在差異性服務上的一個具適應性封包預先標註方法
An Adaptive Packet Pre-marking Method in DiffServ

陳耀庭* 黃文祥* 柯志亨** 謝錫堃**
Yao-Ting Chen Wen-Shyang Hwang Chih-Heng Ke Ce-Kuen Shieh

國立高雄應用科技大學電機工程系*
國立成功大學電機工程研究所**

摘要

本文針對 TCP 應用軟體提出一個建構在差異性服務網路上以維持端點到端點間 Throughput 的控制方法，名為 PPE(Packet Pre-marking Engine)。PPE 為一個執行在使用者端應用層上的程式，它可依據網路負載的狀態而適當地設定 DSCP(DiffServ CodePoint)欄位內容，使核心網路設備以其指定的服務品質來達到使用者的需求。在網路為輕載時，只須提供 Best-effort 服務便能達到使用者的需求，而當網路負載變重且無法達到需求時，則才需要使用較好的付費服務。本文將在一個差異性服務的實驗網路的 FTP client 上實作 PPE，以證明所提出的控制方法是有效而實用的。

一、 前言

隨著網際網路應用的蓬勃發展，龐大的資料傳輸將造成網路頻寬不足，導致封包傳輸時間的增加及丟棄機率的提高；且由於網際網路原本的傳輸服務 Best-effort 是屬於沒有優先等級的服務，無法滿足多媒體及即時性(Real-time)應用的需求，也不能提供不同服務品質(QoS，Quality of Service)給願意付費的使用者。因此近年來有關如何提供網路服務品質的研究成為一個重要而熱門的課題；目前 IETF(Internet Engineering Task Force)已訂出兩種不同網路服務，分別是整合式服務(Integrated Service；IS)和差異性服務(Differentiated Service；DS)。IS 架構採用 Per-flow 的分類方式，藉由分配及管理網路資源的機制來提供每一條 Flow 所需的服務品質，而所謂的 Flow 是指一連串具有相同來源與目的位址的 IP(Internet Protocol)封包。此架構的缺點是路由器必須去維護每個 Flow 的狀態，因而 Flow 的數量太多會造成路由器負擔太重，即所謂的 Scalability 問題，DS 之所以被提出就是要來改善這個缺點。DS 簡化了 IS 的架構採用 Per-class 的分類方式，其運作模式是將流量在網路邊界路由器(Edge router)上先行分類，依封包的不同特性分成為一個個「行為聚集」(Behavior Aggregate)，每個行為聚集靠著單一 DSCP(Differentiated Service CodePoint)來做識別。在網路核心部分，核心路由器(Core router)則只須依不同的 DSCP 值做所對應之行為 PHB(Per-Hop Behavior)來傳送封包即可。

Transmission Control Protocol (TCP)是目前最廣泛被使用在 Internet 的傳輸層協定，提供許多應用層軟體如 Web 和 FTP 等可靠的網路傳輸機制，補足了 IP(Internet Protocol)協定不可靠的問題。但其提供的 Best-effort 服務並不能完全滿足網路使用者和 ISP (Internet Service Provider)業者的需求。ISP 業者希望能藉由提供比 Best-effort 更好的服務品質來增加收入，相對地使用者也希望透過付費來讓一些緊急重要的資料以較好的服務品質傳輸；因此有一些建構在差異性服務網路上以提高 TCP 效能且又能保證最低的傳輸速率之流量管理和封包標註技術的研究[1-3]相繼被提出。但是這些研究大多把焦點放在網路上，很少針對應用層程式來討論；然而應用層程式卻是決定一個技術是否可以被接受的重要角色。在本文中網路被假定為可提供了三種不同服務等級，它們是依 TOS (Type of Service)欄位來區別；文中將提出一個運用於應用層(Application layer)的控制方法稱為 PPE (Packet

pre-marking engine) 來維持網路端點到端點 (End-to-end) 的 Throughput，及盡可能降低服務使用費用。PPE 是利用 TCP 的特性以測量 TCP 回應封包的來回速率來評估網路目前的狀態;當要傳送下一個封包時,便根據 PPE 所量測到的網路狀態，透過本文提出的「2bits_3levels」方法來動態調整 TOS 欄位值。在網路是輕載且測量到的傳輸速率高於需求的速率，則以 Best-effort 服務品質來傳送封包。若網路負載變重且量測到的傳輸速率低於最小傳輸速率時，才調高的傳輸的服務等級，以計費的方式傳送封包。

以下為本文的章節架構：第二節將敘述 PPE 的原理，傳輸速率監控器，及封包標註判斷器。接著第三節描述實驗的結果,最後為本文的結論及未來工作。

二、 Packet Pre-marking Engine (PPE)

PPE 主要由兩個部分組成,分別是傳輸速率監控器和封包標註判斷器。傳輸速率監控器會去監測並計算出 TCP 應用程式資料從 User space 複製到 Kernel space 的速率。其實資料的複製速率和傳輸速率有密切的關係，可說是同指一件事情，所以在接下來內容中將交替使用兩者。傳輸速率監控器把測量到的資訊傳送到封包標註判斷器,判斷封包是否需要做標註。它是依據量測到的傳輸速率和最低傳輸速率做比較，依據 2bits_3levels 方法來做標註處理,達到使用者需求的傳輸速率及維持最少的封包標註數量以減少使用者的花費。

傳輸速率監控器

傳輸速率監控器主要功能為測量資料從 User space 複製到 Kernel space 的複製速率。其測量的資訊反映了網路端點到端點間的流量負荷情形,這資訊可提供封包標註判斷器做為設定封包標註內容的決定。前面曾提到傳送端用 TCP 來傳送資料，它首先將資料維持於 Kernel space buffer 中，直到收到接收端傳來回應為止。但是這個傳送的 Buffer 是有限，在網路端點之間的負載流量很大時,接收端傳回的回應訊息的速度將會很慢，相對地傳送端的複製速率也會因此而慢了下來；因為 Buffer 已經被那些尚未接收到回應訊息的資料填滿了。相反地，若網路端點間的負載流量不大，則從接收端傳回的回應訊息速度將會很快,傳送端的複製速率也因而變得很快，因為 Buffer 中等待回應訊息的資料很快就被清除了。從一個網路應用程式的觀點來看，資料的複製速率是很容易取得的，在 TCP 上用來傳送資料的函式 write()，每當它被呼叫時將會回傳實際由 User space 複製到 Kernel space 的資料大小(bytes)，然後間隔的傳輸速率就可得知。將全部的資料大小除以整個傳輸時間就可得到平均的傳輸速率，如下面式子所示

$$\text{平均傳輸速率} = \frac{\text{全部資料大小}(bytes)}{\text{傳輸全部資料所花時間}}$$

封包標註判斷器

封包標註判斷器的主要目的是用來根據所測量到的傳輸速率以適當的調整封包被標註的比率,最簡單的實做方法是採用 2 level 的標註策略，在這個策略中,當測量到的傳輸速率高於使用者需求的速率時，採用 Best-effort 的服務；而當測量到的傳輸速率低於使用者需求的速率時,將使用高優先等級的服務。這個方法很容易就可實現，但會產生一個問題就是當使用高優先等級的服務去取代 Best-effort 服務時，當網路流量變較輕載時封包標註判斷器不知道如何變回使用 Best-effort 服務。所以在[1]中提出了一個機率性(probabilistic)的標註方法，在此法中封包會根據 PPE 傳來的資訊隨機的標註封包。而標註的機率(probability，***prob***)則會定期的依據所量測的速率和使用者需求速率之差距來更新，圖 1 為一個使用此法的簡單演算法。以一個例子來說明,假設當傳輸速率低於使用者需求速率時初始被標註比率是 70% ，而使用者需求的速率為 100Kbytes/sec。接著如測量到的速率為 90 Kbytes/sec，則 ***prob*** 會變成 73% ，而如測量到的速率為 110 Kbytes/sec，則 ***prob*** 會變成 27% 。這個

方法的優點是當傳輸速率高於使用者需求速率時能快速的減少 ***prob***，這有助於盡可能性的減少標註的封包數。當每次得到 ***prob*** 時，隨機數產生器會隨機產生一個數去和 prob 比較然後去決定是否要將封包標註。如果需要改變服務等級，則 setsockopt()函式會被呼叫去改變 IP header 中 TOS 欄位值。

```
Every update interval
If ( transfer_rate  < target_rate ){
        scale = ( target_rate- transfer_rate ) / target_rate;
        prob = porb + scale * ( 100 - prob );
}
else {
        scale = ( transfer_rate - target_rate ) / transfer_rate;
        prob = ( 100 - prob ) * ( 1 - scale) ;
}
```

圖 1. 封包標註判斷器的演算法例子

可是這個方法有一個缺點，就是變動太快，導致傳輸速率的震盪將很大，容易影響網路的穩定度。其主要的原因是只要一偵測到傳輸速率不同於使用者需求速率時，就會馬上去改變服務等級，利用這方法雖然能盡量減少標註的封包數，但也產生了震盪的問題。

有鑑於此，為了改善震盪的問題且又能盡可能的有效降低標註的封包數，採用了有別於機率性(probabilistic)標註策略且較簡單的方法，稱為「2bits_3levels」，所謂的 2bits 是指 00、01、10、10 這些狀態標示值，而 3levels 是三種不同的服務等級，由高到低的 DSCP 值分別是 0xa8、0x68、0x00。這個策略的運作原理是當偵測的傳輸速率與使用者需求的速率有差距時，並不會馬上改變服務等級，而是改變狀態標示值。00 狀態標示值代表服務等級將改變為較低一等級，11 則表示將把服務等級改成較高等級，而 01 和 10 狀態值是作為緩衝的觀察區。舉一個例子來說明，假設目前為 Best-effort 服務，狀態標示值為 00，而若現在(第一次)量測到的傳輸速率低於使用者需求速率時，服務等級並不會馬上改變，維持在 Best-effort，但把狀態標示值改為 01，接著若第二次量測的速率值也是低於使用者需求，此時將進入第二個觀察緩衝區，也就是狀態標示值會改變為 10，而服務等級還是維持在 Best-effort，當第三次如量測的速率值也是低於使用者需求，則狀態標示值將變為 11，此時代表了網路狀態確實很差，需要將服務等級提昇以達使用者的需求，開始將封包標註為較高(第二)等級的服務。相對地，如目前在最高服務等級，狀態標示值為 11，唯有當經過兩次狀態標示值改變(由 11 變 10，10 變成 01)，到第三次時若量測到的傳輸速率還是高於使用者需求速率，此時代表網路狀態良好，不需要這麼高的服務等級，所以把服務等級降低一等級。這是一個運用了簡易的緩衝方法來降低震盪的策略。

三、 實驗結果

為了驗證 PPE 及「2bits_3levels」策略和機率性(probabilistic)標註策略的差別，我們運用 Cisco 1700 series router 建構了一個 Differentiated Service 的測試平台。圖 2 是我們的實驗架構圖，測試平台由兩台設定好 Differentiated Service 的 Cisco 1700 系列路由器和 3 部電腦組成，其中 John 和 Mary 兩台電腦為 Sender 端，而 Bob 為 Receiver 端。

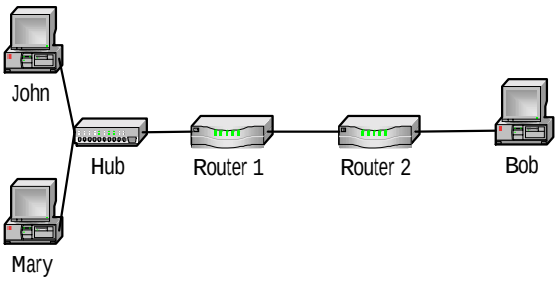

圖 2. 實驗平台

路由器 Differentiated Service model 的設定參考[7]，採用了 WFQ(Weighted Fair Queuing)的佇列方式，兩台路由器間的最高傳輸速率為 250Kbytes/s。WFQ 是一種動態排程機制，它公平地分配頻寬給所有網路的使用者。同時抵達路由器

的封包，會是較小的封包擁有先傳送的權利，接下來再將剩餘的頻寬公平地分配給所有應用程式產生的封包，因此可避免有量大的檔案傳送而影響整個網路運作。在 Cisco 路由器中 WFQ 是 IP Precedence-aware，其可偵測到被標註不同優先等級的封包，並使用其排程機制，使得在網路壅塞的情況下等級越高的封包能夠先被服務。表 1 為系統測試用電腦的設定說明，我們將 PPE 執行於 client 端，其中 Mary 執行了一個封包產生工具名為 rude 來產生傳送到 FTP server 的資料。

主機名稱	性質	作業系統	執行程式
Bob	FTP server(receiver)	Windows98	
John	FTP client(sender)	RedHat6.2R	PPE
Mary	FTP client(sender)	RedHat6.2R	rude

表 1. 系統測試用電腦設定說明

實驗方法是在 Mary 上用 rude 產生 300packets/sec，500bytes/packet，共傳送 30 秒的 UDP 資料，並將這些資料都標註成最高等級（DSCP 值為 0xa8），然後在 John 上分別運用「2bits_3levels」策略和機率性(probabilistic)標註策略來測試傳送一個 3Mbytes 大小的檔案到 Bob，且使用者需求速率(target rate)設定為 100kbytes/sec，來觀察兩個策略所造成的變動情形和資料被標註的情況。

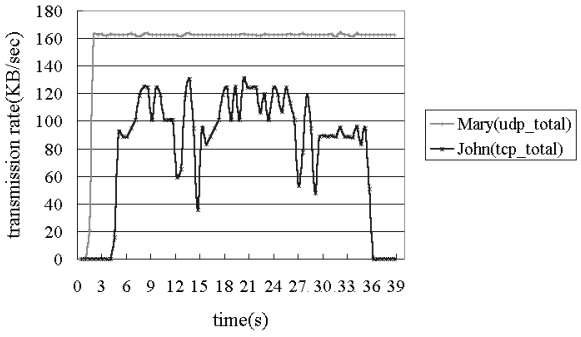

圖 3. 機率性(probabilistic)標註策略的實驗結果

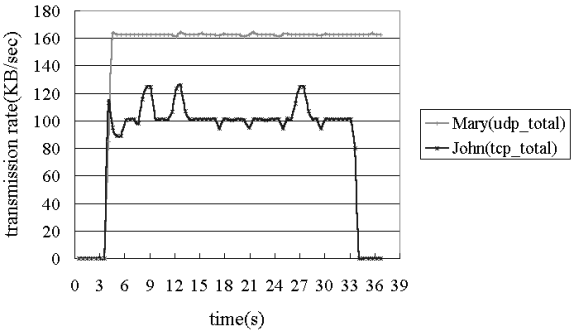

圖 4.「2bits_3levels」策略的實驗結果

圖 3 為採用機率性(probabilistic)標註策略的實驗結果，而圖 4 則是運用「2bits_3levels」策略的實驗結果，由兩張圖中可明顯的發現「2bits_3levels」的震盪較小，且由表 2 中可看到其最高等級封包的標註數量少了將近比率性(probabilistic)標註策略的一半。

	各服務等級被標註封包數 (packets)			封包總數 (packets)
策略名稱	0x00	0x68	0xa8	
機率性	511	751	788	2056
2bits_3levels	125	1589	336	2056

表 2. 兩種標註策略之標註情形

四、 結論及未來工作

本文提出了一個動態地依網路流量變化情況而調整封包標註情況的標註方法，藉由實際的實驗量測得知，本方法不但可以有效的減少震盪且又能降低封包被標註的數量。在接下來的研究中，我們將繼續去探討不同 level 數會有何影響及將此標註策略運用在 UDP 上的結果。

參考資料

[1] Wu-Chang Feng, Dilip D. Kandlur, Debanjan Saha, and Kang G. Shin, “Adaptive Packet Marking for Maintaining End-to-End Throughput in a Differentiated-Services Internet,” IEEE/ACM Transactions on Networking Vol. 7, No. 5, October 1999.

[2] David D. Clark, and Wenjia Fang, “Explicit allocation of best-effort packet delivery service,”

IEEE/ACM Transactions on Networking, Vol. 6, No. 4, August 1998

[3] Xiaoning He; Hao Ch*e*, ” Achieving end-to-end throughput guarantee for TCP flows in a differentiated services network,” Computer Communications and Networks, 2000.

[4] W. Richard Steven, “Unix Network Programming,” Volume 1, Networking APIs:Sockets and XTI, second edition, Prentice Hall Inc., 1998

[5] Sally Floyd and Michael Francis Speer, “Experimental Results for Class-Based Queuing,” http://www-nrg.ee.lbl.gov/floyd/cbq/notes.html, Jan 1998

[6] S. Floyd, V.Jacobson, “Link-sharing and Resource Management Models for Packet Networks,” IEEE/ACM Transactions on Networking Vol. 3, No 4, August 1995

[7] Yao-Tang Chen, Zi-Wei Shu, W.S.Hwang, 2001, "Differentiated Services Packet marking in Cisco Router environment", 2001 兩岸三地無線科技研討會

[8] http://www-hera-b.desy.de/subgroup/network/mgen/UserGuide.html

[9] http://www.csl.sony.co.jp/person/kjc/software.html