國立交通大學

資訊工程學系

碩士論文

無線網狀網路QoS路由技術與傳輸流量平衡策略

A Novel QoS Routing and Load Balancing Scheme for Wireless Mesh Networks

研究生：徐雲婷

指導教授：曾建超 教授

中華民國九十四年七月

無線網狀網路 QoS 路由技術與傳輸流量平衡策略

A Novel QoS Routing and Load Balancing Scheme for Wireless Mesh Networks

研 究 生：徐雲婷　　Student：Yun-Ting Hsu

指導教授：曾建超　　Advisor：Chien-Chao Tseng

國 立 交 通 大 學

資 訊 工 程 系

碩 士 論 文

A Thesis

Submitted to Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science and Information Engineering

July 2005

Hsinchu, Taiwan, Republic of China

中華民國九十四年七月

無線網狀網路 QoS 路由技術與傳輸流量平衡策略

研究生：徐雲婷　　指導教授：曾建超 博士

國立交通大學

資訊工程所碩士班

摘要

在本篇論文中，我們針對 multi-homed 無線網狀網路(Wireless Mesh Network)的環境下，提出一個傳輸流量平衡(load-balanced)的閘道伺服器(gateway)選取演算法和一個考量到傳輸服務品質(Quality of Services，QoS)狀態的路由協定。multi-homed 無線網狀網路是一個由多個可以連到網際網路(Internet)的閘道伺服器構成的無線網狀網路。由於無線網路技術的蓬勃發展，無線網狀網路已成為最有可能被用於建立都市網路和解決寬頻服務的最後一哩(last mile) 問題的方法。因此，無線網狀網路已成為最近熱門的研究項目之一。另一方面，多媒體服務像是 Voice over IP 或是 Video on Demand 等服務，已經在網際網路中逐漸變的越來越普遍了。然而，網際網路使用的是盡力而為(Best Effort)的資料傳輸方法，它並無法滿足這些多媒體服務的頻寬以及低延遲的要求。為了提供 QoS 給多媒體應用程式，網際網路工程小組(Internet Engineering Task Force，IETF)也有針對有線網路提出許多保證 QoS 的規範。因為無線網路的普遍性越來越高，許多 IETF 的工作小組目前也在研究該如何在無線區域網路，ad-hoc 和網狀網路中提供 QoS 的服務。在本篇論文中，我們針對 multi-homed 無線網狀網路的環境下，針對網狀網路的 backhaul routers 提出一個傳輸流量平衡的閘道伺服器選取演算法和考量到傳輸服務品質的最短路由機制。

目前的傳輸服務品質的路由方法，大都是針對已知的一對傳送和發送端，如何在所有的可能路徑中選擇一條滿足使用者 QoS 需求的路徑。為了要選擇一條最為適當的路徑，發送端需要知道網路上所有連線(links)的狀態。因此，每當有一個連線的狀態改變，連線上的節點(node)就需要將改變廣播給其他的節點知道。對於在無線網狀網路的環境中，這樣的資訊交換的負擔太大。除此之外，在 multi-homed 網狀網路中存在有多個閘道伺服器，而閘道伺服器的選取演算法會影響到路徑的選擇。然而，以往的傳輸服務品質路由演算法並沒有將這樣的環境考量進去。

為了減少交換連線資訊的流量，我們首先提出一套閘道伺服器選取演算法來決定替每個移動用戶端(mobile node)服務的閘道伺服器。再來，配合逐節點的路徑選取(Hop-by-hop Path Selection)機制來找出一條到閘道伺服器的最短可能路徑，也就是說路徑上的連線必需滿足移動用戶端的頻寬的要求。閘道伺服器選取演算法不僅僅能透過將移動用戶端的網路流量導到不一樣的閘道伺服器來平衡網路的流量，並且也能幫助逐節點的路徑選取機制消除掉廣播連線資訊的需要，因為每個 backhaul router 只需要知道和鄰居 backhaul routers 的頻寬狀態而已。最後，我們進行模擬分析，模擬的結果顯示我們的方法在不同的網路傳輸流量下，都有達到流量平衡的狀態並且可以提供服務給更多的使用者。

A Novel QoS Routing and Load Balancing Scheme for Wireless Mesh Networks

Student: Yun-Ting Hsu Advisor: Dr. Chien-Chao Tseng

Department of Computer Science and Information Engineering

National Chaio Tung University

Abstract

In this thesis, we propose a load-balanced gateway selection algorithm and a QoS-aware routing protocol for multi-homed wireless mesh network, that is wireless mesh networks with multiples access gateways to Internet. As wireless network technology advances, wireless mesh networks have become the most promising approach to building the metropolitan and last mile access networks, and thus have become one of the most attractive research topics. On the other hand, multimedia services, such as Voice over IP and Video on Demand, have gradually become more and more popular in Internet. However, the best effort data transport method of Internet cannot fulfill the bandwidth and latency requirements of multimedia services. In order to provide Quality of Services (QoS) for multimedia applications, IETF has also proposed several QoS assurance standards for wireline Internet. Because wireless access networks become more and more prevalent, several IETF working groups also working on providing QoS for Internet with wireless LAN, ad-hoc or mesh networks. In this thesis, we focus on multi-homed wireless mesh networks and propose a load-balanced gateway selection algorithm and QoS-aware shortest routing mechanism for the backhaul routers of the mesh networks.

Current QoS routing algorithms aim to select a path that can fulfill user's QoS requirement for a given pair of source and destination. In order to select an appropriate path, the source node needs to know the bandwidth statues of all links. Therefore each time when a link state changes, the nodes on the link have to broadcast the changes to all other nodes over the network. For wireless mesh networks, the link state broadcast may cause too much traffic flow. Furthermore, there exist multiple gateways in a multi-homed mesh network, and the gateway selection algorithm may have effects on route selection. However, current QoS routing algorithms do not take this conditions into consideration.

To decrease link state information exchange traffic, we present a gateway

selection algorithm to select a serving gateway for each mobile node first. We then propose a hop-by-hop path selection algorithm that selects a shortest feasible path, i.e., a path where each link on the selected path fulfills the mobile node's QoS requirement, to the selected serving gateway. Gateway selection algorithm can achieve network load balancing by guiding mobile nodes' traffics to different gateways. Furthermore, it also helps hop-by-hop routing eliminate the necessity of link state broadcast since each backhaul router needs to know the link bandwidth of the neighbor backhaul routers only. Simulation results show that our algorithm can indeed achieve load balance and serve more users under various traffic loads.

誌謝

本論文能順利的完成，首先要感謝我的指導教授—曾建超博士，在過去這兩年來對我的指導與口試期間的細心指點。感謝我的口試委員：楊人順博士與曹孝櫟博士感謝他們細心的審查我的論文，並提供寶貴的意見，讓我得以更完善我的論文。另外，也很感謝徐元瑛學姐和王瑞堂學長在研究的過程中所給予我的建議與啟發，讓我得以完成本篇論文。還要感謝實驗室所有的學長姐、同學、學弟妹們的幫忙與支持，讓我在這兩年之中過得很充實，謝謝你們。最後，我要感謝我的父母與朋友們的陪伴，給予我精神層面上的鼓勵，讓我得以繼續堅持下去。

僅以此論文獻給我親愛的家人，以及所有關心我的師長和朋友們。

目錄

中文摘要 i
英文摘要 iii
誌謝 v
目錄 vi
圖目錄 viii
表目錄 ix
第一章 緒論 1
1.1 研究動機 1
1.2 研究目標 1
1.3 論文內容 2
第二章 研究背景與相關論文研究 3
2.1 無線網狀網路（Wireless Mesh Network） 3
2.2 QoS Reservation機制（RSVP） 5
2.3 路徑選擇演算法 7
2.3.1 路徑選擇時的兩種不同的approaches 7
2.3.2 路徑選擇算法 8
2.4 閘道伺服器選取法 9
第三章 Path Reservation in Wireless Mesh Network 12
3.1 系統架構與前提假設 12
3.1.1 整體設計方向 12
3.1.2 系統架構 13
3.1.3 前提假設 14
3.2 Mesh routing protocol 15
3.2.1 Mesh routing protocol設計概念 15

3.2.2 Mesh routing table 17
3.2.3 Mesh routing protocol 運作流程 19
3.2.3.1 Mesh routing table建立 19
3.2.3.2 Routing table update 22
3.2.3.3 不能傳送routing table的原因 25
3.3 Path selection 27
3.3.1 Path request message 28
3.3.2 Path selection 29
3.3.3 Gateway selection 30
3.3.3.1 閘道伺服器端 31
3.3.3.2 backhaul router端 32
第四章 simulation 34
4.1 模擬模型假設 34
4.1.1 網路拓樸架構 34
4.1.2 模擬架構 36
4.1.3 performance metrics 37
4.2 模擬結果分析 38
4.2.1 Balanced環境模擬 38
4.2.2 Unbalanced環境模擬 41
4.2.2.1 左右區域的MN attach rate分別為0.7和0.3 41
4.2.2.2 左右區域的MN attach rate分別為0.5和0.5 45
4.2.2.3 左右區域的MN attach rate分別為0.3和0.7 47
第五章 結論與未來工作 50
5.1 結論 50
5.2 未來工作 50
參考文獻 51

圖目錄

圖 2.1 Wireless Mesh Network Architecture 4
圖 2.2 RSVP在Hosts和Routers的運作關係圖 6
圖 2.3 RSVP message flow圖 7

圖 3.1 mobile node連上internet範例 14
圖 3.2 routing protocol選取閘道器的原因 17
圖 3.3 routing information message 20
圖 3.4 mesh routing table更新流程圖 22
圖 3.5 BR-BR link中斷狀況下step2 的Routing table更新流程圖 25
圖 3.6 routing information message不傳routing table的原因 27
圖 3.7 routing path reservation流程圖 28
圖 3.8 path request message格式 29

圖 4.1 Topology Model 1 of Simulation 35
圖 4.2 Topology Model 2 of Simulation 36
圖 4.3 Balanced下，三種選取法的閘道伺服器負載 40
圖 4.4 Balanced下，Call Blocking Rate比較圖 40
圖 4.5 Balanced下，Bandwidth Blocking Rate比較圖 41
圖 4.6 (1)Unbalanced下，Load Balance的閘道伺服器負載圖 42
圖 4.7 (1)Unbalanced下，Random的閘道伺服器負載圖 43
圖 4.8 (1)Unbalanced下，Nearest的閘道伺服器負載圖 43
圖 4.9 (1)Unbalanced下，call blocking rate比較圖 44
圖 4.10 (1)Unbalanced下，bandwidth blocking rate比較圖 45
圖 4.11 (2)Unbalanced下，Load Balance的閘道伺服器負載圖 45
圖 4.12 (2)Unbalanced下，Random的閘道伺服器負載圖 46
圖 4.13 (2)Unbalanced下，Nearest的閘道伺服器負載圖 46
圖 4.14 (2)Unbalanced下，call blocking rate比較圖 46
圖 4.15 (2)Unbalanced下，bandwidth blocking rate比較圖 47
圖 4.16 (3)Unbalanced下，Load Balance的閘道伺服器負載圖 47
圖 4.17 (3)Unbalanced下，Random的閘道伺服器負載圖 48
圖 4.18 (3)Unbalanced下，Nearest的閘道伺服器負載圖 48
圖 4.19 (3)Unbalanced下，call blocking rate比較圖 49
圖 4.20 (3)Unbalanced下，bandwidth blocking rate比較圖 49

表目錄

表格 3.1 mesh routing table 18

表格 3.2 圖 3.4 的routing table情況 27

第一章　緒論

1.1　研究動機

由於網路技術的蓬勃發展，所以人們使用網路的頻率愈來愈頻繁，每個人花在電腦上的時間也相對地增加。最初，網路只是用來傳遞、查詢資料用，但慢慢地人們已經開始利用網路來進行溝通互動。到了後來，愈來愈多的網路通訊應用軟體相應而生，像是使用 MSN 和 BBS（Bulletin Board System）透過丟訊息聊天，或是使用網路電話（Voice over IP）可以直接聽到彼此講話的聲音；使用互動式遠距教學（Interactive Distance Education）的方式讓需要修課的學生不再受限於一定要在坐教室裡聽課，透過網路也一樣能在家裡或是其它地方聽課等各式各樣的通訊服務。隨著通訊複雜度增加，所需的網路頻寬和傳輸品質也跟著提升。

然而，傳統的資料傳輸並沒有提供傳輸服務品質（Quality of Service, QoS）[1]的保證，而是以所謂盡力而爲“Best Effort”的方式來提供頻寬服務。這樣的方式對於以往如網頁瀏覽、E-mail 等服務倒還可以被接受，但在面對電子商務、VoIP、視訊會議(Video Conference)等多媒體資訊傳輸服務，則無法滿足這些應用的頻寬以及低延遲的要求。目前這種 QoS 要求不但在有線網路（wired infrastructure network）和基礎建設無線區域網路（Infrastructure wireless LAN）迫切需要，甚至在無線隨意區域網路（Ad-hoc wireless LAN）和無線網狀網路（Wireless Mesh Network）[2][3]也極受重視。

1.2　研究目標

在研究過無線網狀網路的拓樸架構，並針對 mobile node 使用網路的狀況進行分析後，我們發現當 mobile node 在無線網狀網路裡要使用網際網路的資源時，最終路由會走過一個閘道伺服器再連上網際網路，因此閘道伺服器會是一個 bottleneck。若是閘道伺服器和與他有連線的 backhaul router 的 link 的頻寬都滿了的話，即使 backhaul router 和 backhaul router 的 link 都還有多餘的頻寬可以滿足 mobile node 的頻寬需求，mobile node 還是沒辦法使用網際網路資源。因此我們在此設計一個 routing

protocol，它會針對閘道伺服器的負載狀況，計算出一個適當地權重來作爲選擇最終會走哪一個閘道伺服器的依據。

我們希望設計一套完整的路由機制，來幫助 mobile node 在無線網狀網路中做 QoS 路由。我們的研究目標如下：

- 設計出 routing table 的建立方式以及路由的運作流程。
- 分析閘道伺服器權重該如何設定才能提供最佳的使用效能，提升無線網狀網路所能提供的 QoS 服務品質。
- 嘗試分析權重設定對於網路的影響。

1.3 論文內容

關於這一篇論文的內容簡述如下：在第二章「研究背景與相關論文研究」裡，我們會對無線網狀網路作一個介紹，說明無線網狀網路的特性和組成元件。再來我們會介紹要做 QoS 的 reservation 時，會採用的 reservation protocol，也就是 Resource Reservation Protocol (RSVP)[4]，並說明他的大致運作流程，最後在介紹制訂 QoS 路由演算法時所會考量的問題，以及目前最常見的路由演算法。我們將在第三章中提出我們的方法:在 3.1 節裡介紹我們提出的系統架構以及設計的一些想法與架設前提。而 3.2 節則要開始介紹我們訂出的路由機制，3.3 節則介紹我們是怎麼作路由的。在第四章裡，我們會針對我們的方法作模擬，並和一些其他的方法作比較。最後在第五章「結論與未來工作」的部分，將對此篇論文作一個總結，並對未來能夠繼續研究的方向提出一些心得與看法。

第二章　研究背景與相關論文研究

2.1　無線網狀網路（Wireless Mesh Network）

無線網狀網路是一個 Multi-hop Network，是由一些具有路由功能的 nodes 所構成，這些裝置都需要插上電源。每一個 node 會盡量用較少的功率去傳輸可以讓 neighbor node 聽到的訊號，而其 neighbor node 會將收到的資料再轉送，直到資料到達目的地。所以無線網狀網路可以擴展到很長的距離，可以應用在佈線不方便的地區，像是地勢高地起伏過大的地方。

和目前的 single-hop WLAN 比較，Wireless Mesh Network[5]的優點有：

- Low power

  Multi-hop networks 由於只需和鄰近的裝置溝通，所以所需比較少的 power，和其他裝置的無線訊號干擾也會比較少。因此，當可以使用的頻道有限時，multi-hop network 可以因爲重複使用頻道，而增進 capacity。而 single-hop network 中，每個裝置需要使用足夠讓其他任何裝置收聽到的 power 來進行資料傳輸。當兩個裝置同時要進行傳輸時，需要競爭頻道的使用權，因而導致 capacity 受到限制。

- Reliable

  在無線網狀網路中，每一個裝置都和多個裝置相連。當某一個裝置因爲硬體故障或其他問題而脫離網路，則他的鄰居裝置會找到其他可以路由的路徑。

- Traffic Balancing

  在網路連線密集的網狀網路中，每個裝置會有多個鄰居裝置，因此在任兩個裝置中會存在有多個不同的路徑。當有局部的干擾產生時，就可以將路由繞往其他的路徑。或是當一個裝置需要很大的頻寬，則可以動態將流量路由到其

他的網路裝置去，避免經過擁塞的裝置。目前的 single-hop 網路當遇到干擾或擁塞的 network access points 並沒有辦法去做動態的調適。

由於 Wireless Mesh Network[6]主要是要應用在網路的骨幹架構，所以需要比較大的頻寬和傳輸距離，故目前 intel 在推廣將 802.16（WiMax）[7]的規格應用在 Wireless Mesh Network 上面，其內容主要定義 physical layer 和 MAC layer 的規格。現在已經有一個 IEEE 802.11s 的 team 在制訂 Wireless Mesh Network 的 common baseline support。

Wireless Mesh Network的拓樸如圖 2.1所示，每個可以直接互相溝通的裝置都有一條線相連。但即便所有的裝置都是固定不動的，網路拓樸還是可能會因爲無線頻道的配置和干擾而不斷地在變動。主要的組成單元描述如下：

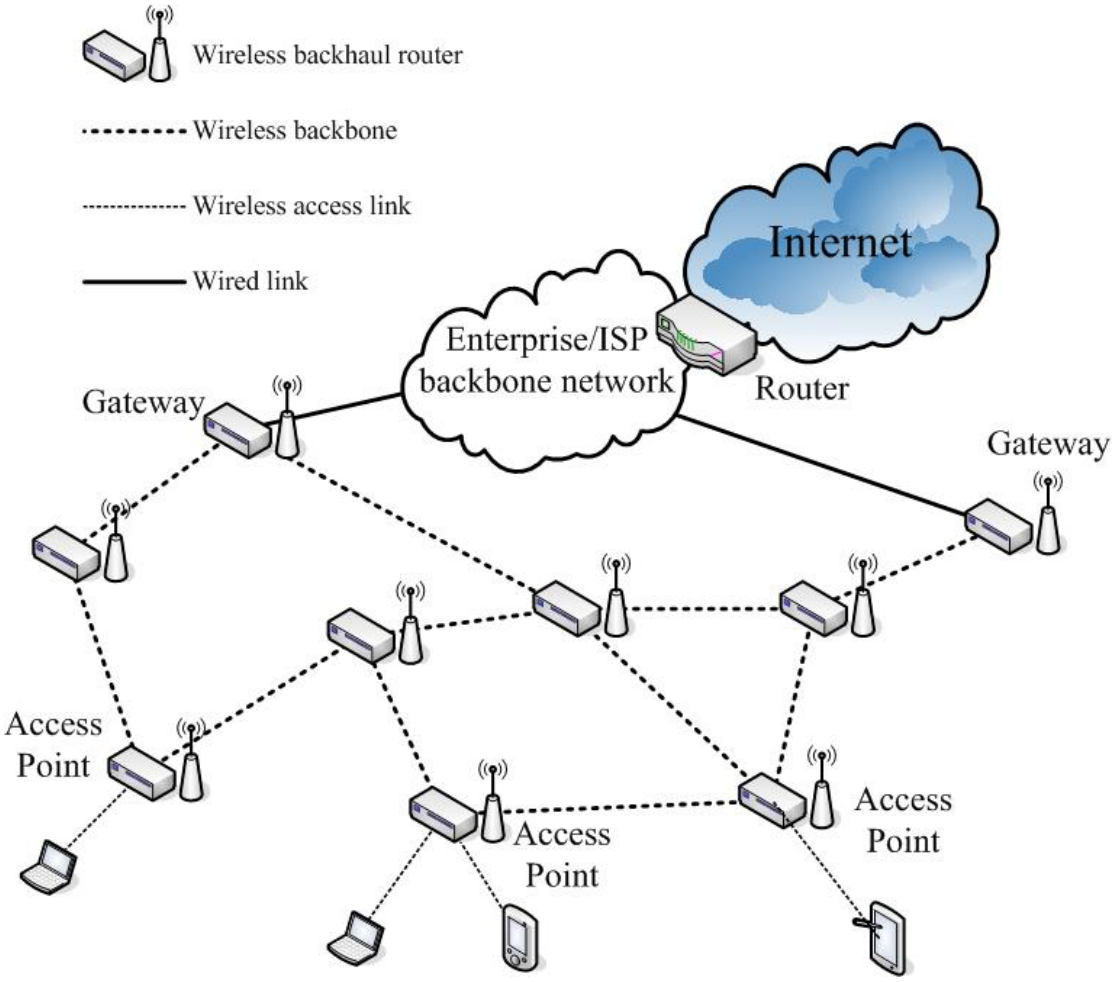


圖 2.1 Wireless Mesh Network Architecture

- 閘道伺服器（Gateway ，縮寫 GW）：一邊的 connection 連到 Wireless Mesh Network，另一邊則是連到網際網路的 router。
- Backhaul router (BR)：提供 mobile node 網際網路連線服務的 router，會連到其他 Backhaul router 或 Gateway，沒辦法直接連上網際網路。Backhaul router 可能有 Access point 提供使用者連線服務，也可能沒有。
- Initial backhaul router：mobile node attach 上的 Backhaul router。

2.2 QoS Reservation 機制（RSVP）

Resource reservation protocol (RSVP)是一個 resource reservation setup protocol。當一個 host 需要特定的服務品質來在網路上傳輸資料時，就會使用 RSVP。RSVP 也會被 router 使用，在路由路徑上的 router 之間會需要傳遞 QoS request，用以建立和維持 QoS 服務的狀態。RSVP 的功能在讓路由路徑上的所有 nodes 都 reserve 資源下來給 host 使用。

RSVP爲單一方向的request做reserve資源的動作。因此，雖然在同一個應用程式下，一個host可能會同時扮演傳送端和接受端的角色，但RSVP會將傳送端和接收端視爲不一樣。RSVP可以在IPv4 和IPv6 上面的Transport protocol階層運作，但是RSVP並不傳送應用層的資料而是比較像Internet control protocol，像是ICMP或routing protocols。跟一般的routing protocols一樣，RSVP是在background的部分做運作，參見圖 2.2。RSVP本身並不是一個routing protocol，而是設計成會和routing protocol合作運作。Routing protocol負責決定封包要傳遞到那裡，而RSVP只關心那些會被路由的封包的QoS需求。

見圖 2.2，QoS是透過集合而成的一些機制來決定什麼時候要傳送特定的封包，我們稱這些集合爲traffic control。它所包含的機制有：

（1）packet classifier：決定每個封包的 QoS class。

（2）admission control：判斷 node 是否有足夠的資源滿足需求的 QoS。

（3）packet scheduler：確保資料傳輸時達到所承諾的 QoS。

（4）其他 link-layer-dependent 機制：確保資料傳輸時達到所承諾的 QoS。

在做 reservation setup 時，RSVP 的 QoS request 會用到兩個 decision 模組，除了上述介紹到的 admission control 模組外，還會用到 policy control 模組。Policy control 模組是用來檢查 user 是否有足夠的權限來做 reservation。如果兩個模組的檢查都通過了，則會設定好能滿足 Qos 需求的參數，參數是設定在 packet classifier 和 link layer interface（像是 packet scheduler 中的）裡面的。如果任一模組的檢查沒通過，則 RSVP 會回復一個錯誤的通知給起始這個 request 的應用程式。

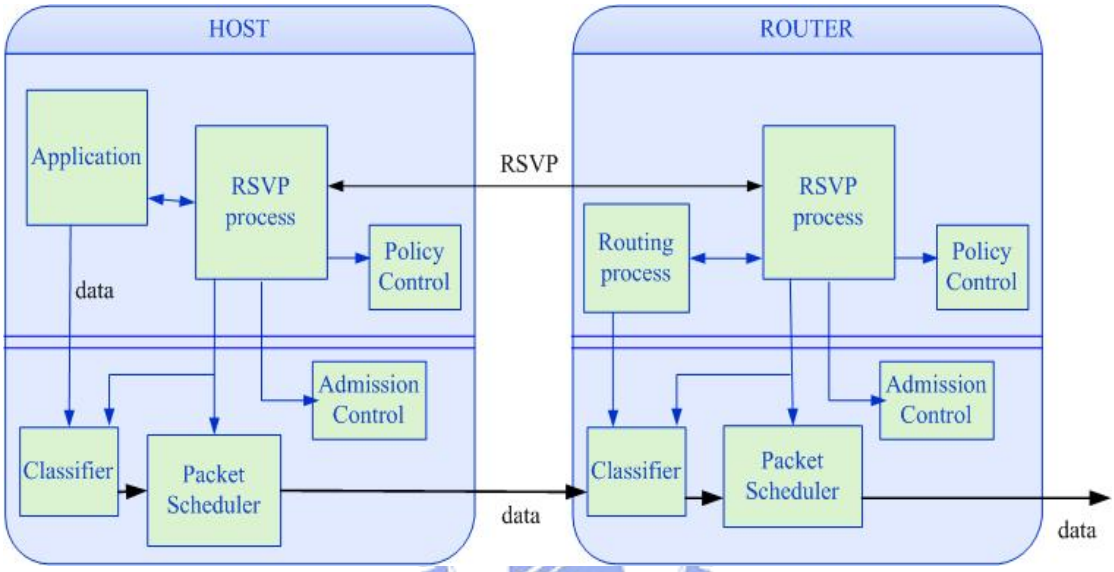


圖 2.2 RSVP 在 Hosts 和 Routers 的運作關係圖

RSVP 的 message type 有兩種，分別是 Resv 和 Path。RSVP 運作流程如下所述：最初傳送端產生一個 RSVP Path message 一路往接收端傳遞過去，Path message 會在沿途經過的 nodes 上紀錄一個關於此一連線的”path state”。Path state 會包含 previous hop 的 ip address，這個資訊會在未來當收到反方向的 Resv 的 message 時，用來做 hop-by-hop 的路由。當 path message 一路經過 internal nodes 傳送到接收端後，接收端會回送一個 Resv message，這個 message 必須沿著剛剛 path message 走過來的路徑走回去。Resv message 會在沿途經過的 internal nodes 上面產生並維護此一 path 的 reservation 狀態，此時才真正有做好 path reservation。最終 Resv message 會送達到傳送端，就做好了整條路徑的 reservation。

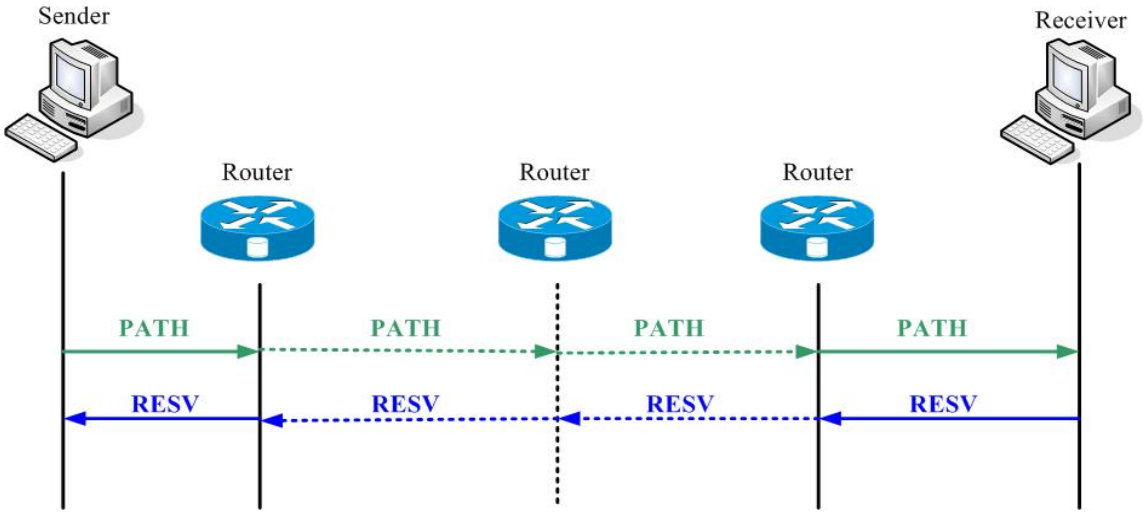


圖 2.3 RSVP message flow 圖

2.3 路徑選擇演算法

2.3.1 路徑選擇時的兩種不同的 approaches

以下將介紹在做 QoS 的路徑選擇時，會考量到的兩個不同的方向[8]：

- Resource Conserving Approach（RC）[9]

  Routing protocol 通常都是在做路徑選擇時，透過減少 hop 數目來來達到節省資源的目的。使用 RC 方法有下列幾個優點：

  - 節省資源：因爲使用最短的路徑，所以會節省資源。較長的路徑使用較多的 link，所以消耗多餘的資源，可能會阻礙到未來的其他使用者的頻寬需求。

  - Robust：因爲在做路徑選擇的主要的 link cost metric（hop count）是固定的，所以當網路的狀態資訊不正確的時候，不會受到很大的影響。

  - Traffic 變動機率少：traffic 變動會出現的原因是因爲當某一個 link 廣播說自己有一個低的使用率，則可能會導致許多新進入的連線

都偏好使用那個 link，因此在下一個更新期間內，會致使這個 link 有很高的負載。再一次更新時，這個 link 就會廣播說自己有很高的使用率，則少數的連線從這個 link 經過，整個循環就會再一次的重複。因爲 RC 方法主要是考慮 hop 數目，所以比較不會有這樣的狀況出現。

除了上述敘述的特性外，RC 方法有一個主要的缺點，那就是當我們限制住了 hop 數目，我們只會使用那一些屬於最短路徑的 link，所以我們會沒法利用那一些有比較常的 hop 數目但是負載量很低的 link。

- Load Distributing Approach（LD）[10][11]

  通常 load balancing 是透過將 link 的 cost metric 表達成一個函式，當 link 的使用率增高的時候，函式得出的數值也會跟著增高。使用 LD 方法有下列幾個優點：

  - 避免擁塞的 link。
  - 將流量分配到整個網路上，所以可以充分地利用比較長的路徑。

  他也以下幾個缺點：

  - 由於他會使用較長的路徑，所以消耗多餘的資源，可能會影響到未來的需求。但在網路負載比較輕的時候影響不大。
  - 因爲他所使用的主要 cost metric 是根據 available 資源的資訊來做出判斷的，所以需要常常做狀態的更新，不然就會判斷錯誤。

2.3.2 路徑選擇算法

以下將介紹最常見的兩個 QoS routing algorithms[12]：

- Widest-shortest path（WSP）[13][14]

歸屬於 Resource Conserving 的方法。從所有可以選擇的路徑中，選擇一個有最小 hop count 的路徑。如果有多個 hop count 都是一樣最小路徑，則選擇擁有最大 reservable 頻寬的路徑，也就是看路徑上的所有 link 中擁有最大的 available 頻寬是多少。如果有多個路徑符合這個條件，則隨便選擇一條路徑。

- Shortest-widest path（SWP）[15]

  歸屬於 Load Distributing 的方法。從所有可以選擇的路徑中，選擇一個有最大頻寬的路徑。如果有多個路徑的最大頻寬都相同，則選擇有最小 hop count 的那一個路徑。如果這樣選擇後，又有多個有同樣 hop count 的路徑，則隨機選擇一條路徑。

2.4 閘道伺服器選取法

這一小節將介紹我們 survey 到的兩個關於閘道伺服器的選取法的演算法[16]。

- Shortest-Path (SP) algorithm

  顧名思義就是選取距離最近的閘道伺服器。有圖例可以看出，此拓樸可以分成三個區塊，分別代表每一個閘道伺服器所負責的區域。至於正中間下面的那一個介於兩個區塊中間的 node，是因爲他到兩個閘道伺服器的距離都相同，所以他可以任選一個閘道伺服器作爲他連上網際網路的媒介。

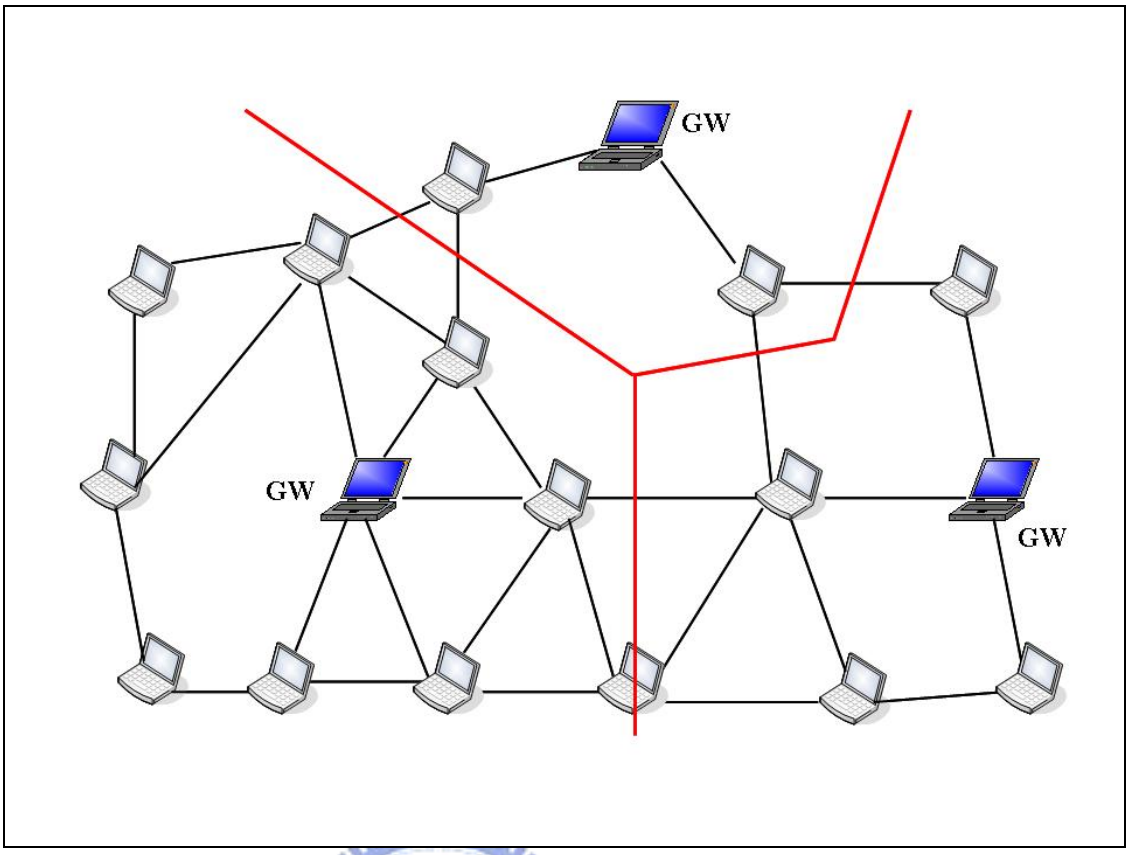


圖 2. 4 Shortest-Path algorithm 圖例

- Minimum Load-Index (MLI) algorithm

在這個方法中，每個閘道伺服器會定時的廣播閘道伺服器的負載量(load index)。每個 host 會保留目前正在爲他服務他的閘道伺服器。當 host x 聽到閘道伺服器 g 的廣播(L_g)時，會做以下的判斷看是否需要更換閘道伺服器：

1. 當 x 尚未選定服務的閘道伺服器，則選取閘道伺服器 g 來爲他服務，並記錄他的負載量，再將訊息轉送出去。
2. 若閘道伺服器 g 是目前的服務閘道伺服器，則 host x 紀錄的負載量，並將訊息轉送出去。
3. 若 host 目前選定服務的閘道伺服器是 g'，則 host 會檢查選擇閘道伺服器 g'是否有超過一個時間範圍 τ 和 $L_g + \frac{T_x}{C_g} + \Delta_l < L_{g'} - \frac{T_x}{C_{g'}}$，其中 Δ_l 爲轉換閘道伺服器的流量臨界限制，T_x 爲 host x 的流量，C_g 爲閘道伺服器 g 的容量。如果通過檢查，則 x 會根據更換閘道伺服器的機率 P_{MLI} 來決定是否要更改其服務伺服器爲 g'。此時並不會再將廣播訊息傳送出去。

這個選取閘道伺服器的演算法，雖然考量的很周到，但他並不適用於 QoS routing。首先在 QoS 中，MN 的流量並不適合做 redirection 的動作，因爲在之

前建立好 QoS routing path 的時候沿路的頻寬皆已被保留，要重新做一次 path reservation 並不符合效益，也不保證一定會成功。此外，若將這個演算法討用到無線網狀網路的環境中，則在每個廣播的區間內，backhaul router 選取哪一個閘道伺服器當作 MN 的服務閘道伺服器是固定的，若有一個區間內，有很多個 MN 都連到某一個 backhaul router，則會導致 backhaul router 選定的閘道伺服器流量暴增，導致那個閘道伺服器的 reject rate 變的很高。

第三章Path Reservation in Wireless Mesh Network

3.1 系統架構與前提假設

3.1.1 整體設計方向

要在Wireless Mesh Network做到QoS可以分成兩種不同的研究方向，一是Resource management across the network，另一個則是Resource management at a single node and link level。

- Resource management across the network－路由（routing）和頻道配置（channel assignment）

  考慮到整個網路的情況下，可以分成路由問題和頻道配置問題這兩種不同的問題。第一個問題主要是考量對於一個已知的網路連線拓樸，當給予一個user link capacity需求時，該如何決定一個有效率的路由路徑。頻道配置則是考量到當已知網路連線拓樸和已知的一些流量，該如何將頻道資源（包含時間和頻率的使用）配置的更有效率。而拓樸中由任一個node做的選擇都會影響到網路上的其他nodes。

- Resource management at a single node and link level

  當路由已經完成而end-to-end路徑也已經設定好了，對於進出一個node的所有flow，node必須要滿足flow在link上所要求的QoS需求。

在本論文中，Resource management across the network部分，我們著重在路由方面的設計，而不去管頻道配置的部分。而Resource management at a single node and link level部分，我們將使用已經定義好的Resource Reservation Protocol（RSVP）。

在前面2.3節當中，我們介紹了一下先前再探討QoS路由時，所採用的機制和常見的路由方法。但之前的QoS路由，都是在探討針對一對已知的發送和接收端，要如何選

擇一個適當地路由路徑。在他們的演算法設計下，所有的發送端都必須要知道所有路徑上的 link 狀態式如何。因此，所有的 node 都必須要廣播出他們的 link 狀態，一旦有所變動，就必須要立刻廣播給大家知道。這樣的流量對於無線網路的環境中，是很重的負擔，所以我們希望設計出的路由機制可以避免這樣繁重的流量資訊廣播出去。

我們設計的方法是採用透過選取閘道伺服器的方法，來做到 load balancing 的效果，而不是透過如何選取 initial backhaul router 與閘道伺服器之間的路徑。我們透過將 mobile node 的連線導向不同的閘道伺服器，來達到分散流量的目的。在減少路由資訊的傳遞方面，我們是透過 hop-by-hop 的選取路徑會走的 link，會針對選定的閘道伺服器，選擇最短的可走路徑，這樣 backhaul router 就不需要廣播他的 link 狀態。

3.1.2 系統架構

在此所針對的環境是設定在一個無線網狀網路架構下。當使用者（mobile node）想要從使用網際網路資源時，他們會透過無線網路介面連上 backhaul router，再從連上的 backhaul router 路由到某一個與網際網路相連的閘道伺服器（gateway），從而能夠到網際網路上抓取所需要的資訊。

以下是一個範例（見圖 3.1）。一個 mobile node 想要使用 internet 資源時的流程大致如下：

1. Mobile node 首先會先連上離他最近的 backhaul router1。
2. Backhaul router1 再將 mobile node 的 request，從 backhaul router1 路由到某個適當的閘道伺服器，在此例子中是 gateway1。
3. 當 request 的訊息到達 gateway1 後，gateway1 會再將 request 路由到網際網路上面去。

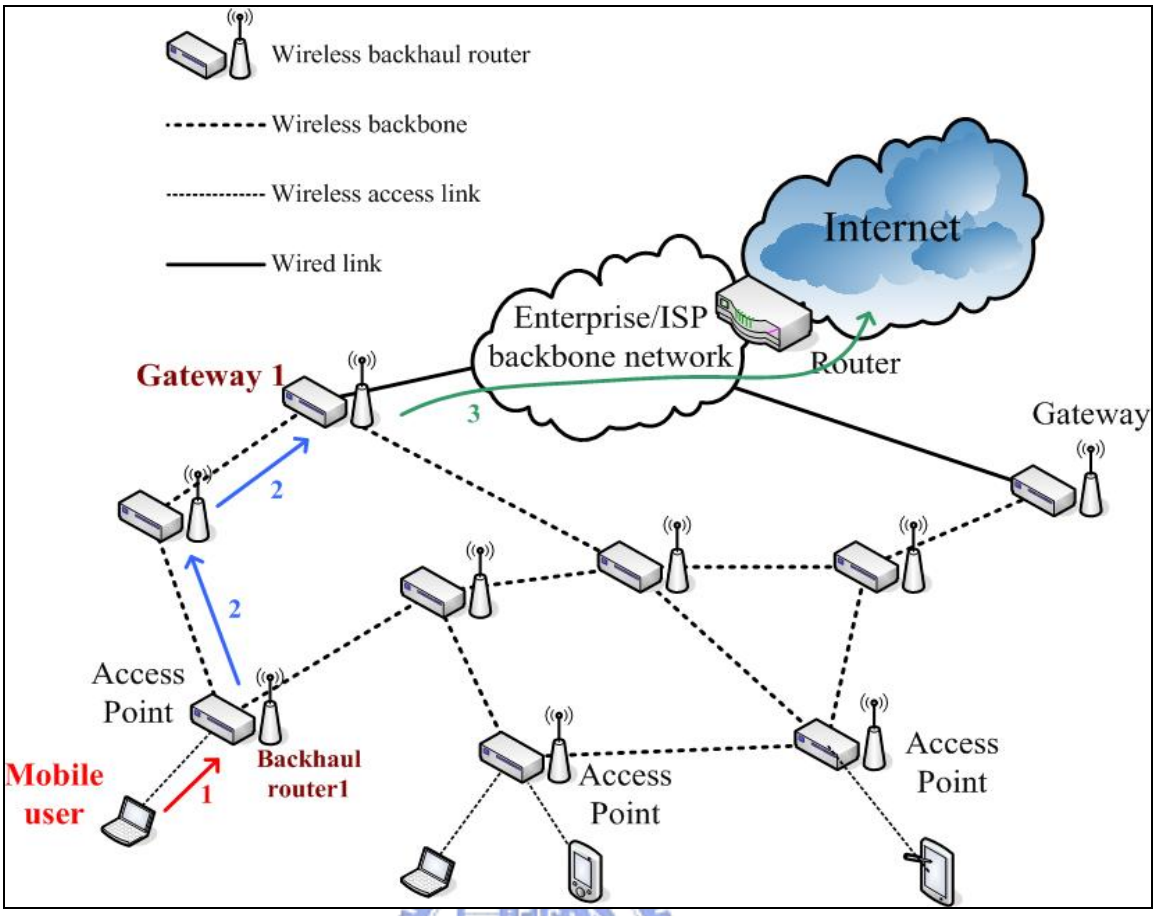


圖 3.1 mobile node 連上 internet 範例

3.1.3 前提假設

針對上述的系統架構，我們設定了兩個前提：

第一個前提就是 backhaul router 是提供 mobile node 連上網際網路的設備。故 mobile node 連上 backhaul router 的最終目的是與網際網路（internet）建立連線，所以不考慮 mobile node 會需要與在同一個 wireless mesh network 裡面的 mobile node 相連的情況;**即不會有流量不經過閘道伺服器(gateway)的情況**。若是在同一個 wireless mesh network 的 2 個 mobile node 需要互聯，則可以直接套用現有的 ad-hoc routing protocol 去做路由。

另一個假設就是 **backhaul router 的 data-link layer(layer 2)知道與其 neighbor backhaul router 的 bandwidth 使用情況**。在本論文中，我們不關心 backhaul router 如何與其 neighbor backhaul router 建立 link、不考慮他是如何挑選與 neighbor 溝通時所使用的 communication channel，我們只要 layer 2 能告知目前所存在的 link 以及其 bandwidth 的使用狀況，其中最主要的是需要知道還剩下多少可以被 reserve 的

bandwidth。

3.2 Mesh routing protocol

由於 wireless mesh network 的特性是，想要連上網際網路就一定需要通過閘道伺服器，只有閘道伺服器有連上網際網路的連線，所以可以說一個 wireless mesh network 的區域範圍就是以閘道伺服器圍起來所能服務的範圍。由於本論文的討論的是針對前面所述的 wireless mesh network 環境，所以建立 path 的部分只討論從 mobile node 連上的 backhaul router 到閘道伺服器這一段 path 的 reservation。至於從閘道伺服器到 internet 的這一部分因爲是在有線網路的環境下，不屬於 wireless mesh network 所以我們不在此討論。

3.2.1 Mesh routing protocol 設計概念

Routing path 可以視爲是以閘道伺服器爲 root，向 wireless mesh network 裡建構出 tree 出來，而封包路由的方向是向 tree 的 root 方向走的，不過不一定是走最短路徑。當有 mobile node 想要上網際網路，而最短路徑的頻寬不夠 mobile node 使用時，則 mobile 去找稍微會繞一點路且有足夠頻寬的 path。

一般的路徑選擇可以分成是 centralized 和 distributed path selection。採用 distributed 的路徑選擇，會將複雜度分散到整個網路中，會增加網路的複雜度。但分散路徑選擇，可以避免掉需要做 network-wide 的 synchronization，可以增加網路的 scalability。同時 distributed 的路徑選擇對於環境的改變有較快的反應，而 wireless mesh network 的連線拓樸並不是固定的，所以我們將採用 distributed path selection，也就是說**路由的決定是 hop-by-hop** 做的，而不是藉由事先知道整個網路的狀況再運算出來的，因此比較不會因爲 link 狀態資訊不正確而受到的影響。但這樣有一個缺點，也就是做出來的決定不一定是 optimal。

本論文所提的 routing 方法大致運作情況如下所述：

1. 當一個 mobile node 連上 backhaul router 後，他會傳送 path request 的 message 給 initial backhaul router，封包其中會敘述他所需要的 bandwidth。

2. Initial backhaul router 會根據mobile node 要求的bandwidth 去建立 routing path。首先會挑選出要經過的閘道伺服器，會依據閘道伺服器的負載情形來做出選擇。
3. 從 initial backhaul router 依據選好的閘道伺服器，去 routing table 中選擇一個有足夠 bandwidth 的並擁有最短 distance 的 next hop，將 path request 往那個 next hop 傳遞。如此重複，直到 path request 的 message 抵達閘道伺服器。
4. 當抵達閘道伺服器確定了 path 後，message 會走原本 path request 走過的路一路往回走並順便把路徑上的 link 的頻寬 reserve 起來(採用 RSVP protocol)。
5. 最終抵達 initial backhaul router 即表示 path reservation 已經完成，mobile node 可以開始使用網路的資源了。

其中，本方法的重點所在就是在第二步驟中，initial access router 要**選擇 path request 要到達的閘道伺服器**。之所以會要先做選擇的原因是，因爲在 wireless mesh network 中，想要連上網際網路一定要經過閘道伺服器，即使 backhaul router 與 backhaul router 中間的 link 的 bandwidth 足夠 mobile node 使用，但是只要 backhaul router 到閘道伺服器那一個 link 的 bandwidth 不夠，那一切都不用談。所以我們希望可以先選擇一個可能可以成功建立連線的閘道伺服器。

另一個原因是，如下圖 3.2 所示範例，當 mobile node 連上 BR1 時，一般的 routing 方法就是會預設往最近的閘道伺服器送，則 packet 會一路經過 BR4 和 BR6。但當 path request packet 到達 BR6 時發現 BR6 和 GW1 的 link 其 bandwidth 已經滿了。則回將 packet 往回傳給上一個 BR4 但不幸的 BR4 也沒有其他的 link 可以選擇，就會再一次退回 BR1，BR1 只剩下往 BR2 送的 link，則接下來對於 BR2 而言，次近的 GW2 就會被當成傳送的目標一路順利的送過去。當到達 Gw2 時，已經花費了 8 個步驟。但若是我們一開始 BR1 就知道 Gw1 的 loading 已經滿了，則 BR1 就會選擇要往 GW2 送 path request packet，則此時所花的步驟數目可以簡化爲 4 步即可。所以說若是可以知道或是推論出閘道伺服器的 bandwidth 不夠的話，在路由 path request 的 packet 時，就可以少走一些冤枉的路。

需要路由到其他閘道伺服器，除了因爲最近的閘道伺服器 bandwidth 滿了的原因之外，也有因爲想要讓每個閘道伺服器的 loading 平均分配，而將流量導到其他的閘道伺

服器的考量存在。不要讓某一特定的閘道伺服器 loading 過重，則可以避免掉可能會發生的擁塞情況。所以雖然最近的閘道伺服器可以成功建立連線，但是還是可能會希望將連線建立到一個稍遠一點的閘道伺服器。

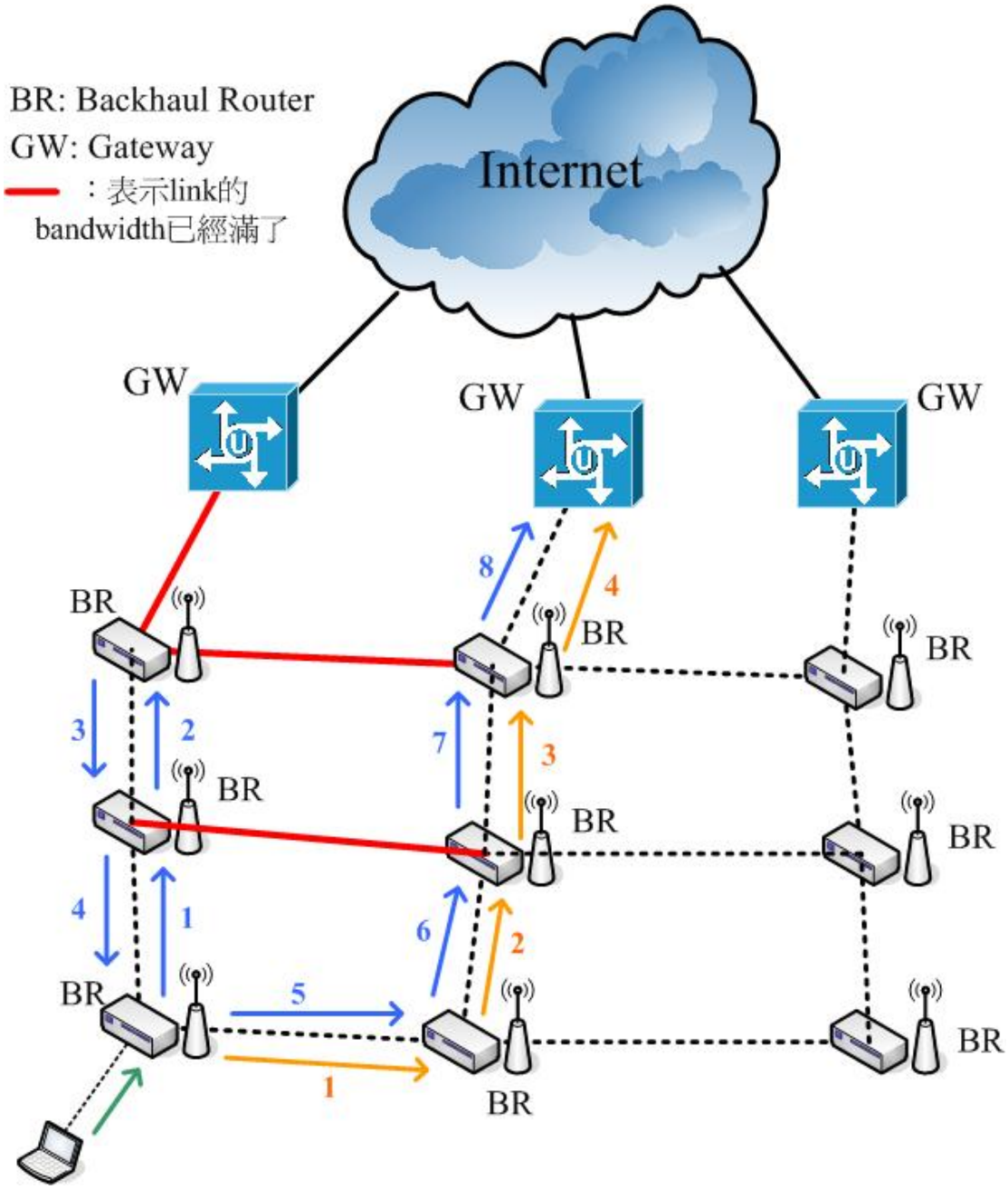


圖 3.2 routing protocol 選取閘道器的原因

3.2.2 Mesh routing table

由於 backhaul router 在向閘道伺服器建立 routing path 時，可能會繞路不一定要走最短的路徑，或是最短的路徑的剩餘頻寬可能會不夠使用，即 routing 的路徑不是

唯一的，而路由選擇又是 hop-by-hop 做的，所以在建立 routing table 時，就 backhaul router 要有比較詳細的 routing 資料，這樣才能夠做選擇。

Network prefix	Hop count	Next hop	Gateway router

表格 3.1 mesh routing table

- Network prefix

  目的地的 network prefix，在此會是指 internet 的所有 network prefix。

- Hop count

  以閘道伺服器爲 root，其 hop count 爲 0，向 mesh network 延伸出去。表示從目前的 backhaul router 到達直到閘道伺服器所需要經過的 hop 數目。

- Next hop

  要傳遞 Data packet 時的下一個 backhaul router。

- Gateway

  Backhaul router 記錄當想要到達指定閘道伺服器時，透過每個 neighbor（next hop）所能需的最小 hop count。選 routing path 時，會要先考慮最後會往哪一個閘道伺服器走。

Backhaul router的mesh routing table需要針對每一個可以到達的閘道伺服器，都需要紀錄透過所有的neighbor backhaul router到達閘道伺服器所需的最小hop count。就算透過其他neighbor backhaul router有比較小的hop count也還是一樣需要記錄。下圖 3.3是一個範例，圖中可以見到backhaul router3 和backhaul router1 的mesh routing table是長的如何。BR3 所能到達的閘道伺服器有兩個（Gw1 和Gw2），而BR3 有三個neighbor backhaul router（BR1、BR4 和BR5），由於所有backhaul router都有路徑可以到達Gw1 和Gw2，所以可以看到針對Gw1，BR3 有三筆紀錄，分別是透過BR1、BR4 和BR5 到達Gw1 的最小hop count。針對Gw2 也是相同的情況。而BR1 由於到達Gw1 一定要經過他自己，所以雖然有兩個neighbor backhaul router但到達Gw1 最短的路徑只有一筆，即是由BR1 直接到達Gw1。但通過Gw2 就有兩筆記錄，分別可以透過BR3 和BR6 到達

Gw2。

BR5

Lv	NH	Gw
1	Gw1	1
2	BR6	2
4	BR3	2

BR3

Lv	NH	Gw
2	BR5	1
4	BR4	1
6	BR1	1
3	BR5	2
3	BR4	2
5	BR1	2

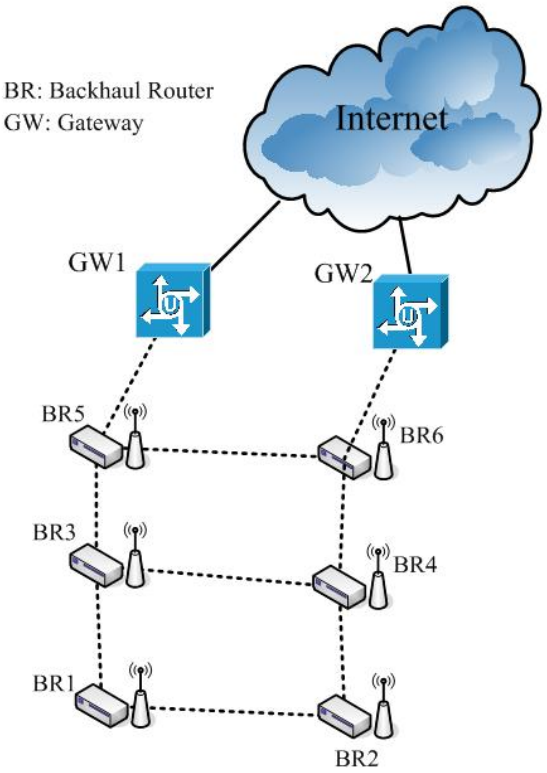


圖 3.3 routing table 實例圖

3.2.3 Mesh routing protocol 運作流程

3.2.3.1 Mesh routing table 建立

Mesh routing table 是藉由閘道伺服器會 periodically 發送 routing information message 來建立的，message 中會記錄下列五項資訊：

- sequence number: 用來判斷是否是新的 routing information message。
- expire time: 記錄多少時間後這一筆記錄就會失效。當閘道伺服器無預警斷線時，可以判斷出需要刪掉關於此閘道伺服器的資訊。設成比發送 routing

information message 的 period 多一些的時間。

- Distance: 代表到達閘道伺服器所花費的 hop count。
- gateway: 表示此 routing information message 是由那個閘道伺服器發出的。
- routing path record: routing path 紀錄 routing information packet 所經過的 backhaul router。Path record number 爲 routing path 中有多少個 backhaul router。最後走過的 backhaul router 是放在 path 的最後面。

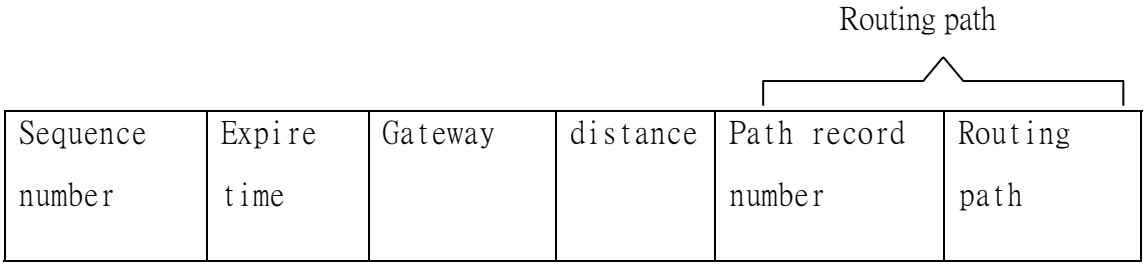


圖 3.4 routing information message

Routing information message 一開始是由閘道伺服器開始發送的，在 backhaul router 之間傳送。Routing information message 中會被 backhaul router 更動的記錄有 distance（即到達閘道伺服器所花費的 hop count）和 routing path record（閘道伺服器會將他自己的 address 放進 routing path record 中）。Routing path record 會記錄更新 message 所經過的 backhaul router 的 address，此 address 會用來判別是否需要再將 message 傳給 neighbor backhaul router。在此所指的 address 可以是 layer 2 的 MAC address，也可以是 layer 3 的 ip address。

如圖 3.4 所示，當 backhaul router 收到 routing information message 時，首先會將 message 的 distance 加上 1，並將 routing path record 的最後面加上目前所在 backhaul router 的 address。接著，就要檢查是否需要更新 routing table。Backhaul router 會先檢查 message 中的閘道伺服器，會出現以下兩種狀況：

1. 如果沒有與這個閘道伺服器相關的資料，則可以肯定這是一筆需要記錄的新資訊。
2. 如果有與此閘道伺服器相關的資訊，則就再看 message 的 routing path record 的倒數第二個 address（即指 previous backhaul router）是否有在 table 中有

資訊。

（1） 如果沒有，就表示之前沒有紀錄過經過 previous backhaul router 到閘道伺服器的 hop count 資料。

（2） 如果有，先比較 sequence number 是否有比較新，若是 message 的 sequence number 比較舊，則將 message 直接 drop 掉。若是 sequence number 比較大或一樣，再繼續下面的比較。接著比較記錄中的 hop count 是否比 message 中的大。若是比較大，則將記錄刪掉，重新記成目前 message 中的資料。若是比較小的不需更動。

Routing table 記錄的資訊有 5 個，包含閘道伺服器、sequence number、expire time、hop count、和 next hop ，其中 next hop 設定成 previous backhaul router（即 routing path record 倒數第二個 address）。

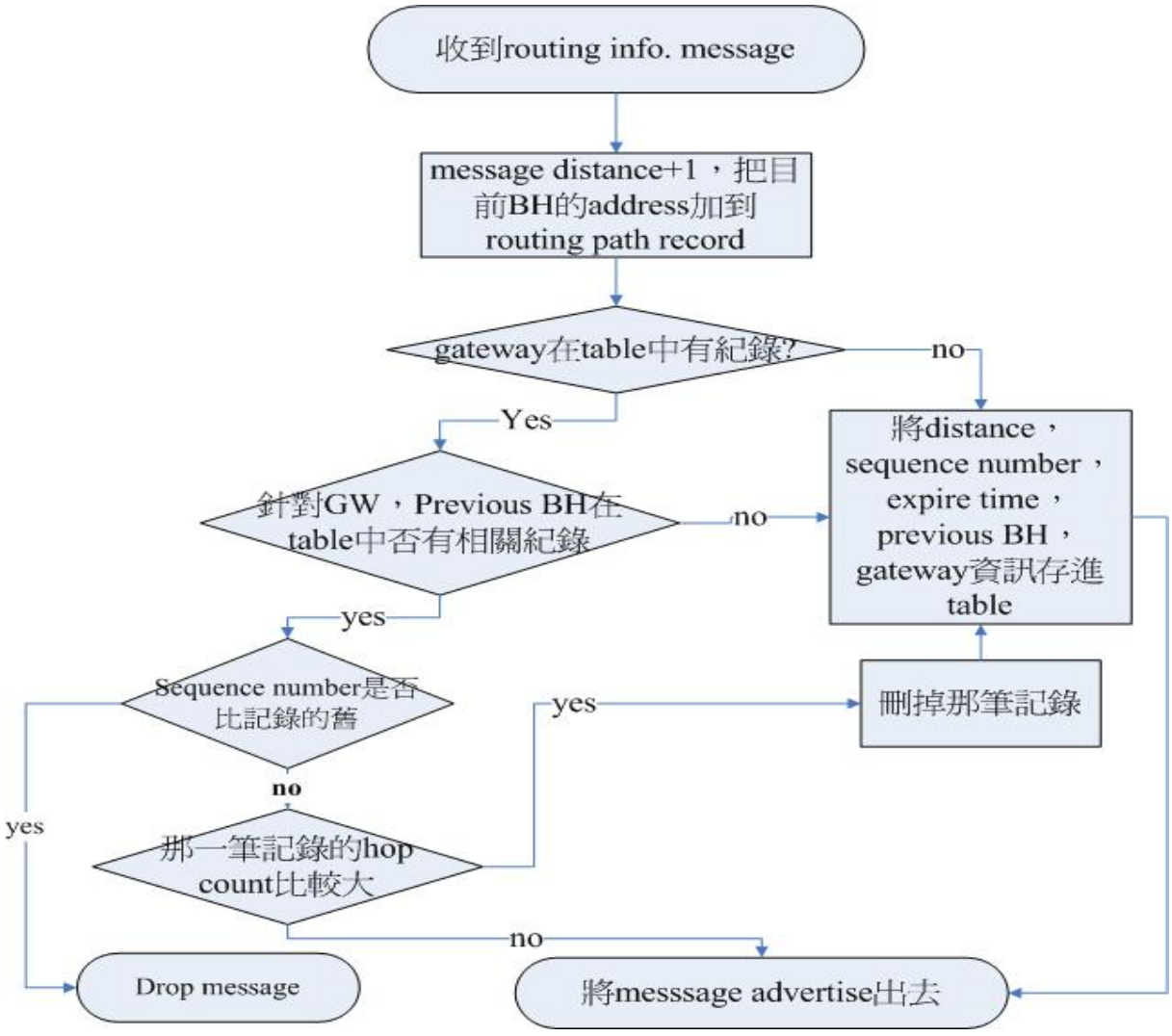


圖 3.5 mesh routing table 更新流程圖

當 backhaul router 要 advertise 收到的 routing information message 時，他首先會核對其 neighbor 是否有出現在 packet 的 routing path record 中，如果沒有就會將 packet 傳送到 neighbor，若是有則不會傳送。

3.2.3.2 Routing table update

在傳送 routing information message 的 period 內，若網路的 topology 有更改，則 routing table 有需要做更新。爲了能即時偵測到有 link 中斷，所以 backhaul router 會不定時偵測 link 是否還有連線。需要更新的 case 可以分成以下四種情況：

- 有新的 GW-BR(gateway-backhaul router) link 連上 mesh network
- 有新的 BR-BR(backhaul-backhaul router) link 連上 mesh network
- 有 BR-BR link 中斷
- 有 GW-BR link 中斷

以下將詳細敘述當發生這些情況時，Backhaul router 會如何處理。

- Case 1. 有新的 GW-BR link 連上 mesh network

  GW-BR link 的閘道伺服器會直接透過那一個新的 link 發送 routing information message，這樣就可以更新所有 backhaul router 的路由資訊。

- Case 2. 有新的 BR-BR link 連上 mesh network

  1. link 上的 Backhaul router 可向周圍的 Backhaul router 發送 request 要求他們的 routing 資訊。
  2. 依據接受到的 routing table 資訊設定好自己的 routing table。接著，向鄰近的 backhaul router 發送自己的 table 資訊，看鄰近的 backhaul router 是否因自己的加入而需要更新他們的 table。

3. 若鄰近的 backhaul router 有更新到他的 routing table，則 backhaul router 會再像他周遭的 router 發送他的 routing table，如此類推，直到沒有做任何更新爲止。

■ Case 3. 有 BR-BR link 中斷（假設是 BR1-BR2 的 link）

Step 1. 需要將 BR1 table entry 中 next hop 是 BR2 的資料砍掉，BR2 也是做相同的動作。

1. 若是正常的中斷 link，假設是 BR1 要做中斷，則 BR1 可以告知 BR2 要中斷的消息，讓 BR2 可以刪除他 table entry 中的關於 BR1 得資料。
2. 若是無預警中斷，則 BR1 是知道自己已經與 BR2 失去連線了，但 BR2 就要靠自己不定時偵測到與 BR2 的連線中斷。在去刪除他 table entry 中的關於 BR1 的資料。

Step2. BR1 和 BR2 需要通知他們現有的 neighbor backhaul router 檢查是否有因爲 BR1-BR2 link 的斷線而需要更新他們的 routing table。

假設目前正在做檢查的 backhaul router 是 BR1，而他刪掉的 entry 是（hop count:n，nexthop:BR2，gateway：Gw(i) ）

1. 檢查除了刪掉的entry外，是否還有其他通到GW(i)的entry，而且hop count小於等於n。（流程參考圖 3.6）
   - 1-1：若有兩筆比以上的 entry，則什麼事都不做。
   - 1-2：若只有一筆 entry，假設此筆 entry 是（hop count:m，nexthop:BR(k)， gateway：Gw(i))，m<n。

     （代表**只有 BR(k)才會需要更新他的 table**）

     則將關於 Gw(i)的 table entry 傳給 BR(k)，讓 BR(k)做 table 更新；若是沒有，則告知 BR(k)通過 BR1 無法到達 Gw(i)。只有 BR(k)需要更新的理由如下所述:

     刪除資料之前，hop count(m)是最小的 hop count，而

hop count(n)是次小的 hop count。所以 BR1 除了 BR(k)外的其他 neighbor backhaul router 的 table 都會記錄當 next hop 是 BR1 的時候，最小 hop count 爲 m+1。而刪除 entry 後，BR1 的其他 neighbor backhaul router 仍是可以走 BR1->BR(k)->…->Gw(i)的最短路徑，此時這個 path hop count 也還是最小的 hop count(m+1)。但 BR(k)透過 BR1 的最小 hop count 是不能考慮 next hop 是 BR(k)的紀錄，所以 BR(k)原本會記錄他走 BR1 的最小 hop count 會是次小的 hop count(n+1)，path 會變成 BR(k)->BR1->BR2->…->GW(i)，但此時 BR1-BR2 的 link 已中斷,所以只有 BR(k)才會需要更新他的 table。

- 1-3：若沒有任何一筆這樣的 entry

  （代表**鄰近的所有 backhaul router** 都需更新 routing table）

  表示原本從 BR1 到 Gw(i)的最小 hop count 是 n,因此 BR1 的所有 neighbor backhaul router 原本紀錄就會是透過 BR1 能到達 Gw(i)的最小 hop count 爲 n+1，而此時當 next hop 爲 BR1，已經無法透過這樣的 hop count 到達 Gw(i)。

  因此需要將關於 Gw(i)的 table entry 傳給通知鄰近的 BR，讓鄰近的 BR 做 table 更新；若是沒有 entry，則告知通知鄰近的 BR 通過 BR1 無法到達 Gw(i)。

2. 若鄰近的 backhaul router 有更新過他的 routing table，則需向自己鄰近的 backhaul router 發送自己更新後的 table 資訊，看鄰近的 Backhaul router 是否因此而需更新。
3. 重複 2.直到沒有更新爲止。

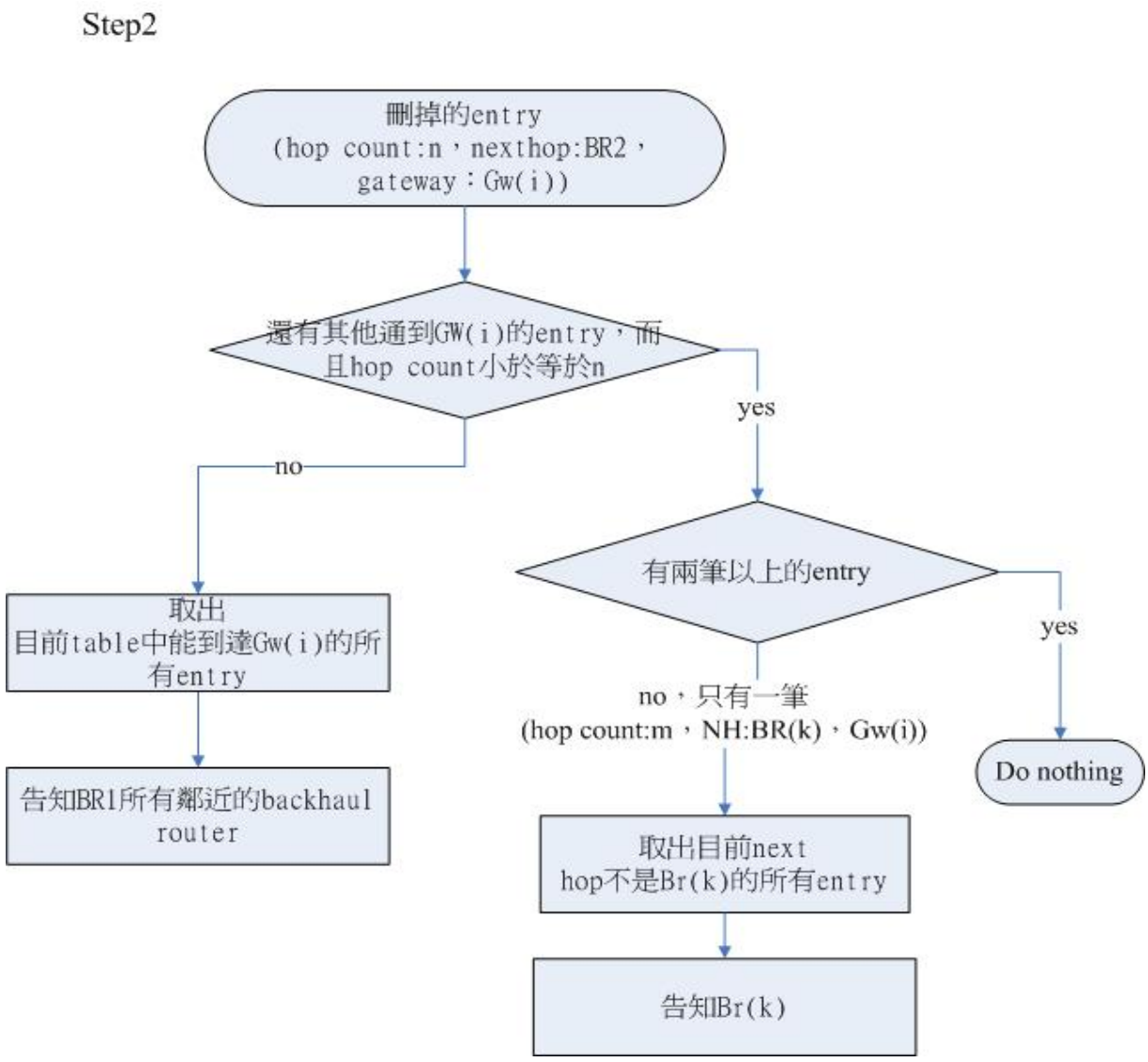


圖 3.6 BR-BR link 中斷狀況下 step2 的 Routing table 更新流程圖

- Case 4. 有 GW-BR link 中斷

  步驟同 Case 3。

3.2.3.3 不能傳送 routing table 的原因

在此提出的 routing protocol 並不像一般常用的 routing protocol 是傳送閘道伺服器或是 backhaul router 的 routing table，而是用傳送 routing information message 出去，並記錄沿途經過的 backhaul router。因爲若是使用傳送 routing table 會出現計算 hop count 錯誤問題。

參考圖 3.7和表格 3.2 所示的範例，由此例子可以看出傳送routing table爲什麼會出現hop count錯誤的問題。其中，表格 3.2 有顯示以下會討論到的routing table，但圖中顯示的只有部分的routing table並非完整的，只是爲了凸顯出問題，所以特別寫出有問題的部分。

當 BR4 傳 routing table 給 BR2 時，BR2 選擇了最短 hop count 的資訊，即 hop count=3+1 的資料儲存起來，此時 hop count＝3+1 是由 path:Gw1→BR5→BR3→BR4→BR2。接著 BR2 會將 routing table 傳給 BR1，由於 BR2 的 routing table 只有一筆資料，所以很自然的 BR1 就會記錄當 next hop＝BR2 時，最小 hop count＝4+1，此 hop count 的 path 是 Gw1→BR5→BR3→BR4→BR2→BR1。BR1 的另一個 neighbor BR3 也會傳送他的 routing table 給 BR1，此時 BR1 會選擇記錄 hop count 最短的那一筆，即儲存 hop count ＝2+1，此 hop count 的 path 爲 Gw1→BR5→BR3→BR1。之後，當 BR1 的 routing table 傳給 BR3 的時候，因爲 hop count＝3 的那一筆的 next hop 是 BR3 自己，所以不考慮。BR3 會選擇 hop count＝5+1 的資訊儲存在 routing table 中。此時就會導致 BR3 的 routing table 的錯誤，因爲 hop count=6 所構成的 path 是 Gw1→BR5→BR3→BR4→BR2→BR1→BR3，可以看出 BR3 重複出現在 path 中。正確的 hop count 應該是要設定成 hop count＝7，此時的 path 爲 Gw1→BR5→BR6→BR7→BR4→BR2→BR1→BR3 才是正確的。但這個 path 在 BR2 的時候，就由於還有比他 hop count 小的 path 所以就沒有儲存起來，因此在傳遞 routing table 的時候就不會被 BR1 得知。

會出現這樣的錯誤，主要的原因是因爲 backhaul router 所選取的最短 hop count 的 path，已經有包含這個 backhaul router 在 path 中，可是光看 routing table 是看不出那個 hop count 的 path 有經過哪些 backhaul router，所以會導致最短 hop count 計算錯誤的問題。已往的路由演算法之所以交換 routing table 不會有回圈問題，是因爲他們只記錄到達某一點的最短路徑，而在此我們是由目前所在的 backhaul router 透過所有可能的 link 所能到達的最短路徑都需要紀錄。

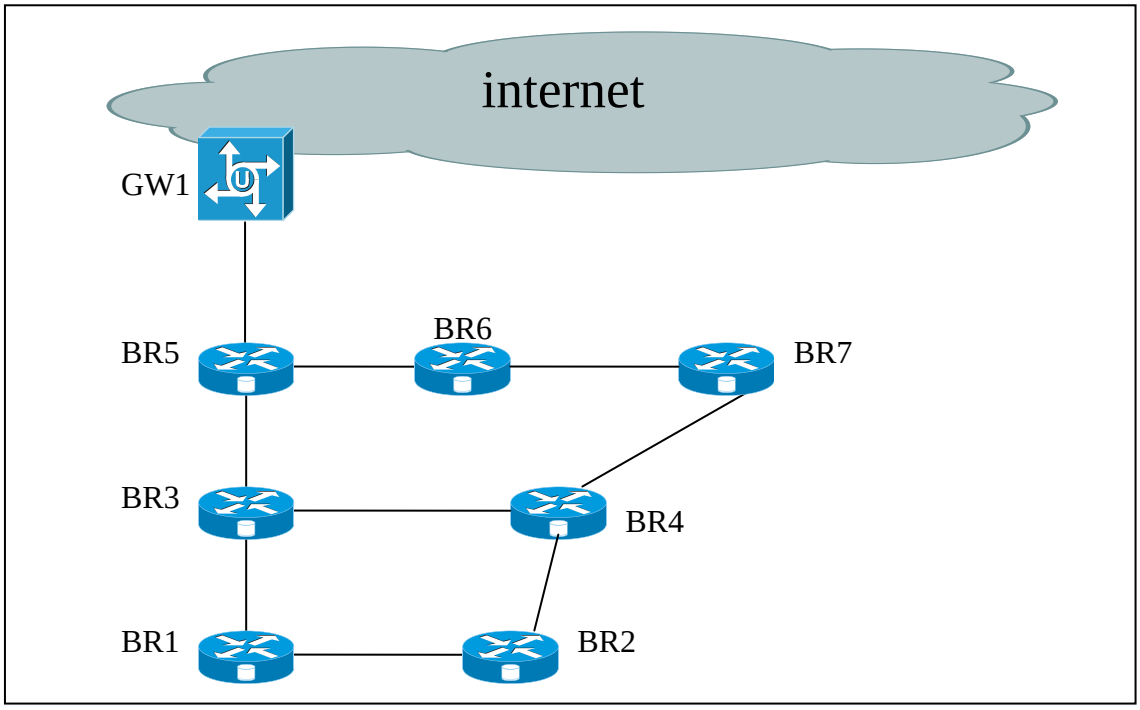
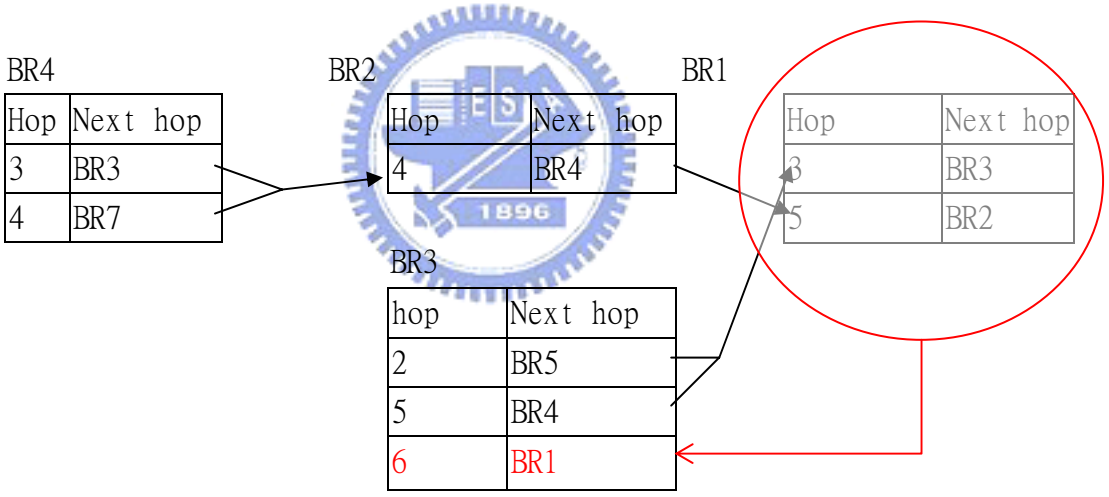


圖 3.7 routing information message 不傳 routing table 的原因



表格 3.2 圖 3.7的routing table情況

3.3 Path selection

要做 routing path reservation 主要分成下列步驟（參考圖 3.7）：

Step 1. mobile node 向 initial backhaul router 發 path request 的 message。

Step 2. initial backhaul router 要選擇 routing path 最終會經過的閘道伺服器。

Step 3. 鎖定 Step 2 所選擇出來的閘道伺服器去做 routing。

Step 4. 當 Step 3 最後到達閘道伺服器後，沿著 initial backhaul router 到

閘道伺服器的反方向，一路 reserve routing path 回去。(採用 RSVP reservation)

Step 5. initial backhaul router 將 path request 成功的 message 傳給 mobile node。

Step 6. mobile node 可以開始使用網際網路資源。

以下將詳細的敘述 step 2 和 step3 的部分。

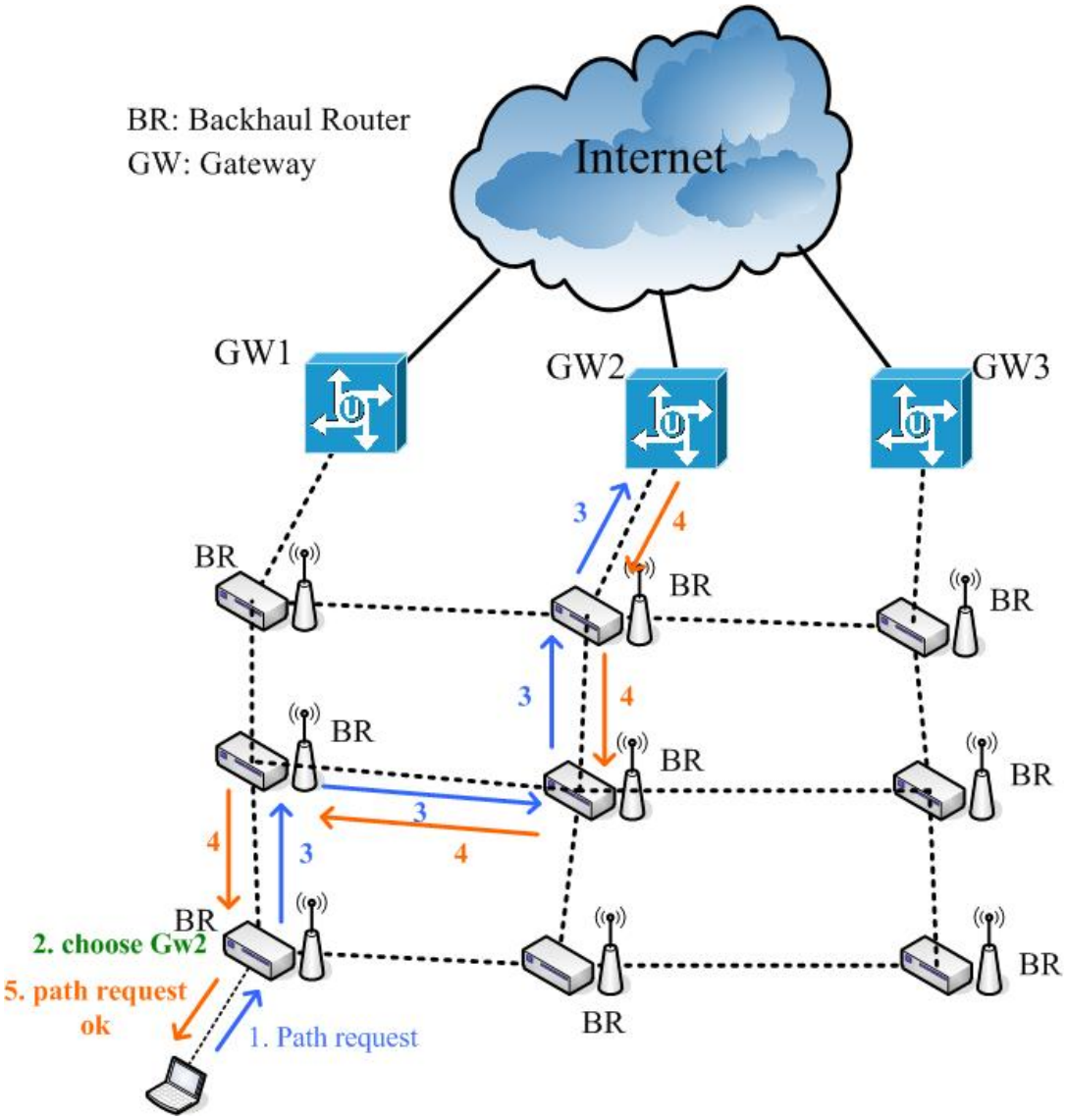


圖 3.8 routing path reservation 流程圖

3.3.1 Path request message

當 mobile node 告知 initial backhaul router 他所需要的路徑的頻寬需求後，initial backhaul router 會發送 path request message，幫 mobile node 建立一條從

initial backhaul router 到達選定閘道伺服器的 reserved path。Path request message 的資訊是放在 ip header 的 option 部分，格式如下圖所示：

type	length	Bandwidth request	Return flag	Distance threshold	Gateway	Path record	Fail backhaul router

圖 3.9 path request message 格式

- Type 和 length: 自訂的 routing protocol option type 和 option header 總長度。
- Bandwidth request：mobile node 所需要的 bandwidth 大小。
- Return flag：表示上一次選擇的 next hop 之後無法再走下去，要另外在選擇其他的 next hop。
- Distance threshold：選擇 next hop 時可以容忍的最大 hop count。
- Gateway：path request 要經過的閘道伺服器。
- Path record：記錄路徑已經走過的 backhaul router。
- Fail backhaul router：記錄走失敗的 backhaul router，即找不到可以走的 next hop 的 backhaul router。

3.3.2 Path selection

Initial backhaul router 一開始會依據各個所能到達的閘道伺服器的負載量，選定適合的閘道伺服器(詳細的選擇方式請參照 3.3.3gateway selection)。由於 routing table 在建立時，考慮到 backhaul router 要可以從多個不同的 link 到達所選的閘道伺服器，所以路由時可能會繞各式各樣不同的路徑。爲了避免路由時，會一直往 tree 的 children 繞去，離閘道伺服器越來越遠，那樣的路徑的成功率可能很低，即使成功建立連線也浪費太多的資源。所以 initial access router 會設一個 distance threshold。當 path request 到達某一個 backhaul router，而那個 backhaul router 可選擇的 next hop 其 hop count 超過 distance threshold，就算頻寬足夠，也被視爲在此 backhaul router 上所有的 next hop 都沒有符合需求，因此將會退回到 previous backhaul router 去做其他路徑選擇。Distance threshold 取法如下：

$$\text{distance threshold}(\text{gw}) = \text{maxlevel}_{ibr(gw)} + \Delta lv \qquad (1)$$

$\max\_ dist_{ibr(gw)}$: initial backhaul router 的 routing table entry 中，針對閘道伺服器 gw 的最大 hop count .

Δlv : 爲一個用來制衡 hop count 的變數，會隨著 topology 的改變而有所變動，通常 topology 越大，Δlv 也會越大；topology 小，Δlv 也會跟著變小 .

設定好閘道伺服器與 distance threshold 之後，initial backhaul router 就會開始要選擇下一個要走的 link。Backhaul router 選擇 next hop 的方式是先找出符合下列條件的 next hop，將其 hop count 從小到大排列，優先從 hop count 小的先選擇。

（1）滿足 mobile node 的 bandwidth request
（2）hop count 不超過 distance threshold
（3）不屬於 fail backhaul router：因爲之前已經嘗試過 fail backhaul router 的所有可以走的 next hop 了，所以再走一次的結果也不大可能有變動，還是會失敗。

若找到適當的 next hop，會將 next hop 放到 message 中的 path record 後面，再將 message 傳給 next hop。當沒有找到合適的 next hop 時，會先將目前所在的 backhaul router 記錄到 fail backhaul router 中，再將 path request message 傳回給 previous backhaul router。若已經是 initial backhaul router，不存在 previous backhaul router，則 initial backhaul router 將 reserve path fail 的 message 傳給 mobile node，表示不能夠爲 mobile node 提供網際網路服務。

當 path request 到達閘道伺服器時，則會一路從走過來時的路徑 reserve link bandwidth 回去 initial backhaul router。若是中途有某一個 link bandwidth 無法 reserve，則會通知 initial backhaul router reserve path 失敗，initial backhaul router 再通知 mobile node reserve path fail 的 message。

當 path 一路從閘道伺服器 reserve 回 initial backhaul router 後，initial backhaul router 就會通知 mobile node reserve path success 的 message，mobile node 就可以開始使用網際網路的服務了。

3.3.3 Gateway selection

想要做到閘道伺服器的 load balancing，就會需要將往網際網路的流量平均分配到

每個閘道伺服器上。當流量平均分配時，就可以避免網路在某一個區域特別擁塞。

在 backhaul router 端會運算出選擇每一個他所能到達的閘道伺服器的機率，依據機率丟擲骰子決定要選取哪一個閘道伺服器。選好閘道伺服器後就去做 path selection，找出到達那個閘道伺服器的路徑。若是沒有找到可以到達那個閘道伺服器的路徑，則會再重新執一次骰子在選一次閘道伺服器，此時也是有可能還是選擇到相同的閘道伺服器。重新嘗試的次數取決於 wireless mesh network 的 topology 大小和閘道伺服器的總數。嘗試次數的決定並不在本論文中討論。

在本論文中會提出兩個不同的公式，分別在閘道伺服器端和 backhaul router 端使用。Backhaul router 選取閘道伺服器的機率是根據閘道伺服器算出的權重以及 backhaul router 與閘道伺服器的 distance 大小。而閘道伺服器的權種則是依據量流以及 capacity 來運算出。

3.3.3.1 閘道伺服器端

在閘道伺服器端，會根據目前本身的流量和其他閘道伺服器的流量，以及本身和其他閘道伺服器的 capacity，用下面的公式(2)算出一個權重出來，並將權重廣播出去給所有與之有路徑可走的 backhaul router 知道。所有的閘道伺服器都要用一樣的公式去算出權重，這樣 backhaul router 收到的權重才會是公平的。在閘道伺服器端的權重公事如下所示：

$$Gweight_i(t) = Gweight_i(t-1) + [\frac{B_{gw}(i)}{\sum_j B_{gw}(j)} - \frac{RB_i(t)}{\sum_j RB_j(t)}] \quad (2)$$

$Gweight_i(t)$ ：閘道伺服器 i 在第 t 次運算出的閘道伺服器權重。

初始 $Gweight_i(0) = 0$

$B_{gw}(i)$ ：閘道伺服器 i 本身的頻寬有多少。

$RB_i(t)$ ：閘道伺服器 i 在目前共被 reserve 多少頻寬。

本公式的目的是希望閘道伺服器能夠根據他本身所具備的頻寬，適當的分配到網路

流量。所以對於閘道伺服器 i 來說最理想的數值是 $Gweight_i(t) = B_{gw}(i) / \sum_j B_{gw}(j)$，也就是閘道伺服器 i 本身的頻寬佔所有閘道伺服器的總頻寬的比例。但有可能在權重更新的 period 內，流量分配的並不是如我們所預期的理想值，表示權重並不是設定得很好需要做修改。目前流量分配狀況的表示式是 $RB_i(t) / \sum_j RB_i(t)$，所以理想值與真實流量分配的差異值就會是 $B_{gw}(i)/\sum_j B_{gw}(j) - RB_i(t)/\sum_j RB_i(t)$。我們把差異值加到上一次運算出來的權重 $Gweight_i(t-1)$ 去做修改，就可以得出這個 period 內閘道伺服器 i 所要用到的權重是多少才比較恰當。

由於閘道伺服器的權重公式需要知道同處在同一個 wireless mesh network 的其他閘道伺服器的網路情況，所以閘道伺服器也是要聽其他閘道伺服器發送的 routing information message，並記錄他所能聽到的閘道伺服器。當需要算出權重時，可以向其他的閘道伺服器詢問它所需要的資料。算出權重後，他會廣播一個 gateway weight 的 message 給所有在同一個 wireless mesh network 下的 backhaul router。或是也是可以將閘道伺服器權重包裹在 routing information message 中，就不需要另外做特別的廣播出去。

3.3.3.2 backhaul router 端

在 Backhaul router 端，收到更新後的閘道伺服器權重後，會針對現有的權重做一次運算，算出每一個他所能到達的閘道伺服器的選取機率是多少。列入運算考慮的權重除了 3.3.3.1 中介紹的閘道伺服器權重（Gweight）外，還有一個就是 distance 權重。

Distance 權重是用目前所在 backhaul router 與閘道伺服器的距離關係所得出的權重。Distance 權重的距離可以純粹看 routing table 中的最小 hop count，也可以較爲複查的考量到 backhaul router 之間的距離與雜訊狀況等環境因素，但在此先簡化爲單純看 routing table 中的最小 hop count。Distance 權重公式如下所示：

$$Dweight_i = \left(1 - \frac{\min\_dist_i}{\sum_j \min\_dist_j}\right) \quad (3)$$

$\min\_dist_i$：閘道伺服器 i 在 backhaul router 的 routing table 中最小的 hop count。

運算在 backhaul router 選取閘道伺服器 i 的機率公式如下：

$$GProb_i = \frac{(\alpha * \text{NGweight}_i + (1-\alpha) * \text{NDweight}_i)}{\sum_j (\alpha * \text{NGweight}_j + (1-\alpha) * \text{NDweight}_j)} \quad (4)$$

公式(4)的設定考量是想要針對閘道伺服器權重和distance權重算出一個適當的選取機率，而兩個不同權重是藉由α參數來做調節，換句話說，α參數決定 backhaul router 在運算閘道伺服器選取機率時，閘道伺服器權重和 distance 權重的比重大小。α參數會依據網路狀況不同而有所不同，通常來說 wireless mesh network 的流量越大，α值也會越小；也可以說是閘道伺服器的負載越大，α值也會越小。相反地，wireless mesh network 的流量越小，α值也會越大。

公式（4）中的 NGweight 和 NDweight 的定義如下：

NGweight_i：閘道伺服器 i 在目前 backhaul router 上面真正用作運算的閘道伺服器權重。取 Normalized Gateway Weight 縮寫成的。

由於每一個 backhaul router 可以連到的閘道伺服器不一定會完全相同，所以 backhaul router 收到閘道伺服器的權重後不能直接用於運算，需要先做好 normalization，讓 backhaul router 所能到達的閘道伺服器權重總和加起來爲 1。因此，在此會將閘道伺服器 i 的權重和目前 backhaul router 的其他伺服器權重做 normalization。

$$\text{NGweight}_i = \text{Gweigth}_i \; / \; \sum_j Gweight_j \quad (4.1)$$

$NDweight_i$：爲閘道伺服器 i 在目前 backhaul router 上的 distance 權重。取 normalization distance weight 縮寫而成的。

$$NDweight_i = \frac{Dweight_i}{\sum_j Dweight_j} \quad (4.2)$$

第四章　simulation

4.1　模擬模型假設

在本章節中，我們將比較我們選取閘道伺服器的方法對網路的影響。除了我們自己提出的 Load Balancing 選取法外，將與我們提出的方法比較的樣本分別是：

（1）Nearest 選取法：選取離 initial backhaul router 最近的閘道伺服器

（2）Random 選取法：隨機選取閘道伺服器。

我們將針對閘道伺服器的負載狀況和建立路由路徑的失敗率進行比較與分析。

4.1.1　網路拓樸架構

我們將針對平衡（balanced）的環境與不平衡（unbalanced）的環境兩種不同的狀況作分析，balanced 的環境主要是作爲比較的樣本。而圖 4.1 和圖 4.2 則分別顯示我們模擬所使用的 wireless mesh network 拓樸模型。拓樸模型主要是一個由十六個 backhaul router 和兩個閘道伺服器的網狀網路，每一個 backhaul router 與其上下左右相鄰的 backhaul router 相連，每一個 link 的頻寬都設定成 54Mbps。

圖 4.1可以明顯看出是用在balanced的拓樸和環境變數下的。

圖 4.2則是unbalanced的拓樸和環境變數，和圖 4.1的差異在於少了GW1 到BR13 的那一條link，因此閘道伺服器的接受流量能力也就不會相同，即兩個閘道伺服器的capacity不同，我們可以從這個拓樸環境下，再去探討當有unbalanced的環境變數的情況會是如何。unbalanced的環境變數設定在後面的 4.1.2 有詳細的講解。

由於在現實環境下，受到使用者的使用習慣，環境中使用群組差異性，時間上的因素等等各式各樣的外在變數的影響，balanced 的環境變數是不大可能出現的情況，所以 balanced 的拓樸和環境變數的比較其實沒有很大的意義。因此，本論文比較注重在當在 unbalanced 的狀況下，選取法的效能的比較方面。

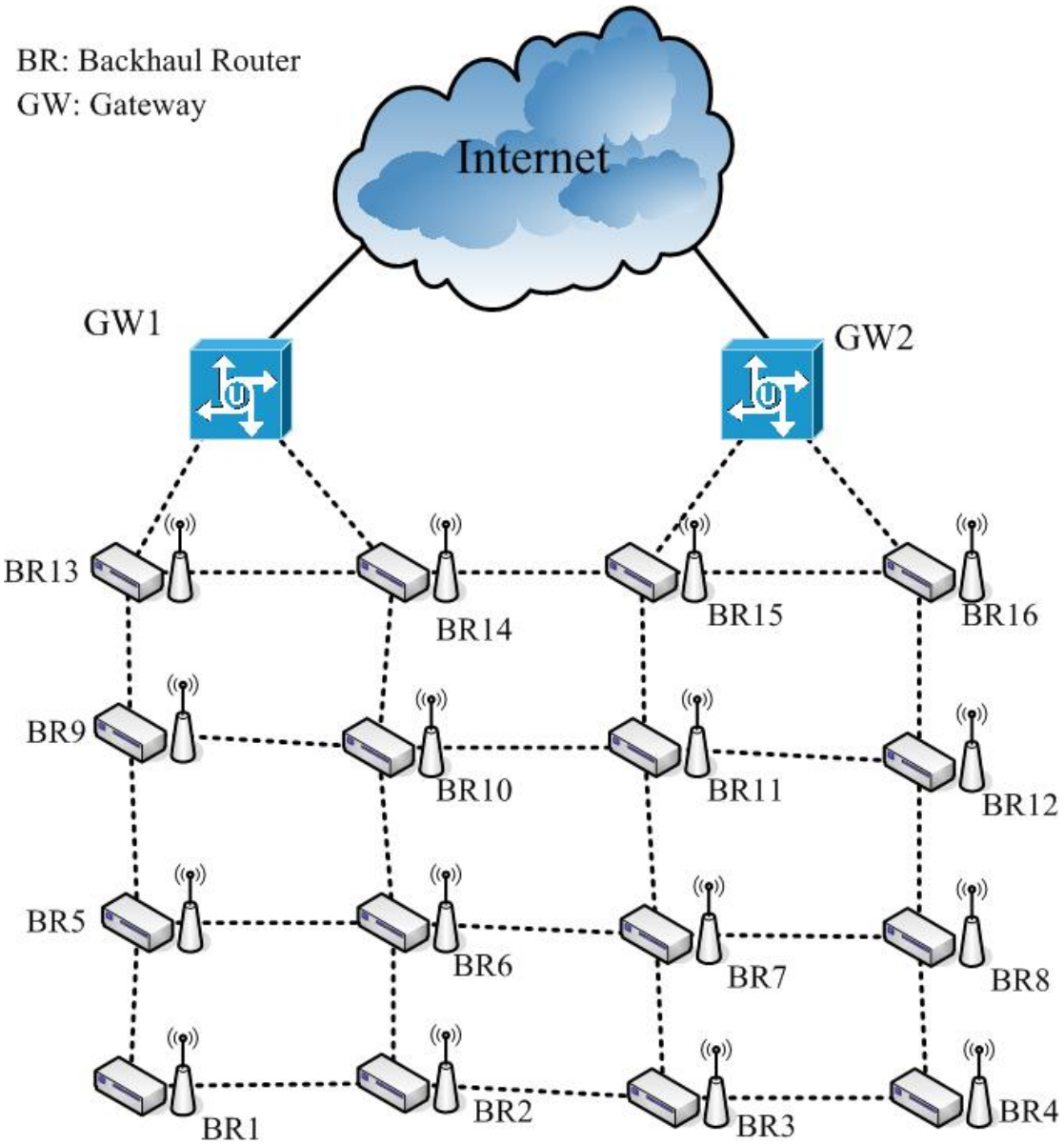


圖 4.1 Topology Model 1 of Simulation

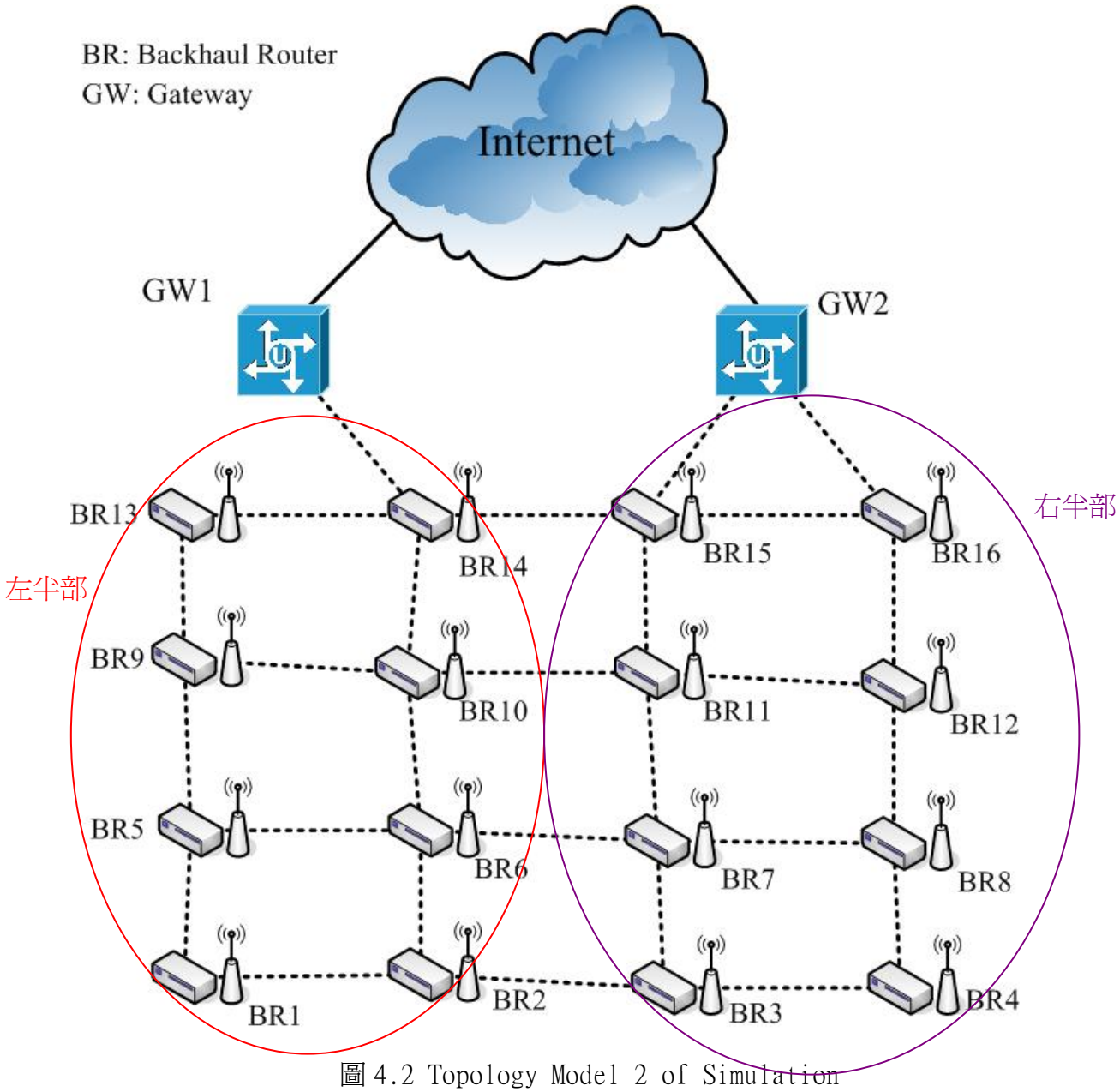


圖 4.2 Topology Model 2 of Simulation

4.1.2 模擬架構

我們的模擬是以 event-driven 模擬器爲架構，主要分成兩個階段。在 session 階段，會做路由以及 resource reservation；在 packet 階段，則會開始傳送 data 封包。我們模擬的架構中包含有兩種不同的 traffic 種類，他們有不同的 bandwidth 需求，一個是 3Mbps，另一個則是 8Mbps，而 mobile node 需求哪一種 bandwidth 是隨機選擇的。至於 mobile node 的 departure rate 則是採用 poisson distribution 來決定的，其λ值設成一個人平均在系統中會呆上 3.33 個單位時間，也就是說每單位時間有 0.3 個人

會離開。

由於在現實世界中，不大有機會會是 balanced 的環境變數，所以就可以針對現實世界的狀況來做出有許多不同的考量，設定出多樣的環境變數，並藉由調整他們來控制 unbalanced 的狀況。但在本論文中，爲了避免太多的變數交互影響，所以目前只有設一個環境變數，那就是 **mobile node attach 到特定區域的 backhaul router 的機率**。因爲 path request 是從 mobile node attach 到的 initial backhaul router 起始到 initial backhaul router 所選定的閘道伺服器，所以這個環境變數對於不同的選取法和網路的流量狀況會有很大的影響。

在此，我們針對不同的區域設定不同的 mobile node attach rate。像是在校園網路中，白天和晚上不同區域的使用者數目會有很大的差異。白天因爲學生要上課，或是學生會去實驗室做研究，不會待在寢室中，所以宿舍區的網路流量就會變小。相對地，教學區有白天有教職員工要上班會使用到網路，又有實驗室的網路使用量，所以教學區在白天的網路流量就會很大。但相反地，當晚上的時候，學生都回寢室了，而教職員工也下班了，所以教學區的流量就會變少，而宿舍區的流量就會變大。鑑於有此種狀況會出現，所以我們這一次的模擬，就將拓樸區域分成左右兩半部，分別給予不同的 mobile node attach 機率，來作爲設定 balanced 或 unbalanced 的環境變數參數。

4.1.3 performance metrics

這邊將先介紹一下會用作效能評估的 metrcis。除了閘道伺服器的負載（loading）狀況外，接著比較重要的就是連線失敗率（call blocking rate）。連線失敗的情況包含因爲路由演算法沒有找到有足夠資源的路徑，或者是選定好的路徑在做 reservation 時沒有足夠的資源。後面這種狀況發生原因是因爲在找適當的路徑時沒有一邊找一邊就 reserve 下來，而是找好路徑後，再一路從閘道伺服器端 hop-by-hop reserve 回 initial backhaul router，所以可能是尋找路徑的時候有足夠的頻寬，但當要 reserve 的時候在這段時間差內已經被別的路徑 reserve 去了，導致 reservation 失敗。

雖然連線失敗率可以看出 mobile node 被拒絕的機率，但由於在我們的架構中，mobile node 可已有兩種不同的 request bandwidth，所以另外增加了一個頻寬連線失敗率（bandwidth blocking rate）。Bandwidth blocking rate 的定義是失敗的頻寬佔總共 request 的頻寬的百分比。因爲當 mobile node 需求不同的頻寬時，一個低的 call

blocking rate 並不一定代表說他的效率就比較高，可能是他靠拒絕頻寬需求比較大的 mobile node 來服務更多的低頻寬需求的 mobile node，這樣是不合公平原則的，每個 mobile node 的使用網路權力應該是相同的，不能有偏好。

$$\text{call blocking rate} = \frac{\sum \text{reject request number}}{\sum \text{total request number}}$$

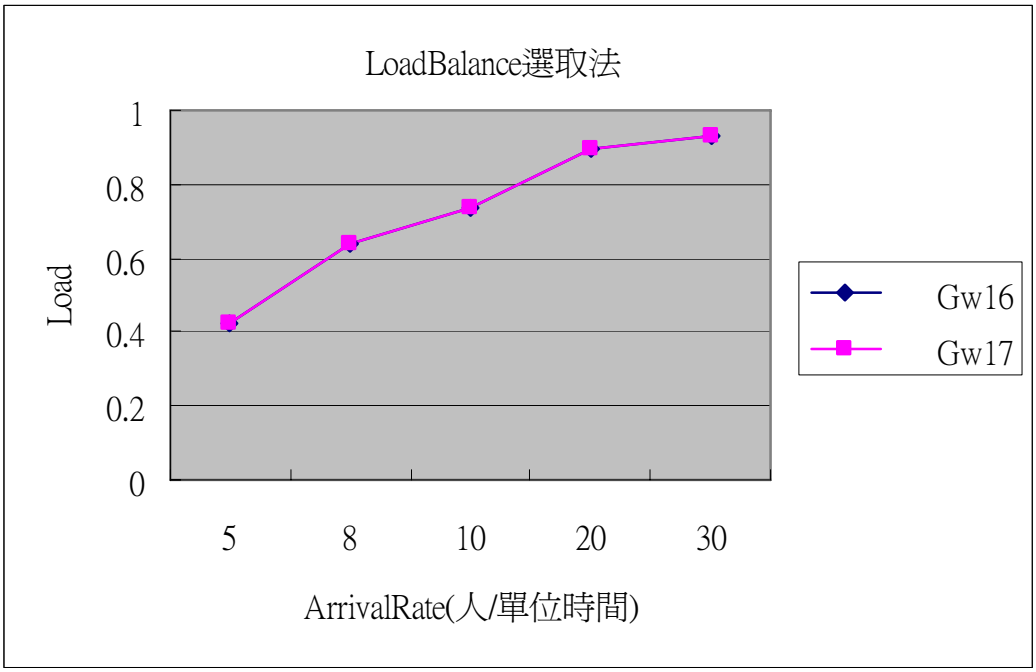
$$\text{bandwidth blocking rate} = \frac{\sum \text{reject bandwidth}}{\sum \text{total request bandwidth}}$$

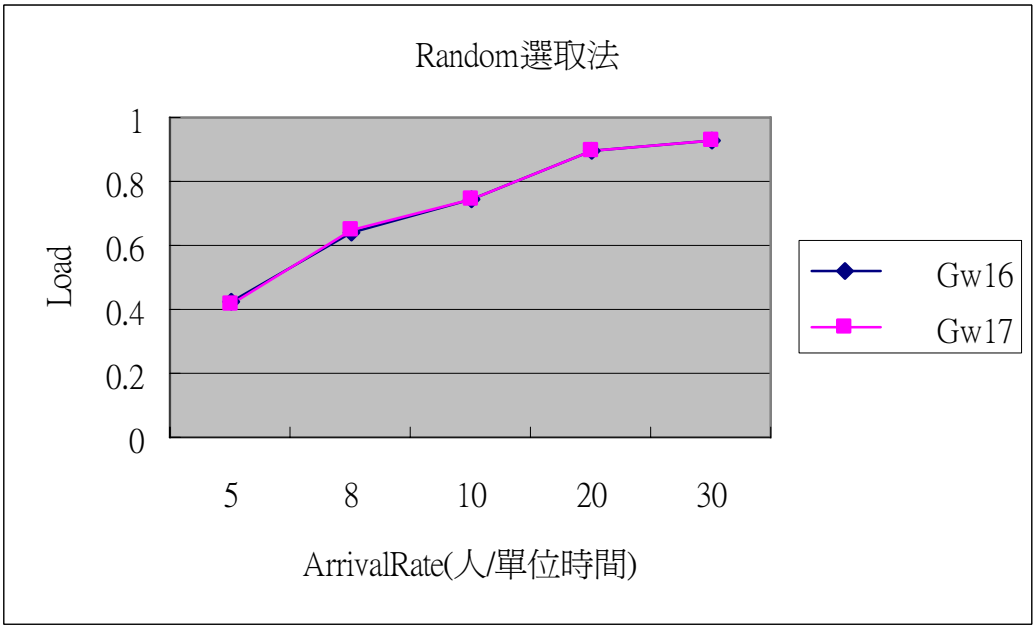
4.2 模擬結果分析

4.2.1 Balanced 環境模擬

本小節比較在balanced的拓樸和環境變數下，即圖 4.1的架構，三種不同的閘道伺服器選擇方法的運作狀況。

如下圖所示，三個選取法的效能都差不多，沒有很大的差異性。因爲 mobile node attach 到每一個 backhaul router 的機率都相同，所以導致從每個 backhaul router 起始的 path request 數目和頻寬需求平均下來都是相同的。在這樣原本就流量就 balanced 出現的情況下，大家的選取法出來的結果都會差不多。


LoadBalance選取法
Load
1
0.8
0.6
0.4
0.2
0
5
8
10
20
30
ArrivalRate(人/單位時間)
Gw16
Gw17


Random選取法
Load
1
0.8
0.6
0.4
0.2
0
5
8
10
20
30
ArrivalRate(人/單位時間)
Gw16
Gw17

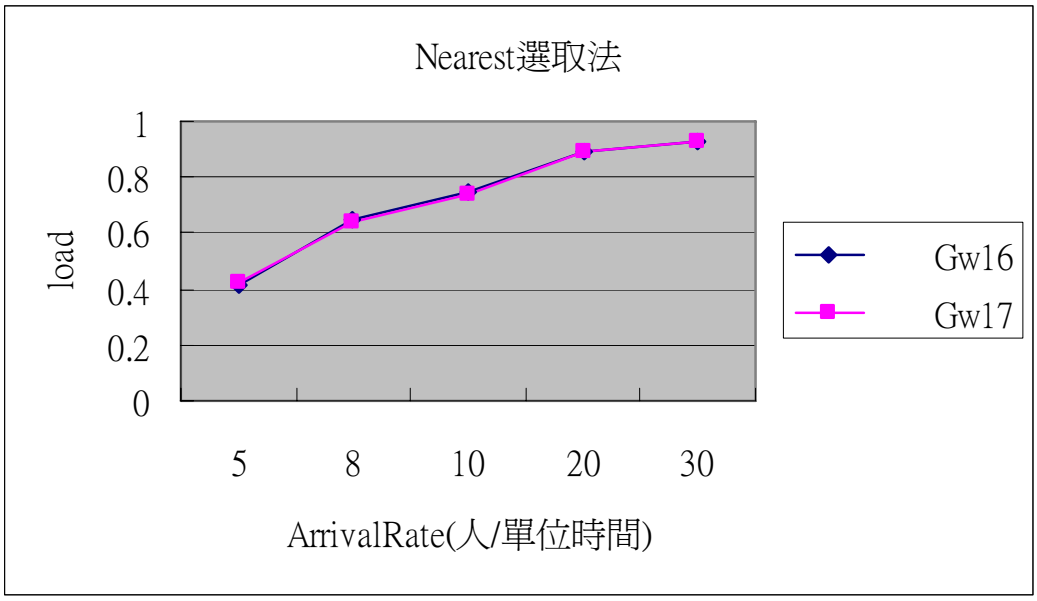


圖 4.3 Balanced 下，三種選取法的閘道伺服器負載

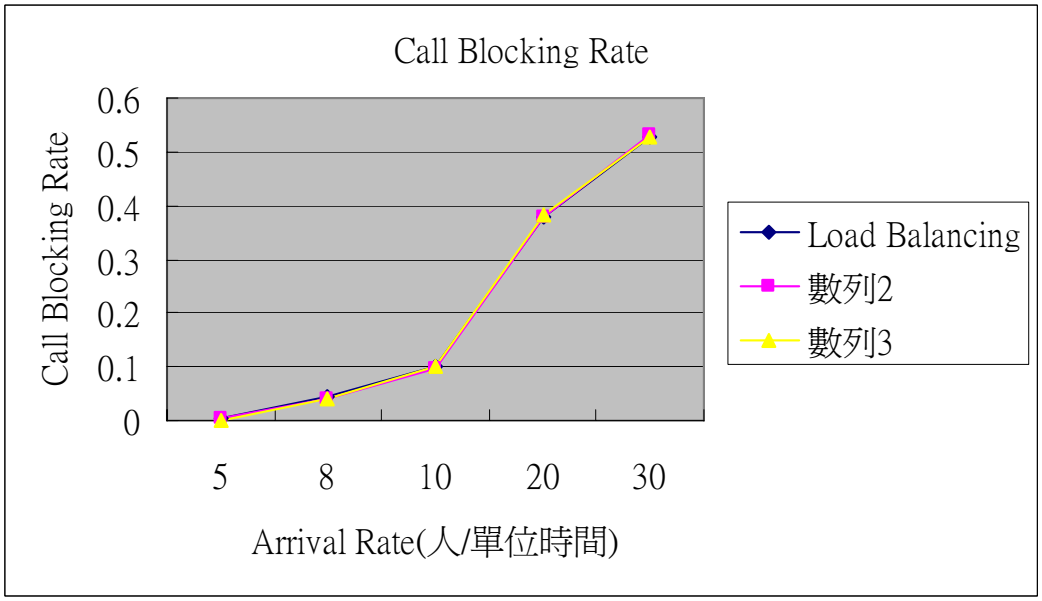


圖 4.4 Balanced 下，Call Blocking Rate 比較圖

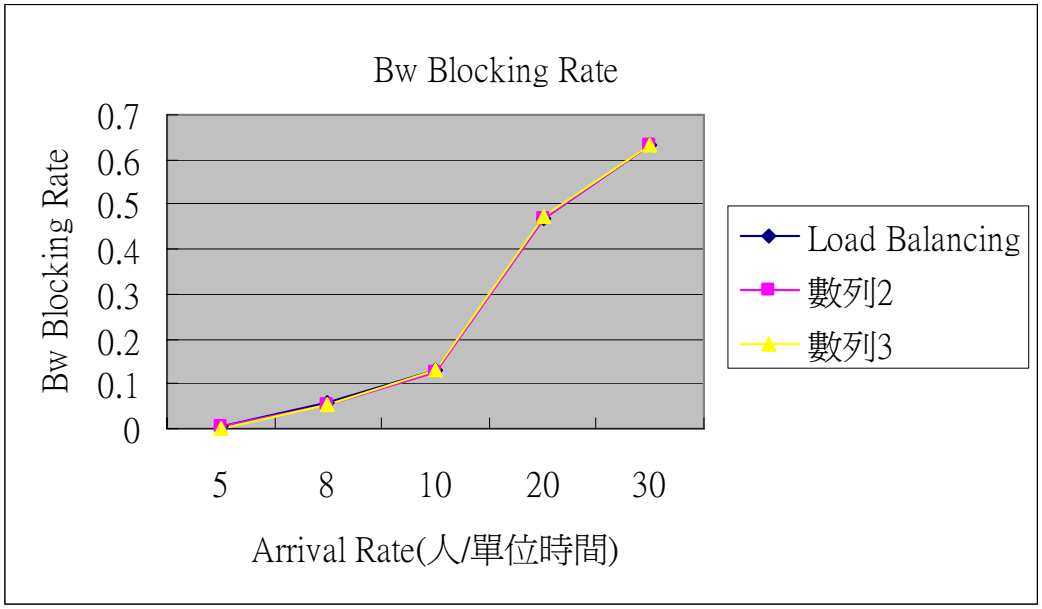


圖 4.5 Balanced 下，Bandwidth Blocking Rate 比較圖

4.2.2 Unbalanced 環境模擬

本小節將比較在unbalanced的拓樸和環境變數下，即圖 4.2的架構，三種不同的閘道伺服器選擇方法的運作狀況。

我們的設定的 unbalanced 環境變數，是將左半部與右半部的 mobile node attach rate 設定成不一樣的狀態。分別測三種不一樣的狀況下的效率，其中有兩個極端的，一個平均的。極端的狀況是，一個設定成左半部的 mobile node attach rate 爲 0.7，換句話說，mobile node 會有 70%的機率連到左半部的 backhaul router；而右半部則相對地設成了 30%的 attach 機率。另一個則是剛好相反的狀況，即左半部的 mobile node attach rate 爲 0.3。平均的狀況則是左右半部的 mobile node attach rate 各爲 0.5。

4.2.2.1 左右區域的 MN attach rate 分別爲 0.7 和 0.3

下面三個圖中，顯示出三種不同選閘道伺服器的方法對於閘道伺服器的負載的影響。由圖 4.6可以看出，採用Load Balancing的選取法的閘道伺服器不管mobile node的arrival rate是多少，兩個閘道伺服器（Gw1 和Gw2）的負載都幾乎相同，沒有什麼差異。而Random選取法和Nearest選取法兩種的閘道伺服器負載都有不小的差距，尤其是當arrival rate小的時候差異就會比較大，當arrival rate大的時候兩個閘道伺服器的

差異就會比較小。其中，又以Nearest的差異比較大，這是因爲當拓樸定下來的時候，每一個backhaul router會選擇的閘道伺服器就固定了，除非backhaul router距離多個閘道伺服器都是一樣近。在本次的架構中（圖 4.2），每一個backhaul router都只有唯一一個閘道伺服器選擇，左半部的會選擇Gw1，而右半部的會選擇Gw2。所以當左半部的流量大的時候，Gw1 的負載就會很重，因爲他不像Random選取法，會隨機從兩個閘道伺服器中做選擇。而Random選擇法雖然是會平均分配流量到兩個閘道伺服器，但因爲兩個閘道伺服器他們本身的capability不一樣，Gw2 的接受能力比Gw1 好，所以會導致Gw1 的負載相對地比較重。

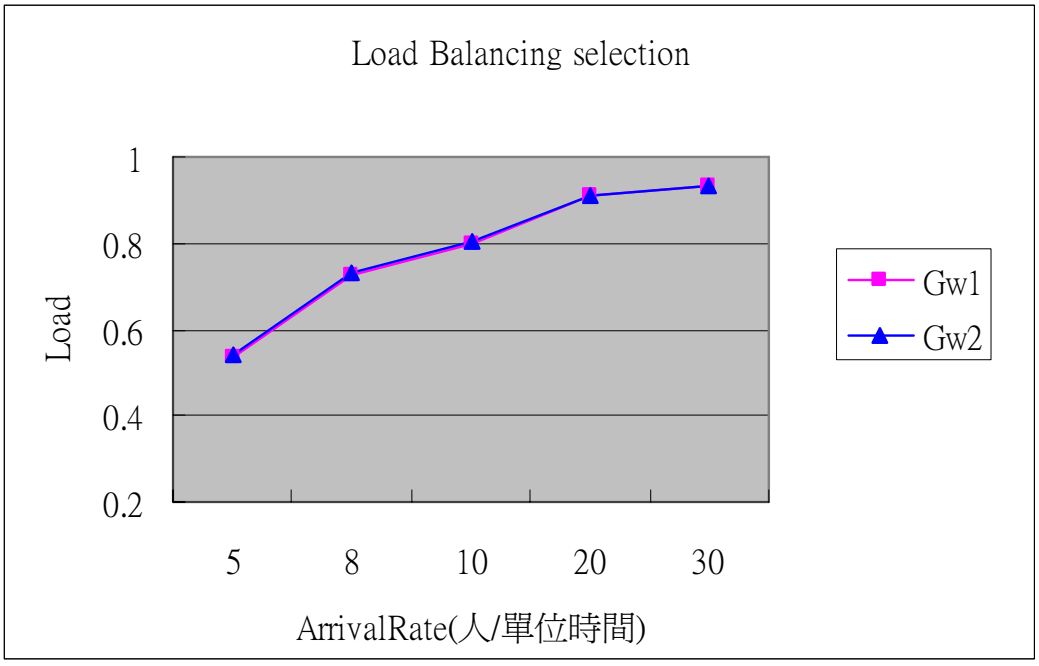


圖 4.6 (1)Unbalanced 下，Load Balance 的閘道伺服器負載圖

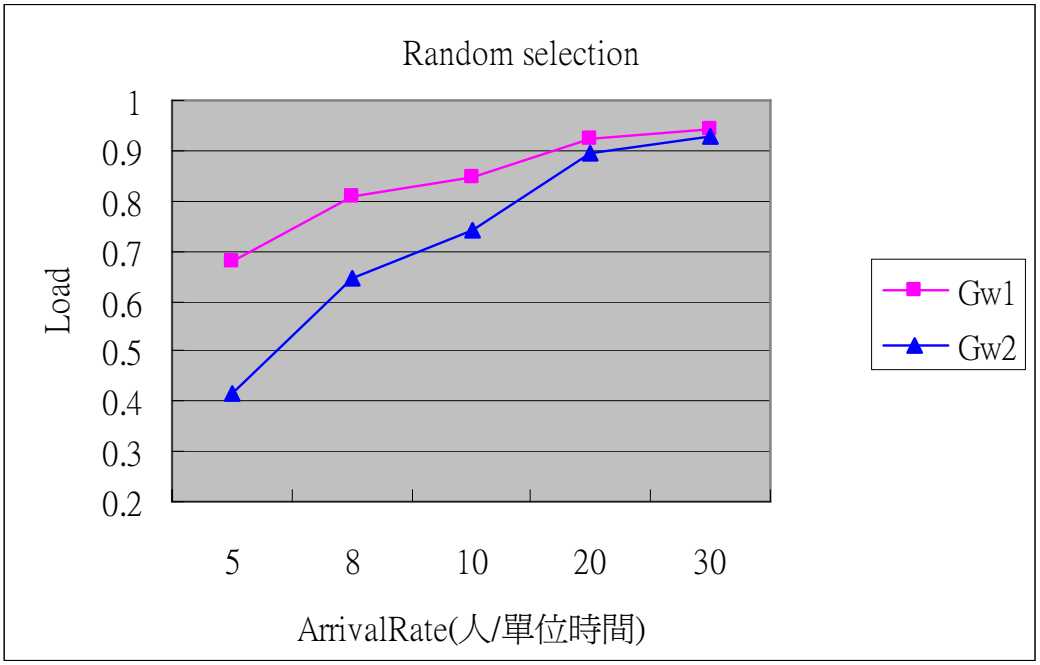


圖 4.7 (1)Unbalanced 下，Random 的閘道伺服器負載圖

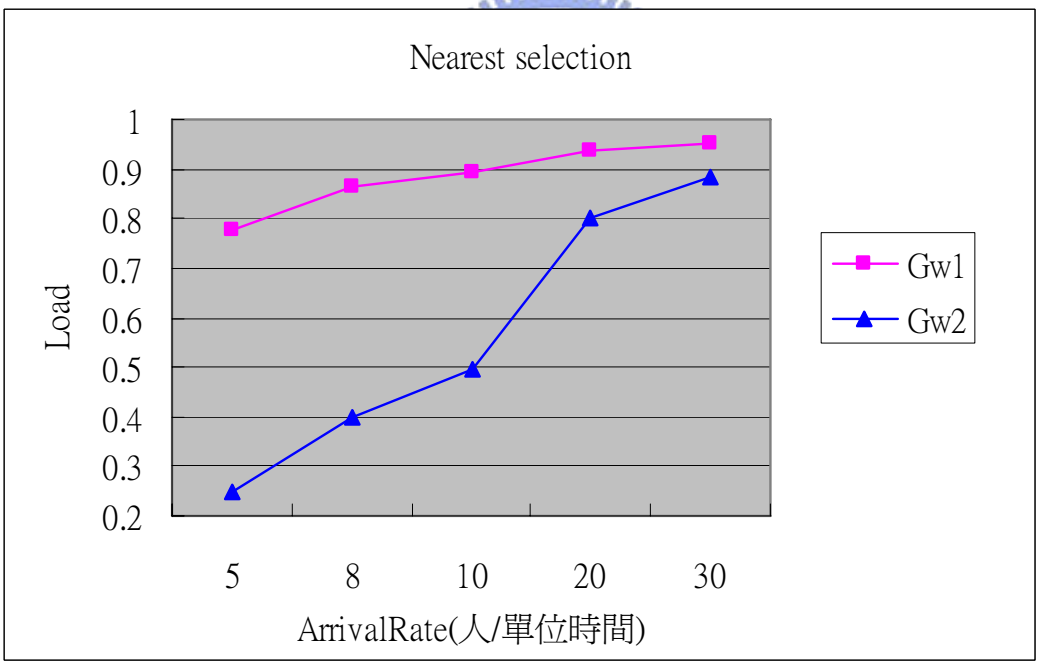


圖 4.8 (1)Unbalanced 下，Nearest 的閘道伺服器負載圖

當arrival rate比較小，也表示著閘道伺服器比較容易滿足mobile node的頻寬需求，因此具有較低的連線失敗機率（call blocking rate）。圖 4.9和圖 4.10顯示出在目前模擬的unbalance環境下，考慮load balancing的選取法在arrival rate較低的時候效率比較好，而當arrival rate比較高的時候，三個選取法都差不多，因爲頻寬嚴重不足，所以連線的失敗率很高，大家都差不多。會出現這樣的結果主要是因爲nearest

的選取法由於會選擇最近的閘道伺服器，而這一個模擬環境的左半部mobile node attach量較大，所以會導致Gw1 很快就會頻寬不足，而他又不會另外選擇較遠的閘道伺服器，所以很多attach到左半部的mobile node會被拒絕連線。至於隨機選取法是因爲Gw1 和Gw2 所能接受的流量不一樣，而隨機選取法會將兩個閘道伺服器視爲同一個等級，選取機率一樣，而事實上Gw1 只有Gw2 一半的接受能力，所以Gw1 也會比較快面臨頻寬不足的問題。考慮到load balancing的選取方法則是沒有這樣的問題，因爲他會根據目前閘道伺服器的流量來考慮下一個period內該如何分配流量，所以表現出的結果也會最好。

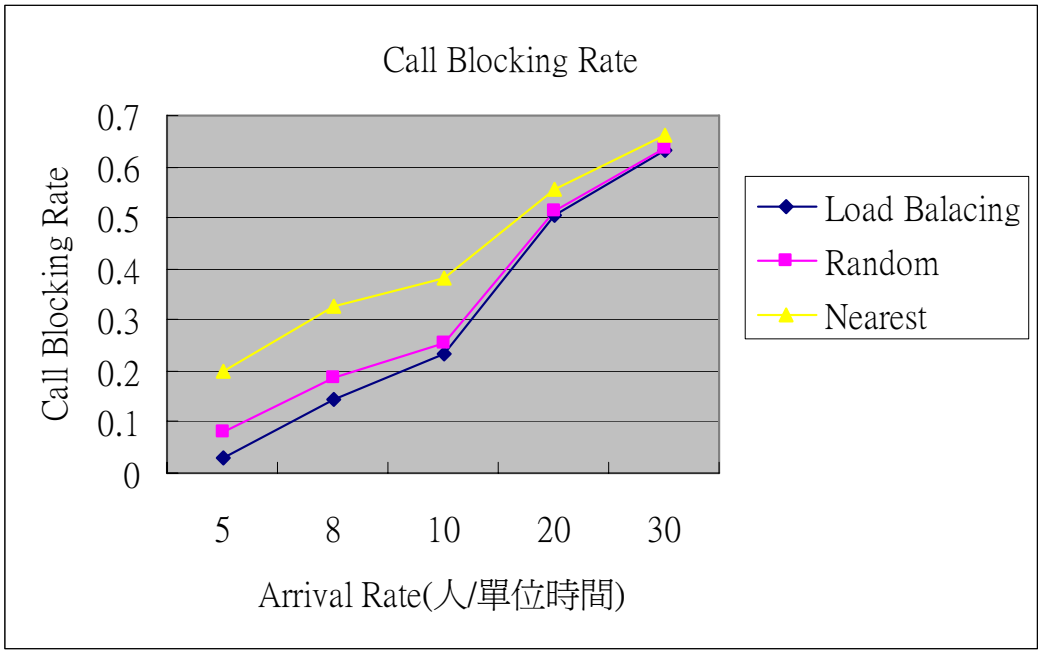


圖 4.9 (1)Unbalanced 下，call blocking rate 比較圖

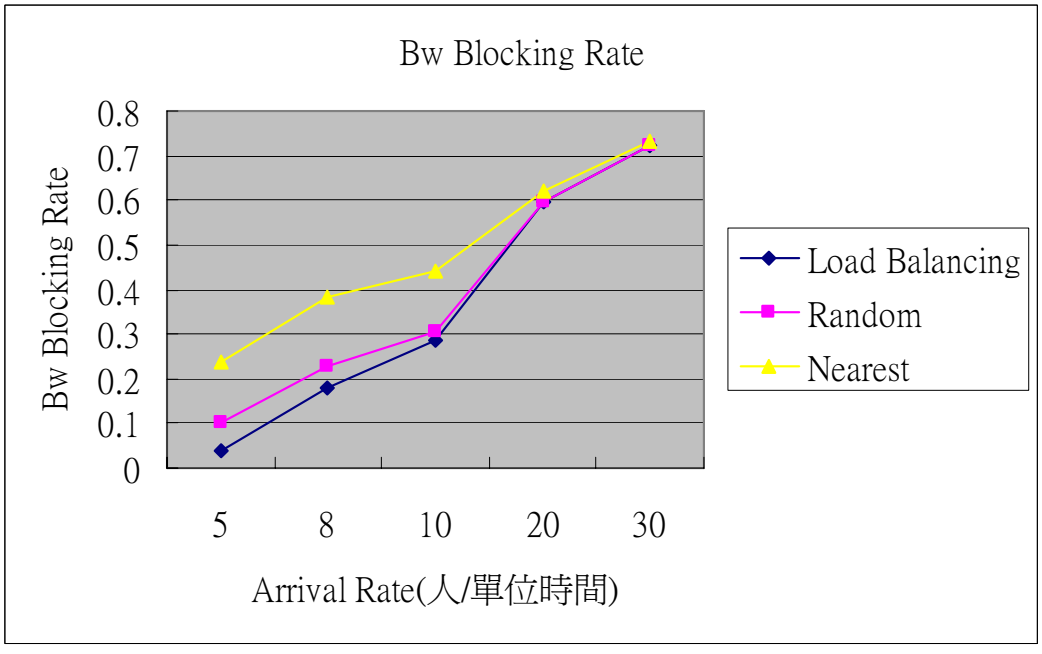


圖 4.10 (1)Unbalanced 下，bandwidth blocking rate 比較圖

4.2.2.2 左右區域的 MN attach rate 分別爲 0.5 和 0.5

Load Balancing 和 Random 的效率與之前的環境差異不大，這是因爲 mobile node attach rate 的變動對於有考慮到負載狀況和隨機選取的方法並沒有影響。但對於 Nearest 則會有影響，因爲目前的流量的分配是平均的左右各爲 0.5，所以兩個閘道伺服器的負載情形並不像之前的有那麼大的差異性。

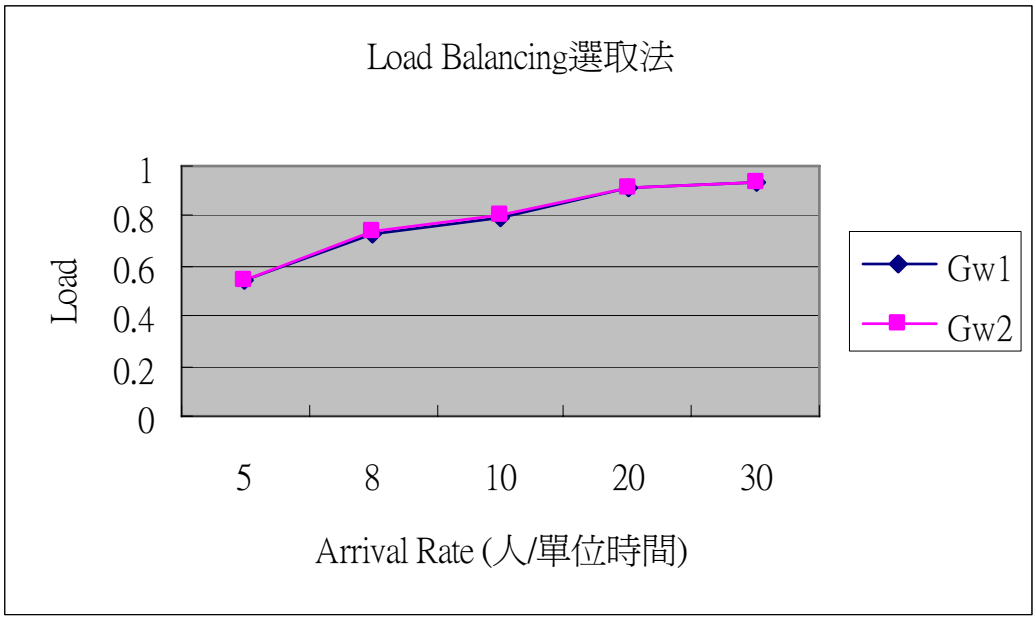


圖 4.11 (2)Unbalanced 下，Load Balance 的閘道伺服器負載圖

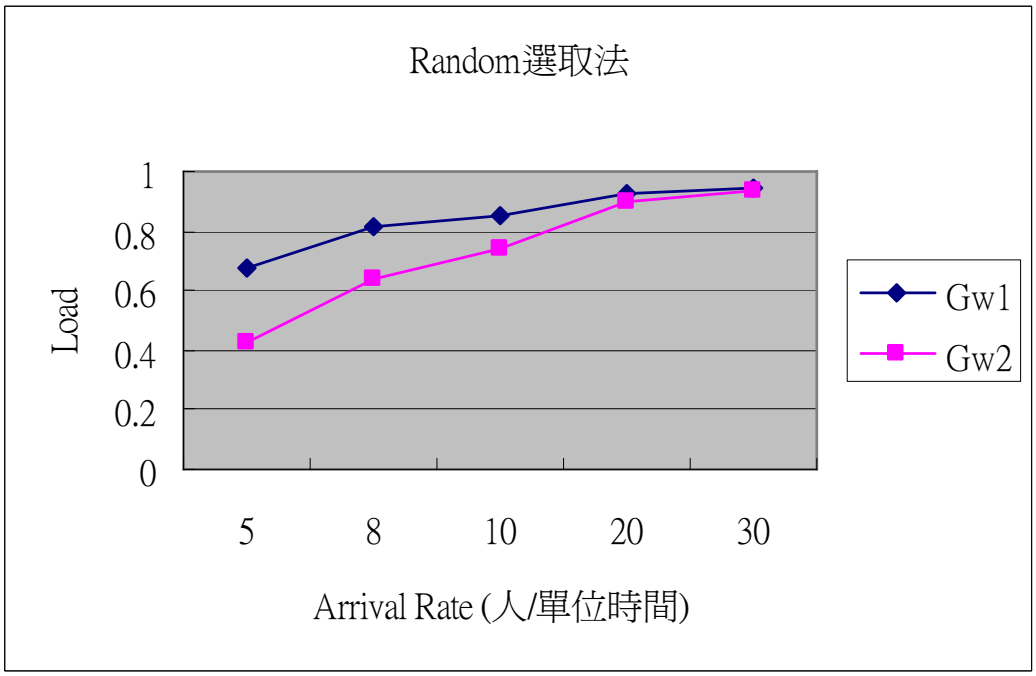


圖 4.12 (2)Unbalanced 下，Random 的閘道伺服器負載圖

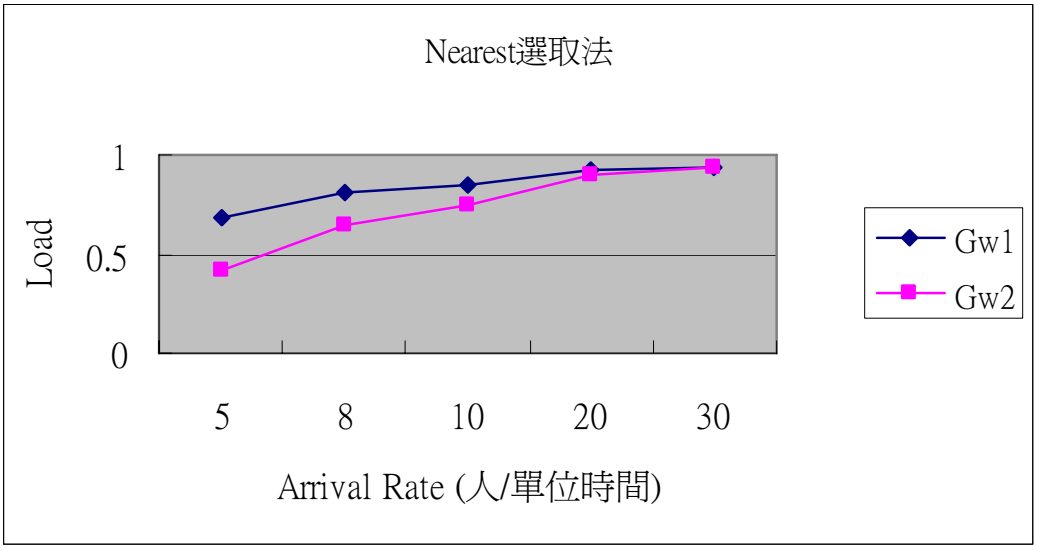


圖 4.13 (2)Unbalanced 下，Nearest 的閘道伺服器負載圖

由於此環境比之前的均衡，不像之前的流量很極端的會往左邊的閘道伺服器流，導致左邊的閘道伺服器很容易超出負載而拒絕使用者連線，因此Nearest的效果也就比之前的更好，他的call blocking rate跟random的方法很接近，但還是沒有Load Balancing選取法好。

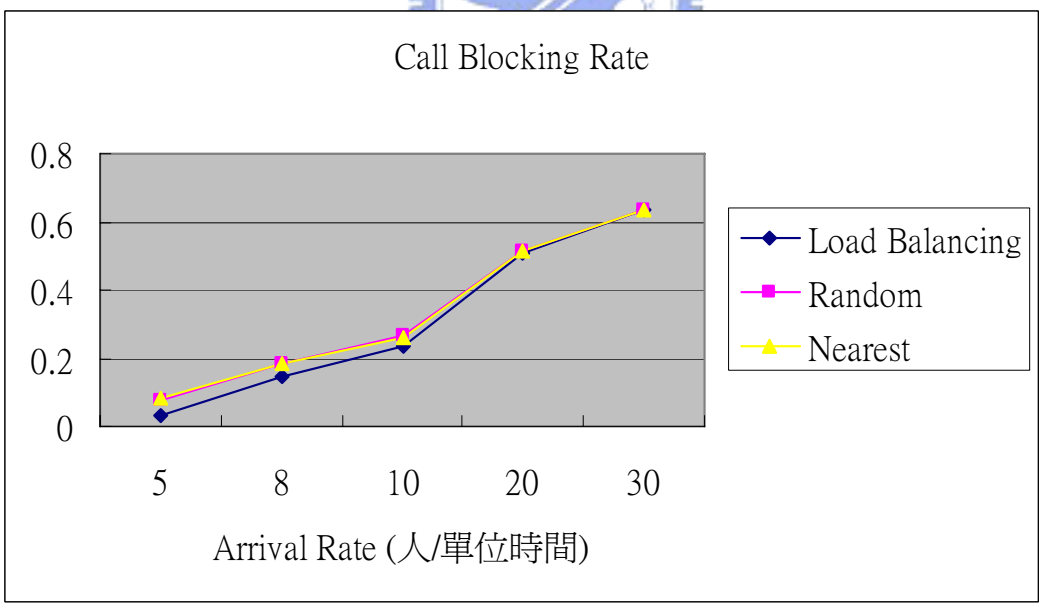


圖 4.14 (2)Unbalanced 下，call blocking rate 比較圖

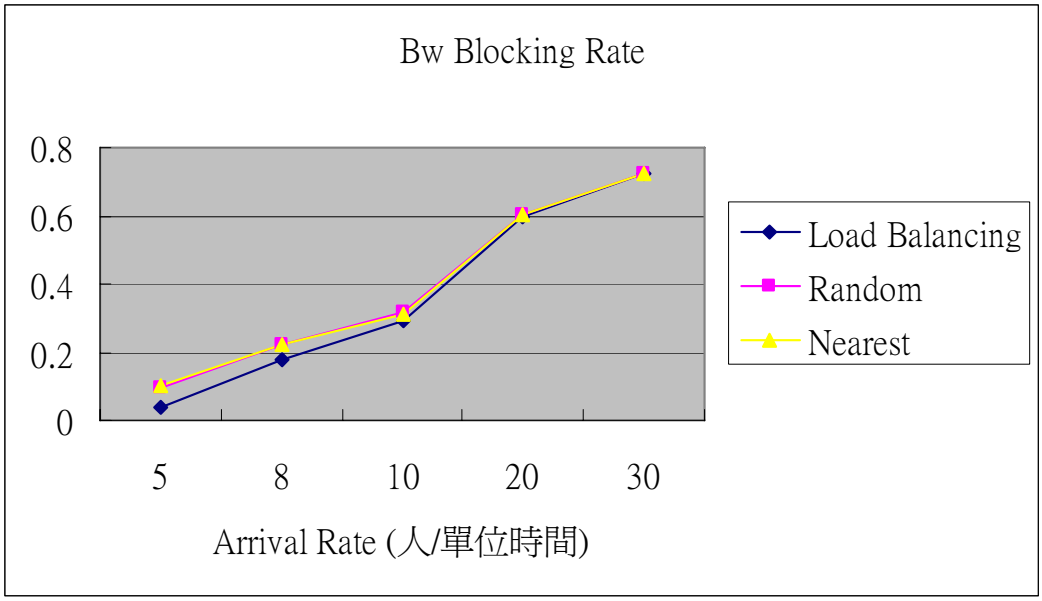


圖 4.15 (2)Unbalanced 下，bandwidth blocking rate 比較圖

4.2.2.3 左右區域的 MN attach rate 分別爲 0.3 和 0.7

由下圖可以看出，因爲 Load Balancing 的選取法的閘道伺服器考慮到負載狀況所以兩個閘道伺服器（Gw1 和 Gw2）的負載都幾乎相同。而 Random 選取法和 Nearest 選取法兩種的閘道伺服器負載都有一些小差距。Random 的結果和他在其他狀況測出的負載都是一樣的，而 Nearest 的差異性則變的比較小，這是因爲在這個拓樸中，左邊的閘道伺服器的 capacity 比右邊的小，而在此環境下的流量也是左邊會比右邊小，所以兩個閘道伺服器的負載就會比較相近。

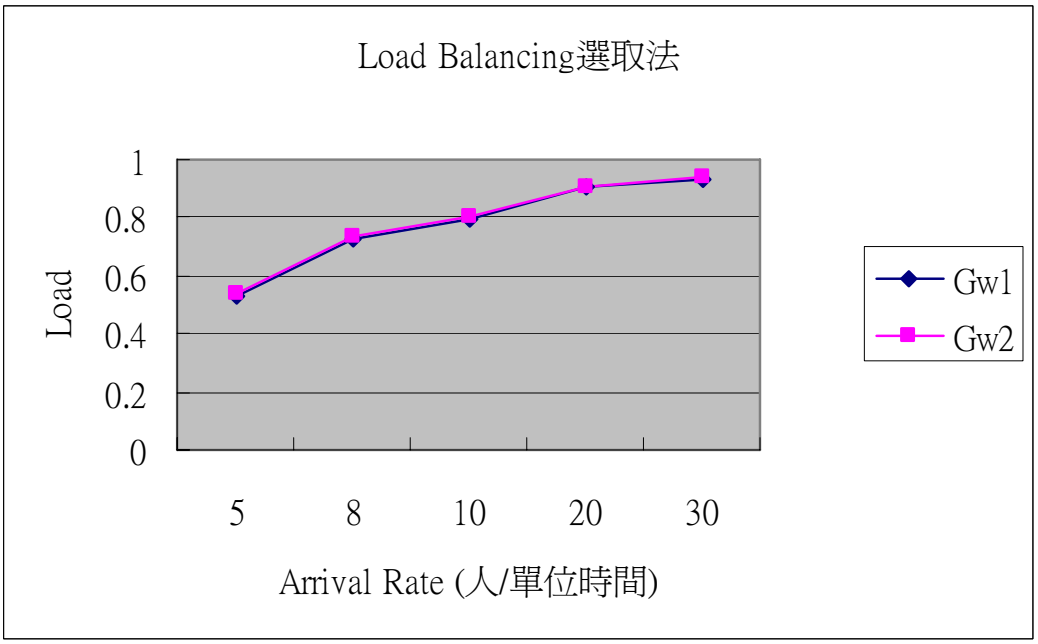


圖 4.16 (3)Unbalanced 下，Load Balance 的閘道伺服器負載圖

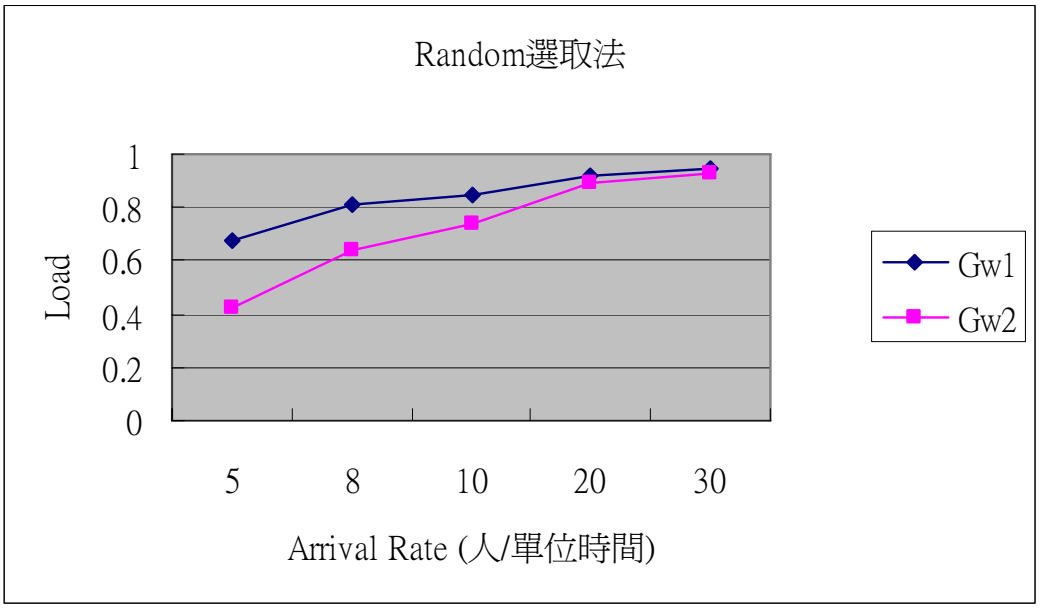


圖 4.17 (3)Unbalanced 下，Random 的閘道伺服器負載圖

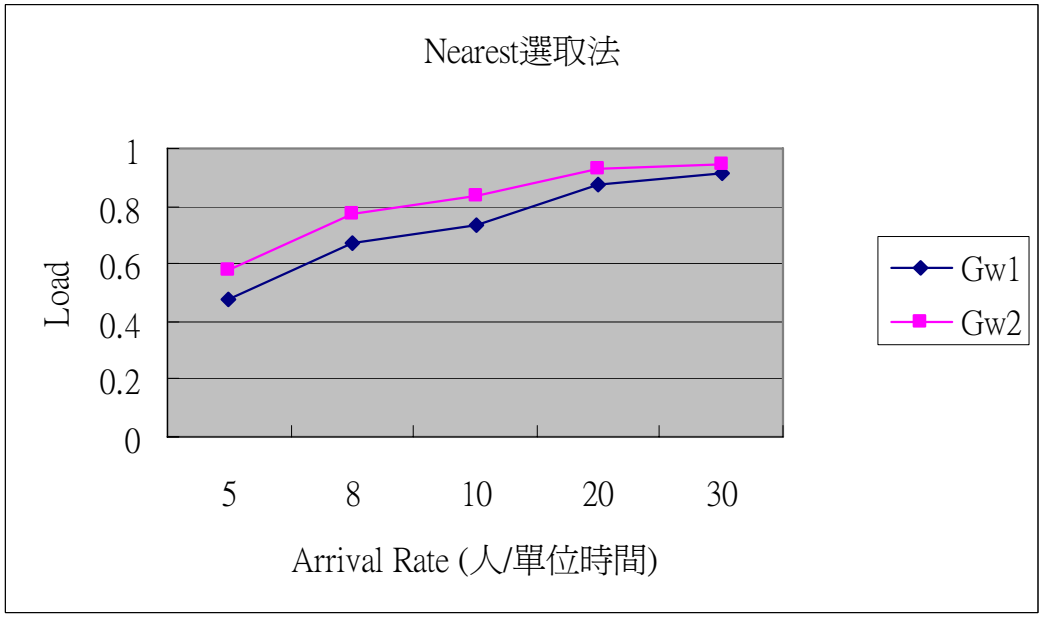


圖 4.18 (3)Unbalanced 下，Nearest 的閘道伺服器負載圖

圖 4.19可以看出三個不同選取法在這個環境下的效率。Random還是和其他的環境下執行效率相同，而在此Nearest和Load Balancing的選取法的效率很接近，這是因爲左右兩個閘道伺服器的capacity比例是 1：2，與mobile node attach rate的 0.3：0.7 是相近的，所以跑出來的結果和有考慮到負載狀況的選取法效能差不多。更由於他所走的是最短距離，即不會浪費多餘的資源，所以Nearest選取法的call blocking rate還會略小於Load Balancing選取法。

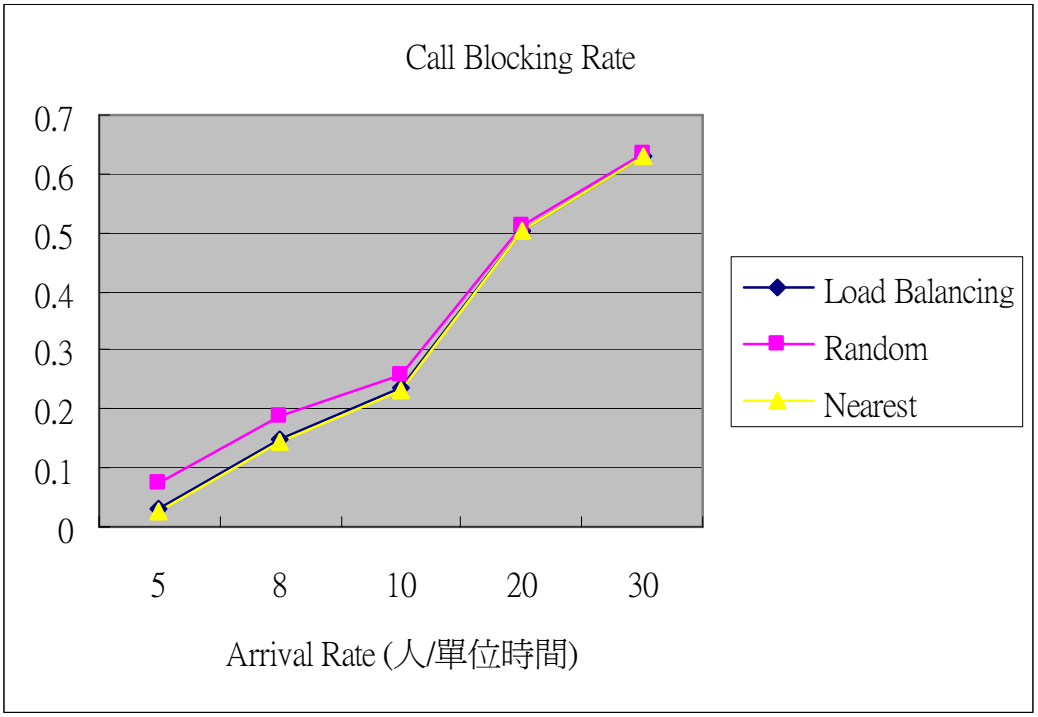


圖 4.19 (3)Unbalanced 下，call blocking rate 比較圖

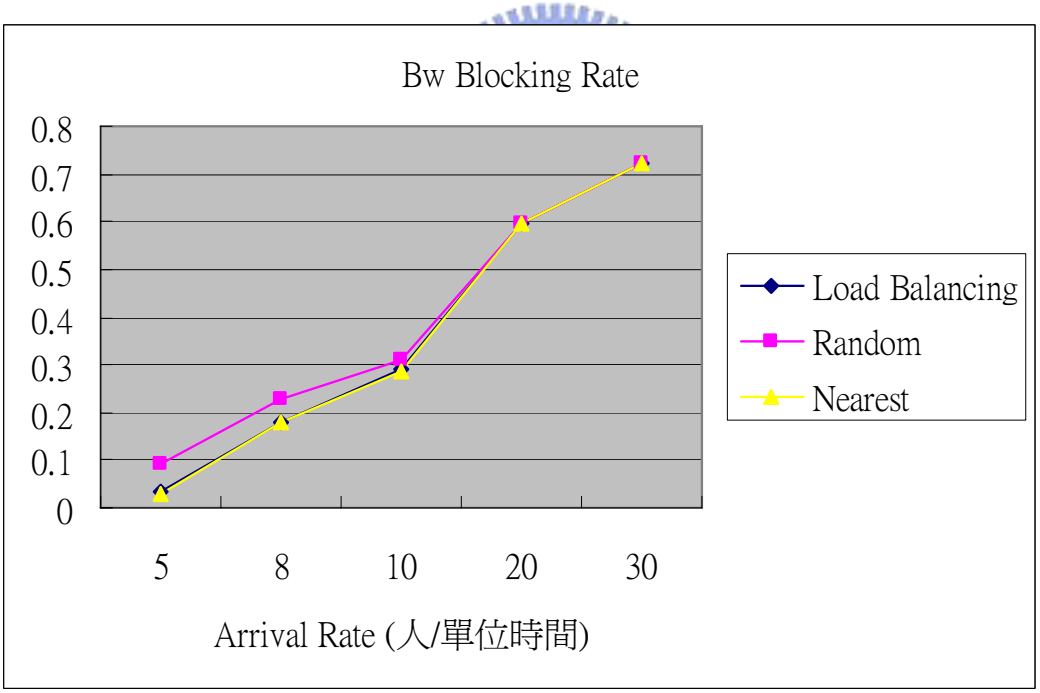


圖 4.20 (3)Unbalanced 下，bandwidth blocking rate 比較圖

第五章　結論與未來工作

5.1　結論

在本篇論文中，我們提出一個考量到傳輸流量平衡並支援 QoS 的無線網狀網路路由方法。我們利用連上網際網路需要先透過閘道伺服器的特性，在建立連線時考量到將網路的傳輸流量平均分散到整個網路中，期望能更有效率的利用有限的頻寬，讓網路能提供服務給更多的使用者。

在此，我們將路由機制分成兩個步驟，分別爲閘道伺服器的選取步驟和 hop-by-hop 路徑選取步驟。首先，我們使用了一個閘道伺服器的選取機制，來將使用者的流量導向不同的閘道伺服器，避免閘道伺服器過渡擁塞。這樣一來，由於在閘道伺服器的選取時已經考量到流量平衡的問題，所以只要再配合 hop-by-hop 的路徑選取的機制來找出適當的路徑。由於 hop-by-hop 的路徑選取方法，只考量到符合 QoS 需求的 next hop，所以很可能目前找到的最適當的節點並沒有辦法到達閘道伺服器。所以當到達一個找不到適當 next hop 的節點時，會將 path request 的封包再上一個節點傳送，讓上一個節點再找另外一個適當的節點，這樣在嘗試建立連線的時候比較不容易失敗。在最後的效能分析結果裡可以發現，我們的機制在不一樣的環境變數下，可以提供給使用者的服務都有良好的品質。

5.2　未來工作

未來，我們希望能夠時做出一個無線網狀網路的環境，觀察並統計使用者移動的頻率與模式，針對使用者換手時，該如何維持或新建立一個連線路徑設計一套規則。此外，目前的閘道伺服器選取機制，只有考量到 short term factor，所以在未來的工作裡，我們希望能夠在選取權重中加入 long term factor，看是否可以將閘道伺服器的效能利用的更好，讓網路能夠服務更多的使用者。

參考文獻

[1] QoS, http://www.objs.com/survey/QoS.htm.

[2] Intel,"Multi-Hop Mesh Network", http://www.intel.com/technology/comms/cn02032.htm.

[3] Nortel,"Wireless Mesh Network Solution", http://www.nortelnetworks.com/solutions/wrlsmesh/index.html.

[4] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin,"Resource ReSerVation Protocol (RSVP)", IETF RFC-2205, September 1997.

[5] Intel,"Meeting the Demands of the Digital Home with High-Speed Multi-Hop Wireless Networks", ftp://download.intel.com/technology/itj/2002/volume06issue04/art06_wireless/vol6iss4_art06.pdf, November 2002.

[6] Intel,"Netting more net with wireless mesh", http://www.intel.com/employee/retiree/circuit/wirelessmesh.htm

[7] IEEE Working Group, "IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems", IEEE Std 802.16-2004, October 2004

[8] K. Kowalik and M. Collier,"Should QoS routing algorithms prefer shortest paths?", IEEE International Conference on Communications, IEEE - ICC, May 2003.

[9] A. Shaikh, J. Rexford, and K. S. Shin, "Evaluation the impact of stale link state on quality-of-service routing," IEEE/ACM Transactions on Networking, April 2001.

[10] A. Kamath, O. Palmon, and S. A. Plotkin, "Routing and Admission Control in General Topology Networks with Poisson Arrivals", SODA: ACM-SIAM Symposium on Discrete Algorithms, 1996.

[11] R. Gawlick, A. Kamath, S. Plotkin and K. Ramakrishann, "Routing and Admission

Control in General Topology Networks", Technical Report STAN-CS-TR-95-1548, 1995.

[12] Q. Ma and P. Steenkiste,"On Path Selection for Traffic with Bandwidth Guarantees", In Processings of IEEE International Conference on Network Protocols, October 1997.

[13] M.-C. Yuen, Weijia Jia, C.-C. Cheung,"Efficient Path Selection for QoS Routing in Load Balancing", The 9th Asia-Pacific Conference on Communications, pp988–992, Sept. 2003.

[14] G. Apostolopoulos, S. Kama, D. Williams, R. Guerin, Orda A., Przygienda T.,"QoS Routing Mechanism and OSPF Extensions", RFC 2676, August 1999.

[15] Wang Z., Crowcroft J.,"Quality-of-Service for Supporting Multimedia Applications", IEEE JSAC, vol. 14, issue 7, pp.1288-1234, Sept 1996.

[16] C.-F. Huang, H.-W. Lee, and Y.-C. Tseng,"A Two-Tier Heterogeneous Mobile Ad Hoc Network Architecture and Its Load-Balance Routing Problem", ACM Mobile Networking and Applications (MONET), Special Issue on Integration of Heterogeneous Wireless Technologies, Vol. 9, pp. 379-391, 2004.