

基於封包壓縮之網路分享器設計與實作

黃一軒¹ 林純守² 陳景菘² 楊正仁¹

元智大學資訊工程系

{ihhuang,czyang}@syslab.cse.yzu.edu.tw¹
{s932209,s932216}@mail.yzu.edu.tw²

摘要

近年來網路分享器被廣泛使用在各種網路環境中，一方面是為了共享網路頻寬，一方面是為了解決 IPv4 位址不足的問題。不過因為網路分享器是所有內部網路封包的中轉站，網路分享器很容易成為整體網路的效能瓶頸。為了解決這個問題，我們提出一個基於封包壓縮的網路分享器設計。在我們的設計中，我們利用封包壓縮技術提高資料傳輸率，使資料傳輸率可以高於實際的網路頻寬。另一方面，為了兼顧網路服務品質，我們加入品質保證服務的機制，讓高優先等級的網路服務使用壓縮時間少的壓縮演算法，而低優先等級的網路服務使用壓縮能力高的壓縮演算法。我們並且在 Intel IXP425 平台上實作網路分享器，根據我們的實驗結果，我們的網路分享器可以有效增加 15% 的資料傳輸率。

關鍵詞：網路分享器、封包壓縮、品質保證服務。

Abstract

Recently, NAT routers have been widely applied in network environments. One reason is for sharing the Internet bandwidth and the other reason is for solving the IPv4 address exhaustion issue. However, since the NAT router is the transfer port of every internal packet, the NAT router is very likely to become the performance bottleneck of the whole network. To solve this problem, we present a NAT router design based on the packet compression technique. In our design, we utilize the packet compression algorithm to increase the data transmission rate such that the data transmission rate can be higher than the actual network bandwidth. On the other side, to maintain the network service quality, we apply the QoS in our NAT router design. The packets of high priority applications are compressed using a fast compression algorithm. The packets of low priority applications are compressed using a slow but high compression ratio algorithm. To evaluate our design, we implemented the NAT router on the Intel IXP425 platform. According to the experimental results, the data transmission rate of our NAT router increases 15%.

Keywords: NAT router, packet compression, QoS.

1. 前言

近年來網路分享器成為市場上網路設備的寵兒。究其原因，主要是因為 (1) 為了節省購買網路服務的成本，公司或家庭多半讓電腦共享網路頻寬。因此，公司或家庭往往購買網路分享器來提供網路共享服務。(2) 現行網際網路的定址方式主要採用 IPv4 方式，不過隨著網路普及，IPv4 位址空間已經不敷使用。為了讓沒有分配到 IPv4 位址的電腦仍然可以連接網際網路，我們可以利用網路分享器上的網路位址轉譯服務 (NAT, network address translation) [4] 來協助沒有分配到 IPv4 位址的電腦連接網際網路。

不過，由於網路分享器是內部網路封包與網際網路間的中轉站，網路分享器往往成為網路傳輸的瓶頸，導致使用網路分享器直接犧牲了網路傳輸效能。為了解決效能低落的問題，根據我們的觀察，目前有三種方法可以在使用網路分享器的同時仍然保有網路服務品質。

- 直接擴充網路分享器的對外頻寬
使用網路分享器造成的網路壅塞通常是因為內部網路有太多封包要傳送到網際網路，但是網路分享器的對外網路頻寬不夠大，使得內部網路封包必須在傳輸佇列中等待。對於這個問題，最直覺的解決方法就是擴充網路分享器的對外頻寬。不過這個作法，首先，需要支付高額的網路頻寬費用。其次，受限於網路傳輸科技的發展和實際佈建的進度，我們能夠擴充的頻寬仍然無法滿足所有使用者的需求。
- 實作負載平衡路由 (load balancing routing)
負載平衡路由 [2][7][8][10] 的使用已經行之有年，將其概念套用在網路分享器，我們可以在網路分享器上安裝多個網路介面，形成平行網路連結 (parallel links)。當內部網路封包要轉送到網際網路時，網路分享器可以根據平行網路連結的即時情形動態地改變傳輸路徑，使得內部網路封包不會因為單一網路連結的壅塞導致整體網路效能嚴重下降。不過，具有負載平衡路由功能的網路分享器通常價格不菲，不適合用在低階市場。

■ 使用品質保證服務 (quality of service)

除了上述兩種以「開源」(增加頻寬)為目標的方法外,我們可以透過使用品質保證服務 [1][5][6] 來達到「節流」(有效分配頻寬)的效果。在品質保證服務架構中,我們不用添增更多的網路頻寬。我們先將所有的網路應用程式按照優先等級排序。為了維持高優先權應用程式的網路服務品質,我們犧牲低優先權應用程式的品質,將更多的頻寬分配給高優先權應用程式。不過雖然使用品質保證服務的作法可以避免購買頻寬的成本,其整體網路效能仍然不變,因此,當如果內部網路的總頻寬需求大幅度成長時,此一機制就不適用。

綜合以上三種作法,在解決網路分享器效能問題上,我們希望可以避免購買網路頻寬成本,又當整體網路頻寬需求成長時仍然能夠提供一定服務品質,因此提出一個兼具低成本增加頻寬(開源)和品質保證服務(節流)的機制是有必要的。

在網路傳輸的技術中,運用壓縮演算法將傳輸的資料壓縮可以提供比系統實際頻寬更大的資料傳輸率 [9][12],達到低成本增加頻寬的要求。同時,為了保證網路服務的品質,我們可以結合品質保證服務,區分網路服務的等級,然後配置不同壓縮時間和壓縮比的演算法給不同優先權的網路服務。對高優先權的服務來說,我們配置低壓縮時間的演算法,這些高優先權的服務不會因為壓縮而造成大量的延遲。對低優先權的服務來說,我們配置高壓縮能力的演算法,然後數個封包一次傳送,低優先權的服務可以享有比過往方法更高的資料傳輸率。

在本論文中,我們提出一個基於封包壓縮的網路分享器設計。在我們的網路分享器設計中,為了同時達到低成本增加頻寬和有效利用頻寬的目標,我們在網路封包的傳遞過程中加入封包壓縮演算法,根據網路服務的優先權給予不同壓縮演算法,並利用不同演算法間的壓縮時間差來提供各種網路服務不同大小的網路頻寬。在我們的設計中,高優先等級的網路服務使用壓縮時間少的壓縮演算法,而低優先等級的網路服務使用壓縮能力高的壓縮演算法。任何一個封包只要壓縮完成,資料立刻傳送到網路上。由於高優先等級服務的資料會先完成壓縮,所以高優先等級服務傳送的次數較多,並分配到較大的頻寬。而在我們提出的方法中,低優先等級服務也享有比過往作法更高的資料傳輸率。

我們在 Intel IXP425 網路處理器平台 [13] 上實作提出的網路分享器設計。當網路分享器收到不同優先權的封包時,將依據優先權去選擇相對應的壓縮處理。我們所使用的壓縮演算法為 LZSS [11],此壓縮演算法可以依所設定的 sliding window size 去決定壓縮比及壓縮時間。Sliding window size 設定越大,壓縮比越大且壓縮時間也越長,反之,

sliding window size 越小,壓縮比變小,壓縮時間也縮短。我們期望達到兩個主要目的,讓有即時性需求的服務可以儘快完成,並且讓網路的總輸出量提升。故優先權越高的服務使用壓縮時間較小的壓縮處理,可以達到第一個目的;而優先權越低的服務使用壓縮比越好的壓縮處理,則可以達到第二個目的。根據我們的實驗結果,目前我們的系統雛形可以增加 15% 的資料傳輸率。

我們所提出的基於封包壓縮的網路分享器擁有兩個優點。第一,可以讓傳輸速率比實際上所能提供的更快;第二,可以節省大量的硬體成本。一般擴充網路頻寬需耗費大量的硬體成本,而我們的方法可以利用壓縮演算法使系統的整體網路傳輸速度高於實際的網路頻寬,藉此可以達到與擴充網路頻寬一樣的效果。

在接下來的章節裡,第二章將進行文獻回顧,討論負載平衡路由、品質保證服務和封包壓縮演算法的相關研究。第三章則是詳細描述所提出的網路分享器設計。第四章描述提出的網路分享器實作和實驗效能量測。最後,第五章是本研究論文的結論。

2. 相關文獻回顧

在本節,我們先討論三種可以用以解決網路分享器頻寬不足問題的作法。接著,我們討論網路封包壓縮的相關研究。

在解決網路分享器效能瓶頸的問題上,購買新的網路頻寬是最直覺的作法。不過根據台灣主要 ISP 業者中華電信在 2007 年 7 月的資料 [14],如果要由 45Mbps 網路升級到乙太網路速度的 155 Mbps,我們每個月必須付出至少兩倍以上的費用。在網路頻寬更高以及接線距離更遠的服務中,費率更是增加許多。例如,同一區域內 30 公里以上 155 Mbps 升級到 655 Mbps,每增加一倍頻寬每個月得增加 75% 的費用。因此,找到一個低成本的增加頻寬方法是有必要的。

負載平衡路由 [2][7][8][10] 也是一種可行的作法,相較上述直接購買更大頻寬的作法,我們可以改購買許多小頻寬的網路線路,然後透過負載平衡路由機制來實作大頻寬的網路分享器。在負載平衡路由的機制中,我們可以將多條低頻寬的線路整合成一條虛擬的大頻寬線路。在實際傳輸資料時,由於內部網路傳輸封包的資料量也許會超過單一條小頻寬線路能夠快速傳輸完畢的大小,此時我們必須將大封包切割成小封包再分送各個網路線路傳輸。切割的大小可以是 byte-level [2]、packet-level [8][10] 和 connection/call-level [7]。另外,由於有許多封包等待傳輸,我們可以用各種不同的排程演算法將封包分配給這些小頻寬的平行網路線路。最常見的作法就是使用輪詢 (round robin) 演算法來分配封包。

不過雖然使用負載平衡路由可以降低購買頻

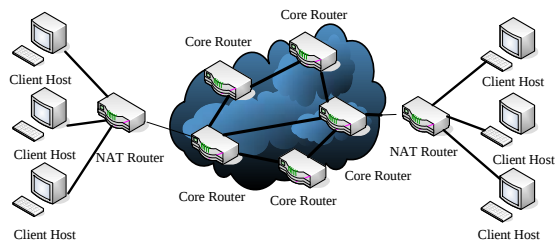


圖 1：網路環境

寬的成本，由於具有負載平衡路由功能的網路分享器多半相當昂貴，因此並不適合使用於小型公司、個人工作室或家庭。

第三種可行的解決方法是在網路分享器上採用品質保證服務。現行品質保證服務的作法是先為各種網路服務定義優先等級，然後為高優先權的網路服務保留足夠的頻寬。舉例來說，Fineberg 在 2002 年提出的系統利用 RSVP (resource reservation protocol) 來管理網路頻寬 [5]。路由器可以利用 RSVP 通訊協定通知客戶端設備降低網路速度，此外，路由器上運行優先權佇列 (priority queue)，並按照優先權傳送佇列的內容。

相較於我們提出的品質保證服務架構，過往的品質保證服務效能仍受限於實際的網路頻寬大小。而我們利用封包壓縮則可以提供比實際網路頻寬更大的資料傳輸率。因此，低優先權的服務能夠有更高的資料傳輸率。

根據我們的研究，將網路封包壓縮的概念近年來也相當受到關注。Shajeemohan 和 Govindan [9] 提出 Intelligent Dictionary Based Encoding (IDBE) 壓縮演算法，使封包的傳輸時間最多減少為未壓縮封包傳輸時間的 7%。而 Subathra、Sethuraman 和 Vinosh Babu [12] 進一步分析各種字典壓縮演算法在高速網路的效能表現。他們比較 LZ77、LZPI 和 MSR 三種演算法，並發現 MSR 為其中表現最佳的演算法。在本論文中，為了提供多版本的壓縮演算法，我們採用 LZ77 的改良版本 LZSS 演算法。未來我們希望能夠針對 MSR 演算法提供多版本的實作，並將之應用在我們的網路分享器上。

3. 基於封包壓縮之網路分享器設計

3.1 網路環境

如圖 1 所示，我們的網路環境中包含客戶端設備 (client host)、網路分享器 (NAT router) 和核心路由器 (core router)。由於客戶端到網路分享器的網路頻寬往往遠大於網路分享器到核心路由器的頻寬，為了妥善分配頻寬，網路分享器負責執行封包壓縮和品質保證服務通訊協定。

在我們的網路環境中，網路分享器負責壓縮和解壓縮傳送的封包，而核心路由器和客戶端設備的功能與現行網路中一致。

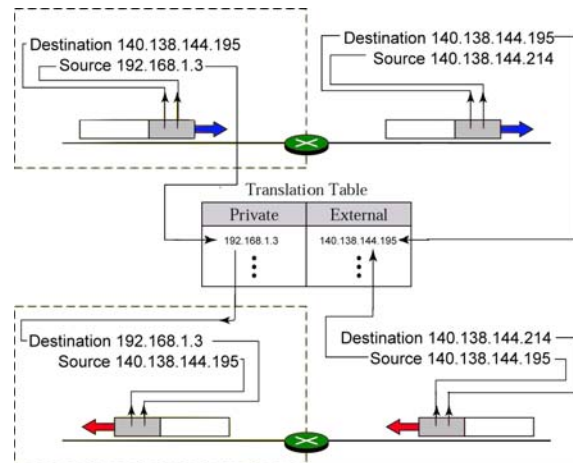


圖 2：NAT 演算法

3.2 NAT 演算法

在網路分享器的設計中，最基本的功能就是 NAT 演算法的設計。在我們的設計中，網路分享器利用一個表格來儲存內部網路電腦的網路位址與網際網路電腦的網路位址，並且記錄它們連接埠的對應。舉例來說，如果內部網路電腦 S (位址為 192.168.1.3) 透過連接埠 1024 連接到網際網路上的電腦 D (位址為 140.138.144.195) 連接埠 80，而當網路分享器 R (位址為 140.138.144.214) 收到來自內部網路電腦而目的地是網際網路上的電腦的封包時，網路分享器會在表格中新增一筆記錄 (192.168.1.3:1024, 140.138.144.195:80)，並且修改封包標頭部分，將來源 IP 欄位 (src_IP) 改為 140.138.144.214 後重新計算 checksum 值，並轉送封包給電腦 D。當電腦 D 收到封包後，會回傳訊息。在此訊息中，目的 IP 欄位 (dst_IP) 是 140.138.144.214，來源連接埠 (src_port) 是 80。當網路分享器 R 收到回傳的封包後，它會根據表格找出 140.138.144.195:80 對應到 192.168.1.3:1024，並將 dst_IP 改為 192.168.1.3，dst_port 改為 1024。然後重新計算 checksum 並轉送封包給電腦 S。到這裡為止為一個封包的往返過程。

3.3 LZSS 壓縮演算法

在我們的網路分享器設計中，我們採用 LZSS 壓縮演算法。LZSS 是 Storer 和 Szymanski 在 1982 年提出的 LZ 演算法改良版本 [11]，特點是利用指針的概念避免重複的字串組合，並且適合二進位資料的壓縮。為了提供不同壓縮時間和壓縮比的版本，我們以 sliding window 作為參數來控制壓縮時間。

壓縮過程中，為了達到較好的壓縮比，我們依照封包個數的多寡及 sliding window size 的大小去改善壓縮效率。因收集封包的個數可以讓資料的重複字元量上升，而 sliding window size 的大小則影

表 1：LZSS 演算法效能

Telnet + HTML		
W	Time	R
8	2	67.91
11	14	56.49
18	574	48.13

Online Game		
W	Time	R
7	1	68.38
11	6	43.23
15	74	43.46

Random		
W	Time	R
7	2	91.08
11	21	71.18
17	1172	65.47

響 LZSS 的字典大小，重複字元越多及字典大小越大可以讓壓縮比上升。因此，優先權服務越低的封包收集的個數越多，且所使用的 sliding window size 也越大，使壓縮比上升。但 sliding window size 亦不可過大，否則其效能沒有提升反而下降。

我們實際測試了不同的封包類型對於壓縮演算法的效能，我們收集網路上的封包並儲存成檔案，利用 LZSS 演算法壓縮後計算所得的數據，如表 1。在相同收集封包個數的條件下，分別以 sliding window size (W) 作為控制壓縮時間的參數，time 為平均每一個封包需要的壓縮時間，單位為毫秒 (ms)，R 為壓縮比 (壓縮後檔案大小/壓縮前檔案大小)。我們分別針對三種不同的網路環境進行封包取樣，第一個樣本的網路環境內的網路封包大部分屬於 TELNET 和 HTML 通訊協定，第二個樣本的網路環境內的網路封包大部分屬於網路遊戲，第三個樣本的封包則隨機分屬於各種網路服務。根據我們的初步研究，LZSS 演算法確實可以提供很好的封包壓縮比。同時，sliding window size 的大小不能太大，否則壓縮的效果也不好。

3.4 品質保證服務通訊協定

為了維持網路應用程式的傳輸品質，在我們的設計中，網路分享器負責利用品質保證服務通訊協定管理網路服務的優先權、LZSS 壓縮演算法的 sliding window 值、以及解壓縮時使用的字典。圖 3 為使用品質保證服務來傳輸封包的流程。每當客戶端設備送出封包，網路分享器會將封包收集到記憶體中。當累積到特定數量，網路分享器則用 LZSS 演算法將這些封包壓縮，並將解壓縮用的字典附在壓縮後的封包中，一併傳送到目的地。核心路由器不對封包做任何處理，僅僅負責將封包轉送到目的地。最後，目的端的網路分享器負責在將封包解壓縮後分送給各個客戶端設備。

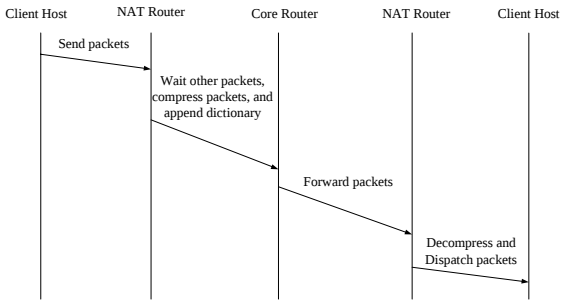


圖 3：封包傳輸流程

3.5 封包傳送排程演算法

為了對不同優先權的網路服務執行不同版本的壓縮演算法，網路分享器必須為各個優先權的網路服務提供各自的暫存佇列。一旦任何一個佇列的封包完成壓縮，網路分享器馬上將壓縮後的封包轉送到核心路由器。因為高優先權的網路服務的壓縮時間較短，因此有較低優先權網路服務更多的機會被轉送。換句話說，高優先權服務得以分配到更多的實際頻寬。

4. 網路分享器實作與效能分析

在本論文中，我們在 Intel IXP425 網路處理器開發平台上實作提出的網路分享器。Intel IXP425 處理器相容於 ARM V5T Thumb 指令集，主要應用在有線和無線網路環境。我們的硬體 I/O 介面包括三個乙太網路埠、一個 JTAG 埠、一個 RS-232 埠與數個 LED 燈號。我們在 uCLinux [3] 上自行開發此平台上的軟體，包括 NAT 演算法、壓縮演算法和品質保證服務。我們將實作分為下列四個步驟：

4.1 實作 NAT 演算法

實作網路分享器的第一個步驟是實作 NAT 功能。我們在 uCLinux 即時作業系統中，利用作業系統提供的 socket API 來開發應用程式層的網路分享器程式。我們將程式分為接受內部網路封包和接受外部網路封包。這兩個程式都利用 ioctl() 將網路卡設定為 promiscuous 模式，然後再將封包資料修改後轉送到目的網路區域。

4.2 實作 LZSS 演算法和品質保證架構

在我們的網路分享器中，我們採用不同的 LZSS 演算法來壓縮封包和實作品質保證服務。因為 LZSS 演算法的特色是壓縮的單位為 $2^n + 2$ bytes，即字典檔大小為 2^n 。我們就利用不同 n 值的 LZSS 演算法作為品質保證服務的核心程式。除了利用不同大小的 n 值來控制壓縮時間，我們為了增加出現相同資料的機會，在低優先權應用程式的壓縮演算

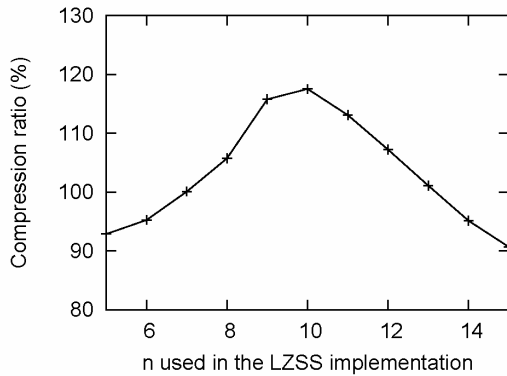


圖 4：網路分享器之封包壓縮效能

法中，網路分享器會收集 p 個封包後才一起用 LZSS 演算法壓縮。

在我們的品質保證服務中，我們將封包分成即時性封包與非即時性封包。即時性封包不需要經過壓縮處理，因此享有極高的傳輸效能。一般封包不能直接傳送，而必須累積到 p 個封包後，由網路分享器一次壓縮後送出。這樣才能改善壓縮率。而有即時性需求的封包則不在此限，當封包到達網路分享器馬上就會送出。

4.3 測試效能

由於我們利用 LZSS 演算法對網路封包進行壓縮和解壓縮，我們的系統會增加資料傳輸率和造成封包處理的延遲，壓縮演算法對效能的影響如圖 4 及圖 5 所示。圖 4 是使用 LZSS 演算法後網路封包的壓縮率（原本封包大小/壓縮後封包大小）。從實驗結果看來，基於壓縮演算法的品質保證服務可以增加 15% 的資料傳輸率。同時， n 值介於 9 到 11 擁有較佳的壓縮率。

圖 5 是網路分享器在不同壓縮封包單位 p 值與封包延遲的關係。結果顯示，當 $n=10$ ，隨著 p 值的

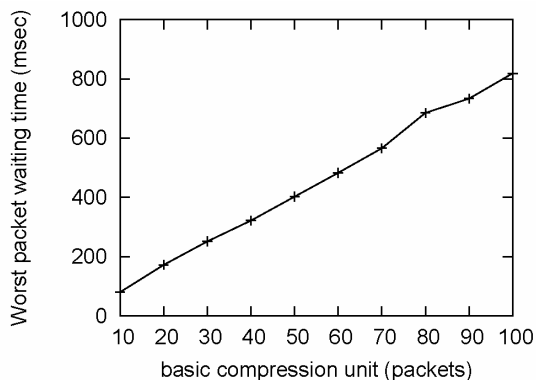


圖 5：網路分享器之封包傳輸延遲

增加，封包延遲的情形也漸趨成長。當以 100 個封包為處理單位時，這一百個封包中最長的延遲約達 0.8 秒。

5. 結論

網路分享器是目前網路設備中不可或缺的成員，主要原因是因為大家希望可以共享網路頻寬，節省上網成本，以及解決 IPv4 位址不足的問題。不過，由於扮演內部網路封包與網際網路間的傳輸橋樑，網路分享器往往成為網路傳輸的瓶頸，導致整體網路傳輸效能受限於網路分享器的頻寬。在解決此效能低落的問題上，我們希望可以避免花費購買網路頻寬的成本，又希望當整體網路頻寬需求成長時仍然能夠提供一定服務品質，因此，提出一個兼具低成本增加頻寬和品質保證服務的機制是有必要的。

在網路傳輸的技術中，運用壓縮演算法壓縮傳輸的封包可以達成避免花費購買網路頻寬成本的目標。同時，為了保證網路服務的品質，我們可以結合品質保證服務，為不同等級的網路服務配置不同壓縮時間和壓縮比的演算法。對高優先權的服務來說，我們配置低壓縮時間的演算法，這些高優先權的服務不會因為壓縮而造成大量的延遲。對低優先權的服務來說，我們配置高壓縮能力的演算法，然後數個封包一次傳送，低優先權的服務可以享有比過往方法更高的資料傳輸率。

在本論文中，我們提出一個基於封包壓縮的網路分享器設計。在我們的網路分享器設計中，我們在網路封包的傳遞過程中加入封包壓縮演算法，根據網路服務的優先權給予不同壓縮演算法，並利用不同演算法間的壓縮時間差來提供各種網路服務不同大小的網路頻寬。在我們的設計中，高優先等級的網路服務使用壓縮時間少的壓縮演算法，而低優先等級的網路服務使用壓縮能力高的壓縮演算法。任何一個封包只要壓縮完成，資料立刻傳送到網路上。由於高優先等級服務的資料會先完成壓縮，所以高優先等級服務傳送的次數較多，並分配到較大的頻寬。而在我們提出的方法中，低優先等級服務也享有比過往作法更高的資料傳輸率。

為了驗證我們的網路分享器效能，我們在 Intel IXP425 網路處理器平台上實作提出的網路分享器設計。我們所使用的壓縮演算法為 LZSS，此壓縮演算法可以依所設定的 sliding window size 去決定壓縮比及壓縮時間。Sliding window size 設定越大，壓縮比越大且壓縮時間也越長，反之，sliding window size 越小，壓縮比變小，壓縮時間也縮短。根據實驗的結果，當 LZSS 演算法的壓縮單位為 2^9+2 bytes 到 $2^{11}+2$ bytes 間會有最佳的壓縮比，約可增加 15% 的資料傳輸率。

目前我們的系統採用的 NAT 演算法僅考慮到一般 HTTP 傳輸等等僅牽涉傳輸層和網路層封包標頭修改的網路應用。對於像 FTP PASV 模式或其他

需修改應用層標頭內容的網路封包來說，我們希望在接下來的時間根據個別網路應用來補強我們的 NAT 演算法。此外，在我們的研究過程中，我們也發現 MSR 壓縮演算法可以擁有較 LZSS 演算法更加優異的壓縮效能。我們希望接下來可以針對 MSR 演算法提出多種壓縮時間的實作版本，並結合我們的網路分享器，提供更好的效能。

參考文獻

- [1] C. Aurrecoechea, A. Campbell, and L. Hauw, "A Survey of QoS Architecture," *Multimedia Systems*, vol. 6, no. 3, May 1998, pp. 138-151.
- [2] J. Duncanson, "Inverse Multiplexing," *IEEE Communications Magazine*, vol. 32, no. 4, Apr. 1994, pp. 34-41.
- [3] M. Durrant and J. Dionne, uCLinux, <http://www.uclinux.org/>.
- [4] K. Egevang and P. Francis, "The IP Network Address Translator (NAT)," **RFC 1631**, May 1994.
- [5] V. Fineberg, "A Practical Architecture for Implementing End-to-End QoS in an IP Network," *IEEE Communications Magazine*, vol. 40, no. 1, Jan. 2002, pp. 122-130.
- [6] Y. Jia and M. Chen, "A New Architecture of Providing End-to-End Quality-of-Service for Differentiated Services Network," *Proc. of 2001 Military Communications Conference (MILCOM)*, Oct. 2001.
- [7] N. Maxemchuk, "Dispersity Routing on ATM Networks," *Proc. of IEEE INFOCOM*, Mar. 1993.
- [8] Y. Liu and L. Ni, "Dynamic Dispersion Framework for Router Load Balancing," *Proc. of 11th International Conference on Parallel and Distributed Systems (ICPADS)*, July 2005.
- [9] B. Shajeemohan and V. Govindan, "Compression Scheme for Faster and Secure Data Transmission over Networks," *Proc. of 2005 International Conference on Mobile Business (ICMB)*, July 2005.
- [10] C. Staff, "Gigabit Network Testbeds," *IEEE Computer*, vol. 23, no. 9, Sept. 1990, pp. 77-80.
- [11] J. Storer and T. Szymanski, "Data Compression via Textual Substitution," *Journal of the ACM*, vol. 29, no. 4, Oct. 1982, pp. 928-951.
- [12] S. Subathra, M. Sethuraman, and J. Vinosh Babu, "Performance Analysis of Dictionary based Data Compression Algorithms for High Speed Networks," *Proc. of 2005 IEEE Indicon*, Dec. 2005.
- [13] Intel Corp., "Intel XScale Technology Overview," <http://www.intel.com/design/intelxscale/>.
- [14] 中華電信, "市內數據電路月租費," <http://www.cht.com.tw/BusinessCat.php?Module=Fee,Describe&CatID=360&PageID=792>