行政院國家科學委員會補助專題研究計畫成果報告

寬頻網際網路邊緣路由器支援差別性服務品質保證的封包排程與緩衝區管理之設計與實作

計畫類別：□個別型計畫　■整合型計畫

計畫編號：NSC 89－2213－E－002－171

執行期間：89 年 8 月 1 日至 90 年 7 月 31 日

計畫主持人：國立台灣大學資訊管學系　孫雅麗副教授

本成果報告包括以下應繳交之附件：

- □赴國外出差或研習心得報告一份
- □赴大陸地區出差或研習心得報告一份
- □出席國際學術會議心得報告及發表之論文各一份
- □國際合作研究計畫國外研究報告書一份

執行單位：國立台灣大學資訊管學系

中　華　民　國　90　年　9　月　28　日

計畫名稱: 網際網路服務品質保證之促成工具（II）

子計畫四：寬頻網際網路邊緣路由器支援差別性服務品質保證的封包排程與緩衝區管理之設計與實作

Design and Implementation of Packet Scheduler and Buffer Management for Broadband Internet QoS Router Supporting Differentiated Services

計畫編號：NSC89-2213-E-002-171

主持人：孫雅麗　台灣大學資訊管學系　副教授

E-mail: sunny@im.ntu.edu.tw　　Fax:02-23621327

中文摘要：(關鍵詞：差異化服務、傳輸服務品質保證、資源管理)

在寬頻網際網路邊緣路由器支援差別性服務品質保證，我們前兩年已對封包排程與緩衝區管理完成相關之研究，今年我們研究主題是封包分類(Packet Classification)。許多的網路服務需要有封包分類的功能，像防火牆、提供不同品質的服務(Quality of Service)，網路計價(Internet Pricing)……等，都需要路由器決定進來的封包屬於那一個服務等級、並且決定如何去處理它。傳統封包分類器無法有效支援彈性的分類規則，規則優先權之設定，以及無效率的規則管理。

在本研究中，我們針對這些缺點提出了一個新的封包分類器架構叫做索引搜尋樹及其資料結構；它可以處理區間的比對(range matching)、支援彈性擴充至多個欄位的問題，並允許使用者可以設定彈性的規則－包括規則之間的優先權、"don't care"與 not 運算子的使用。另外我們針對規則管理提出解決方式，包括偵測規則衝突、如何解決規則衝突，規則更新時的 lock 機制。一個好的規則管理方式有助於網路管理者訂定出符合期望的網路政策。

我們也實作所提出的架構與方法在過去兩年所開發的 QoS router 上，其效能測試結果顯兩種方法在大部分情況下都可以表現良好。我們另外用上萬的規則測試我們的架構，在規則數目很多時仍然很有效率。

英文摘要：(Keyword: Differentiated Services, QoS, packet classification)

There are a number of network services that require packet classification, such as routing, access control in firewalls, policy-based routing, provision of differentiated qualities of service, and traffic accounting/billing. In each case, it is necessary to determine which flow an arriving packet belongs to so to determine how to treat the packet. This categorization function is performed by an important component of a router called packet classifier.

In the past packet classification research has focused on IP packet routing. The new focus is moving towards to the more general problem of packet classification or fast content classification. Traditional packet classification algorithms have several drawbacks: a) no support of flexible rule setting; b) no support of precedence order between rules; and c) poor efficiency in rule addition and deletion. In this project, a new packet classification scheme is proposed. It supports range match and allows network manager to define fields in Layer 4 and Layer 7 protocols in the matching rules. Since a packet may obey more than one rule in at least one rule base. Policy rules defined by network administrators often conflict with each other. Packet classifiers must be able to pick out all of them and apply the most appropriate one(s). The proposed scheme also supports precedence order between rules and the negation (i.e. NOT) operator. We

have implemented the scheme on our router prototype. The testing results have shown that the scheme can effectively support large-scale rule sets.

1. 研究動機與目的

當網際網路上商業應用愈來愈普遍，服務提供者會希望網路設備如路由器能提供差異化的服務。但是傳統的路由器並無法提供這樣的服務，因為它們對於所有通過的封包都視為同一個服務等級。還有很多其它的網路服務也需要網路設備能夠有封包分類的功能，例如像防火牆對所通過的IP封包以及不同應用的控管、以政策為基準的封包傳送、提供不同傳輸品質的服務、VPN、或是網路計價等等。對未來以封包內容為主的網路傳輸路由器，這些服務的提供都需要封包分類的功能來決定收到的封包要如何去處理 — 例如是否要傳送它（防火牆），它是屬於那一個傳輸服務等級，傳送這個封包應該收費多少（網路計費）等等。

封包的分類主要是依據封包header上的欄位或者是封包payload內容的資料來加以區分、辨別。一般而言第三層的路由器依據IP封包的source IP 與 destination IP，第四層的路由器會多參考第四層的通訊協定如source port, destination port, protocol type, 第七層的路由器會看更多特定應用相關的資料如TCP SYN/ACK、HTTP GET message URI, etc.

傳統的封包分類器有下列的缺點。第一，他們多無法支援彈性的規則： 很多傳統的封包分類器，規則的設定只支援IP address prefix的設定方式，或是只能設定source IP和destination IP這二個欄位而已。彈性的規則需要能夠支援IP header和TCP/UDP header各項欄位的設定，設定的值可以是任意的區間。第二，必須要能支援優先權的設定。我們知道當同時有很多的分類規則時，網路管理者所設定的規則常常會有衝突(conflict)的現象。現有的路由器的封包分類多採用第一個符合規則(first match)；優先權的設定可以幫助使用者訂定不同規則的優先等級，使得政策的訂定更為彈性。第三，在分類規則新增或是刪除的運作效率很差 - 新增、刪除新的規則都需要重新調整封包分類器架構。這對寬頻網路同時有高速、大量資料的傳輸環境來說，是無法達到所需的功能的。我們需要快速的新增或刪除規則。除此之外，未來的封包分類不但要查看封包的header，像防火牆這樣的應用程式還需要查看封包的資料部分才能決定這個封包的服務等級。

在寬頻網際網路邊緣路由器支援差別性服務品質保證，我們前兩年已對封包排程與緩衝區管理完成相關之研究，今年我們研究主題是封包分類(Packet Classification)。

2. 相關研究

過去的文獻提出了不少封包分類器的架構，不同的架構通常是在搜尋速度和所花費的記憶體之間採取一個平衡點[1, 2,3,4,5,6,7,8,9,10,11,12,13,14,15,16]。例如在[1]所提出的封包分類器之架構，他們基本上是利用Trie這種資料結構來建構一個封包分類器。在[2]提出利用大量記憶體來幫助搜尋的演算法：建造一個很大的陣列儲存不同IP、Port時對應到那一個服務等級，分為數個phase以處理Multi-Dimension的情況。在[3]，使用者訂定的封包過濾器規則可以視為一個個的區間，使用陣列來紀錄一個區間之內有多少的規則。當封包進入時，分別就每一個欄位找到所屬區間符合的規則，如果有數個符合的規則，則選取第一個符合的規則。

傳統的封包分類器的架構有下列的缺點：

(1)沒有彈性規則：過去的文獻只能利

用 prefix 來設定的 source IP和 destination IP，如果封包分類器要適用於各種應用程式，必須支援多種欄位的設定(如source port、destination port、type of service and protocol type.....等)，而且給定值的方式不單只限於prefix的方式，應該要能夠支援任意的區間設定和邏輯運算元的運作。

(2)沒有優先權的設定：網路管理者所設定的規則常常有衝突的現象[6]，當有封包同時符合規則A、B時，傳統的封包分類器在搜尋時通常以第一個找到作為這個封包的服務等級，但是管理者在遇到規則衝突時，希望能給予不同的規則不同的優先權。

(3)新增、刪除的效率很差：傳統的封包分類器往往著重要快速的搜尋，忽略了這一方面的需求。

除了以上的缺點外，封包分類器的架構我們著重於能夠應用於大型、高速的網路如ISP的骨幹網路，並且支援大量的規則的設定。我們設計的封包分類器目的解決過去封包分類器的缺點，提供彈性的規則設定，使用者能訂定不同規則的優先權和良好的新增、刪除規則的效能，能夠針對封包的header和資料部分 (content)作分類，並且能有處理大型網路的能力。

3. Range Match

一個封包分類器含有一個 rule set 包含一條條定義的規則。規則是由一個個欄位所組成的，每一個欄位都對應到 IP/TCP/UDP 上封包的某一個欄位，每一個規則的欄位數目都是相同的。每一個欄位值都被視為一段連續的正整數區間，每一個區間都有起始值和終止值，終止值必須大於等於起始值。如 source IP 140.112.8.1- 140.112.8.254，IP 的資料型態可以視為一個 32bit 正整數，一段連續的 IP 區段可以表示為連續正整數區間，如果訂定的欄位值是單一值(single value)，如 source IP 為 140.112.8.1，將其轉換為起始點等於終止點的區間。

每一個規則並不是每個欄位有給定值.。在本計畫我們考慮欄位值是"Don't care" or "Any". 在 rule set 的搜尋,我們利用欄位間的特性來加速分類之搜尋。也就是說當收到一個封包要做分類時，並不需要比對每一規則的所有欄位。整個設計的理念是要想辦法避免不必要、多餘的比對以加速搜尋的速度。

假設有 N 個規則，每一個規則都有 d 個欄位，每一個規則的每一個欄位都定義一區間。For each field f_i 1<=i<=d, let $V_i=\{v_{i,j}\}$ contains the values contained in the current ruleset of field i： $\phi(v_{i,j})=y_{i,j}$，$y_{i,j}$ is the number of rules matched for $v_{i,j}$。We will choose the field that gives the minimum $M=\min_i\{\sum_j \phi(v_{i,j})\}$, $M \geq N$ ，it should give the best 搜尋效能。

Let us look at a field. Consider N rules. Let $(b_{i,k}, e_{i,k})$ be the value of rule k. For example in Figure 3.2, there are six rules (2,4)、(3,7)、(5,8)、(13,18)、(14,20)、(16,22)。Each rule-element corresponds to a node in the corresponding Index Tree denoted by T_i.

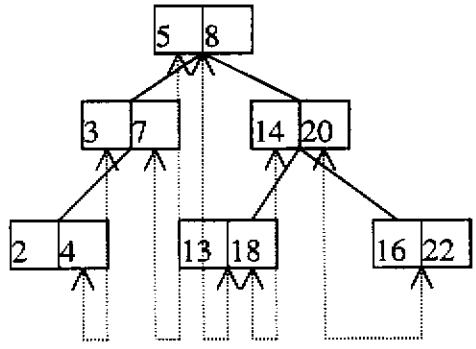

圖 1 Example of threaded binary search tree

Given $\{<b_{i,k}, e_{i,k}>\}_{k=1,N}$,

1. Construct a balanced binary tree

according to the begin (or end) value. Either one is ok. Here we assume the begin value.

2. Construct "threads" to link nodes in the tree

圖1表示將上面的區間都以一個樹結點表示，實線表示parent和child之間的關係，虛線表示thread的串起方法。

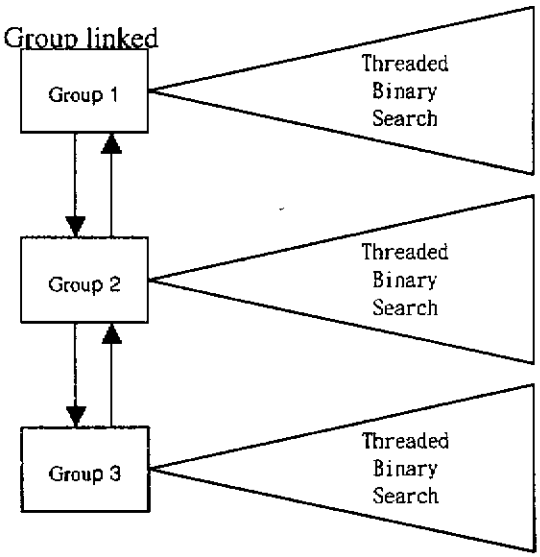

圖 2、An example of group threaded binary search tree

Rule Matching Algorithm

先利用 binary search 只比對樹結點的起始值，找尋起始值大於x的點，找到後利用thread之間的關係往 LeftThread 的方向回去比對找尋是否符合區間。使用這個方法的複雜度為O(log(n)+k)，log(n)為找尋開始比較的結點，k為延著 LeftThread 找尋符合的區間。

把一組區間分為數個群組，每一個群組之間的大小關係不論利用起始點或終止點來排都是固定，屬於同一群的區間利用 threaded binary search tree 來儲存，使用 threaded binary search tree 好處是，我們可以利用 thread 依大小串起每一個區間，搜尋時可以使用 binary search tree 的優點來搜尋。不同的群組利用 Double linked list 串接起來，如圖 2 所示。當加一個新的區間時，搜尋所有的群組查看是否可以加入某一個群組，如果有則加入。如果沒有的話，就獨立出來成為另外一個群組。當要刪除某一個區間時，搜尋所有的群組查看區間位於那一個群組內，再刪除區間。當封包進入時，分別進入每一個群組去搜尋是否有符合的規則。

B. 區間之間的關係

我們在以下定義區間之間的關係

- 定義一、.如果區間 A 大於區間 B，則區間 A、B 要滿足下列關係

(A.begin>=B.begin and A.end>B.end) Or
(A.begin>B.begin and A.end>=B.end)

- 定義二、如果區間 A 等於區間 B，則區間 A、B 要滿足下列關係

(A.begin=B.begin and A.end=B.end)

- 定義三、如果區間 A 小於區間 B，則區間 A、B 要滿足下列關係

(A.begin<=B.begin and A.end<B.end) Or
(A.begin<B.begin and A.end<=B.end)

- 定義四、如果兩個區間之間存在大於/等於/小於的任一關係，我們稱這兩個區間是可比較的(comparable)；反之如果都不存在大於/等於/小於三個關係，我們稱這二個區間為不可比較的(incomparable)。任意兩個區間之間的關係只有可比較和不可比較二種。

- 定義五、群組(Group)是一群可比較區間的集合。

對於一組區間(a set of ranges)，我們可以把依據彼此可比較或不可較的關係把它們分為數個群組。

4. 效能評估

在效能評估方面我們所提出的方法，在規則的新增、刪除管理以及搜尋都有好的效能，特別是在處理動態接收值域內的任何一個值。此外，我們也對大量的規則的搜尋速度做實作評估。我們分別產生4、8、16、32、64、128、256個class C 的IP rule，對應到的規則數量分別為1024、2048、4096、8192、16384、32768、65536。表1記錄了測試的結果。隨著規則數目的增加，我們新增一個規則和刪除一個規則都呈log(n)的比例成長。在搜尋的時間方面我們量測的結果隨著規則數目的快速增加，時間並不會增加很多。

Class C數目	Add time(us)	Del time(us)	Search time(us)
4(1024 IP hosts)	4.69	3.36	0.84
8(2048 IP hosts)	5.43	3.79	0.86
16(4096 IP hosts)	5.88	4.13	0.88
32(8192 IP hosts)	6.29	4.79	0.89
64(16384 IP hosts)	7.13	5.28	0.9
128(32768IP hosts)	7.76	5.94	0.91
256(65536 IP hosts)	8.44	6.53	0.96

表 1、使用 Rule Set E 測試 index search add/del/search 的效能

5. 結論

在本研究中，我們針對過去封包分類的缺點提出了一個新的封包分類器架構叫做索引搜尋樹及其資料結構；它可以處理區間的比對(range match)、支援彈性擴充至多個欄位的問題，並允許使用者可以設定彈性的規則－包括規則之間的優先權、"don't care" 與 not 運算子的使用。另外我們針對規則管理提出解決方式，包括偵測規則衝突、如何解決規則衝突，規則更新時的 lock 機制。一個好的規則管理方式有助於網路管理者訂定出符合期望的網路政策。

我們也實作所提出的架構與方法在過去兩年所開發的 QoS router 上，其效能測試結果顯兩種方法在大部分情況下都可以表現良好。我們另外用上萬的規則測試我們的架構，在規則數目很多時仍然很有效率。

參考文獻

[1] H. Adiseshu, S. Suri, and G. Parulkar, "Detecting and resolving packet filter conflicts," In Proc. INFOCOM, volume 3, pages 1203--1212. IEEE, March 2000.

[2] M. M. Buddhikot , S. Suri , and M. Waldvogel, "Space Decomposition Techniques for Fast Layer-4 Switching," Proc. Conf. Protocols for Hight Speed Networks, Aug. 1999, pp. 25-41.

[3] W. Cheswich and S. Bellovin, "Firewalls and Internet Security," Addison-Wesley, 1995

[4] D. Decasper, Z. Dittia, G. Parulkar, B. Plattner, "Router Plugins: A Software Architecture for Next Generation Routers," In Sigcomm, 1998.

[5] Mikael Degermark, Andrej Brodnik, Svante Carlsson, and Stephen Pink, "Small forwarding tables for fast routing lookups," In Proc. ACM SIGCOMM Conference (SIGCOMM '97), pages 3-14, October 1997.

[6] A. Feldmann and S. Muthukrishnan, "Tradeoffs for packet classification," In Proc. INFOCOM, volume 3, pages 1193- 1202. IEEE, March 2000.

[7] P. Gupta and N. McKeown, Algorithms for Packet Classification, IEEE Network , March/April 2001, vol. 15, no. 2, pp 24-32.

[8] P. Gupta and N. McKeown, "Packet classification on multiple fields," in Proceedings of ACM SIGCOMM'99, ACM, August 1999.

[9] Sundar Iyer, Ramana Rao Kompella and Ajit Shelat "ClassiPI: An Architecture for Fast and Flexible Packet Classification," IEEE Network , March/April 2001, vol. 15 , no. 2 , p33-41.

[10] T.V. Lakshman and D. Stiliadis, "High Speed Policy-based Packet Forwarding Using Efficient Multi-dimensional Range Matching," Proc. ACM SIGCOMM '98, 1998.

[11] L B. Lampson, V. Srinivasan and G. Varghese, "IP Lookups using Multiway and Multicolumn Search," In Infocomm, 1998.

[12] Peter Newman, Greg Minshall, and Larry Huston, "IP Switching and gigabit routers," IEEE Communications Magazine, January 1997.

[13] M. H. Overmars and A.F. van der Stappen. "Range Searching and Point Location Among Fat Objects," Journal of Algorithms, vol 21, no. 3. Nov. 1996, pp.629-56.

[14] V. Srinivasan, S.Suri, and G.Varghese, "Packet Classification using Tuple Space Search," in Proceedings of ACM SIGCOMM, Sept 1999.

[15] V. Srinivasan, G. Varghese, S. Suri, and M. Waldgovel, "Fast and scalable layer four switching," In Proceedings of SIGCOMM, October 1998.

[16] V. Srinivasan and G.Varghese, "Fast IP Lookups using Controlled Prefix Expansion", in Proc. ACM Sigmetrics, June 1998.