利用決策樹改善以 FPGA 為基礎之入侵偵測系統資源利用

魏雅笛
國立中央大學
Email:weiyati@gmail.com

陳奕明
國立中央大學
Email:cym@mgt.ncu.edu.tw

***摘要*—網路的應用對於目前個人及企業越來越重要，但網路威脅趨日增加，網路入侵測系統基於特徵比對便成為企業不可或缺的基礎防護。然而目前入侵偵測系統大多架設在軟體的架構之上，趨漸無法應付目前網路現況；相反地，硬體具有高速及平行比對能力，能夠快速比對，尤其 FPGA 能重覆燒錄及快速製作雛型，相當合適設計入侵偵測系統。但 FPGA 內所能使用的資源有限，而特徵資料庫卻不斷的更新及擴張，故本研究基於以上動機，利用 FPGA 設計入侵偵測系統，以決策樹處理規則的標頭，再依規則標頭比對架構建置多字串比對群組來進行封包內容的比對。本研究提出的架構平均可以降低 56%的電路資源使用率，故能擁有更多資源來擴充新的規則，具有可擴張性，而且採用多字串比對群組，可以使用特徵字串平行比對增加效能，實驗證明本系統架構可以使用較少的資源，且較其它 FPGA 設計更具效能。**

***關鍵詞*—FPGA, 入侵偵測系統, 樹狀架構, NFA**

一、研究背景

網路的應用與服務不但改變了溝通、生活以及企業的商業模式，幾乎所有的商業活動都需要基於網路服務，網路的安全及保護就更顯重要，賽門鐵克 2009 年的最新統計，每年節節增加新的威脅[15]，因此各企業或機構無不採取更嚴謹的防禦措施，以保障網路安全。

SNORT[13]是一個廣泛使用的開放式入侵偵測系統，利用規則比對可以過濾已知的攻擊，通常為資訊安全守護的第一條防線，然而其規則庫會不斷的進行更新、增加規則以及目前網路頻寬不斷增加，入侵偵測系統需要進行大量的字串比對，但網路型入侵偵測系統其效能只能達到網路流量在 60Mbps 左右[12]，當效能開始跟不上網路流量速度時，就會開始進行放棄封包的動作，任何惡意的攻擊者就可以利用此缺陷來躲避封包的檢查。而基於依據 TWNIC 的 2009 年 1 月網路頻寬調查中，88 個單位(營利、非營利、交換中心)中有 75 單位的網路頻寬使用超過 100Mbps [1]，為能夠符合目前及日後的網路環境，一個具有效率快速比對封包的規則型入侵偵測系統架構是不可或缺的，因此基於硬體擁有運算速度快及平行處理的二大優點，即可解決上述問題，在硬體上實作網路入侵偵測系統是非常合適。

Field Programmable Gate Array(FPGA)，現場可規劃邏輯閘陣列，是一種半訂製(Semi Customize)型 IC [3]，具有可程式陣列邏輯元件及邏輯閘陣列邏輯的規劃彈性，擁有高效能與高度的變化性，目前將字串規則比對運用於網路入侵偵測系統並實作在 FPGA 為一趨勢[2] [10][14] [16]，但入侵偵測系統實作硬體上只考慮設計特徵字串比對[7] [8]，卻顯不夠完整，因此一個完整的入侵偵測系統應加入標頭比對，才能完整的偵測攻擊，減少誤判(false positive)。此外特徵資料庫會因為新的攻擊的出現而需要進行更新的動作，因此規則的更新會使得規則不斷的增加，其所需要使用在硬體設備上資源也越多。

本研究設計開發一個完整的入侵偵測系統在 FPGA 硬體架構上且能夠節省資源使用，在相同

硬體設備下容納更多規則。利用 Snort 的規則具有固定格式，其格式為規則標頭(Rule Header)與規則選項(Rule Option)，前者有許多規則會共享相同的規則標頭，其中包括協定(Protocol)、來源位址(Source IP address)、來源埠(Source port)、目的位址(Destination IP address)、目的埠(Destination port) 這五項，後者根據不同的規則有不同相字串比對[4]。利用規則標頭建立決策樹，減少需要重覆比對電路，規則選項依據規則標頭的樹狀架構產生多字串比對群組。

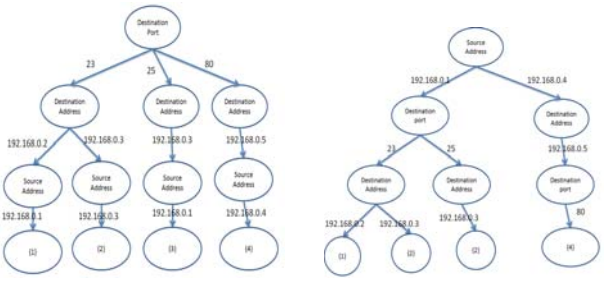
本論文可有效的降低電路資源的使用平均達56%，故能擁有更多資源來擴充新的規則，具有可擴張性。採用多字串比對群組，可以使用特徵字串平行比對增加效能，實驗証明本系統架構可以提供較低資源的使用，且較其它設計更具效能。

本論文分為五節。第二節將會介紹決策樹、FPGA 的概念。第三節為本論文所提出的方法架構，說明如何以利用決策樹的方式來減少需要比對的規則以及說明如何架設在 FPGA 架構之上，並詳細描述主要模組的運作。第四節則以提供模擬本實驗架構在 FPGA 之上，以驗證本研究所提出模型之運作。最後於第五節提出結論及未來研究方向。

二、相關研究

(一) 規則標頭

封包分類可設計實作於軟體及硬體當中，在軟體上的解決方案，其執行效率方面弱於硬體設計，但硬體的設計又過於複雜或是實作成本昂貴。軟體實作以文獻[9]提出將規則標頭建構成決策樹，每個節點代表規則屬性的比對，葉節點為可能比對規則，藉由快速搜尋可能符合規則，再進行其它規則標頭屬性和規則選項比對，建置決策樹採用 ID3 演算法[11]。在軟體架構上無資源使用限制卻有效能問題，但相反的在硬體架構上其運算速度高，卻限制於硬體的資源使用，因此 ID3 演算法並不符合其資源節省的目的，圖 1 中相同的規則標頭以不同建置方式所產生的決策樹，圖 1-b 為本論文基於屬性最多共用所建置的決策樹，其樹狀架構較圖 1-a 所使用的節點數較少。



A. ID3　　B. 本論文

圖 1 ID3 與本論文建置規則標頭

Song et al. [14]提出一個新的封包分類架構 BV-TCAM 是基於硬體實作，將規則標頭分為二個部份，協定、來源位址和目的位址利用 BV(Bit vector)-TCAM 進行比對，產生的結果會放置在位元向量(Bit vector)，另一部份則是利用 Tree Bitmap 針對規則中來源埠和目的埠的比對，最後結合所有屬性的比對結果。

(二) 規則選項

在字串比對電路當中，採用[8]中所提出的非決定性有限狀態自動機(Nondeterministic Finite Automaton, NFA)，可分享狀態以減少電路，並且降低資源的使用，NFA 是利用狀態轉換並紀錄目前比對到的狀態來進行比對封包酬載。以圖 2 為例，其特徵字串為：&&*、*port1、*port2、 *pass，將特徵字串建立有限狀態自動機再將相同字元進行合併，共用其狀態。

三、系統設計

在本論文的系統架構中利用規則標頭建立決策樹進行封包標頭比對，利用位元向量來儲存結果，再依符合條件的規則標頭進行規則選項的

比對，並提供工具可自動分析規則以產生規則標頭決策樹。

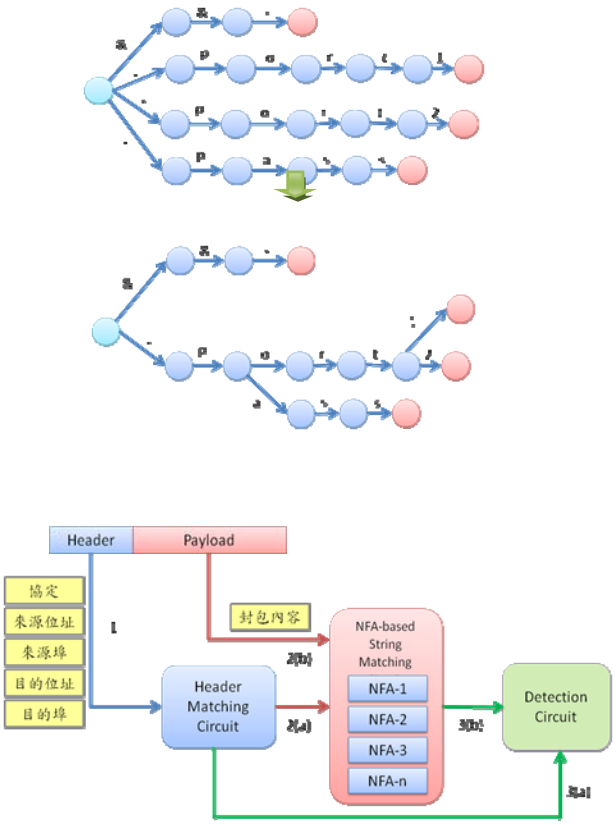


圖 3 本研究系統架構

(一) 系統架構

本系統設計如圖 3，以下詳細說明系統設計。

1. 將封包標頭送入標頭比對電路(Header Matching Circuit)進行標頭比對，將比對結果放入相對應位元向量(bit vector)當中，圖 4 中以決策樹進行封包標頭比對，每個節點代表一個屬性比對，比對結果放入位元向量，其長度為決策樹中葉節點的數目。產生結果後將位元向量與封包酬載一起送入字串比對電路(String matching circuit)。
2. 字串比對電路，會接收由標頭比對電路所送出的結果和讀入封包酬載(payload)，在字串比對電路當中，依據規則標頭中決策樹的建置，切割成多個字串比對群組電路。圖 5 每個位元向量會對映到相對應的字串比對群組，每個字串比對群組包含多個特徵字串，特徵字串比對結果會放入相對應的位元向量當中。

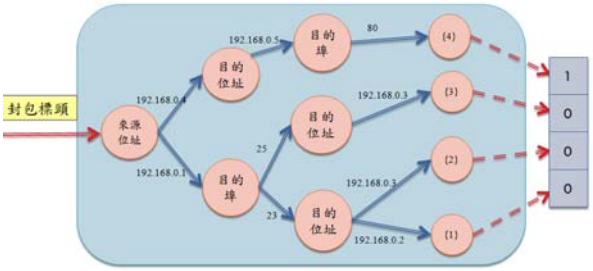


圖 4 標頭比對電路中決策樹比對結果

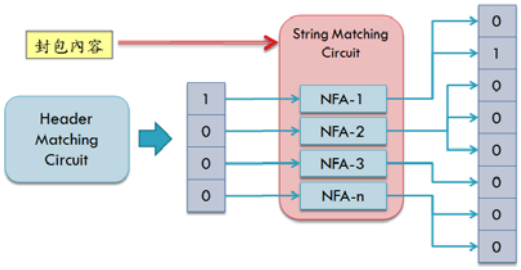


圖 5 規則標頭產生結果對應字串比對電路

3. 偵測電路(Detection circuit)會讀入由標頭比對電路和字串比對電路所產生的結果，圖 6 偵測電路會根據原有的規則內容，組合規則標頭和字串比對電路的比對結果，將結果儲存在位元向量當中，偵測電路的位元向量長度即為原有規則的數量，偵測電路的位元向量可以顯示多重比對的結果。

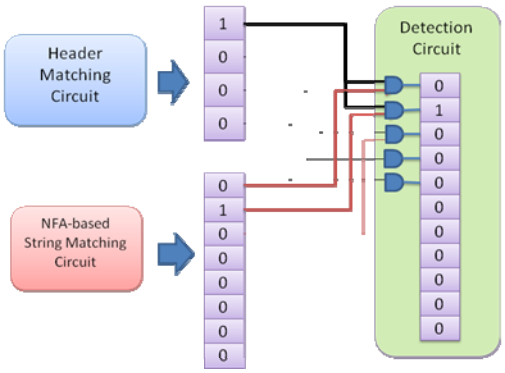


圖 6 偵測電路綜合規則標頭和字串比對電路

(二) 樹狀規則標頭比對建置

標頭比對電路的設計是基於決策樹的建置，尋找分割節點的原則是取所有屬性中具有最大共用及最具價值的屬性，代表該屬性所產生的分群數目最少，即表示在該屬性下的數值，是最多規則共用的屬性。樹的建立會合併相同的電路比對，減少需要重覆比對的電路，在決策樹內一個節點即為一個規則屬性比對，從決策樹的根節點走訪到葉節點的路徑即是一個規則標頭比對，規則標頭中則包含有五個屬性：協定、來源位址、來源埠、目的位址、目的埠。建置步驟如下：

1. 若是從樹的根節點開始建置，則到步驟 2 進行計算。若非根節點，則從緩衝區(Buffer)中讀出節點，從緩衝區中讀出節點，再進行步驟 2 的分割計算。
2. 讀入節點後，目前所在的節點為 N，找出目前節點到根節點的路徑上尚未進行比對的屬性，以 A0 ~An-1 表示，n 為未比對屬性的數目。設計一個副程式 GN，GN (N，A) 代表計算 N 節點下屬性 A 的分割數，再用 Min { GN(N , A0) , GN(Node, A1)…., GN(Node, An-1)}，找出分割群組數目最少的屬性，其屬性即為該節點下的分割屬性。
3. 依分割屬性進行規則分割，將分割後的規則群組標記代號後放入緩衝區當中，再反覆回步驟一，直到每個路徑的根節點到葉節點，都具有完整的五個屬性(即代表一個完整的規則標頭)才停止動作。

在圖 7 的例子當中，計算根節點下所有屬性的分割數，取最小分割屬性，其屬性為「協定」，依此屬性分為二個群組，再依照各群組下選擇最小分割的屬性，建立成最大共用的決策樹，圖 8 的決策樹是由圖 7 所建置而成。

(三) 多字串群組的建置

字串比對電路，是根據規則標頭進行字串分群，群組數目即為標頭比對電路結果的向量長度，由每一個位元來對應一個字串比對群組。

	協定	來源位址	來源埠	目的位址	目的埠	特徵字串
0	UDP	Any	80	Any	6000	&first&
1	TCP	$Home_net	1024	192.168.0.0/16	2589	++Conectado
2	TCP	$Home_net	10101	192.168.0.0/16	any	|07|author
3	TCP	Any	Any	$Home_net	443	shell boun
4	TCP	192.168.0.0/16	Any	$Home_net	110	|00 00 00 00 00 00 00 00|
5	TCP	$External_net	146	$Home_net	1000:1300	|03 00 1C 00 00 00 01|Furax
6	TCP	$Home_net	10101	192.168.0.0/16	any	&rapor
7	UDP	Any	80	Any	6000	&rapor

圖 7 轉換規則為樹狀規則檔頭比對

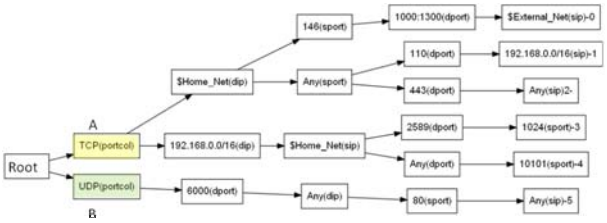
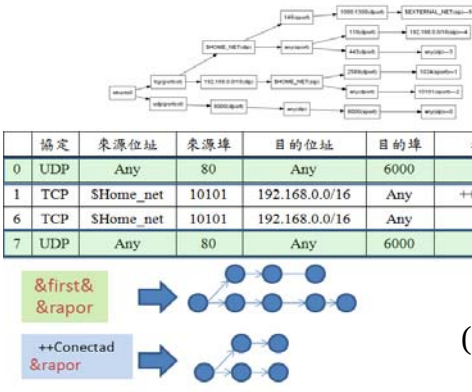


圖 8 轉換規則為樹狀規則檔頭比對

字串群組的建立是依據前面的標頭比對電路(圖 9-a)，根據相同的規則標頭，特徵字串會分相同的群組當中，以圖 9-b 中規則編號 0 和 7，二者的規則標頭是相同的，則將後方的字串合併為同一群組內(圖 9-c)，將分組後的字串轉換成基於 NFA 的字串比較電路。

字串比對電路設計採用非決定性有限自動機來進行字串比對，是基於狀態轉換的方式來進行字串比對，使用此方法的優點在於可以共用相同的字元狀態，節省重覆的字元，降低資源使用。



	協定	來源位址	來源埠	目的位址	目的埠	特徵字串
0	UDP	Any	80	Any	6000	&first&
1	TCP	$Home_net	10101	192.168.0.0/16	Any	++Conectado
6	TCP	$Home_net	10101	192.168.0.0/16	Any	&rapor
7	UDP	Any	80	Any	6000	&rapor

圖 9 基於規則檔頭建立多字串群組

(四) 自動化產生工具(Automatic Generation Tool)

入侵偵測系統的特徵資料庫包含的規則數目龐大，單以人力撰寫硬體描述語言花費時間浩大，且特徵資料庫需時常進行更新，若每次進行更新規則時，都採用人力方式重新撰寫不符合其時間效益，容易失去更新的時效性，導致出現安全漏洞，也不符合原本使用 FPGA 的原意(即是可以立即更新規則)，因此，為方便快速的更新規則，讓不會撰寫硬體描述語言的使用都也可以進行規則更新的動作，本研究開發自動化產生電路工具，將 Snort 中的規則自動轉換成基於本研究架構的硬體描述語言，只需再進行測試電路是否正確，即可對於 FPGA 進行規劃，因此即使不具有撰寫硬體描述語言能力的使用者也能夠自行更新規則，縮短更新規則的時間。

此工具可以提供自動讀取 Snort 的規則，進行分析，並依照本研究架構產生入侵偵測系統電路，提供測試文件，方便使用者在燒入 FPGA 前可測試電路的正確性，由圖 10 顯示從自動化工具從規則到入侵偵測系統電路產生流程。

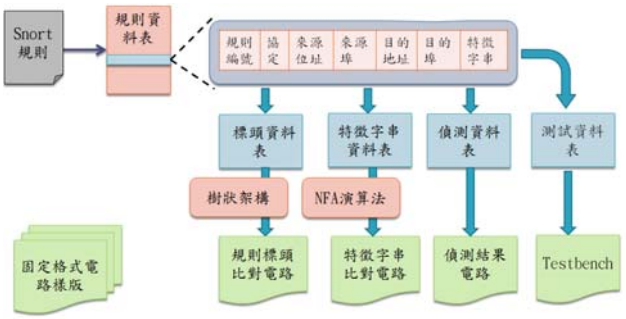


圖 10 自動化產生電路架構工具系統架構

規則資料表內存放所有 Snort 的完整規則，標頭資料表內則存放群聚後的規則標頭並予以編號，特徵字串資料表則是放置特徵字串並給予編號，偵測資料表則是依照規則資料表產生相對應的標頭資料表編號以及特徵字串資料表編號，此外依照規則內容，產生測試資料表，資料表內儲存測試規則的資料，做為功能模擬時用來評估功能是否正確，產生的設計是否符合規則的需求，以確保設計自動化的產生無誤。

四、實驗討論

(一) 實驗環境

本論文以 Altera [5] 的 StratixII EP2S180F1508C3 為設計標的，利用模擬器對本研究架構來進行實驗分析。軟體的部份使用 Alerta Quartus II 8.0 及 ModelSim 6.5 當作撰寫 Verilog 硬體描述語言的平台，Quartus II 可以提供語法檢查、時序分析、邏輯元件的配置、產生規劃檔案、電路合成以及繞線佈局等強大功能，ModelSim 則是可以提供進行大量資料測式功能，模擬器執行的硬體設備採用的 CPU 為 Pentium4 2.0GHz 及 3G 記憶體。

表 1 FPGA 規格

EP2S180F1508C3	
Core voltage	1.2V
ALUTs	143520
User I/Os	1171
Memory bits	9383040
DSP	96
PLL	12
DLL	2
Global clocks	16

(二) 實驗架構

本實驗為評估本系統架構的效能及所能節省資源，將建立三種不同的實驗架構進行比對：

1. 模型一：採用平行架構，標頭比對及特徵字串比對平行架構，偵測電路需同時接收到標頭比對電路和特徵字串比對電路的結果，圖 11 所示。
2. 模型二：採用循序架構，標頭比對電路的結果控制特徵字串比對電路是否需要進行比

對，標頭比對電路的設計則是採取非樹狀架構，圖 12 所示。

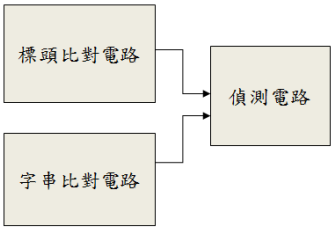


圖 11 模型一系統架構

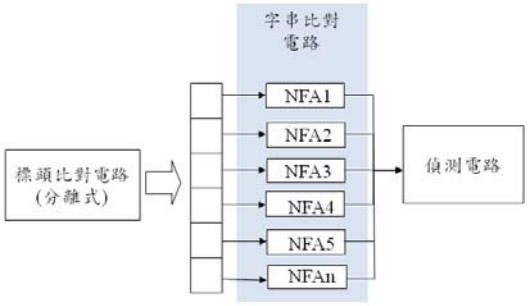


圖 12 模型二系統架構

3. 模型三：採用循序架構，利用標頭比對電路來控制特徵字串比對群組，與模型二的差別在標頭比對電路採用決策樹建置以減少電路使用，模型三為本論文架構，圖 13 所示。

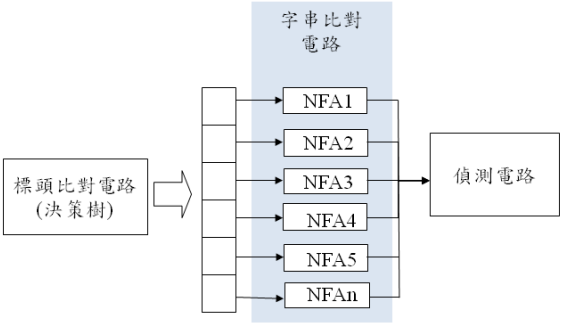


圖 13 模型三系統架構

模型一主要是顯示平行架構與循序架構的差別，模型一特徵字串比對中，只建置單一字串群組，相較於模型二、三的則是使用多特徵字串比對群組。模型二跟模型三的差別，只有在規則標頭比對電路中，分離式及決策樹的差別。

(三) 實驗與討論

本研究實驗的規則來源是採用 Snort 2.6 版，總規則數為 9,261 筆，實驗可分為三部份，實驗一隨機選取規則中的攻擊類型，再依三種不同模型下進行效能和資源的評估，實驗二則是評估規則數目不同下，其三種不同模型的變化，實驗三則以五個不同的規則集合來評估。

1. 實驗一：本研究在實驗中針對三種不同的模型架構下進行效能和資源的評估，實驗中所使用的規則集合，是從 Snort 規則中 47 個規則類型，在其中隨機取出 30 個規則類型，總共有 1023 條規則。表 2 顯示在 1023 個規則當中，將規則標頭相同的進行合併後共有 257 個，字串數去除重覆後有 597 個特徵字串，因樹狀架構產生重覆比對的字串數目為 141 個字串，表 2 整理規則內容的分析。

表 2 實驗規則數分析

規則數	**1023**
標頭數	267
字串數	597
樹狀比對數	738
額外增加字串比對	141

表 3 為實驗結果，三種不同的模型下其邏輯利用率，以模型三(本研究架構)最少，與模型二相比，規則標頭以樹狀架構比對可以有效的降低資源利用，與模型一相比其資源利用率又更少。在效能的評估方面，模型三的效能高於模型一的架構，其效能提高的主因，在於字串比對電路提供平行比對多個字串群組，對效能有所提升，也可以達到資源的節省，對於在日後規則數提高，可容納更多的規則數目。

表 3 三模型效能及資源使用率比較

規則數=1023	模型一	模型二	模型三
架構	平行式	循序式	循序式
規則標頭	分離式	分離式	決策樹
字串特徵	單一 NFA	多分群 NFA	多分群 NFA
Combinational ALUTs	12516/143520 (9%)	10240/143520 (7%)	7067/143520 (5%)
Operating Frequency	96.28MHz	151.08 MHz	148.54MHz

2. 實驗二：本階段的實驗針對於評估不同規則數目下，在三個模型中效能及資源的變動分析，表 4 列出實驗中所採用的規則數目，其規則來源取自實驗一中的規則集合，再從其中取其子集合，在表 4 顯示規則數及取出規則分析。

表 4 不同規則數實驗規則分析

規則數	58	333	561	810	1023
標頭數	26	150	146	226	267
字串數	32	229	303	463	597
樹狀比對數	38	261	347	554	738
額外增加字串比對	6	32	44	91	141
不需進行字串比對	6	34	125	129	146

不同規則數在三種不同模型的評估下，圖 14 中顯示其結果，隨著規則數目增加，操作頻率會趨漸下降，整體曲線顯示，模型三的效能仍高於模型一和模型二，雖在規則數 1023 下曲線驟降，其原因為比對的額外字串數增加許多，造成曲線下降。

針對於資源使用，在不同規則數下所產生的變化，可由圖 15 所示，資源的使用量隨著規則數的增加而趨於成長，但模型三所使用資源數量較另外二個模型少。

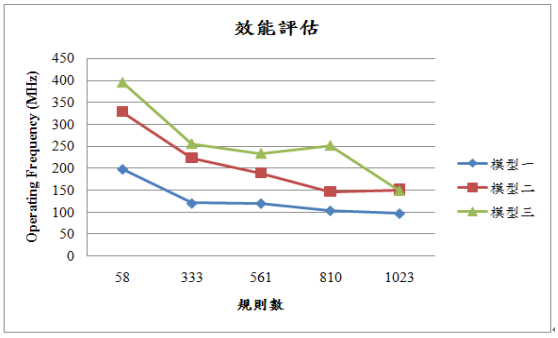


圖 14 在不同規則數下效能評估

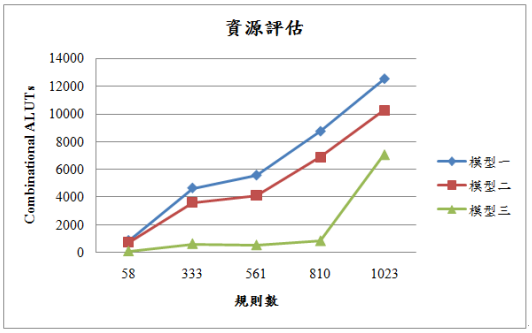


圖 15 在不同規則數下效能評估

3. 實驗三：評估不同的規則組合是否會影響效能和資源數據，並計算平均效能和資源的節省比例，在 RS_1 的採用的規則為實驗一中所使用的規則集合，RS_2 到 RS_5 的規則集合是採用隨機的方式從 9,261 個規則中取出 1023 筆規則，採用完全隨機挑選。表 5 中顯示編號 RS_1 到 RS_5 規則集合的分析。

在表 5 中，發現以隨機取出規則數目，雖然規則數目相同，但在 RS_2 到 RS_5 所

需比對規則標頭和字串數目都較 RS_1 多，因此在圖 16 中 RS_2 到 RS_5 的效能較 RS_1 中略低一些，但曲線並未產生非常大的變動，模型三的仍維持其效能高於模型一和模型二。

表 5 不同規則數實驗規則分析

規則編號	RS_1	RS_2	RS_3	RS_4	RS_5
規則數	1023	1023	1023	1023	1023
標頭數	267	385	388	387	387
字串數	597	808	809	808	807
樹狀比對數	738	919	919	917	917
額外增加字串比對	141	111	110	109	110

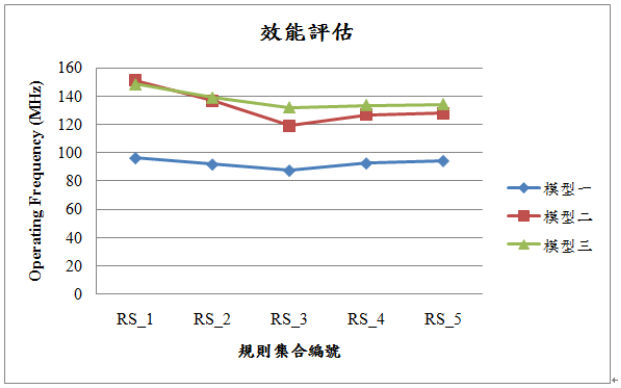


圖 16 不同規則組合下效能評估

對於不同規則集合下資源利用的評估，以 RS_2 到 RS_5 所使用的規則標頭數及字串比對數目較 RS_1 多，在圖 17 中模型一和模型二的曲線在資源使用量會上升，但相反的採用本論文架構的模型三，卻反而資源使用量略為下降，顯示本論文架構能夠大幅減少資源的使用。

利用公式 1 來計算 RS_1 到 RS_5 中模型三對於模型一和模型二模型的資源改善，在表 6 中顯示 RS_1 到 RS_5 中資源的改善比例，本論文架構針對於模型一可以節省 44% ~ 60%的資源使用，其平均能節省 56%資源。

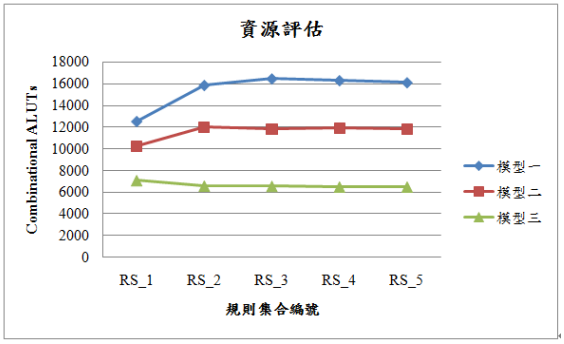


圖 17 不同規則組合下資源使用

$$\text{資源改善比例} = \frac{(\text{原有資源數} - \text{改善後資源數})}{\text{原有資源數}} \times 100\% \quad (1)$$

表 6 不同規則數實驗規則分析

單位：%	RS_1	RS_2	RS_3	RS_4	RS_5
模型三對模型一之資源改善	44	59	60	60	60
模型三對模型二之資源改善	31	45	45	46	45

(四) 實驗結論

以電路資源來討論，由以上實驗可以發現隨著規則數目的增加，所使用的電路資源也提升，但從實驗數據可以發現模型三(本研究架構)對於模型一和模型二相比改善資源比例平均可達 56%，這對於日後入侵偵測系統的規則更新，可在相同的設備之下容納更多的規則數。

效能的評估上，可以明顯的發現到，多字串群組的字串比對，可以提升比對效能，因為當字

元進入字串比對電路中，可平行對多個群組的有限狀態自動機進行比對，另外，我們利用標頭比對的結果來對映每個字串群組，若是標頭比對的結果，已未符合規則，則不需要再進行特徵字串的比對，可以節省掉不必要的比對。因此本研究架構在相較於模型一和模型二，不但能夠提供低資源的使用以及具有較佳的比對效能。

五、貢獻與未來研究

(一) 研究貢獻

本研究貢獻在於提出以樹狀架構為基礎的標頭比對電路能夠有效的節省電路資源的使用，對於入侵偵測系統日後規則更新，規則數不斷的擴張下，能夠有更多容納規則的空間，具有可擴充性，也可以重覆的使用相同的硬體設備。

特徵字串比對電路依據標頭比對電路產生多個分群字串，標頭比對電路的結果對映到相對的字串群組，封包酬載可平行比對多個字串群組加快比對效能，且依據規則標頭的比對結果，來控制每個字串群組的比對，在實驗中驗証本研究架構不但可以有效節省資源達 56%，且具有較佳的比對效率。

此外本研究為支援日後規則更新的時效性，提出自動化產生電路工具，對於日後規則，不需要重新找人編寫，即使不會撰寫硬體描述語言的使用者，都能夠輕鬆的產生電路進行規則更新，並提供完整的測試劇本，來驗証產生電路的正確性，對於一般使用者即可以輕鬆進行更新，增加本研究的方便性以及廣泛性。

(二) 未來研究

未來研究，可以分為以下幾個方向及改進行研究：

1. 針對於字串比對電路的部份，在本研究架構中，以每次輸入一個字元進行比對，但卻不夠充分展現硬體能夠平行處理的特性，若能將字串比對電路採用多個字元平行輸入比對，則可有效的提高效能。
2. 本研究架構當中，採用以一個封包內容處理完成後再進行下一個比對，雖然此方式較為謹慎處理，但其效能較低，因此可採取使用管線式(Pipeline)來進行比對，當規則檔頭比對完畢產生結果，則下一個封包即可以進入比對，減少中間等待比對時間，加速處理比對。
3. 本研究採用模擬器做繞線佈局最佳化，但若是針對規則的特性來進行規則佈局配置或順序變動，其所造成影響及改變，足以留為探討。
4. 實驗討論中所使用規則數目最高為 1023 筆，其受限本研究架構的設計是採用位元向量方式顯示結果，其優點為多個規則符合結果的呈現，但也使得輸出引腳 (pin)數目不足，而使得特徵規則數目受限，因此可以改變其比對結果的呈現，例如採用編碼方式，即可增加特徵規則數量。
5. 本研究主要利用規則標頭比對，再對於特徵字串進行分群，此種做法對於字串會產生額外的比對，若是改以字串分群為基礎出發，或是依照特徵字串的一些特性來延伸比對，是否更能提升其系統效能，其議題留予其它研究學者思考。

六、參考文獻

[1] TWNIC-台灣網路資訊中心網路使用調查，2009。

[2] 黃威智. “在可程式化系統晶片中實現網路入侵偵測系統之高效能封包分類與比對電路.” 國立臺灣師範大學資訊工程學系研究所碩士論文, 2006.

[3] 鄭信源，Verilog 硬體描述語言數位電路-設計實務，儒林出版社，台灣，2007。

[4] 蘇建中，Snort Tracing and Implementation 程

式追蹤與模組實作指引，網路電子文件 ，2003。

[5] Altera, http://www.altera.com/.

[6] SchimmelR. Clark and David E.Christopher, “Efficient Reconfigurable Logic Circuits for Matching Complex Network Intrusion Detection Patterns,” Field-programmable Logic and Applications, Vol.2778, P.956-P.959, 2003.

[7] B. L. Hutcchings, R. Franklin, and D. Carver, “Assisting network intrusion detection with reconfigurable hardware,” Proceedings of the 10th Annual IEEE Symposium on Field-Programmable Custom Computing Machines, P.111-P.120, 2002.

[8] Toshihiro Katashita, Atusi Madeda, Kenji Toda, and Yoshinori Yamaguchi, “Highly Efficient String Matching Circuit for IDS with FPGA,” Proceedings of the 14th Annual IEEE Symposium on Field-Programmable Custom Computing Machines, P.285-P.286, 2004.

[9] Christopher Kruegel and Tomas Toth, “Using Decision Trees to Improve Signature-Based Intrusion Detection,” Proceedings of the 6th International Workshop on the Recent Advances in Intrusion Detection, Vol.2820, P.173-P.191, 2003.

[10] Toshihiro Katashita, Yoshinori Yamaguchi, Atusi Madeda, and Kenji Toda, “FPGA-Based Intrusion Detection System for 10 Gigabit Ethernet,” The Institute of Electronics, Information and Communication Engineers Vol. E90-D, No.12 , 2007.

[11] J.R. Quinlan, “Induction of Decision Trees,” Maachine Learing, Vol.1, P.81-106, 1986.

[12] T. Ramirez, C. D. Lo, “Rule Set Decomposition for Hardware Network Intrusion Detection,” International Computer Symposium, Taipei, Taiwan, 2004.

[13] SNORT official web site, http://www.winsnort.com/

[14] Haoyu Song and John W. Lockwood, “Efficient packet classification for network intrusion detection using FPGA,” Proceedings of the 2005 ACM/SIGDA 13th International Symposium on Field-programmable Gate Arrays, P.235-P.242, 2003.

[15] Symantec Global Internet Security Threat Report, http://www.symantec.com/business/theme.jsp?themeid=threatreport, access at June, 2009.

[16] Nicholas Weaver, Vern Paxson, Jose M Gonzalez, “The Shunt: An FPGA-Based Accelerator for Netwrok Intrusion Prevention,” International Symposium on Field Programmable Gate Arrays, P.199-P.206, 2007.