Distributed Netflow Analysis for Limiting User Data Rates

林鳳銘　　吳守豪　　李蔡彥
國立政治大學電算中心
l96in@nccu.edu.tw, swu@nccu.edu.tw, li@nccu.edu.tw

摘要

隨著網際網路的發展，網路資源的有效管理與運用，一直都是十分重要的課題。網路流量收集與分析的工具已日趨成熟，但大部分此類工具的系統架構，多將收集與分析的工作集中在同一台主機上。當病毒肆虐或在網路流量較高的主幹上，此類系統架構常有擴充性不足的情形發生，使得正常的收集與分析工作無法順利進行。因此在本論文中，我們提出了一個分散式計算的架構，重新實作流量分析程式，以依照需要將某一網段或某些特定封包的流量資訊，即時傳送給其他專屬的資料分析伺服器。此方式可將流量收集與分析的負載，分散至多個伺服器，以有效提高流量分析品質。另外，我們嘗試將此分析所得的資料，做初步的流量管制應用。換言之，我們將分析出具有異常流量的主機資訊，回饋給路由器，以自動進行限速的設定，並主動通知主機使用者改善。我們以此實做出的系統進行實驗，證明這個自動調整流量的方式，能有效改進網路管理品質。

關鍵詞：分散式流量分析，Netflow，流量限制，降速。

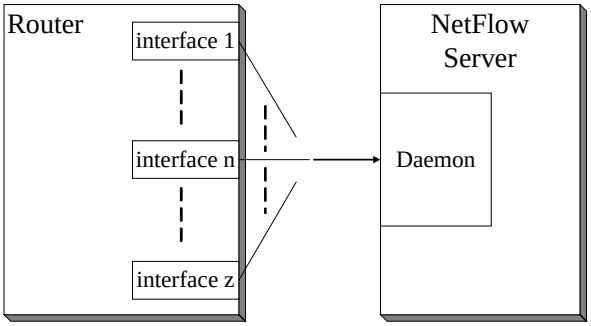
1. 簡介

由於網際網路不斷發展，連結上網際網路的機器數量以驚人的速度成長，而一般的區域網路中連接的主機也在不斷的增加。隨著主機數量的增加，網路的流量亦自然隨之提高，而如何分析網路的使用情形，以善用網路資源，便成了十分重要的課題。通常為了計算網路上的流量，許多網路中心都會架設流量伺服器，以收集網路上流量的資訊[2]；之後再根據此流量資訊進行詳細的分析，以找出網路上發生的一些異常狀況[1][3][9][10]。根據不同的異常狀況，管理者可以再設計不同的處理方式以因應之。這些網路流量統計可以用來做為網路計價的依據[12][13]，也可以用來分析網路的特性[8]。例如，[4]就以流量研究出網路的一些不正常的使用狀況，而 [7] 則提出了利用網路流量來進行被動式的資訊流行為偵測。

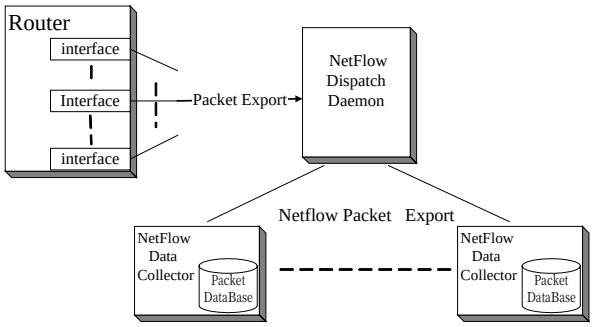
目前這些網路流量分析的工作，多藉由收集流量資訊的主機來執行。對於流量不大或是網路規模較小的單位而言，單一主機的架構或許足以應付；但是當網路流量成長到一定的程度時，要在同一部機器上同時收集並計算網路流量，就會顯得有些吃力。例如，網路上發生病毒、 spam、或是想要即時將流量資訊存入資料庫以便稍後分析時，大量的運算將會拖慢這台伺服器，甚至發生來不及運算的情況。

為了解決上述的狀況，在本論文中，我們提出了一個分散式流量分析系統架構，以將某一網段或某些特定封包的流量資訊，即時傳送給其他專屬的伺服器上。在文獻上，已有許多流量資訊收集與分析的系統被提出來，其中多數是以單一主機收集 Netflow 功能所提供的流量資訊；其中在[6]可以找到以 C 語言實做的這一類流量收集程式。為了達到分散式流量處理的目的，我們必須將單一主機收集到的流量資訊傳送到一個以上的流量分析主機。為此，我們重新實做了收集 Netflow 流量的程式。如此一來，我們不但能有效降低接收路由器封包的伺服器之負載，同時也可以即時針對一些異常的網路行為或主機執行更詳盡的流量分析。

過去在網路管理（特別是學校宿網的管理）上，對於即時流量過大的主機，所採取的制裁手段通常是「鎖卡」（禁止該主機連上 Internet）。這個處理的方式最有效也最直接，可以即時阻止網路頻寬被該主機大量佔用。雖然管理者可以事先以 E-mail 通知該主機的使用者，不過由於鎖卡後剝奪了該主機的上網能力，因此一旦該主機的使用者沒有在事前收到 E-mail 就被停止連上 Internet 的能力時，該主機的使用者可能認為網路出了問題，以致無法連通 Internet，因此也無法透過網路瞭解該主機無法上網的原因。為了解決這個問題，本論文提出了一個比較折衷的處理方式。我們根據分散式流量分析，對於流量過大的主機採取降速（降低該主機使用網路頻寬的速度）的措施，並將降速的資訊即時於網上公布。此時由於只是上網的速度變慢了而不是再也無法連接網路，因此該主機的使用者可以很快的察覺到異狀，並且透過網路查知該主機



圖一、集中式流量資訊的收集



圖二、分散式流量資訊的收集

是否已被限速，而採取必要的補救措施。如此一來，將可以同時保障該主機與同一網段底下網路使用者的權益。以下謹就分散式流量資訊收集及即時限速的方式進一步提出我們的作法。

2. 分散式網路流量資訊的收集

2.1 系統架構

路由器是網路封包的交換中心，因為網路上的訊息需靠路由器的傳送以達到目的地。因此，在適當的路由器上，我們可以有效的收集到網路上的流量資訊。一般而言，常見的流量資訊收集架構如圖一所示[1]。此流量收集架構，屬於集中式架構。圖中的 Netflow 伺服器負責收集路由器送出的封包資訊，同時該伺服器也負責路由器封包的分析運算。這個架構適用於規模較小而且網路負載較輕的網路，用來做統計計價或偵測網路異常使用現象。然而，對於區域網路中心而言，除了需要分析自己的流量資訊外，也必須一併分析底下所連接網路的流量，因此所需處理的資料量極大。這些流量資訊的大小，可以大到每十分鐘數百萬位元組之多。為了避免佔用過多的磁碟空間，我們常會以即時壓縮的方式處理接收到的流量資訊。如果我們同時想在這部 Netflow 伺服器上即時進行流量資訊分析，則必須將此壓縮的流量資訊作解壓縮後方能進行。即時運算與分析這麼大的流量資訊將會使得該伺服

Source IP Address	Destination IP Address	Source Port	Destination Port	Protocol	Packets	Octets

Source IP Address：資訊流的來源端 IP 位址。
Destination IP Address：資訊流的接收端 IP 位址。
Source Port：資訊流發送端程式所使用的埠號。
Destination Port：資訊流接收端程式所使用的埠號。
Protocol：資訊流所使用的通訊協定。
Packets：資訊流等於多少個封包。
Ocets：資訊流的大小。

圖三、所收集的特定流量資訊格式

器的負載提高，甚至發生漏接路由器所送出流量封包的情況。如果同時又遭受病毒的攻擊（例如不斷以小封包偵測可攻擊的 web 伺服器），流量的資訊就更可觀了。為了解決這種情況，同時使用資料庫來儲存經過計算後的即時流量資訊，我們提出了圖二所示的分散式流量分析架構。

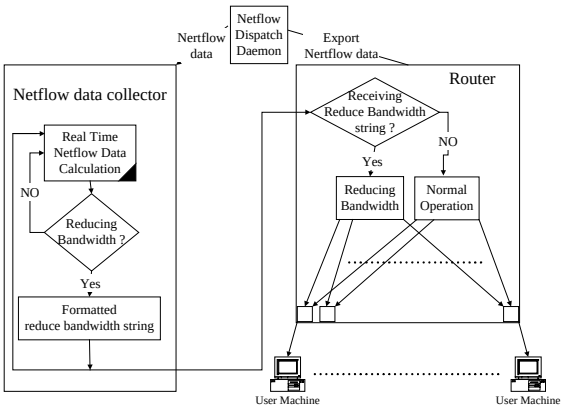
根據圖二的分散式的流量分析架構，我們可以發現這個架構與[1] 所使用的流量收集方式最大的差別在於 Netflow 伺服器上只執行了一個 Netflow Dispatch Daemon，這個 Daemon 負責即時收集路由器送傳送出來的流量資訊，同時它可以根據使用者需求將一些特定流量資訊透過網路傳送給遠端的 Netflow Data Collector 上。由於這些 Netflow Data Collector 所收到的流量資訊屬於篩選過的特定流量資訊（如下一小節說明），因此流量資訊較小，有助於稍後的分析與計算。

2.2 流量資訊的格式

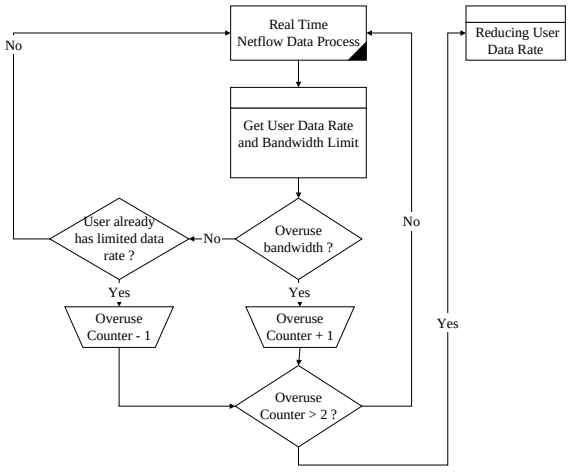
Netflow 是由 Cisco 所提出的專屬技術規格[5]，因而所傳送出來封包的內容也受到該規格的規範。我們所重寫的分散式流量分析程式，僅擷取了 Netflow 規格中的部分必要欄位，所抓取出來的資訊欄位如圖三所示。

我們所收集的流量資訊與[1]不同之處在於分散式流量資訊少了資訊流發送端路由器介面編號與資訊流接收端路由器介面編號。對於流量分析而言，我們並不在意所分析的流量流經路由器那一個介面編號(interface number)。

我們已實做完成此一分散式流量資訊收集與分析系統，並完成了初步的實驗測試。為了縮短系統開發所需要的時間，此系統雛形係以 Perl 語言[11]撰寫，以簡化程式開發時間，換取執行上的效率。



圖四、限制使用者網路頻寬之架構圖



圖五、網路連線降速之演算法

3. 使用者上網速度的限制

網路是一個開放式頻寬共享的環境，任一位連上網路的使用者，都期望能盡可能地使用網路的頻寬。由於網路的頻寬是固定的，因此在同一時間內，如果大家同時競相使用網路頻寬，則必有排擠效應。使用到較少網路頻寬的使用者甚至會以為網路變慢或故障了。網路管理者最不希望發生的事，便是在同一網段上有人架設非法的各式網站。由於架站的伺服器是以背景的方式來執行，因此只要頻寬許可，這個架站的伺服器最有希望使用較多的網路頻寬。倘若發生了這種情況，同一網段內的使用者將會發現網路變慢了。如果這個網段又是以集線器(hub)來串接，則碰撞(collision)的效應，將使網路變慢情形更加嚴重。

為了找出使用過量網路頻寬的使用者，我們可以藉由分散式流量統計來完成。我們將特定網段的流量資訊匯出到某一部 Netflow Data Collector 上，同時在這部伺服器上詳細分析流量資訊，以找出過量頻寬的使用者。當我們找到了使用過多頻寬的機器時，為了避免同一網段上的使用者受到影響，我們通常會採取的手段是「鎖卡」，以切斷該部機器上網的管道，使其與網際網路隔絕。這個方式可以解決了同一網段上其他使用者上網速度慢的問題，但卻不見得可以透過網路通知到使用者進行改善。因此，在這裡我們提出的另一個以「降速」為主的解決方式。由於我們不斷即時計算網路流量，一旦發現有機器開始佔用大量的網路頻寬時，我們的流量統計伺服器就可以開始即時監控該主機接下來的動作。如果此主機一直都使用過量的頻寬，我們就會主動地限制該主機可以使用的網路頻寬大小。

3.1 系統架構

為了達成即時自動降低使用者佔用大量網路頻寬的動作，我們所採用的網路架構，如圖四所示。圖中的多個 Netflow Data Collector 持續接收 Netflow Dispatch Daemon 所傳送出來的流量資訊，並即時計算流量資訊。一旦發現有使用者過量使用網路頻寬時，立即將流量過大的使用者 IP 記錄下來，然後再依據路由器所需要的設定方式加以封裝，透過網路傳送到路由器上。而路由器除了為底下串接的使用者傳送一般封包外，還會根據 Netflow Data Collector 所送過來的指令，對需要降速的主機，進行限制頻寬的降速動作。降速是透過路由器的設定達到的。由於資料流速檢查需要路由器上的 CPU 執行額外的計算，因而加重路由器的負載，因此限速的主機數量需有所限制。以實驗中的 Cisco 6509 路由器而言，由實際的經驗得知當需要被降速的 IP 在不超過 100 個的情況下，路由器的 CPU 負載仍可低於 10 %，因此額外的計算負載仍在可接受的範圍內。

雖然上述的降速設定是針對主機 IP 而做的，但比較嚴謹的說法應是根據 MAC 位址來降速。只是目前實驗的環境是宿舍網路，而我們宿舍上網的先決條件是必須由住宿生上網登記 IP 與 MAC 位址的對照關係。在這種情形下，由於 IP 位址與 MAC 位址是一對一的關係，所以在這裡執行的降速動作對 IP 與 MAC 位址而言是相同的。

3.2 降速演算法

就目前而言，一般的區域網路都屬 Ethernet 的網路環境，串接的方式多為以集線器(hub)串連。因此以下我們將以 Ethernet 為例說明。對一個網路上的主機而言，需要被降速的原因包含以下兩點：第一、當使用者所使用的流量超過 Ethernet 頻寬的一定比例(例如 30 %，則初始流量上限為 10Mbps * 30%= 3 Mbps)；第二、當該主機或許由於受到病毒感染，而不斷地在網路上傳送攻擊性的小封包，導

致網路上到處充斥著一些沒有機器回應的封包資訊。兩個原因只要其中之一成立，便構成了降速的起始條件。

本研究所用的降速演算法，如圖五所示。假設某一主機 i 的即時流量經過計算後為 f_i。我們首先檢查該主機 IP 是否存在於被降速表格中。如果是，則系統就會一併計算該主機被降速後的頻寬上限 h_i；否則將 h_i 設定為該主機之初始流量上限 h_{max}。如果該系統超量使用頻寬(f_i>h_i)，則我們會把該系統的超量計數器(c_i)增加，以得到連續超量的次數。如果 c_i 未超過連續超量次數上限 c_{max}，則系統目前毋須改變設定；但是如果 c_i 大於或等於 c_{max}，(c_i >= c_{max})，則表示這個 IP 連續超量使用網路頻寬。此時系統會將頻寬上限再度打折(h_i=max(h_{min}, h_i*r), 0<r<1, r 是速度調降的比例)，並將此降速資訊格式化成路由器可以接受的格式，再傳給路由器以即時執行降速的動作。

假設目前即時收集的主機流量顯示該 IP 未超量使用頻寬上限，則超量計數器(c_i)可以歸零，並且系統會針對該 IP 是否已經被降速而採取必要升速措施。換言之，h_i=min(h_{max}, h_i/r), 0<r<1。如果 h_i = h_{max}，則代表該主機已升速至起始速度，此時我們可以將此 IP 從限速表中剔除。

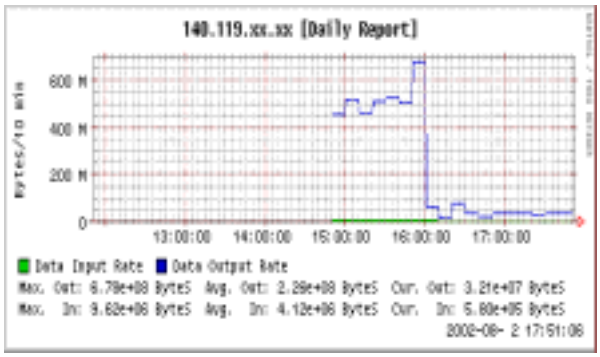
3.3 實際範例

我們架設了一台測試用的主機來驗證我們所提出架構的可行性。我們在這部主機上架設了一個 FTP 伺服器，然後透過網路中的其他台主機不斷地從這台 FTP 伺服器上抓取資訊，以造成這部測試主機過量使用網路頻寬的條件。經過 Netflow Data Collector 計算後發現該主機的行為異常，因此需降速以限制頻寬。我們將降速的條件格式化後，傳送給相對應的管理路由器上，再由路由器根據降速條件執行相對應的降速動作。

圖六是我們實驗結果的圖形輸出。圖六中的 Y 軸表示每十分鐘該主機的流入與流出量(藍色代表流出量而綠色表示流入量)，X 軸則為時間。圖六中的主機(IP:140.119.xx.xx)，在 15:00 與 16:00 一小時內不斷地流出資訊，平均每十分鐘的流量都大於 400 MB(等於 400 * 8 * 10^6/600 = 5.33 Mbps)，當我們所設計的系統發現這個狀況時，在 16:00 通知相對應的管理路由器進行降速的動作。從圖六看出該測試主機從 16:00 之後每十分鐘的資料傳出量已經小於 25 MB (25 * 8 * 10^6/600 = 0.33 Mbps)，這部主機的資料傳輸量已明顯地降低。

4. 結論

分散式流量統計的優點在於可以將接收到的



圖六、顯示網路連線降速之流量圖

流量資訊計算分散到一個以上的主機，而不必侷限於 Netflow 伺服器的運算能力，同時能夠降低 Netflow 伺服器的系統負載。除了減低系統負荷外，也能較輕易做到異質性的資料分析工作；例如，可以為特定網段或某一部主機，採取不同方式的流量計算方式。這種分散式流量統計，對於流量極大的區域網路中心而言，十分重要。它可以將所串接的連線單位事先分類，然後分別將流量資訊傳送到不同的 Netflow Data Collector 上作個別計算，以避免因資訊量過大而發生無法即時運算的問題。

網管人員在面對使用者過量使用網路頻寬時，除了將該超量主機隔離外，還可以根據需要即時地降低該主機可以使用的頻寬。本論文便提出了這樣一個即時降低流量上限的方法。網路存取的速度在降速後會變慢，如果該主機的使用者此時也在線上操作這部主機，他將可以明顯地感覺到差異性。這個作法的優點是讓該使用者在仍可上網的情況下，透過網路瞭解自己經過量使用網路資源，並且尋找解決之道。

本文中所建立的分散式流量分析機制，是根據來源端與目的地端的 IP 位址來分析流量，降速的動作則交由路由器來執行。這樣的作法多少會加重路由器的負載。由於需要被降速的 IP 是以存取表單的方式交由路由器來循序執行，因此當表單越大時，路由器的負載也會增加。所以如何決定一個適合的表單大小，以不嚴重影響路由器效能，將是一個有待後續實驗的重要問題。

參考文獻

[1] 林鳳銘、吳守豪、李蔡彥, “Intrusion Detection: a Network View”, in *Proceeding s of TANet 2001 Conference*, pp. 34-48, 2001.

[2] 范修維, “網路流量分析系統之建置及應用,” in *Proceedings of TANet 2001 Conference*, pp. 419-424, 2001.

[3] 黃文穗、林守仁, “利用 Netflow 建置 Code Red Worm 偵測系統,” in *Proceedings of TANet 2001 Conference*, pp. 471-474, 2001.

[4] P. Barford and D. Plonka. “Characteristics of Network Traffic Flow Anomalies”, in *Proceedings of ACM SIGCOMM Internet Measurement Workshop*, Nov. 2001.

[5] Cisco’s IOS NetFlow Feature, http://www.cisco.com/warp/public/732/netflow/

[6] ftp://ftp.eng.oar.net/pub/flow-tools

[7] C. Fraleigh, C. Diot, B. Lyles, S. M., P. Owezarski, D. Papagiannaki, and F. Tobagi. “Design and Deployment of a Passive Monitoring Infrastructure,” in *Proceedings of PAM 2001 Workshop*, April 2001.

[8] A. C. Gilbert, Y. Kotidis, S. Muthukrishnan, M. Strauss, “QuickSAND: Quick Summary and Analysis of Network Data,” *DIMACS Technical Report 2001-43*, 2001.

[9] D. Moore, G. Voelker, and S. Savage, “Inferring Internet Denial-of-Service Activity,” in *Proceedings of 2001 USENIX Security Symposium*, Washington, DC, August 2001.

[10] B. Nickless, J.-P. Navarro, and L. Winkler, “Combining Cisco NetFlow Exports with Relational Database Technology for Usage Statistics, Intrusion Detection, and Network Forensics”, *Usenix*, LISA, 2000.

[11] Perl, http://www.perl.com.

[12] S. Shenker, D. Clark, D. Estrin, S. Herzog, “Pricing in Computer Networks: Reshaping the Research Agenda,” *Telecommunications Policy*, vol. 20, No. 3, April 96.

[13] F. Travostino, “Towards an Active IP Accounting Infrastructure,” in the *Proceedings of the OpenArch 2000 Conference*, 2000.