200826571

發明專利說明書

（本說明書格式、順序及粗體字，請勿任意更動，※記號部分請勿填寫）

※申請案號：PS 14+P74

※申請日期：95、12、8　　※IPC 分類：H04L 12/66　H04L 12/24

一、發明名稱：（中文/英文）

應用於點對點閘道器上之辨別及管理系統及其方法

二、申請人：（共 1 人）

姓名或名稱：（中文/英文）

國立交通大學

代表人：（中文/英文）黃威

住居所或營業所地址：（中文/英文）

新竹市大學路 1001 號

國　籍：（中文/英文）　中華民國 TW

三、發明人：（共 5 人）

姓　名：（中文/英文）

1、 林柏青

2、 蔡孟甫

3、 張朝江

4、 林盈達

5、 賴源正

國　籍：（中文/英文）

(均同) 中華民國 TW

四、聲明事項：

☑ 主張專利法第二十二條第二項☑第一款或☐第二款規定之事實，其事實發生日期為：95 年 6 月 9 日。

☐ 申請前已向下列國家（地區）申請專利：

【格式請依：受理國家（地區）、申請日、申請案號　順序註記】

☐ 有主張專利法第二十七條第一項國際優先權：

☐ 無主張專利法第二十七條第一項國際優先權：

☐ 主張專利法第二十九條第一項國內優先權：

【格式請依：申請日、申請案號　順序註記】

☐ 主張專利法第三十條生物材料：

☐ 須寄存生物材料者：

國內生物材料 【格式請依：寄存機構、日期、號碼　順序註記】

國外生物材料 【格式請依：寄存國家、機構、日期、號碼　順序註記】

☐ 不須寄存生物材料者：

所屬技術領域中具有通常知識者易於獲得時，不須寄存。

五、中文發明摘要：

本發明提供一種應用於點對點閘道器上之辨別及管理系統及其方法，其係設置於核心空間（kernel space）中，並於核心空間上外掛一核心模組，使封包的前置處理及應用程式處理皆可在核心空間上完成，不需將資料複製到使用者空間再做處理，另在核心空間中設置一連線快取（connection cache）用以處理所有封包的來源/目的IP位址及連接埠號，識別出相同的重新連線請求封包並將其擋下，因此應用本發明可加速內容過濾閘道器之通過量，並提升封包處理之效率。

六、英文發明摘要：

七、指定代表圖：

（一）、本案代表圖為：第二圖

（二）、本案代表圖之元件代表符號簡單說明：

20 核心空間	202 連線快取
203 第七層過濾器	204 佇列
206 封包處置程式	208 應用程式資料
209 程式區段	22 核心模組
222 應用程式模組	224 第二應用程式模組
24 使用者空間	26 libipq 函式庫

八、本案若有化學式時，請揭示最能顯示發明特徵的化學式：

九、發明說明：

【發明所屬之技術領域】

本發明係有關一種點對點閘道器之管理系統，特別是指一種應用於點對點閘道器上，可增進網路速度及效能之辨別及管理系統。

【先前技術】

在過去幾年中，點對點（P2P）檔案分享在網際網路傳輸方面以驚人的速度成長，因此如何管理點對點通訊之效能便成為一重要之課題。系統管理員通常利用眾所周知的幾個固定連接埠號將網路網路通訊進行分類管理，包括將特定應用程式的通訊傳輸阻擋掉，及在多種內容過濾（如病毒掃描）後重新導向至代理伺服器（proxy）。但是，這種分類方法在點對點通訊上並不適用，因為大多數的點對點應用程式皆使用動態連接埠，也就是自動選取一個連接埠而不是使用固定那幾個眾所周知的連接埠，因此，點對點應用程式應就應用層（application-layer）訊息之特徵來進行分類。傳統上分類步驟係於核心空間（kernel space）完成，因為其簡單特徵與資料內容的前幾位元組相合，然而，點對點分享檔案上所做之檔案過濾與掃毒等管理中，亦包含由封包所組成的資料複雜內容處理，由此觀之，此步驟於使用者空間中進行似乎較為自然。

縱使於使用者空間上執行，諸如 InstantScan 及 P2PADM 等點對點管理工具必須在核心空間及使用者空間之間交換資料，然而資料交換係將核心空間之資料複製到使用者空間，會大量消耗效能，而事實上，此消耗亦存在於網路伺服器套件（web server packages）中，如伺服器 HTTPd。為減少消耗，另一種核心內（in-kernel）套件之伺服器 kHTTPd 將伺服器 HTTPd

移至核心空間中，以直接於核心中掌握回應訊息，可避免資料交換並真正提供比應用使用者空間之伺服器 HTTPd 更高的效能。

底下敘述 P2PADM 之架構及管理方法，其為一種新式作業系統之閘道器結構，管理目的包括：(1)點對點應用程式之連線分類；(2)濾除不想要之點對點應用程式；(3)針對點對點分享檔案進行掃毒；(4)將聊天訊息及傳輸檔案過濾並審查；以及(5)控制點對點通訊之頻寬。如第一圖所示之架構，核心空間 10 中利用第七層過濾器（L7-filter）102 辨認連線分類，並將連線分類之封包儲存於佇列 104 中；代理伺服器中之一主要執行緒(main thread)透過呼叫 libipq 函式庫 122 及在封包處置程式 124 中執行前置處理作業，如總合檢查、封包分類及處理 TCP 序列後，從核心空間 10 的佇列 104 中取得封包；接著，主要執行緒呼叫一特定應用程式之執行緒，以控制與該應用程式協定相關之作業，每一應用程式之執行緒皆負責一特定連線，並決定要連線內之封包要通過或丟棄掉。

P2PADM 從佇列 104 中以 libipq 函式庫 122 取得封包，此 libipq 函式庫 122 為一種應用於 iptable 上之開發函式庫，其提供一應用程式介面以與 ip_queue 核心模組通訊，此 ip_queue 核心模組係利用 Netfilter 功能框架進行登錄，以將封包於核心空間 10 及使用者空間 12 之間傳遞。因此，P2PADM 必須進行核心與使用者模式之間的內容置換，將資料從核心空間 10 中複製到使用者空間 12 來管理點對點通訊，而複製資料會降低 P2PADM 之執行效能。

因此，本發明即針對上述習知技術中之缺失，提出一種應用於點對點

閘道器上之辨別及管理系統，以增加效能，有效克服上述之該等問題。

【發明內容】

本發明之主要目的在提供一種應用於點對點閘道器上之辨別及管理系統，其係設置一核心模組，其外掛於核心空間上，將應用程式模組設置於核心模組中處理通訊協定、過濾及審查等工作，便於修改應用程式通訊協定之處理工作。

本發明之另一目的在提供一種應用於點對點閘道器上之辨別及管理系統，其係設置一連線快取以處理封包之來源/目的 IP 位址及目的/來源連接埠號等資訊，當具有與上述資訊相同之封包時即判斷為重新連線封包，而連線快取可將該封包阻擋下。

本發明之再一目的在提供一種應用於點對點閘道器上之辨別及管理系統，其係利用快速通過（fast pass）機制，在閘道器中將脫序封包複製下來，並讓脫序封包快速通過，用以縮短封包遺失時之不確定性延遲。

為達上述之目的，本發明提供一種應用於點對點閘道器上之辨別及管理系統，其係設置於作業系統下核心空間（kernel space），包含一連線快取(connection cache)及一第七層過濾器(L7-filter)，連線快取接收複數封包，利用第七層過濾器比對封包之特徵進行分類，並於可識別連線之封包上加上一識別記號，再進行前置處理；一核心模組，外掛於核心空間上，該核心模組中包含至少一應用程式模組負責處理相關之封包的通訊協定處理、過濾及審查；以及在一使用者空間（user space）中處理病毒掃描。

本發明另提供一種上述應用點對點閘道器辨別及管理系統之方法，包

括下列步驟：複數封包進入一核心空間中之一連線快取中檢查封包之來源IP位址、目的IP位址及連接埠號；利用一第七層過濾器在核心空間內進行分類連線及特徵比對，並標記一識別記號於可識別連線之封包上；核心空間依據識別記號將不要之封包濾除或進行頻寬控制，再將封包傳送至一封包處置程式進行前置處理；以及利用一核心模組處理封包之通訊協定、過濾及審查後，封包處置程式將封包傳送出去。

底下藉由具體實施例詳加說明，當更容易瞭解本發明之目的、技術內容、特點及其所達成之功效。

【實施方式】

本發明係提供一種應用於點對點閘道器上之辨別及管理系統，如第一圖所示，本發明之應用於點對點閘道器上之辨別及管理系統中包括一核心空間（kernel space）20、一核心模組22及一使用者空間（user space）24，其中核心空間20中更包含一連線快取（connection cache）202、一第七層過濾器（L7-filter）203、至少一佇列（queue）204、一封包處置程式（packet handler）206及至少一應用程式資料208。連線快取202用以檢查來源/目的IP位址、目的連接埠號及通訊協定編號（protocol id），當連線快取202收到具有與上述四點相同之封包，就視為重新連線之封包，則將其阻擋；第七層過濾器203比對封包之特徵進行分類，並於可識別連線之封包上加上一個識別記號，而具有識別記號之封包則依序儲存於佇列204中；封包處理程式206用以檢查封包檢查碼（checksum）、識別連線（connection identification）以及處理TCP序列（TCP handling）等封包前置處理動作；

應用程式資料 208 中將程式碼切成複數個區段(section)以便於做後續處理。

核心模組 22 中至少一應用程式模組 222，其與應用程式資料 208 相對應，用以處理相關的封包，負責設定封包之通訊協定（protocol）進行處理、過濾及審查該封包等判決（verdict）。而封包之病毒掃描工作由於會消耗許多時間，可能中斷核心的運作，故將掃毒工作設置於使用者空間中。而 libipq 函式庫 26 則設置於核心模組 22 及使用者空間 24 之間的介面。

一開始，所有的封包都進入連線快取 202 中，檢查封包之來源 IP 位址、目的 IP 位址、目的連接埠號以及通訊協定編號；接著利用第七層過濾器 203 在核心空間 20 內進行分類連線及特徵比對，首先，第七層過濾器 203 收集開頭最多八個封包重新組合成應用程式訊息（application message），再進行特徵比對，若第七層過濾器 203 可識別此封包中所載連線，則標記一事先定義的識別記號於該封包上，有識別記號的封包儲存於佇列 204 中，核心空間 20 會依據識別記號將不要之封包濾除或進行頻寬控制，再將封包傳送至封包處置程式 206 進行前置處理；當封包前置處理完成後，會呼叫核心模組 22 中特定的應用程式模組 222，利用核心模組 22 處理封包之通訊協定、過濾及審查。

本發明之系統偶爾會呼叫 schedule 函式，把 CPU 控制權讓給其他行程使用，以避免發生餓死（starvation）的情況。schedule 函式是一個位於 schedule.c 中的 Linux 核心函式，其作用係對行程（process）進行排程。如果沒有其他行程需要使用 CPU，則 CPU 控制權會再回到本發明系統中。此外，本發明所提供之系統會呼叫 call_usermodehelper 函式以在使用者空間中

進行病毒掃描的工作，並且會阻擋 Linux 核心的執行直到病毒掃描的工作完成，為了預防長時間的阻擋，檔案資料會被分為許多片段（piece）來進行掃描。掃描完一個片段的資料之後，呼叫 schedule 函式，把 CPU 控制權讓給核心空間 20 或其他行程。

當應用本發明之系統於 Linux 作業系統下，其封包的流程如第三圖所示，首先如步驟 S10 及 S12，在 Linux 核心中喚起 init 行程後，建立一個新的核心執行緒，此核心執行緒用以執行本發明之系統，並且在 Linux 關閉（shutdown）時被終止；核心內的管理架構等待新的連線，以及呼叫 schedule 函式將 CPU 控制權轉移給其他的行程以避免餓死的發生，如步驟 S14 所述。接著如步驟 S16 判斷是否接收到封包，若是，則如步驟 S16 及 S18 所述，從 netlink 取得封包並判斷檢查碼是否正確；反之則回到步驟 S14 再次呼叫 schedule 函式。Netlink 是 Linux 系統中之 IP 服務通訊協定，當檢查碼不正確時，為了避免封包遺失或是反覆送出確認訊號，故如步驟 S22 所述讓封包快速通過，回到步驟 S14 再次呼叫 schedule 函式。

當總合正確時，接受一個新的連線，且如步驟 S24 所述，本發明之系統需維護一份該連線套接口（socket）的資料結構，並可利用這份資料結構進行 I/O 操作，而不必依賴較高層的函式。接著進行前置處理，如步驟 S26 及 S28 所述之封包分類與 TCP 序列處理，當前置處理作業皆完成後，如步驟 S30 至 S36 所述，本發明之系統以訊號通知特定的應用程式執行緒（API thread）處理封包，然後應用程式執行緒將設定封包的判決（verdict），依據判決決定要將該封包丟棄（drop）或接收。

本發明可有效處理脫序（out-of-order）封包，方法為在閘道器中複製那些脫序封包，並讓它們立刻通過，如第三圖中之步驟S22，如此一來，接收端可以早一點收到完整的檔案。在先前技術中，若是有任何封包遺失，這些脫序封包會在閘道器中排隊等候（queue）並由TCP逾時引發重新傳輸，這會延長傳輸時間；而本發明中，接收端會收到脫序封包並送出三個相同ACK訊號給發送端，以引發重新傳輸，由於重新傳輸是由三個相同的ACK訊號，而非由TCP逾時所引發，因此會縮短封包遺失時之不確定性延遲。

第四圖所示為有快速通過及沒有快速通過在不同封包遺失速率下之傳送時間曲線圖，封包遺失率從0%至5%以模擬實際環境。快速通過可減少FTP客戶端與FTP伺服器端之間的傳輸時間，由圖中可知兩點：(1)封包遺失率愈高，則有快速通過及沒有快速通過兩者間之傳輸時間差距愈多；以及(2)延遲時間愈長，則愈多傳輸時間可被減少。造成第一點的原因在於當封包遺失率增加時，閘道器中佇列時間會愈長，因此傳輸時間會更大；第二點是因為當每一封包之延遲增加時閘道器中之佇列時間會變長。簡而言之，當延遲時間及丟棄封包率增大時，快速通過可減少更多的傳輸時間。

通過量及CPU使用率為一閘道器系統中測量效能的兩個主要標準，底下以第五圖及第六圖分別顯示在不同組態下，本發明之系統與先前技術中之P2PADM系統之通過量及CPU使用率之比較，其中第六圖不只是完全地CPU使用率，同時提供核心部份之CPU使用率。由圖中可知，本發明之系統比P2PADM傳輸速率快，其原因不只是因為在核心空間編碼可減少資料從核心被複製到使用者空間，也因為可減少呼叫函式的數目。

第七圖及第八圖顯示當本發明之系統上具有連線快取時之通過量及CPU 使用率。在試驗中，利用阻擋所有從兩台客戶端主機其中之一所傳送來之封包，迫使被阻擋之客戶端必需持續送出重新連線的請求。連線快取可增加大約 15%之通過量，而由於本發明之系統中除了掃毒之外，所有的處理皆在核心空間中執行，故 CPU 使用率會被本發明之系統所佔用，從而使 CPU 使用率永遠可達到約 100%。

綜上所述，本發明提供之應用於點對點閘道器上之辨別及管理系統及其方法可快速掌握重新連線的封包並將之阻擋下，當脫序封包產生時則讓其快速通過以避免不確定性延遲（non-deterministic delays），更將封包前置處理部份搬移到核心空間中，減少在核心空間及使用者空間之間的資料傳遞動作來達到更好的封包處理效能，。

唯以上所述者，僅為本發明之較佳實施例而已，並非用來限定本發明實施之範圍。故即凡依本發明申請範圍所述之特徵及精神所為之均等變化或修飾，均應包括於本發明之申請專利範圍內。

【圖式簡單說明】

第一圖為先前技術中 P2PADM 系統之示意圖。

第二圖為本發明應用於點對點閘道器上之辨別及管理系統之方塊圖。

第三圖為本發明之系統中封包之流程圖。

第四圖為有無快速通過在不同封包遺失速率下之傳送時間曲線圖。

第五圖為在不同組態下，本發明之系統與 P2PADM 系統之通過量示意圖。

第六圖為在不同組態下，本發明之系統與 P2PADM 系統之 CPU 使用率示意

圖。

第七圖為本發明之系統上具有連線快取時之通過量示意圖。

第八圖為本發明之系統上具有連線快取時之 CPU 使用率示意圖。

【主要元件符號說明】

10 核心空間

102 第七層過濾器

104 佇列

12 使用者空間

122 libipq 函式庫

124 封包處置程式

20 核心空間

202 連線快取

203 第七層過濾器

204 佇列

206 封包處置程式

208 應用程式資料

209 程式區段

22 核心模組

222 應用程式模組

24 使用者空間

26 libipq 函式庫

十、申請專利範圍：

1. 一種應用於點對點閘道器上之辨別及管理系統，其係設置於作業系統下之虛擬記憶體空間中，包括：

一核心空間（kernel space），包含一連線快取(connection cache)及一第七層過濾器(L7-filter)，該連線快取接收複數封包，利用該第七層過濾器比對該封包之特徵進行分類，並於可識別連線之該封包上加上一識別記號，再進行前置處理；

一核心模組，外掛於該核心空間上，該核心模組中包含至少一應用程式模組負責處理相關之該封包的通訊協定處理、過濾及審查；以及

一使用者空間（user space），於該使用者空間中處理病毒掃描之工作。

2. 如申請專利範圍第 1 項所述之應用於點對點閘道器上之辨別及管理系統，其中該連線快取於系統剛啟動後，在該封包進入之前係為空的，使所有該封包皆可進入該連線快取中。

3. 如申請專利範圍第 1 項所述之應用於點對點閘道器上之辨別及管理系統，其中該連線快取係檢查該封包之來源 IP 位址、目的 IP 位址及連接埠號。

4. 如申請專利範圍第 1 項所述之應用於點對點閘道器上之辨別及管理系統，其中該連線快取可更新連線資訊。

5. 如申請專利範圍第 1 項所述之應用於點對點閘道器上之辨別及管理系統，其中該核心空間可依據該識別記號將不想要之該封包濾除或進行頻寬控制。

6. 如申請專利範圍第 1 項所述之應用於點對點閘道器上之辨別及管理系

統，其中該第七層過濾器收集至多八個開頭之該封包以重新組合成一應用程式訊息，並進行特徵比對。

7. 如申請專利範圍第 1 項所述之應用於點對點閘道器上之辨別及管理系統，更包括一封包處置程式(packet handler)，進行檢查該封包之檢查碼（checksum）、連線識別及 TCP 序列處理等前置處理動作。

8. 如申請專利範圍第 7 項所述之應用於點對點閘道器上之辨別及管理系統，更包括至少一佇列，將具有該識別記號之該封包儲存於該佇列中，依序送出至該封包處置程式。

9. 如申請專利範圍第 1 項所述之應用於點對點閘道器上之辨別及管理系統，其中該使用者空間中在進行掃毒工作時，該核心空間及該核心模組之動作將會暫停。

10. 如申請專利範圍第 1 項所述之應用於點對點閘道器上之辨別及管理系統，更包括一排程函式，用以將行程(process)排程，呼叫該排程函式以將中央處理器之控制權轉讓給其他行程使用。

11. 如申請專利範圍第 1 項所述之應用於點對點閘道器上之辨別及管理系統，其中該連線快取係判斷接受該封包或丟棄。

12. 如申請專利範圍第 1 項所述之應用於點對點閘道器上之辨別及管理系統，其中該連線快取係過濾重覆連線，以提升系統效能。

13. 一種應用點對點閘道器辨別及管理系統之方法，包括下列步驟：

複數封包進入一核心空間中之一連線快取中檢查該封包之來源 IP 位址、目的 IP 位址及連接埠號；

利用一第七層過濾器在該核心空間內進行分類連線及特徵比對，並標記一識別記號於可識別連線之該封包上；

該核心空間依據該識別記號將不要之該封包濾除或進行頻寬控制，再將該封包傳送至一封包處置程式進行前置處理；以及

利用一核心模組處理該封包之通訊協定、過濾及審查後，該封包處置程式將該封包傳送出去。

14.如申請專利範圍第 13 項所述之應用於點對點閘道器上之辨別及管理方法，其中該連線快取在該封包進入之前係為空的，使所有該封包皆可進入該連線快取中。

15.如申請專利範圍第 13 項所述之應用於點對點閘道器上之辨別及管理方法，其中該連線快取可更新連線資訊。

16.如申請專利範圍第 13 項所述之應用於點對點閘道器上之辨別及管理方法，其中該第七層過濾器收集至多八個開頭之該封包以重新組合成一應用程式訊息，並進行特徵比對。

17.如申請專利範圍第 13 項所述之應用於點對點閘道器上之辨別及管理方法，其中該前置處理動作係包含檢查封包檢查碼、連線識別及 TCP 序列處理等。

18.如申請專利範圍第 13 項所述之應用於點對點閘道器上之辨別及管理方法，其中該第七層過濾器更將具有該識別記號之該封包儲存於一佇列中，依序送出至該封包處置程式。

19.如申請專利範圍第 13 項所述之應用於點對點閘道器上之辨別及管理方

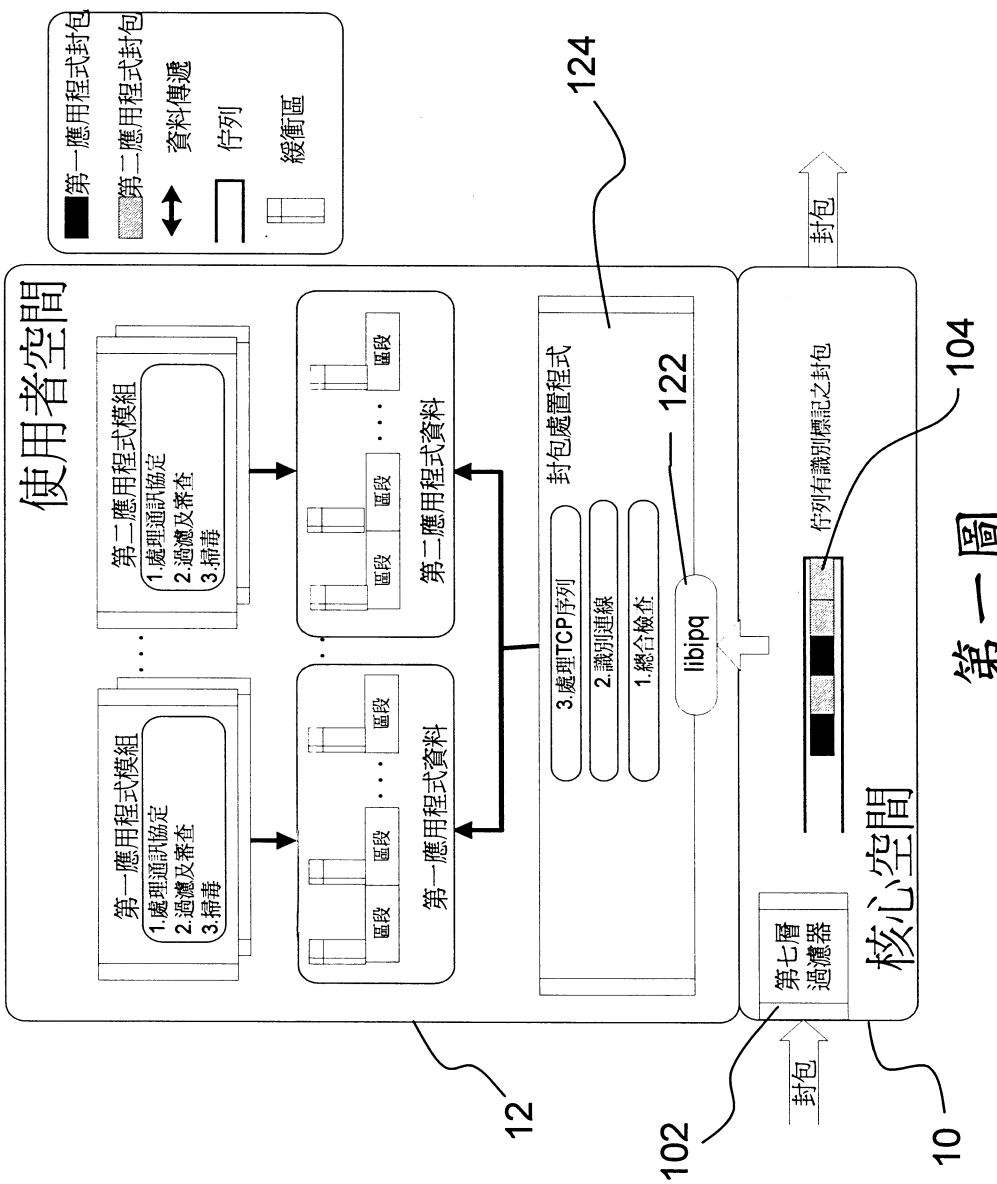
法，其中該封包之病毒掃描動作係在一使用者空間中進行。

20.如申請專利範圍第 19 項所述之應用於點對點閘道器上之辨別及管理方法，其中該使用者空間中在進行掃毒工作時，該核心空間及該核心模組之動作將會暫停。

21.如申請專利範圍第 13 項所述之應用於點對點閘道器上之辨別及管理方法，更包括一排程函式，用以將行程(process)排程，呼叫該排程函式以將中央處理器之控制權轉讓給其他行程使用。

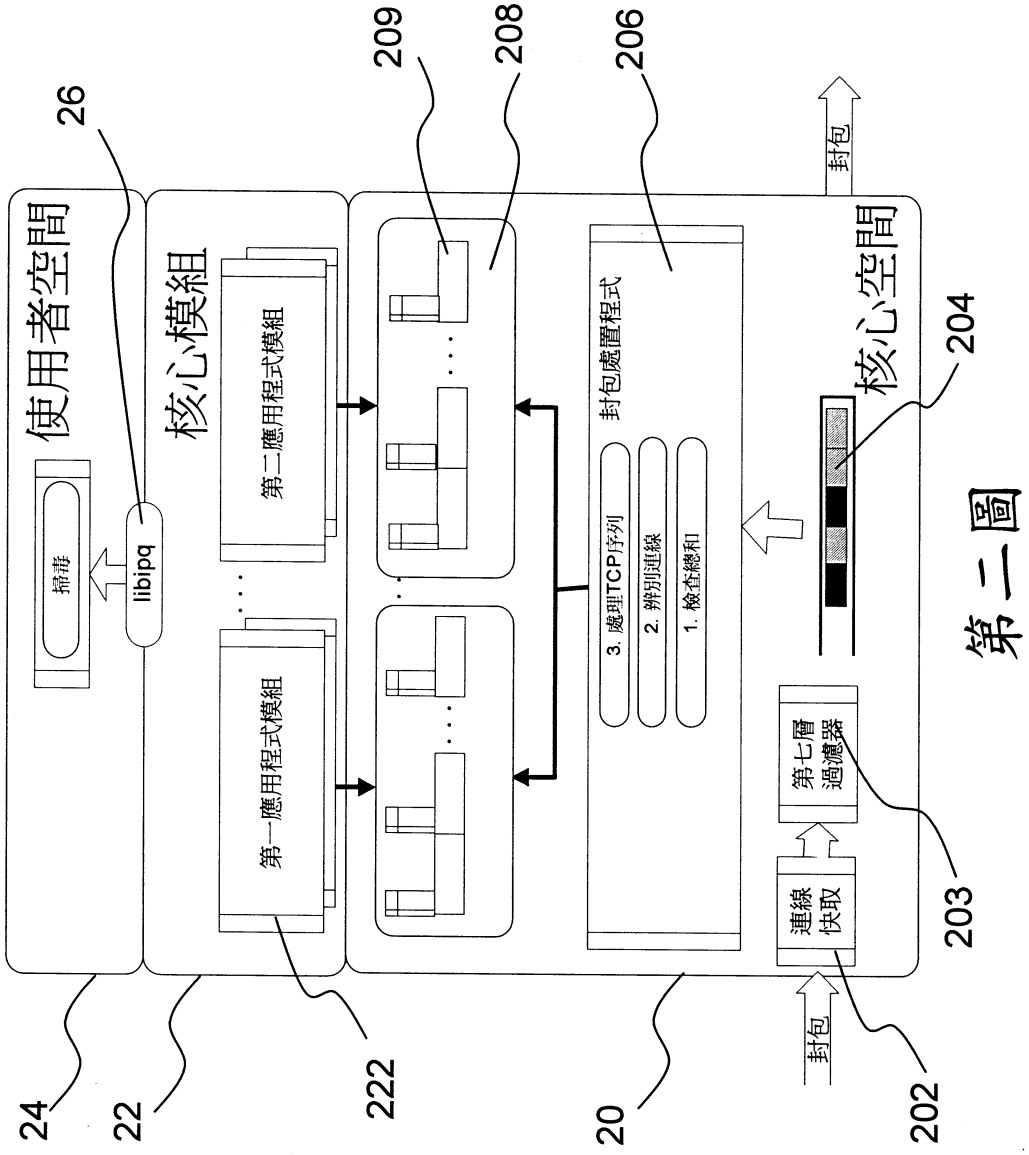
22.如申請專利範圍第 13 項所述之應用於點對點閘道器上之辨別及管理方法，其中該核心模組係外掛於該核心空間之外。

200826571

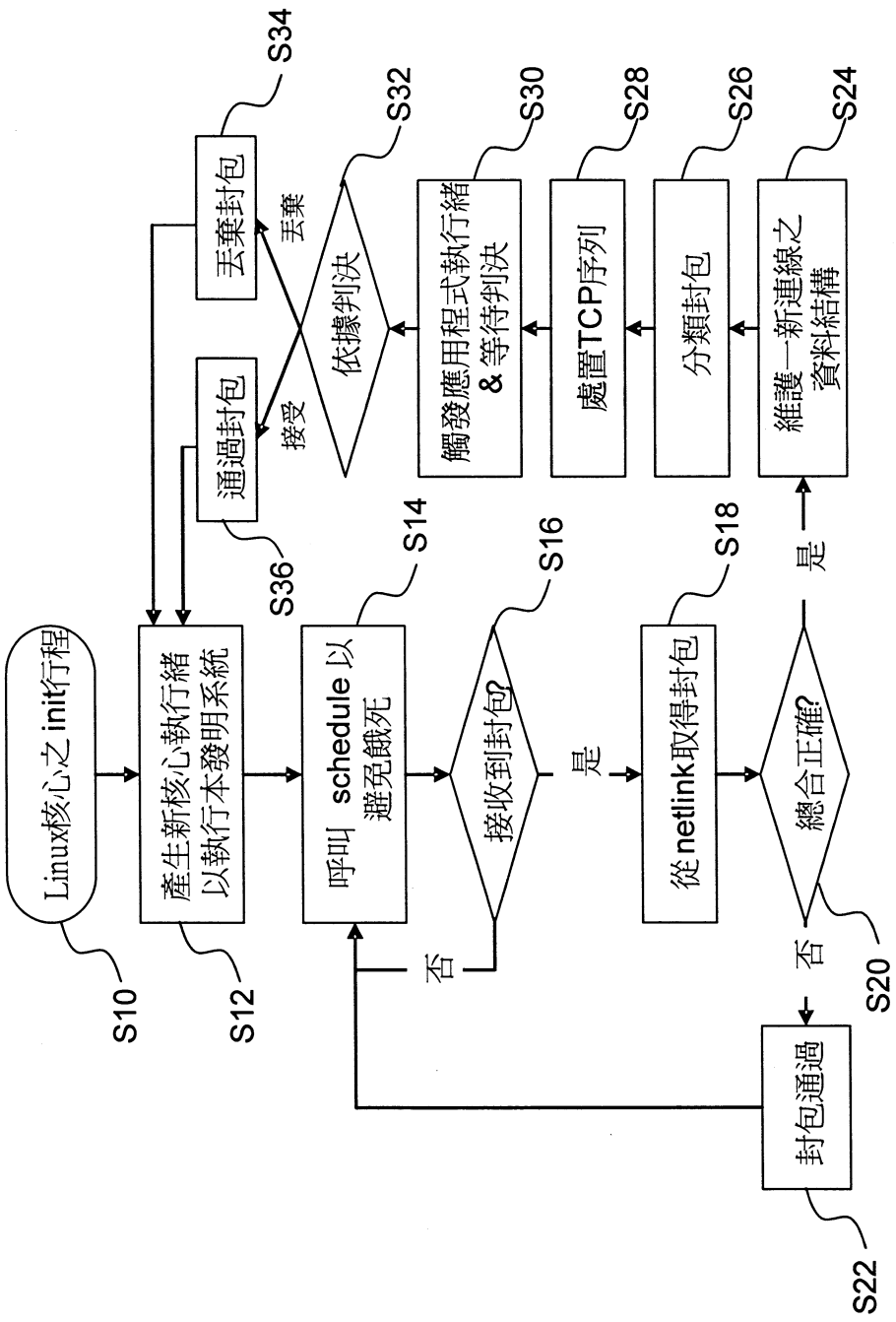


第一圖
（先前技術）

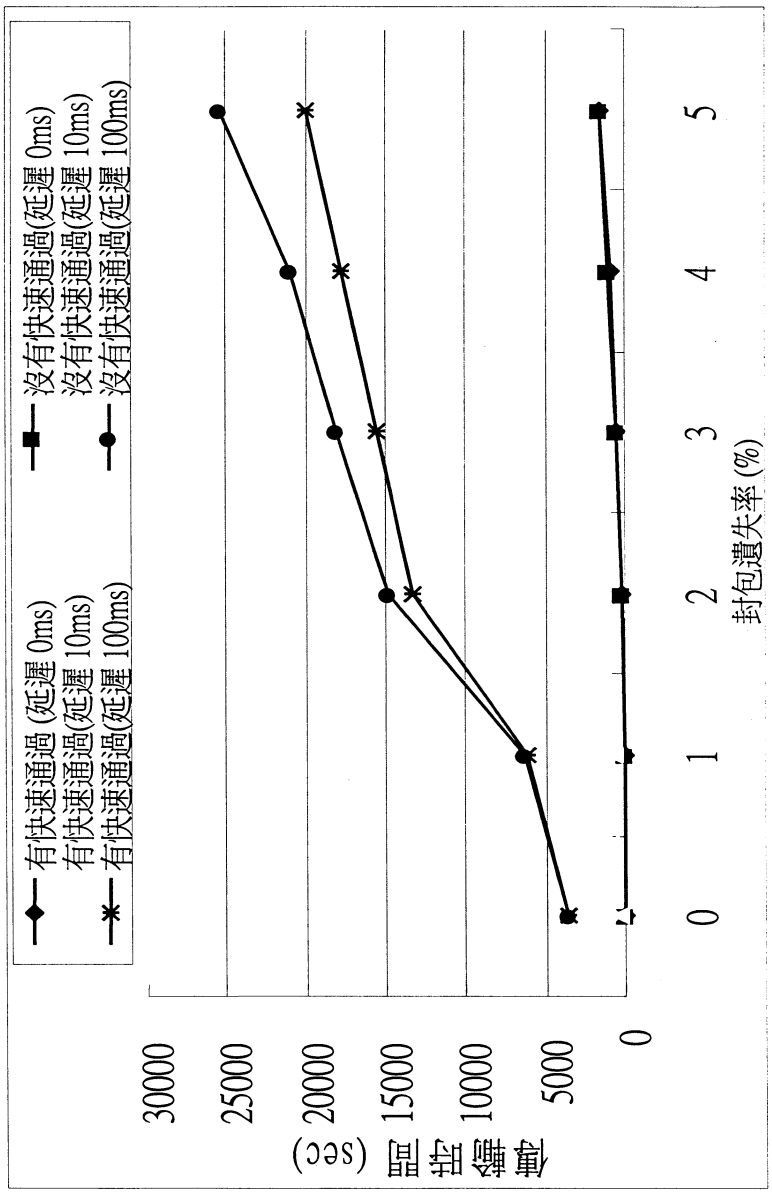
200826571



第二圖


Linux核心之 init行程
S10
產生新核心執行緒
以執行本發明系統
S12
呼叫 schedule 以
避免餓死
S14
接收到封包?
S16
否
是
從netlink取得封包
S18
總合正確?
S20
否
是
封包通過
S22
維護一新連線之
資料結構
S24
分類封包
S26
處置TCP序列
S28
觸發應用程式執行緒
& 等待判決
S30
依據判決
S32
丟棄封包
S34
丟棄
接受
通過封包
S36

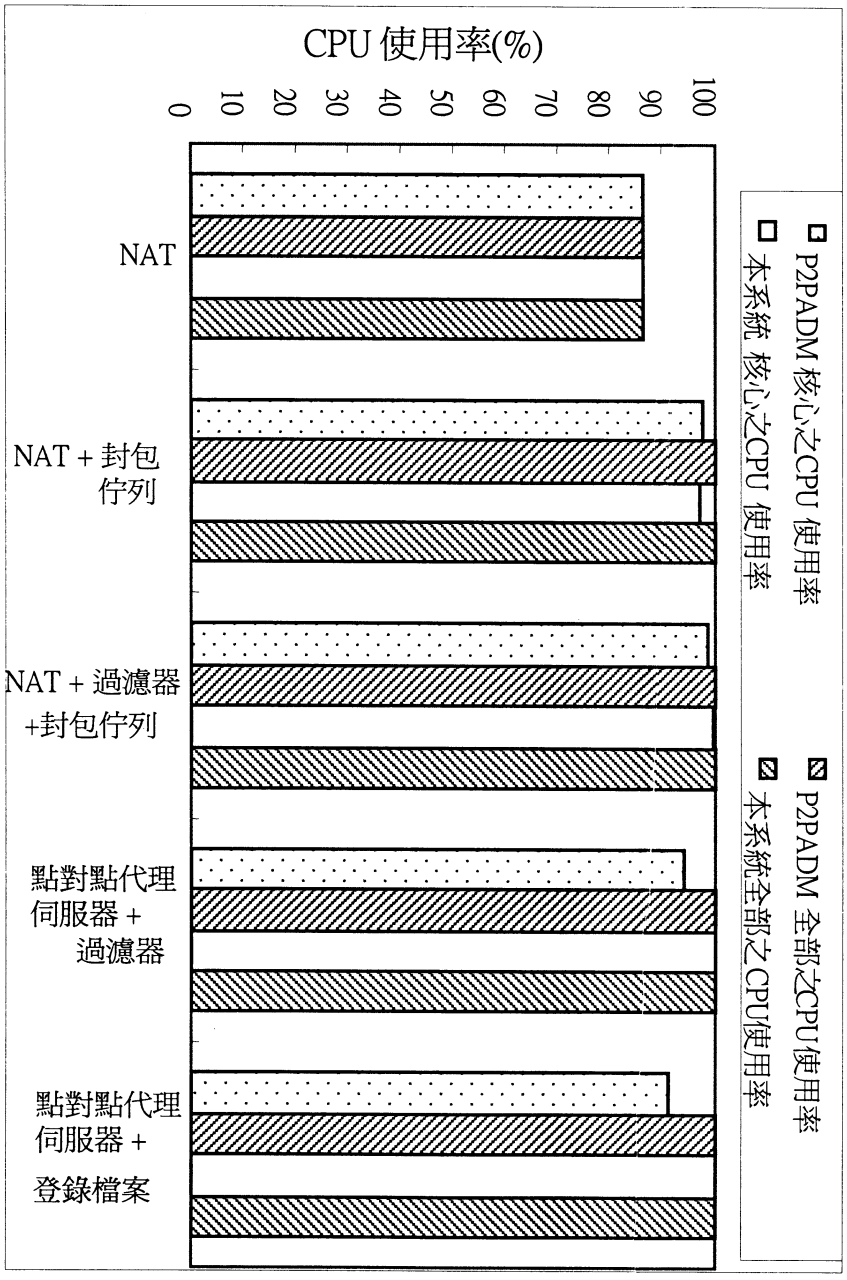
第三圖


有快速通過 (延遲 0ms)
沒有快速通過(延遲 0ms)
有快速通過(延遲 10ms)
沒有快速通過(延遲 10ms)
有快速通過(延遲 100ms)
沒有快速通過(延遲 100ms)
傳輸時間 (sec)
30000
25000
20000
15000
10000
5000
0
0
1
2
3
4
5
封包遺失率 (%)

第四圖

P2PADM
本發明系統
通過量(Mbps)
300
250
200
150
100
50
0
266.13
155.24
223.71
89.25
178.1
85
164.68
69.98
133.17
NAT
NAT + 封包佇列
NAT + 第七層過濾器 + 封包佇列
點對點代理伺服器 + 過濾器
點對點代理伺服器 + 登錄檔案

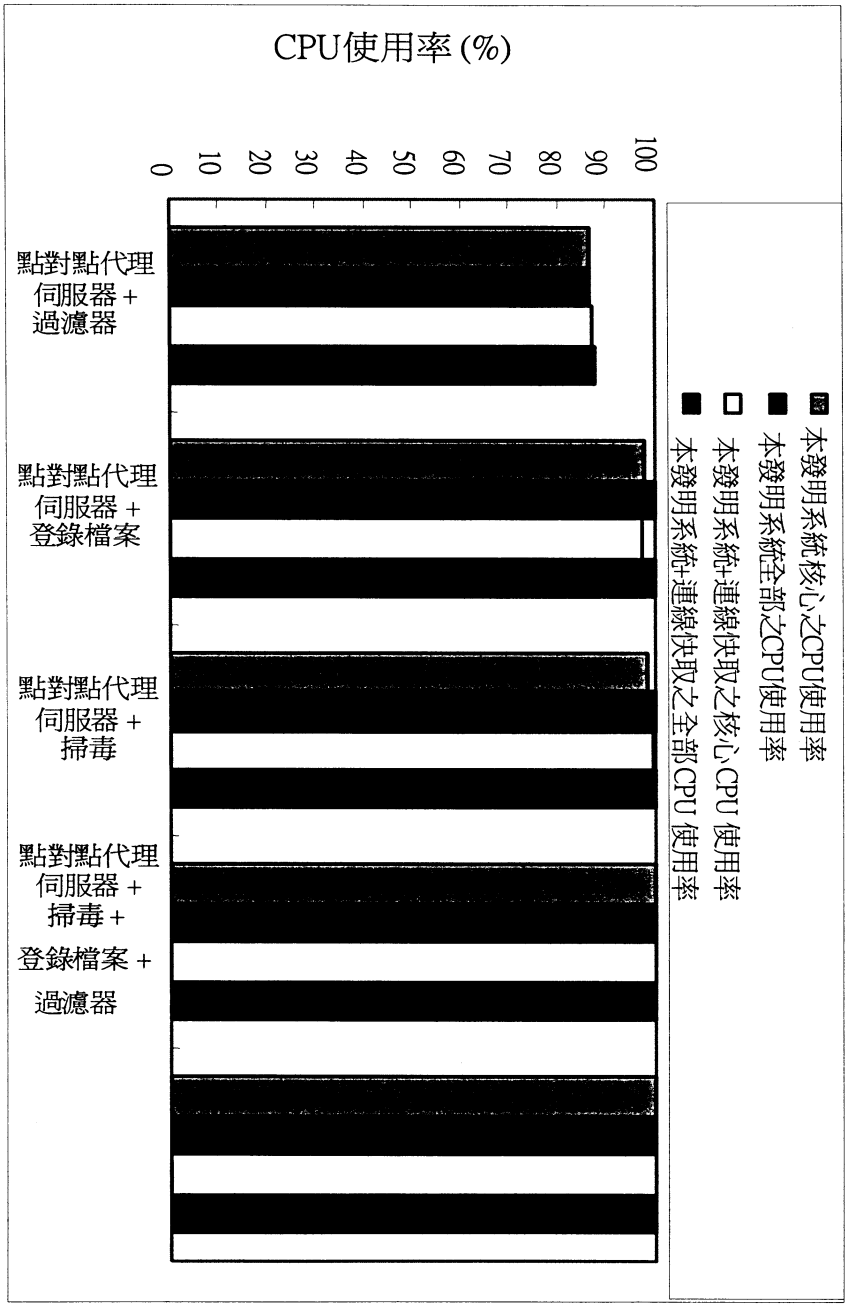
第五圖


CPU 使用率(%)
0
10
20
30
40
50
60
70
80
90
100
NAT
NAT + 封包佇列
NAT + 過濾器 +封包佇列
點對點代理伺服器 + 過濾器
點對點代理伺服器 + 登錄檔案
P2PADM 核心之CPU 使用率
本系統 核心之CPU 使用率
P2PADM 全部之CPU使用率
本系統全部之CPU使用率

第六圖

□ 本發明系統 ▨ 本發明系統 +連線快取
通過量 (Mbps)
300
250
200
150
100
50
0
266.13
223.71
223.7
178.1
199.4
164.68
185.73
133.17
167.12
NAT
NAT + 封包
佇列
NAT + 第七
層過濾器 +
封包佇列
點對點代理
伺服器 +
過濾器
點對點代理
伺服器 +
登錄檔案

第七圖


CPU使用率 (%)
0
10
20
30
40
50
60
70
80
90
100
點對點代理伺服器 + 過濾器
點對點代理伺服器 + 登錄檔案
點對點代理伺服器 + 掃毒
點對點代理伺服器 + 掃毒 + 登錄檔案 + 過濾器
本發明系統核心之CPU使用率
本發明系統全部之CPU使用率
本發明系統+連線快取之核心CPU 使用率
本發明系統+連線快取之全部CPU 使用率

第八圖