重要，因為它實現了網路的**透明化**和**動態化**，是企業在多分支機構環境中建立高效互連的基礎。

1. 站點對站點 VPN (Site-to-Site VPN)

這是 SD-WAN (Software Defined Wide Area Network) 實現**廣域連接**的核心。

- **功能：** 在兩個地理位置分散的網路（例如總部和分支機構）之間建立**加密的安全通道**。
- **優勢：** 確保跨公共網際網路傳輸的企業資料具備**機密性和完整性**。

2. 橋接模式 (Bridge Mode)

這是決定 VPN 網路層次行為的關鍵。

- **功能：** 將兩個遠端區域網路 (LAN) 視為**同一個 Layer 2 網段**。這使得遠端節點可以**直接共享**相同的網路廣播域和子網。
- **優勢：** 允許應用程式（尤其是舊版應用程式）透明地運作，**無需複雜的路由設定或 NAT 轉換**，簡化了網路設計。

3. 動態路由協定支援 (OSPF/RIPv2)

這是實現網路**彈性**和**自動化**的決定性因素。

- **OSPF (Open Shortest Path First) / RIPv2 (Routing Information Protocol v2)：** 這些是常見的**內部閘道協定 (IGP)**。
- **在 VPN 中的作用：** 由於 VPN 橋接模式將兩個遠端網路連接起來，動態路由協定可以在此 VPN 通道上運作，**自動學習和宣告**每個站點的新增或變更的網路路徑。
- **優勢：** 當站點網路拓撲發生變化、有新子網加入或某條連線失敗時，路由器會**自動更新路由表**，**無需**管理員手動干預，極大地提升了 SD-WAN 網路的**擴展性 (Scalability)** 和**高可用性 (High Availability)**。

結論

將 **Site-to-Site VPN Bridge Mode** 與 **OSPF/RIPv2** 結合，是 SD-WAN 領域中確保**網路透明、簡化管理、實現自動化路由決策**的標準且常用的設計模式。

第六章 部署情境範例 (Deployment Scenarios)

在本章中，我們將透過數個實際「部署情境」示例，說明常見的需求與對應的網路條件，並示範如何運用前述章節介紹的各項功能來完成設定。每一個情境都會簡要說明背景、目標以及應套用的功能模組，協助您將前面學到的概念，轉換為可在實務環境中直接套用的設定步驟。

範例 1：具備 / 不具備網際網路存取權限的主機

在本示例中，我們希望將**具備／不具備網際網路存取權限的主機**放在不同安全區域中。

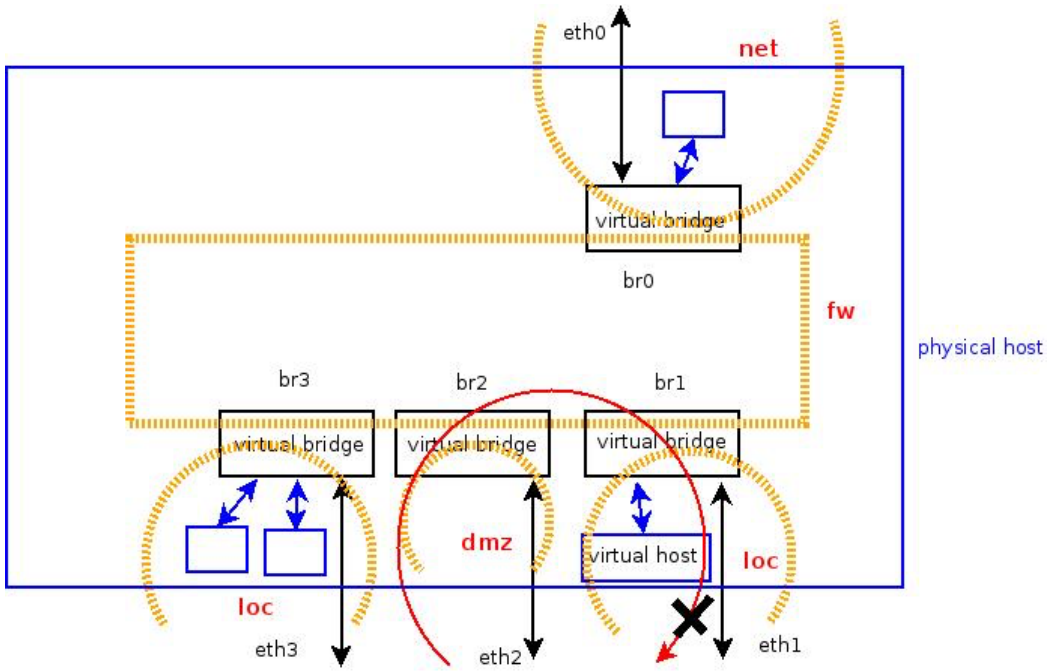
出於資安考量：

- 用於內部營運的主機 **不得直接連上網際網路**。
- 具有網際網路存取權限的主機，也 **不得主動連線** 到這些內部營運主機。

回顧前文對「dmz」區域的說明：
在邊界控制預設規則下，「dmz」區域無法主動連線到「loc」區域。因此，我們可以：

- 將 **需要上網的主機** 放在「dmz」區域；
- 將 **僅供內部營運使用的主機** 放在「loc」區域。

圖例 160：dmz 區域規則



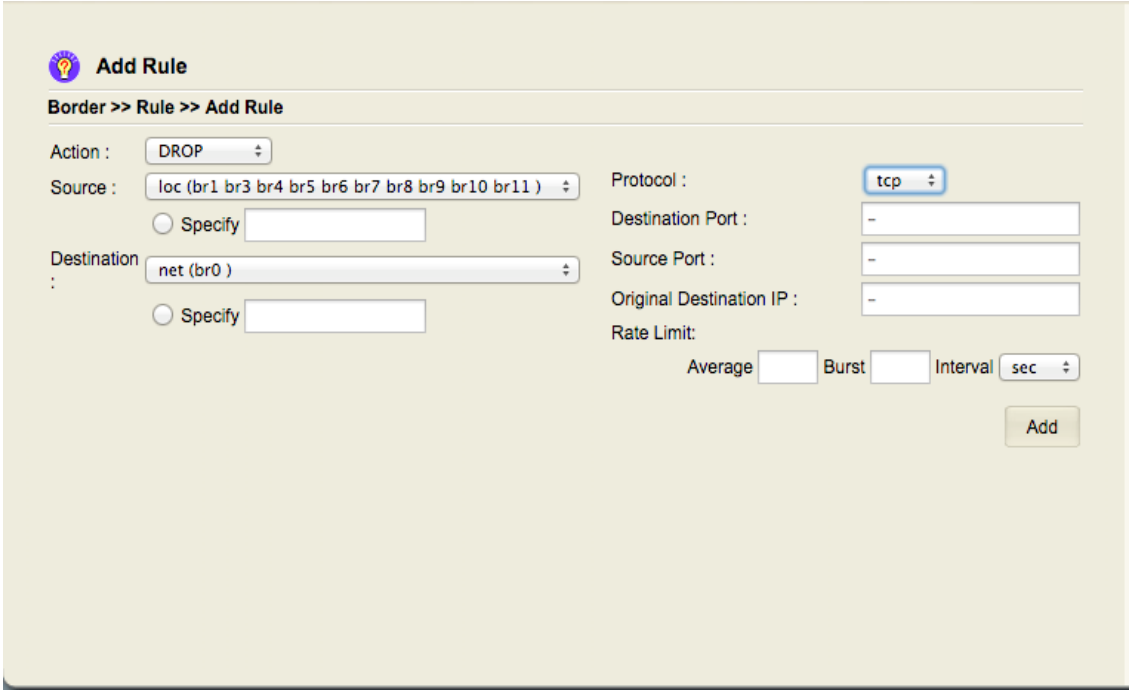


圖 161：阻止區域“loc”的網路存取。

不過，「loc → net」的連線預設是允許的。若要完全禁止內部營運主機直接上網，必須再透過「**Border >> Rule >> Add Rule**」新增規則，封鎖自「loc」至「net」的流量。

許多傳統辦公室做法是：
將對外服務（例如公開的 Web / Mail 伺服器）放在「dmz」區域，其他員工主機放在「loc」區域，並透過埠轉發與存取控制，限制對網際網路的暴露範圍。

在 Azblink NFV 平台下，我們可以進一步支援下列需求：

每位員工使用兩台「邏輯上獨立」的桌面環境：

- 一台專門處理內部業務（不可上網）
- 一台專門用來上網（不可碰內部系統）。

網路隔離與遠端連線:
為確保內部業務系統 (loc 區域) 的安全，避免其直接連網際網路，我們採用雙主機隔離策略。

流程如下：

1. **建立上網虛擬機 (VM)：** 在「dmz」區域建立一台或多台虛擬機 (VM)，作為員工專用的「上網跳板桌機」。

2. **多樣化的遠端連線：** 員工在自己的**內部營運主機**上（位於「loc」區域）啟動以下任一種**遠端桌面客戶端**，連線到「dmz」區域中的對應虛擬機：
    - **核心桌面協議：**
        - VNC Client (Virtual Network Computing): 廣泛適用的跨平台協議。
        - SPICE Client (Simple Protocol for Independent Computing Environments): 常見於 KVM/QEMU 虛擬化環境，提供高效能多媒體支援。
        - Microsoft Remote Desktop (RDP) Client: 最常用於連線 Windows VM，提供高效的桌面體驗。
    - **高性能串流選項：**
        - Sunshine (Server) / Moonlight (Client): 此組合適用於對**影音延遲和畫質要求極高**的場景（如遠端遊戲或圖形應用），Sunshine 作為伺服器端，Moonlight 作為客戶端進行連線。
    - **其他連線選項：**
        - SSH 搭配 X11 Forwarding: 適用於 Linux VM，安全地顯示單一的圖形介面應用程式。
        - HTML5/Web 介面連線: 透過部署遠端桌面**網頁閘道服務**（如 Apache Guacamole），員工只需使用瀏覽器即可連線。
        - TeamViewer/AnyDesk Clients: (若允許使用) 商業化的遠端控制解決方案。
3. **隔離目標達成：** 員工所有的上網行為，都在這台 DMZ 區域的 VM 上完成；內部營運主機本身**完全不直接連出網際網路**，從而**大幅降低受感染風險**。

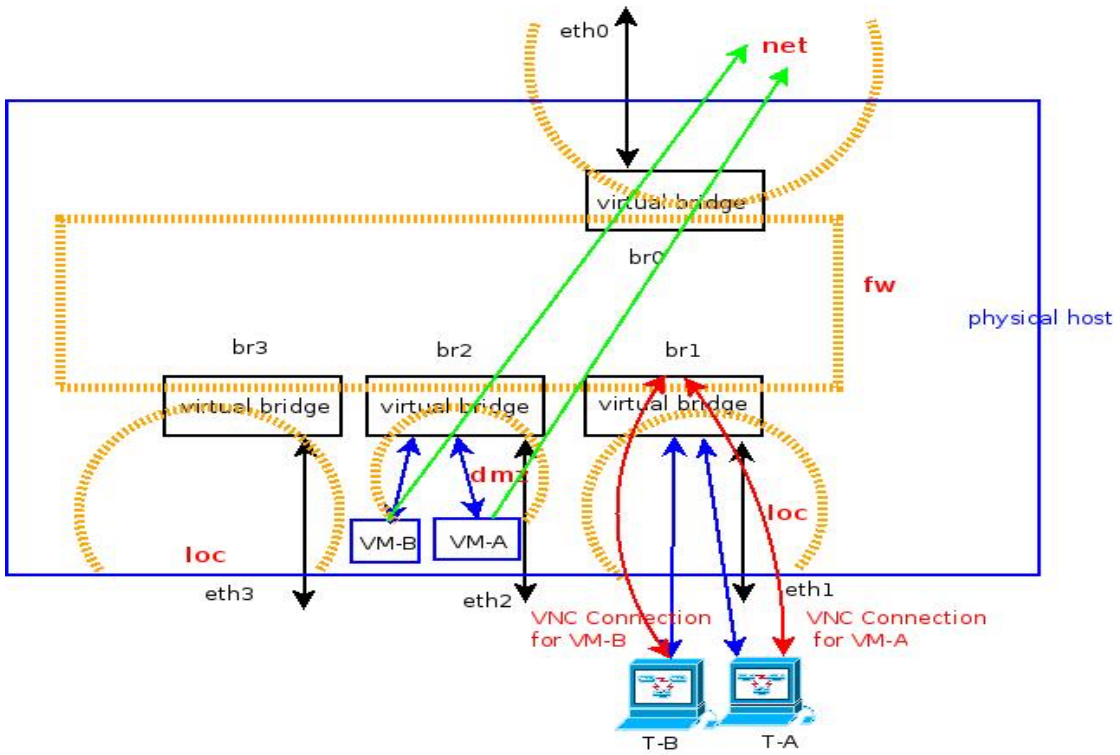


圖 162：使用虛擬機器進行網際網路存取

請特別注意，遠端連線時，VNC / SPICE / RDP 客戶端應連到 **基礎平台的 IP 位址**（例如 br0 的 IP），並指定對應的 VNC/TCP 連接埠。
如圖所示，T-A 這台內部主機的螢幕上顯示的是 VM-A 的遠端桌面（透過 VNC），使用者在 VM-A 的瀏覽器中上網，而非直接在 T-A 上連線，藉此達成**內外網邏輯隔離**。

範例 2 : 使用 VPN 存取虛擬桌面

在使用 VNC 客戶端（或 SPICE 客戶端）連線到基礎平台上的虛擬機時，請特別注意：
VNC/ SPICE 客戶端連線的目標，應該是「基礎平台的 IP 位址」，而不是虛擬機本身的 IP 位址。
原因是：VNC/ SPICE 服務是由基礎平台提供，虛擬機只能透過基礎平台被「轉接」出來。

因此，若您是透過 VPN 連回辦公室使用 VNC，請將 VNC 客戶端指向 VPN 位址池中的第一個 IP 位址（預設為 172.16.38.1），也就是 VPN 用戶端眼中的「基礎平台位址」。

成功取得虛擬機的主控台畫面之後，後續所有的網路存取（例如連到辦公室內其他主機）都會依照既有的防火牆與路由策略來管控。換句話說，您是先「遠端操作一台位於安全區域內的虛擬機」，再由這台虛擬機去存取內部資源，而不是讓 VPN 用戶端直接對內網發起連線。這樣可以在維持便利性的同時，進一步降低內部網路暴露的風險。

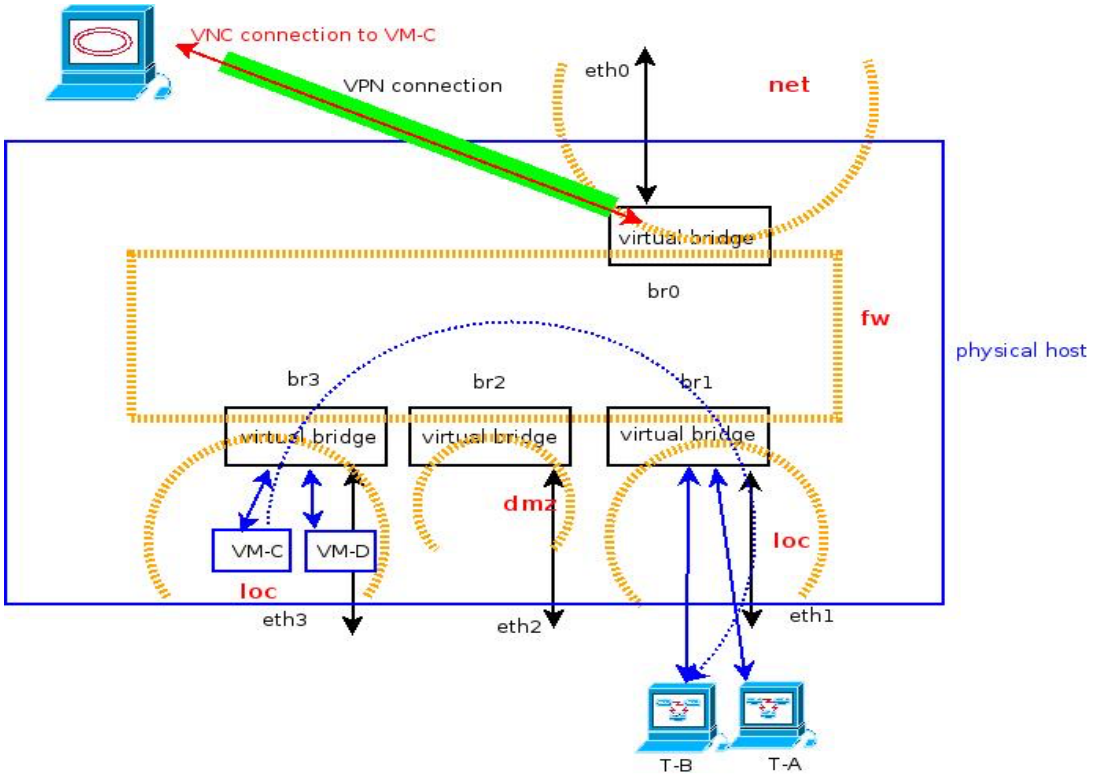


圖 163：透過 VPN 訪問虛擬機

如上圖所示，使用者可以先透過 VPN 連入辦公室網路，再以 VNC 連線到虛擬主機的主控台。取得 VNC 主控台之後，便能以該虛擬主機為跳板，進一步連線到辦公室內其他受管控的主機與服務。

這樣的作法有兩個好處：

1. **無需直接開放** VPN 客戶端對各個內部子網的存取權限，只需允許其連到提供 VNC 服務的虛擬主機所在區域即可。
2. **可在虛擬主機上再加上一層認證機制**（例如帳號密碼、多因素驗證或其他登入流程），將 VPN 存取與內部系統存取分層控管，進一步強化整體安全性。

範例 3：SBC 與防火牆的虛擬化

一般而言，SBC 與防火牆至少會配置兩個網路介面：一個連接至 Internet，另一個連接至內部私有網路。因此，在 Azblink NFV 平台上建立 SBC 或防火牆虛擬主機時，只要將該虛擬機的網路介面分別接到「br0」(作為 WAN 端）與「br1」(作為 LAN 端）即可。
接著，將 **位於防火牆後方之區網內的終端設備**（例如辦公室 PC、實體伺服器或其它網路設備；**不包含本 NFV 平台上的其他虛擬主機**）的預設閘道，設定為此虛擬 SBC／防火牆在 LAN 端的 IP 位址。如此一來，這些區網設備的進出流量便會經由該虛擬主機轉送與防護，達成與實體 SBC／防火牆相同的運作模式。

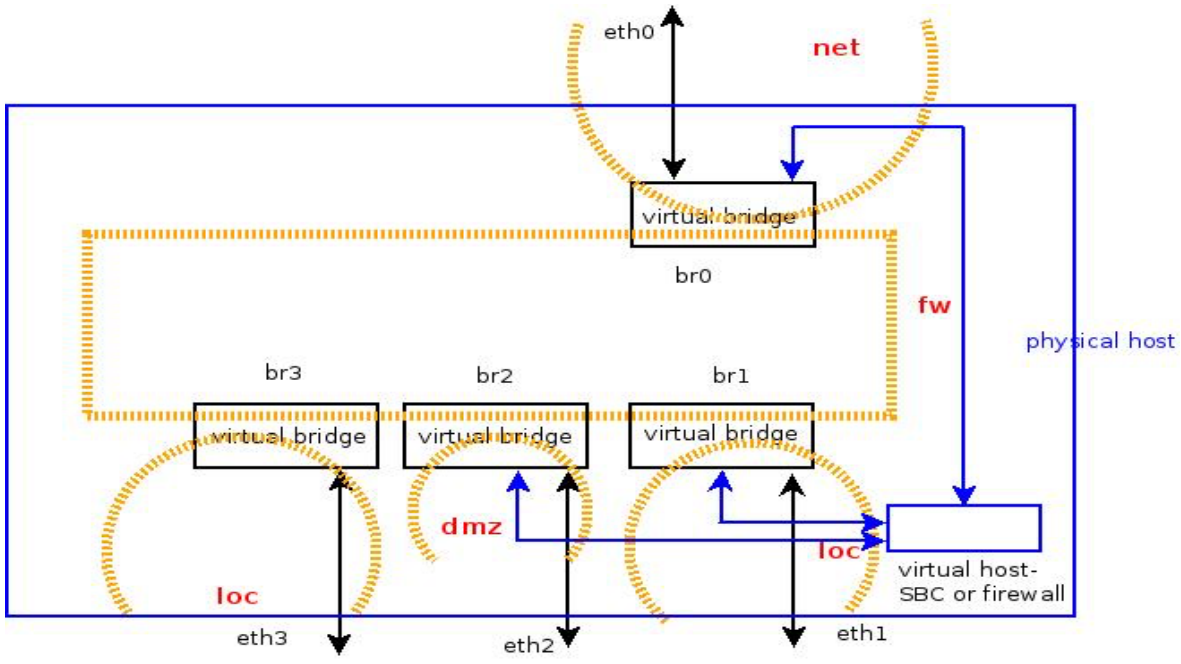


圖 164: SBC 或防火牆虛擬化

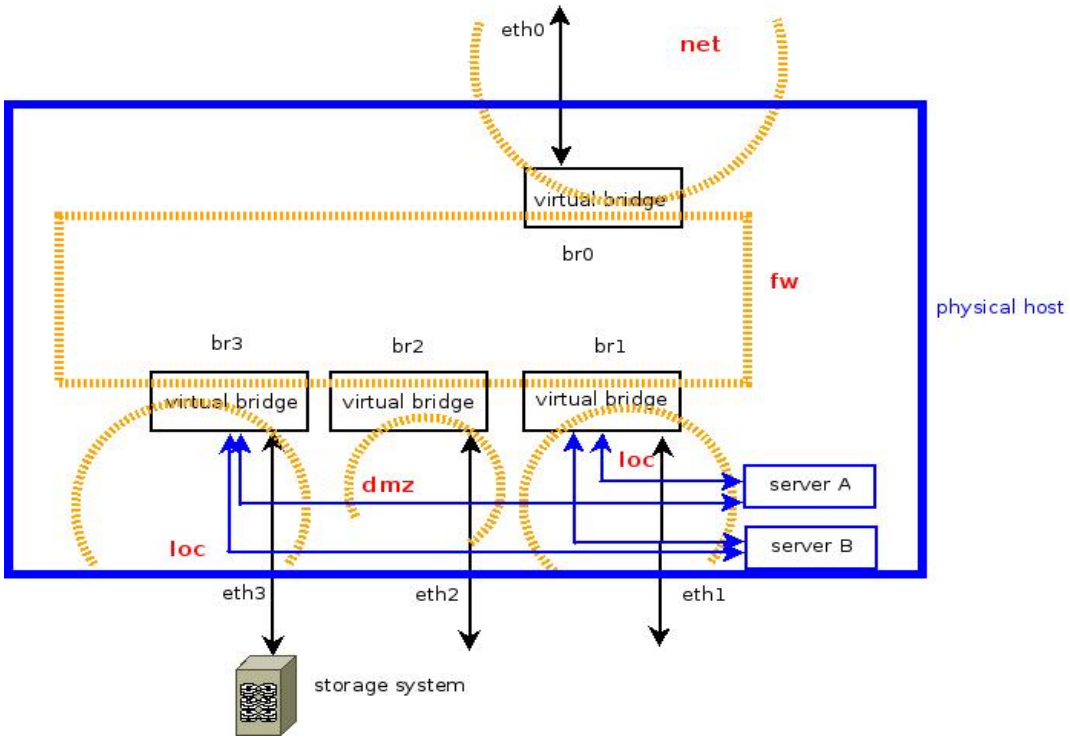


圖 165：使用專用子網作為存儲區網路

範例 4 : 將儲存系統部署在另一個子網中

基礎平台本身的儲存空間，往往不足以同時滿足多台虛擬機的長期使用需求。以上示意圖說明了一種常見作法：Server A 與 Server B 主要透過 br1 連入服務網段，對外提供一般應用服務；同時，每台伺服器額外再透過 br3 連上一個獨立子網，經由 iSCSI 連接外部儲存系統。

換句話說，掛在 br3 上的這個子網是專門用來承載儲存流量的 Storage Area Network (SAN)。這樣一來，就可以在不佔用基礎平台本機磁碟的情況下，為多台虛擬機提供集中式、高擴充性的儲存空間，同時也將儲存流量與一般業務流量隔離開來，較易管理與維運。

範例 5：具備多播路由與發送功能的路由器

下圖示範如何將基礎平台同時作為**多播路由器 (multicast router)** 使用，而多播來源端 (multicast sender) 則位於虛擬機中。範例中已關閉「邊界控制」(防火牆功能)，因此圖中的 net、dmz、loc 子網未特別標示區域角色，所有子網皆視為同一安全等級的網段。
如果在實際環境中啟用了「邊界控制」，則 net 與 dmz 將受到前述區域規則限制，多播封包可能

因此無法在這些區域之間正常傳遞，導致多播路由失效。
在基礎平台上啟用多播路由的重點，是**開啟 PIM（Protocol Independent Multicast）功能**，並在相對應的介面上正確啟用；只要 PIM 正常運作，多播路由便能在各子網之間依設定轉送，讓虛擬機內的多播傳送端與其他子網的多播接收端順利溝通。

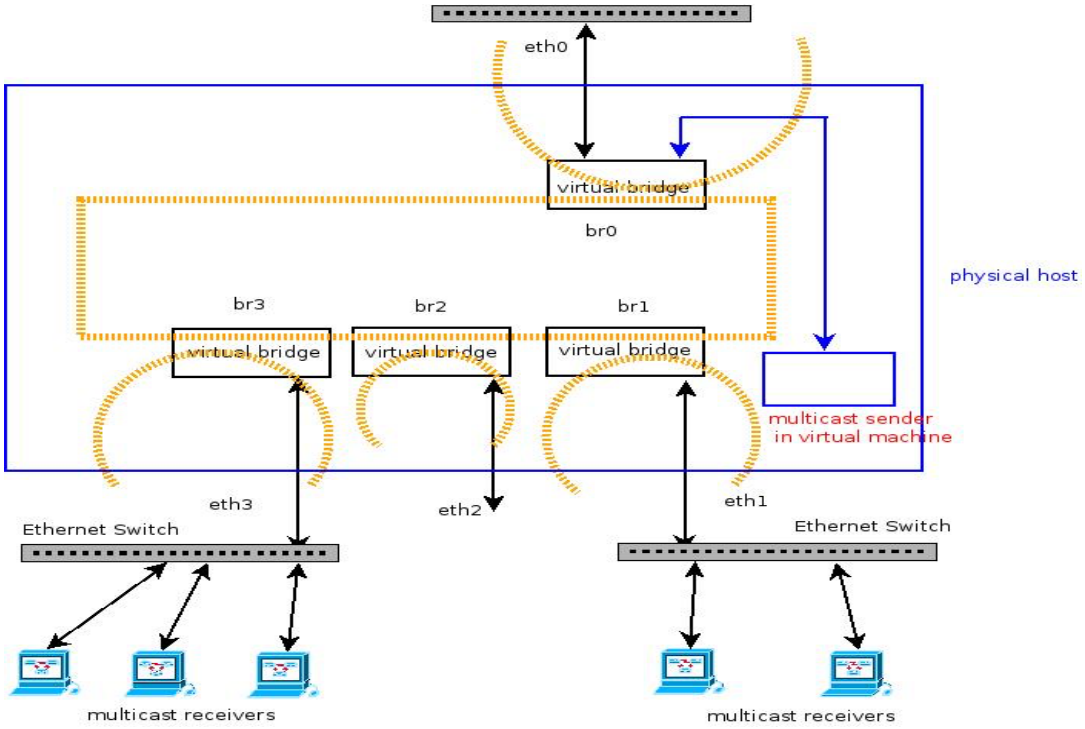


圖 166：具有多播路由器與多播發送器的示意圖。

在上圖範例中，多播發送端安裝在一台連接至 br0 的虛擬機內。
對基礎平台本身而言，只要**啟用 PIM**，就已具備多播路由能力；其餘行為則取決於多播發送端與接收端的設定。要讓多播封包跨越路由器抵達其他子網，封包中的 **TTL（Time To Live，存活時間）** 必須大於 1。每經過一台路由器，TTL 值就會減 1；當 TTL 減到 0 或更小時，路由器就會停止轉送該封包。
實務上，許多人會使用開源軟體 **VLC** 來測試多播傳送與接收。撰寫本文件時，VLC 的預設 TTL 值為 -1，代表多播封包**無法離開本地子網**。若要透過本平台把多播流量送到其他子網，請務必在 VLC 中調整 TTL（例如設為 4 或更大）；否則，多播發送端與接收端將只能位於同一個子網中。

本手冊不打算成為完整的 VLC 教學，因此僅提供上述重點提示。VLC 介面中調整 TTL 的位置，請參考以下螢幕截圖。

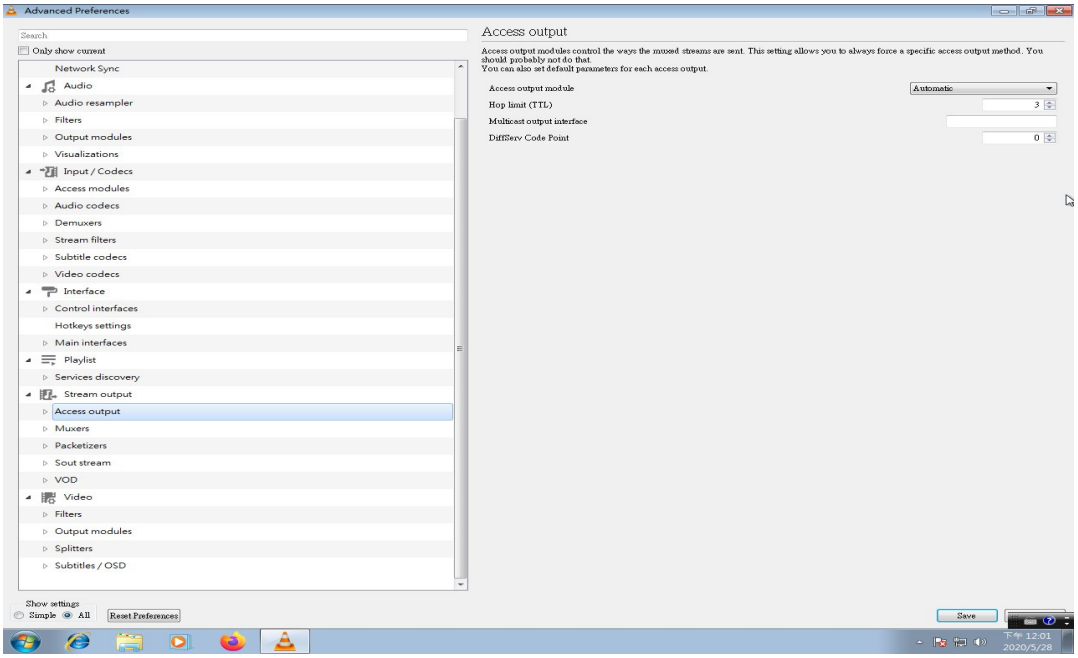


圖 167：VLC 作為多播發送者中 TTL 設置的示例