私立東海大學資訊工程與科學研究所

碩士論文

指導教授：呂芳懌

以模糊理論為基礎的網路故障推論系統

A Network Fault Deduction System Based on Fuzzy Theory

研究生：歐建暉

中華民國九十三年七月

致謝

在職的研究生涯能夠順利取得學位，在這過程中得到的許多幫助與關心都是我畢生難忘的。

首先要感激是呂芳懌老師耐心的指導，從老師身上不但讓我學習到學術研究的精神與方法，在處事上嚴謹的態度也讓我獲益良多，如果沒有呂老師的教誨，此論文是無法順利完成的。

特別感謝我太太幸君及家人在這段期間對我百分百的支持及無怨的付出，讓我無後顧之憂，能夠專心的學習及研究並有此成果，最後的這一年特別要感謝陳玲如醫生的協助，不斷的給我打氣讓我在繁忙及緊張的生活中仍能繼續堅持下去。

最後要感謝學弟士傑全力的協助，若沒有你的努力，我的實驗是沒有辦法完成的，還要謝謝同學維埠在技術及設備上的協助、實驗室的同學義樺、政瑋、學弟妹們及所有幫助過我的人，謝謝大家。

摘要

網路技術的蓬勃發展使得網路的使用非常普遍，而網路設備的更新速度也越來越快，網路也日益複雜及龐大，設備的增加代表網路管理人員工作量的加重，爲了使網管人員可以有效掌握網路現況，如何有效了解網路問題的症狀並適時提出警訊，以提高網管軟體的便利性及多功能性就顯得非常重要。

本論文以網管系統中最普遍的 SNMP（Simple Network Management Protocol）爲基礎，經由自行開發的網路管理主機蒐集各式的網路管理資訊，並透過案例比對及模糊推論的方式，來分析網路上被管理物件可能產生的徵兆，藉此推論出網路已發生或可能發生的錯誤，成爲一個主動式的網路監視架構，使網管人員可即時瞭解網路問題的發生處，並幫助網管人員作相對應的處理機制。

本研究中的主要目標是在模糊理論爲基礎所建構的模糊推論系統（Fuzzy Inference System）來推論網路問題。網路管理人員通常他們是憑藉著以往的經驗以及專業知識，才能在複雜的警訊中找出真正的關鍵警訊，再加上網路不斷擴大，架構越趨複雜化的狀況甚至缺乏網路管理資訊的狀況下，要精確的找出網路問題是有一定程度的困難存在，在不容易掌握的複雜系統下，我們嘗試以模糊理論來解決網路上具模糊特質的事物，希望在問題資訊不完整的狀況下也能推論出最有可能的問題，並將推論時所用的知識，能夠經由自我學習的方式來擴充，是本研究希望在網路錯誤的診斷上達到的功能與目標。

本研究期望建立網路的監控模式外，並能結合案例比對的方式與模糊理論爲基礎的推論機制，整合成一個具有主動式管理且能自我成長的網路管理系統，對網路錯誤的判斷與問題排除提供一個新的處理模式。

章節目錄

圖示目錄

表格目錄

第一章、　緒論

1.1、 研究背景與動機

網路系統就如同企業體一般，有一定的運行法則及規範，一個網路系統如果沒有正確的管理制度及方法，將會引發層出不窮的問題，不只功能不彰，使用效率亦將轉趨低落。網路管理最重要的是必須維持網路的順利運作，管理人員必須充分掌握構成網路的各種元件與彼此間的運作關係，一但問題發生時，管理人員必須利用網路管理工具及網路管理經驗，迅速且正確的查出是哪些網路裝置發生問題，並確實排除。而隨著網路應用的日益發達及電子商務的快速發展，網路服務的持續性、可靠性及品質將日趨重要，不當的網路管理及不穩定的網路服務將對公司營運造成極大的損害。

過去的網路管理以硬體爲主，重點僅局限於網路設備的安裝及設定、設備相連結的狀態、系統的正常運作及資源的共享等方面。隨著網路 e 世代的風行，企業間電子商務與企業對消費者的電子商務市場急遽成長，網路流量更隨之暴增，加上病毒與電腦蠕蟲的影響，網路的服務品質也越來越受到重視。而隨著寬頻網路的蓬勃發展，網路設備的處理速度大幅提昇，也連帶加速所提供服務的複雜化。如何佈建完善而安全的網路環境，高效的網路品質管理，提升服務的效能都是企業網路管理人員被賦予的重責大任。

現今網路管理的五大議題:設定管理 （Configuration Management），效能管理 （Performance Management），安全管理 （Security Management），會計管理 （Account Management），錯誤管理 （Fault Management），再加上服務管理 （Service Management）。這六大議題環環相扣，要做到良好的網路管理六者缺一而不可，本研究係以設定管理、效益管理及錯誤管理爲範籌，而以使用者能夠順利順暢地使用網路爲宗旨，並探究管理人員平日從事網路管理所必須監控的資訊、掌控的方式及其網路瓶頸或錯誤發生時該採取的相關應變措施。

對網路管理人員而言，若是沒有全面監控重要的網路設備與了解設備間的運作狀況，再加上使用者若發現有網路問題卻表達不清、使用者不了解設備的各項設定之意義及程序等，而無法遠端的確認，甚至網路管理人員若是經驗不足則也無法正確判斷網路問題的成因，一般的做法是反覆測試及尋找問題，直到正確地找出問題點，並解決問題爲止。依據統計當網路出現問題時，網路管理人員通常需要花費 70%的時間來確定網路問題發生的原因與地點，另外 30%的時間爲真正處理與解決網路問題。若是沒有規劃一個妥善的網路錯誤管理方式與問題回報途徑，網路管理者通常會是最後一個才知道網路已經發生問題的人員。

本研究計畫的目的在於全面性監控與管理一個區域網路，並在網路問題發生時能以模糊推論方式自動推導出所發生的網路瓶頸或錯誤，且能依據所推論的問題建議管理者如何修復，再以 Qos 的角度建議網路管理者如何改善或是自動調整路由器設定，達到一個可以偵測、修復問題及建議的網路的專家系統，提供給初階管理人員及一般網路管理人員，使其能在最短時間內主動的發現問題並解決問題，進而改善邏輯或實體上的架構。本計畫共包含了，四大部份

1.網路設備的監控

2.網路問題的推論

3.網路問題回復的方式

4.Qos 的掌控及動態調整

網路設備的監控與 Qos 的掌控與動態調整是由另一位研究生邱維埠同學(已於 91 年畢業)所負責，網路問題的推論與網路問題回復建議則是在本文中討論。

由於網路越來越複雜化，所使用的設備也越趨精密，所提供的網路服務也日趨多樣化，網路問題也通常以不明確的形式出現，甚至同樣的網路錯誤會因網路拓樸的不同、網路延遲及網管資料遺失而有不同現象發生，如傳輸速度過慢、服務主機效率不佳、網路封包無故遺失等狀況均有

可能是因爲不同的原因而導致。在此前提下，一個網路管理人員若希望能在最短時間內了解所有問題與設備的關聯性，網路問題發生時希望能在最短時間確認時間與地點，很有經驗的管理人員在實際的執行面上也是有其困難的，常常是在花了數倍的時間與精力之下卻也不一定能完全解決問題。

而模糊理論是當初是爲了掌握定義數據模糊特質的事務而發展出來的，所以它的應用較偏重於人類的經驗及對問題特性的掌握程度，不主張用繁雜的數學分析及數學模式來解決問題。Zadeh 教授曾提出一個互斥性原理：”當系統的複雜性增長時，對系統特性做精確描述的能力亦相對降低，最後精確性與有意義性將變成完全互相排斥的特性”，這充分的說明”當複雜程度越高，有意義的精確化能力便越低”，很高的複雜性意味著只能在一個刻意壓縮的低維因素空間觀察，而無法對問題的全部因素都進行考量，使得本來是明確的觀念變的模糊。而網路問題正符合這個特性，所以採用模糊理論及模糊推論方法來嘗試解決網路問題。

本文旨在以模糊理論爲基礎所建構的模糊推論系統（Fuzzy Inference System）來推論網路問題。模糊推論系統簡單的說，是一個「以模糊知識爲依據，以模糊推理爲方法的智慧型系統」，其主要核心是由知識庫（Knowledge Base）和推論引擎（Inference Engine）所構成。本文將以一連串的實驗及所得數據來分析網路頻寬、網路流量及主機效能三者對使用者在使用者網路上的影響，並以不同的方式製造網路問題，再嘗試以模糊專家系統來推論網路問題，推論正確則提出明確的網路問題與解決方法，並檢核所提出的方式是否能正確的修復網路問題，推論的結果若有錯誤時則依據正確的結果修正推論的參考數據及推論規則。讓網路管理人員能以比較清晰的概念，面對日益複雜的網路環境，除了隨時有效地掌握網路服務的品質之外，也能針對現有網路資源做有效地運用及管理，使其運轉得以最佳化。一方面在網路發生瓶頸及問題時，也可以在最短時間內地找出其可能的發生處。並獲得排除該瓶頸或故障的有效資訊，進而提供網路資源

擴充、更換或升級時的參考。

1.2、 研究範圍與目的

本研究之範圍與目的包括下列五大項

1.網路流量監控與實驗網路環境的建置

a.分析現有網際網路服務系統之特性

b.建置網路流量監控模式

c.建構網路資料庫及計算平台

d.建置產生網路流量之平台

2.蒐集資訊的內容及方式

a.確認受監控網路之元件，包括網路頻寬，網路設備效能及主機效能等。

b. 建置蒐集網路監控數據之相關軟硬體設備。

3.警訊分類及其歸屬函數的定義

a.警訊分類

b.歸屬函數的定義

4.利用模糊方法推論網路問題

a.警訊值模糊化

b.模糊規則庫

c.模糊語意資料庫

d.模糊推論

e.解模糊化

5.解決網路問題，專家系統並能自我成長

a.在模擬網路狀況過程中,測試及分析各項變數對網路所造成的影響。

b.在解決網路問題後，確認的資料能回饋資料庫，提供後續案例作爲資料判斷。

1.3、 論文章節概要

本文章的架構是：第二章介紹網路事件與錯誤的意義、模糊集合理論與模糊專家系統，第三章介紹研究的架構與流程，網路錯誤的資料來源與資料的正規化及如何將資料轉化成知識的表示方式，並定義出模糊知識庫與模糊規則庫，在實驗中的案例比對方法及利用模糊專家系統完整的推論及如何反模糊化得到推論的結果，最後則是案例的回饋與資料庫的參考數據修正，以達到本系統自我成長的目標。第四章則是依據第三章的方法設計實驗，描述整體的運作流程確認實驗步驟並驗證其結果，第五章爲結論及未來的發展方向。

第二章、 文獻探討與理論基礎

網路管理系統的建置重點爲警訊關聯規則，即網路錯誤知識庫，當網路發生問題時，所伴隨發生的網路事件及警訊即爲判斷網路錯誤的最佳來源，網路錯誤所建立的行爲知識與適當的問題推論系統，是構成一套良好的網路管理系統不可或缺的要素，本研究也希望以模糊理論及推論方法，來解決使用精確的比對及推論方式的網路錯誤推論系統誤判率偏高的問題，因此我們將更進一步去了解網路事件及警訊與錯誤的主從關係及定義、模糊集合理論與模糊推論的相關研究。

2.1 網路事件、警訊與錯誤

網路事件、警訊與網路錯誤的定義

警訊是網路管理中錯誤偵測的重要資料依據，A. Bouloutas 在 1990 年提出警訊的結構”警訊可以提供網路狀態資訊給網路管理中心，因此警訊定義的結構與內容是決定錯誤種類的重要依據”。Robert【ROBE97】提出對警訊的定義”當網路元件正處於一個特殊或是不正常的現象時所發出的訊息”。Jakobson 在【JAKO93】中指出”當管理元件中的軟體或硬體不當的使用發生錯誤時，此時警訊則是一種外在的表現”。

一般網路管理者則藉經由網路管理協定所提供的 MIB(Management Information Base)來偵測警訊，因爲即使是不同廠商所製造的網路設備，只有利用相同的網路管理協定就可以表現出相同的警訊形式，可以避免不同網路設備警訊格式亦不同的缺點。若利用網路通訊協定中的 MIB 當成警訊時，可以把單一個 MIB 代表一種警訊，也可以把數個 MIB 集合起來當成一種警訊。

瞭解警訊定義後，若要利用在網路上蒐集到的警訊幫助網路管理者找出發生什麼網路錯誤與發生地點的話，以達到自動化警訊關係推導的目的的話，警訊則必須帶有足夠的資訊以供分析。Nielsen【BOUL94】指出最佳化的警訊應夾帶下列五項資訊：(1)who：發佈警訊的系統名稱；(2)what：在何種情形下發佈；(3)where：從網路何處發出；(4)when：何時發佈該警訊；(5)why：發佈該警訊的原因。一個標準的警訊中包含的資料最少要有以上五項才是，因爲這些資料都是判斷錯誤的重要依據。另一方面也有人從網路”健康”的觀點【CONC97】，勾勒

網路警訊的定義：當網路資源的健康指數(health Index)超過固定臨界值之後，系統應該發出警訊通知，同時應將導致網路不健康的指數伴隨警訊一併到達管理端，以利會診。

在警訊推理系統中，若把每一個警訊看成是單一的事件(Signal Event)，每一個事件都是獨立的話，那必不用去考慮事件混和(Composite Events)的問題，則警訊關係較易描述。然而在網路環境中，事件並不單純，而是複雜的，是互相交錯結合的，事實上，當網路發生一個錯誤時，可能同時會產生數個，甚至數十個警訊，也就是說當同一個時間網管人員收集到數十個警訊時，有可能這些警訊指代表同一件事情。

MIB 的分類

我們將在本研究中，於統計資料裡挑選出在網路行爲異常現象有分析價值的MIB，這些 MIB 被稱之爲事件(Event)，事件跟據其本身的特性可分成 3 種的型態，分別爲數量式(Magnitude)、觸發式(Trigger)及字串字(String)。在本研究中會使用到數種 MIB 來做爲分析網路錯誤行爲的資料來源，介紹如下：

1、MIB 2：

MIB 2 爲 IETF RFC-1213 所定義，其中包含了十個物件群組（Object Group）功能如下：

a. system：整體有關訊息。
b. interface：介面有關的資訊。
c. at：位址轉換相關訊息。
d. ip：實行與執行 IP（Internet Protocol）之相關資訊。
e. icmp：實行與執行 ICMP（Internet Control Message Protocol）之相關資訊。
f. tcp：實行與執行 TCP（Transmission Control Protocol）之相關資訊。
g. udp：實行與執行 UDP（User Datagram Protocol）之相關資訊。
h. egp：實行與執行 EGP（External Gateway Protocol）之相關資訊。
i. transmission：於介面上之轉換與存取架構之資訊。
j. snmp：實行與執行 SNMP 之相關資訊。

2、RMON1 爲 IETF RFC-1757 所定義，也有十個物件群組，功能如下：

a. statistics：代理者之負荷及錯誤的統計。

b. history：對於 statistics 群所做的區段統計。
c. alarm：可讓網管人員設定臨界值，並紀錄在 RMON Probe 中。
d. host：紀錄網段（Subnet）上流經主機的各類型封包。
e. hostTopN：於網段上根據特定流量之主機排名。
f. matrix：以矩陣方式列出對應關係兩位址之錯誤與負載值。
g. filter：依某些設定條件觀察某特定封包。
h. capture：可擷取而送至管理者的封包。
i. event：RMON Probe 所產生事件之紀錄表格。
j. tokenRing：於 Token Ring 子網路上統計與設定之資訊。

3、802.1d Bridge MIB 爲 IETF RFC-1493 所定義，共有五個物件群組：
a. dot1dBase：802.1d 基本資訊。
b. dot1dstp：802.1d Spanning Tree Protocol 相關資訊。
c. dot1dSr：802.1d Source Route Bridging 相關資訊。
d. dot1dTp：802.1d Transparent Bridging 相關資訊。
e. dot1dStatic：根據目的位址所設定的過濾條件之相關資訊。

4、Cisco Proprietary MIB
a、Local MIB，共有八個物件群組，功能如下：
(a). flash：記錄 flash 記憶體相關資訊。
(b). interface：記錄設備介面相關資訊。
(c). ip：記錄設備執行 IP 封包相關資訊。
(d). ip accounting：記錄設備 IP 封包統計之即時資訊。
(e). ip checkpoint accounting：記錄設備 IP 封包統計之累計資訊。
(f). system：記錄設備整體有關之資訊。
(g). terminal service：記錄設備提供終端服務之相關資訊。
(h).TCP：提供 TCP 連線 IN/OUT 流量之統計資訊。

2.2 模糊集合理論

2.2.1 模糊理論的起源

Fuzzy 理論是爲解決真實世界中存在的模糊現象而發展的一門學問，它是美國自動控制學家美國加州柏克萊分校的 Lotfi.A.Zadeh 於 1965 年首先在他主編的「Information and Control」刊物上提出的一種定量表達工具，用來表現某些無法明確定義的模糊概念，尤其是在表現人類語言特有的模糊性現象方面有頗佳的成果。

Fuzzy 理論已經在人工智慧、自動控制、圖像識別、醫療診斷、心理學、決策支援、管理科學、氣象預報、環境評估等各種領域的應用有豐碩的成果。但是，比較廣爲人知的應用還是在自動控制這個領域。1970 年代中期，模糊控制研究已經有相當多的成功實例，1980 年代開始實用化、商品化。最近，模糊理論更與類神經網路、知識工程等互相結合，達成電腦科技很多新的突破，對於新一代電腦的研究及發展助益頗多。

我們熟知而且長期相處的真實世界，普遍存在著各種模糊性現象，本身所表現出來的特徵比二元邏輯（binary logic）的電腦世界更複雜。一些常見的模糊性現象說明如下：

1. 不完整（incomplete）
2. 曖昧性（ambiguity）
3. 不精確性（imprecision）
4. 隨機性（randomness）
5. 模糊性（fuzziness）

以下數點是 Zadeh 提出模糊理論的理由：

1. 現代科學技術的研究對象，往往都是非常巨大的物體或機械，像這種大規模又複雜的系統，若想正確地、精密地掌握大局，就必須從物體的細部做起，它必須非常準確且不允許有些微的錯誤產生。所以更需要有一個更好的理論才行。
2. 過去科學技術的成長，完全取決於它有一個明確的數學定義，若這些研究的對象中，無法了解其數學性質的話，那麼以過去這種科學技術的研究，將顯得束手無策被迫停止。Zadeh 教授認爲無法用數學模式構築的系統，模糊理論將顯得更重要。
3. 人行的知識可說是用語言來表達的，而語言中存在的模糊性，特別是因人而異所產生的主觀性也各有不同。這些模糊性現象無法使用傳統的數學工具，例如，機率等解決，必須尋找另外的替代途徑。

模糊理論正是爲了掌握這類定義具模糊特質的事物而發展出來的，所以它的

應用較偏重於人類的經驗，及對問題特性的掌握程度，不主張用繁雜的數學分析及數學模式來解決問題。模糊理把傳統數學從二元邏輯擴展到連續多值（continuous multi-value），利用歸屬函數（membership function）描述一個概念的特質，而以０和１之間的數值來表示一個元素屬於某一概念的程度，這個值稱爲元素對集合的歸屬度（membership grade）。

模糊理論是以模糊集合(fuzzy set)爲基礎，其基本精神是接受模糊性現象存在的事實，而以處理模糊不確定的事物爲其研究目標，並積極的將其嚴密的量化成電腦可以處理的訊息。

2.2.2 歸屬函數

就是集合論中的特徵函數，雖然模糊集合的範圍（邊界）是模糊不明確的，但是利用歸屬函數在【0,1】封閉區間中取值倒是十分的明確。歸屬函數的縱軸代表函數在封閉區間內的取值，也就是模糊集合中該元素隸屬於此模糊集的程度大小。這在模糊集理論中是一個非常重要的基礎，歸屬函數能讓專家們有最大的彈性來定義他們心中的概念，它比普通集合更客觀化。

歸屬函數因爲要描述模糊量化後的程度大小，雖然有相當的主觀性，不過也可以經由客觀的統計方法來描述眾人或多次發生後的取值規律性，因此也有下述常用的幾種常用形態：

１. 梯形

梯形歸屬函數可以分成三類，其數學描述如下：

(1).左邊梯形－數學描述如式子(1)，外形如圖 2.1 所示

$$\mu_A(x)=\begin{cases}1 & x \le a,\\ \dfrac{b-x}{b-a} & a < x \le b,\\ 0 & x > b.\end{cases} \quad \text{----------(1)}$$

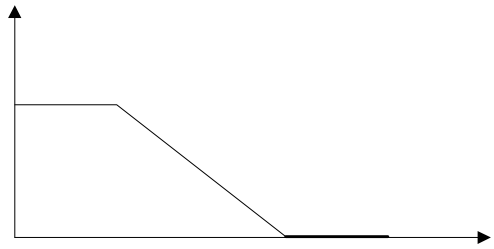

圖 2.1：左邊梯形歸屬函數

(2).右邊梯形－數學描述如式子(2)，外形如圖 2.2 所示

$$\mu_A(x)=\begin{cases}0 & x<a,\\ \dfrac{x-a}{b-a} & 0\le x<b,\\ 1 & x\ge b.\end{cases} \quad \text{----------(2)}$$

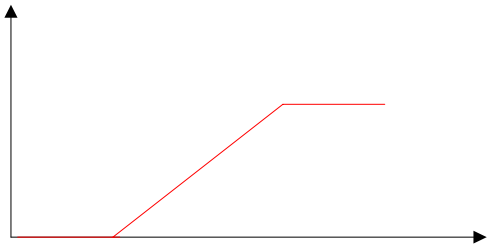

圖 2.2：右邊梯形歸屬函數

(3).中間梯形－數學描述如式子(3)，外形如圖 2.3 所示

$$\mu_A(x)=\begin{cases}0 & x<a,\\ \dfrac{x-a}{b-a} & 0\le x<b,\\ 1 & b\le x<c\ ,\\ \dfrac{a-x}{d-c} & c\le x\le d,\\ 0 & x>d.\end{cases} \quad \text{----------(3)}$$

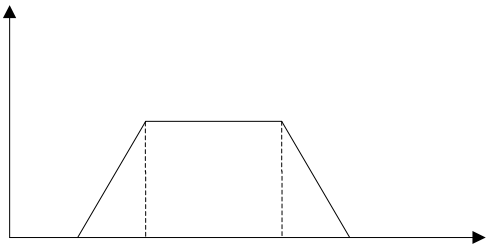

圖 2.3：中間梯形歸屬函數

2. 常態分佈形

常態分佈形歸屬函數的數學描述爲式子(4)，外形如圖 2.4 所示。

$$\mu_A(x) = e^{-k(x-a)^2}, k > 0 \quad \text{---------(4)}$$

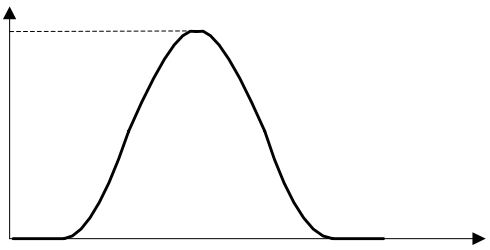

圖 2.4：常態分佈型歸屬函數

3. S 型（S-Shape）歸屬函數：

S 型歸屬函數的數學描述爲式子(5)，外形如圖 2.5 所示。

$$S(x;l,r) = \begin{cases} 0, & \text{for } x \le l \\ 2\left(\dfrac{x-l}{r-l}\right)^2, & \text{for } l \le x \le \dfrac{l+r}{2} \\ 1-2\left(\dfrac{r-x}{r-l}\right)^2, & \text{for } \dfrac{l+r}{2} \le x \le r \\ 1, & \text{for } x \ge r \end{cases} \quad \text{----------(5)}$$

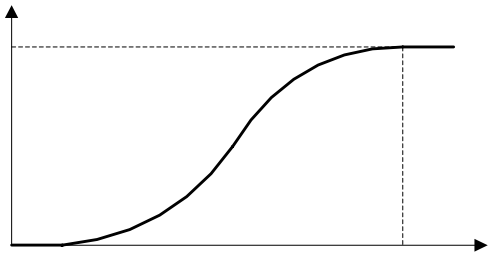

圖 2.5：S 型（S-Shape）歸屬函數

S 型歸屬函數常用於表述諸如重、厚、長、大、熱、胖、高、好、多、昂貴、年老 ... 等代表較高數量級的模糊概念，*l* 和 *r* 分別代表其左、右端點。

4. Z 型（Z-Shape）歸屬函數：

Z 型歸屬函數的數學描述爲式子(6)，外形如圖 2.6 所示。

$$Z(x;l,r)=\begin{cases}1, & \text{for } x\le l\\ 1-2\left(\dfrac{x-l}{r-l}\right)^2, & \text{for } l\le x\le \dfrac{l+r}{2}\\ 2\left(\dfrac{r-x}{r-l}\right)^2, & \text{for } \dfrac{l+r}{2}\le x\le r\\ 0, & \text{for } x\ge r\end{cases} \quad \text{----------(6)}$$

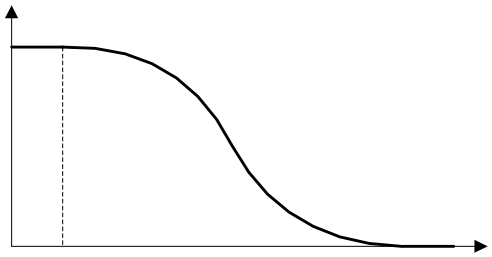

圖 2.6：Z 型（Z-Shape）歸屬函數

顯然地，*Z*(*x*; *l*, *r*) = 1-*S*(*x*; *l*, *r*)，兩者互爲模糊補集。Z 型歸屬函數常用於表述諸如輕、薄、短、小、冷、瘦、矮、壞、少、便宜、年輕 ... 等

代表較低數量級的模糊概念，*l* 和 *r* 分別代表其左、右端點。

Zadeh 認爲某一個元素屬於某集合的程度愈大，則其歸屬度愈接近１，否則愈接近０。

然而，歸屬函數的訂定完全因人而異，當然也可以藉由統計方法獲得，或是請教專家【林信成、彭啓峰，1994】。不論是用何種方式訂定歸屬函數，唯一的原則是要合情合理，才能被接受。

2.2.3 模糊集合與模糊運算

事實上模糊集合的觀念是由傳統的明確集合(crisp sets)所延伸，它是傳統集合的擴大，集合論中的許多關係運算等性質也就自然地推廣到模糊集來。假設 A、B、C、D 爲論域上的四個模糊子集，我們可以定義下述的基本關係：

1. 相等

   若 A=B，則 $\forall_x \in X, \exists_{\mu A}(x) = {}_{\mu B}(x).$

2. 包含

   若 $A \subseteq B$，則 $\forall_x \in X, \exists_{\mu A}(x) \le {}_{\mu B}(x).$

3. 聯集(union)

   若 $C=A \cup B$，則 $\forall_x \in X, \exists_{\mu C}(x) = \max[\mu_A(x), \mu_B(x)]$

4. 交集(intersection)

   若 $C=A \cap B$，則 $\forall_x \in X, \exists_{\mu C}(x) = \min[\mu_A(x), \mu_B(x)]$

5. 補集(complement)

   若 $\overline{A}$ 爲 A 的補集，則 $\forall_x \in X, \exists_{\mu C}(\overline{x}) = 1 - \mu_A(x).$

模糊集合的運算亦有下述的特質：

1. 交換律

$$A \cup B = B \cup A, A \cap B = B \cap A$$

2. 結合律

$$A \cup (B \cup C) = (A \cup B) \cup C, A \cap (B \cap C) = (A \cap B) \cap C$$

3. 分配律

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$
$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$

4. 雙重否定律

$$\overline{\overline{A}} = A$$

5. 德摩根定律

$$\overline{A \cup B} = A \cap B, \overline{A \cap B} = A \cup B$$

但 $A \cup \overline{A} \neq U, A \cap \overline{A} \neq \varnothing$。也就是說集合論中的排中律和矛盾律在模糊集合中是不成立的，這也是模糊集合和普通集合除了描述方法外的另一重要區別，這是起源於模糊集合不確定的邊界所引起的。

$$\mu_A(x) \vee \mu_{\overline{A}}(x) = \mu_A(x) \vee (1 - \mu_{\overline{A}}(x)) \neq 1$$
$$\mu_A(x) \wedge \mu_{\overline{A}}(x) = \mu_A(x) \wedge (1 - \mu_{\overline{A}}(x)) \neq 0$$

2.2.5 解模糊化的方法

解模糊化(defuzzification)的目的，是將規推論的口語結論轉回到數值。大多數的模糊邏輯系統是需要這個步驟，因爲期望的輸出值通常是數值而非口語。

就像有不同的歸屬函數樣式一樣，對於解模糊化也有許多不同的力法。選擇正確的解模糊化方法，需要瞭解解模糊過程中口語上的意義。一般而言，解模糊化的兩種不同語言意義。

(1). 決定最佳的折衷值

(2). 決定最合理的結果

在實際的應用裡是很重要的而解模糊化的方法包括：

A.最大中心法

最大中心法(Center of Maximum)，(CoM)，首先要決定每種術語的最典型值，然後再計算模糊邏輯推論最佳折衷值。

B.面積中心法

“面積中心法(Center-of-Area)，(CoA)”，有時也稱爲“重心法(Center-of -gravity)”，這種方法首先是要在個別術語合法的情形下，切割該歸屬函數圖，如些在所有術語的結論函數之下區域就會重疊，平衡了這些結論區域就可以獲得折衷值。

C.最大平均法

最大平均法(Mean-of-Maximum)反模糊方法的變體，它們是利用歸屬函數的最典型值來計算。

2.3 模糊規則爲基礎的專家系統

模糊控制器的組成大致可分爲模糊化(fuzzification)、知識庫、推理引擎及解模糊化等四部分，模糊控制系統之基本架構如圖 2.7 所示。

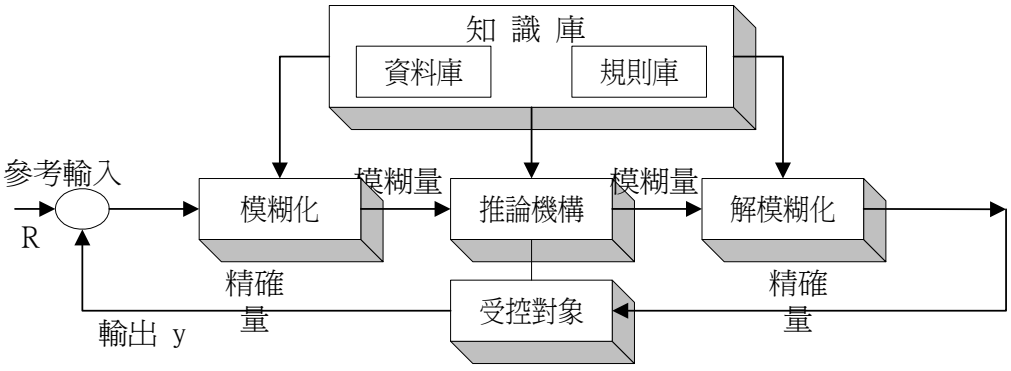

圖 2.7：模糊控制系統之基本架構圖

2.3.1.模糊化

模糊化的設計是將受控對象輸出之精確信號轉換成相對應的模糊量，以供推

論引擎作爲推論之依據。因此，必須定義一個由輸入空間(input space)到某個論域中之模糊集合的映射關係。針對每一個輸入變數劃分爲若干個語言項，並定義每個語言項所對應之模糊集合，而這些模糊集合是以歸屬函數來表示。

2.3.2.知識庫

知識庫是由資料庫(data base)和規則庫(rule base)兩個部份所組成，其主要功能提供依據專家知識或操作人員的經驗而制定之推論規則及相關控制資料，以便提供推論引擎作推論之用。茲將資料庫和規則庫之功能分述如後：

(1)模糊語意資料庫

在資料庫中存放著輸出入變數之語言變量及定義各語言變量的歸屬函數，這些語言變量及其歸屬函數分別提供模糊化介面、解模糊化介面及模糊推論引擎使用，以作爲各模糊變量定義之依據。

語言變量的選擇是依據控制系統的實際需要而制定的，各語言變量的語意也盡可能接近人類的思考方式而訂定

當各輸出入之語言變量選定後，就必須定義每一個語言變量的歸屬函數。歸屬函數形狀的選擇是依據專家的知識和經驗及控制器之控制效果而決定的，比較常用之歸屬函數圖形是三角形(triangle-shaped)、吊鐘形(bell-shaped)等。

(2) 模糊規則庫

規則庫中存放的是模糊推論引擎進行推論時所需要之推論規則，推論規則的型式則是仿效人類的思考推理方式，每個推論規則皆以「若（原因或條件）則動作」(IF condition THEN action)來表示。

$$R_i : IF\ x\ is\ A\ and\ y\ is\ B\ then\ z\ is\ C$$

其中　x,y　是模糊控制器輸入之語言變數

　　z　　是模糊控制器輸出之語言變數

　　A_i B_i C_i爲x, y, z在論域U,V,W中的語言值

一般而言，模糊推論規則可由下列方式獲得：

1.依據專家的知識和經驗。

2.觀察操作員的控制行爲。

3.根據自我學習演算法來調整推論規則。

2.3.3.模糊推論引擎

模糊推論引擎的功能是以模糊化的語言變量搜尋規則集中之適當的規則，加以推論得到所需的結果，俾以進行解模糊化，所以資料庫、規則庫與推論引擎可建立一個控制法則表。假設誤差爲 E 而誤差變動量爲 ΔE，則輸出變動量爲 ΔU。

模糊推論引擎是模糊控制器的核心，推論的方法一般是採用以推論合成規則爲基礎的方法，又稱爲近似推論(approximate reasoning)。若以馬丹尼教授所提出的模糊蘊涵最小運算規則(Mamdani’ s minimum operation rule as a fuzzy implication function)推論方式爲例，茲將模糊推論之過程敘述如：

假設有兩個規則如下所示：

規則 1 ： *IF x is A_1 and y is B_1 then z is C_1*

規則 2 ： *IF x is A_2 and y is B_2 then z is C_2*

假設 x_0,y_0 是由受控對象輸出所取得之精確量，則此精確量經過模糊化過程之後，可以轉換爲相對應之模糊集合，其歸屬函數分別爲 $\mu_{A_i}(x_0)$ 及 $\mu_{B_i}(y_0)$ ，其中 A_i及B_i 分別爲模糊控制器之輸入量 x_0及y_0 所對應之模糊集合。若規則 1 及規則 2 的觸發強度(firing strength)分別爲 α_1 及α_2 ，則 α_1 及α_2 可表示如下：

$$\alpha_1 = \mu_{A_1}(x_0) \wedge \mu_{B_1}(y_0)$$

$$\alpha_2 = \mu_{A_2}(x_0) \wedge \mu_{B_2}(y_0)$$

則每一條規則所得到的結論之實際輸出模糊集合分別以 $C_1^{'}$及$C_2^{'}$ 表示，則其歸屬函數 $\mu_{C_1^{'}}(z)$及$\mu_{C_2^{'}}(z)$ 可由下式求得：

$$\mu_{C_1^{'}}(z) = \alpha_1 \wedge \mu_{C_1}(z) \qquad \text{規則 1}$$

$$\mu_{C_2'}(z) = \alpha_2 \wedge \mu_{C_2}(z)$$ 規則 2

最後將每一條規則所得到的結論C_1'及C_2'聯集，即可獲得所有規則結論之綜合結果C'，其歸屬函數爲$\mu_{C'}(z)$，即

$$\mu_{C'}(z) = \mu_{C_1'}(z) \vee \mu_{C_2'}(z)$$

$$= [\alpha_1 \wedge \mu_{C_1}(z)] \vee [\alpha_2 \wedge \mu_{C_2}(z)]$$

2.3.4.解模糊化

由於模糊推論引擎推論所得之結果爲模糊量，而控制受控對象之控制信號必須是精確量，故推論所得之結果必須經過解模糊化過程，使之轉換爲一精確之控制信號以便控制受控對象。

第三章、研究架構與研究方法

網路發生問題通常是出自硬體、軟體、頻寬達到瓶頸或是網路架構的不協調所造成，網路錯誤可以透過網管資訊加以偵測，藉由警訊的發佈，網路管理人員才能掌握網路錯誤的狀況，通常他們是憑藉著以往的經驗以及專業知識，並輔以各種軟、硬體工具，才能在複雜的警訊中找出真正的關鍵警訊，並依此確認網路問題進而解決問題。但在網路不斷擴大，架構越趨複雜化的狀況下，要快速且正確的解決網路錯誤，藉由人工化的處理方式就顯得相當沒有效率了，因此網路問題判斷的自動化就越顯其重要。

本研究希望能將複雜的警示資訊的發佈與網路問題的診斷程序予以自動化，也就是將意義不夠明確的警示資訊轉換成有參考價值的網路故障資訊，本研究希望在網路錯誤的診斷上達到下列的功能與目標：

(1).在問題的資訊不完整的狀況下也能推論出最有可能的問題。

(2).推論時所用的知識，能夠經由自我學習的方式來擴充。

(3).能夠推論出同時間內發生的多個問題。

基於以上描述之功能目標，認爲網路問題推論系統最主要的重點在於如何描述所使用的知識以及設計推論的方法。

最早被用來解決警訊關聯問題的理論是以規則爲基礎的專家系統，Lewis 在 1993 年提到建構一個規則爲基礎的錯誤推論專家系統，茲簡述如下：

(1) 定義一種用來表達網路問題的描述語言。

(2) 從不同領域的專家及問題解決手冊中萃取出所需要的知識。

(3) 將所萃取出的知識表達成規則爲基礎專家系統推論的格式。

爲了處理複雜且不明確的警訊，我們採用模糊推論來判斷網路問題，而模糊規則爲基礎的專家系統就是結合規則爲基礎的專家系統與模糊推論方式而成，不但兼有規則爲基礎的專家系統的及模糊推論的優點，也因爲模糊推論的關係，不似規則爲基礎的專家系統容易被環境及其他因素干擾而受到限制。

本章節在說明模糊規則爲基的專家系統的架構與研究方法，並描述模糊推論的知識表示及推論方法。

3.1 實驗架構與流程

一、實驗架構：

整個論文的實驗架構主要區分爲兩大部分。

1. 實驗架構一爲模擬各式網路狀況並記錄相關數據，以爲後續推論問題使用。首先，我們架設網路模擬及監控環境，並在控制各種網路環境變數下，刻意製造各種網路問題，以區分哪些狀況及組合會造成網路瓶頸或網路錯誤。除了找出當時的各種警訊組合，也尋求正確的解決方法，並將此資料提供給後續的模糊推論機制，以便在網路發出警訊時，推論出正確的網路問題。

圖 3.1：實驗架構一

2. 實驗架構二爲網路問題推論，重複先前實做的網路實驗與問題，並以第一部份實驗所得的數據所建立的案例資料庫及模糊推論引擎來推論，推論的結果若不正確，則交由網路專家進行人爲判斷並修正推論規則或調整規則庫。實驗架構二是以實驗架構一的成果爲推論的法則，重複相同的實驗內容，並以比對及條件推論的方式來判斷網路錯誤的發生，實驗架構二的步驟如下：

a. 在固定時間區段內取得所監控網路設備的相關 MIB 值。
b. 判斷所取得的 MIB 值是否超過該項 MIB 所定義的臨界值。
c. 將超過臨界值 MIB 的集合與案例資料庫中的案例逐一比對，是否有相符的案例發生?
d. 若有符合案例，則進行錯誤回復，若沒有則進行模糊推論。
e. 以模糊推論與案例資料庫中的所有案例比對，推論出最有可能發生的網路錯誤種類。
f. 以推論出的網路錯誤種類進行錯誤回復。
g. 進行錯誤回復時若能解決網路問題，則進行案例回饋，讓系統能夠自我成長。
h. 錯誤回復時若無法解決網路問題，則請網路專家或是網管人員進行人爲處理，處理後並將結果回饋至案例資料庫，請參考圖 3.2：實驗流程圖。

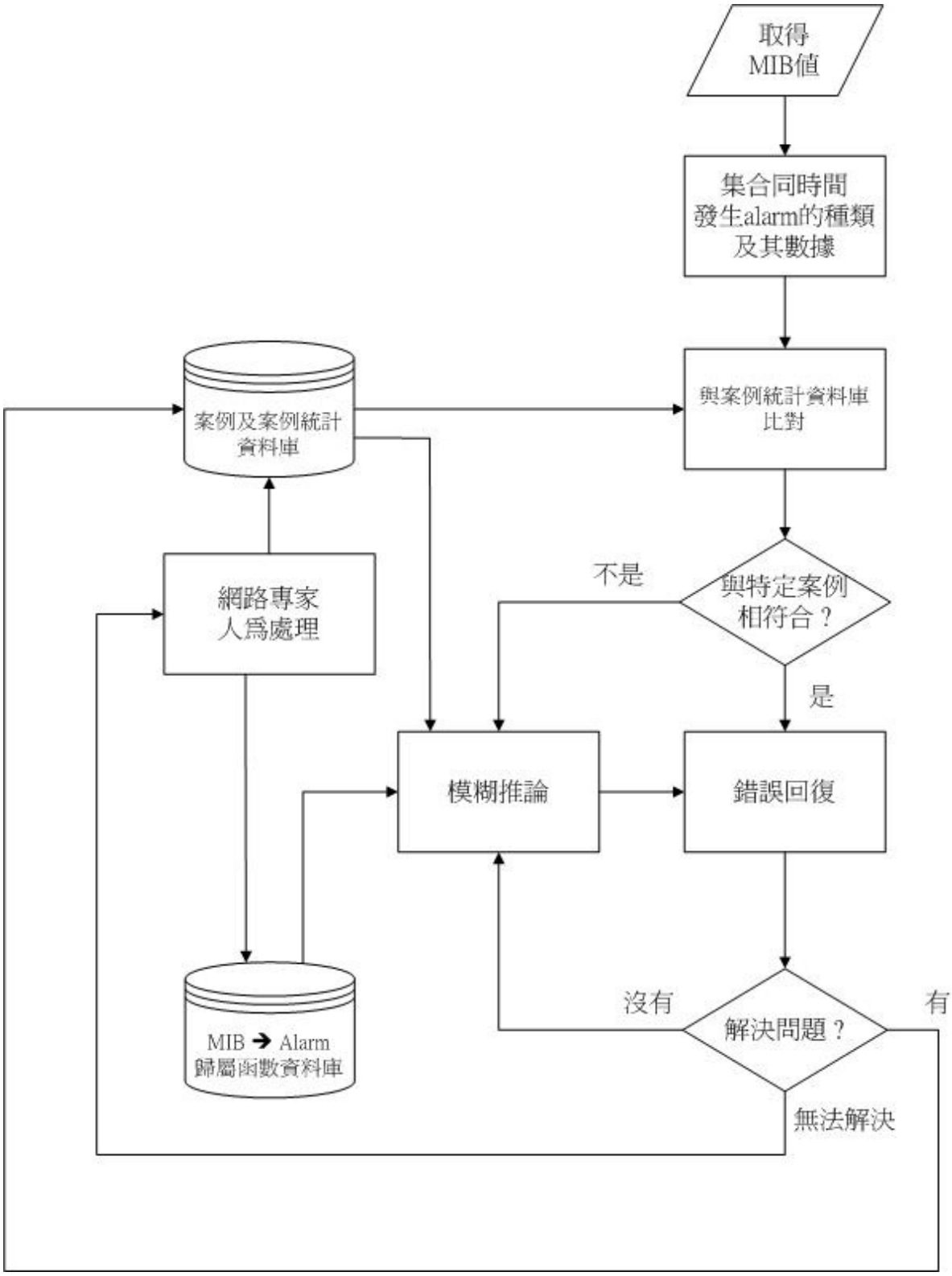

圖 3.2：實驗架構二之流程

二、實驗流程

前一節實驗架構中說明本實驗分爲”模擬各式網路狀況並記錄相關數據”與”網路問題推論”兩大部份，本研究之實驗模式又區分爲封閉型實驗及開放型實驗兩種，整體的實驗流程茲說明如下：

一、封閉型實驗：

首先是以封閉型實驗刻意製造各種網路問題，找出當時的各種警訊組合，方法是將實驗環境架設在實驗室內，目的則是能以相同的環境變數，而以不同設定值及流量負載模擬不同的網路狀況，優點是可以不斷重複實驗過程，封閉型實驗同時也是開放型實驗之前置作業。

二、開放型實驗：

開放型網路實驗，是以實際網路爲環境所進行的實驗，我們以東海大學校園網路爲實驗骨幹，再以自行開發及市售之網管軟體，監控網路的使用狀況，並加以紀錄及分析。

經由封閉型實驗與開放型實驗的結果，藉以建立案例資料庫與模糊資料庫，並確定資料庫的正確性與調整推論規則，最後則是在開放型架構下進行網路錯誤推論。實際實驗中，我們分別爲網路線路及伺服主機製造不同程度的負載或錯誤，使成爲系統之瓶頸，並監控及記錄發生瓶頸時所產生的現象，重複實驗確定產生的結果並存檔成爲案例。

封閉型的實驗環境單純，實驗時不會影響實際系統的正常運作，給予相同的網路負載，則會有相同的結果，但測試的數據及結果會與開放型環境有一定程度的差異。開放性網路系統變數多，網路各元件之負載行爲不易分析，但可實際量測到上線運作時可能發生的狀況。

封閉型實驗和開放型實驗必須完成下列六項工作：

1.以網路區段爲單位，建立網路設備中的網管資訊的蒐集方式與網管資料的分析方法，藉以蒐集及分析區段內各網路節點的使用狀況。

2.網管資訊數據蒐集平台的建立及監控端軟體、網路流量製造軟體的編寫、使用及測試。

3.架設網路模擬及監控環境，在固定各種網路環境變數下，製造網路問題，並記錄網路錯誤發生時的現象,及當時所監控的測試及環境數據。實驗必需重複數次以確定實驗方法無誤，也讓數據更具完整性與可靠性。

4.分析數據及研究解決方案，問題處理後需再重測相關數據，以確認網路問題點確實已經排除。

5.記錄各項監控數據並存入資料庫，以爲後續之研判及分析，並訂立網路設備到達負載瓶頸的臨界值。

6.分析實驗紀錄，並歸納及確認每一種網路錯誤發生時的網路警訊集合及相關 MIB 值，定義各 MIB 值的歸屬函數，供後續模糊推論時使用。

網路錯誤推論則必需完成下列四項工作

1.重複實驗並以模糊方法推論網路問題。

2.檢討推論網路問題的正確性，以修正推論問題的方法與監控的 MIB 數。

3. 若推論正確並順利解決，則將該案例資料建入資料庫，則資料庫可依每次案例的增加，調整各 MIB 的歸屬函數與網路問題的警訊組合，而資料庫也因此修正與成長。

4.若推論錯誤，則以人工方式(網路專家)，確定網路問題並找出解決方法，再將相關資料建入資料庫，以利下次比對使用。

3.2 網路錯誤的資料正規化及知識表示

一、網路錯誤的資料來源與正規化

ANSIT(American National Standard for Information Technology 1994)將一個警訊形成的程序定義如下：『網路問題會引發一個或數個網路事件，並導至警訊的產生』，因此我們蒐集網路事件及警訊來推論發生了何種網路問題。在本研究中使用到以下數種 MIB 來做爲分析網路錯誤行爲的資料來源：

1、MIB 2：用來判斷網路介面狀況、IP（Internet Protocol）、等相關的資訊。

2、RMON1：記錄網路介面各種負荷及錯誤的統計數據。用來判斷網路流量、及各種正常及非正常的封包統計數據。

3、802.1d Bridge MIB：記錄 802.1d Spanning Tree Protocol 相關資訊，主要用來判斷拓樸狀態改變。

在網路管理架構中，本研究所研製之網路管理伺服主機 NMS(Network Manager Server) 係透過網路管理協定蒐集被管理設備的 MIB 值，以分析網路設備的現狀。在封閉型實驗時因環境單純，可以控制單一錯誤的發生，但在開放式環境下，一個錯誤發生時，可能同時發生數個甚至數十個網路事件，又若有網路錯誤同時發生時，哪些事件是造成哪一種網路錯誤的主因，該如何判斷？就是因爲這樣的因素，要如何將有關的事件連結起來，並了解其中的關係是最重要且最困難的，這也是爲什麼我們同時安排封閉型實驗與開放式實驗進行相同的實驗，藉以比對網路錯誤的警訊組合是否正確。

因此資料庫中的案例與每一個 MIB 的臨界值，都必須要有一正規化的方式來建立網路事件警訊與網路錯誤的對應關係，正規化的方法如下：

1. 資料蒐集及分類
   (1)、確認網路範圍與資料蒐集的對象，做爲資料分析的來源。
   (2)、訂定每一項警訊的臨界值，用以判斷所監視的資料是否超過臨界值，如是則產生警訊。
   (3)、記錄並統計在某個時間區段中產生的警訊種類
2. 實驗方法
   (1)、先進行封閉式網路實驗，以取得各種網路錯誤的基準值，再進行開放式網路實驗，以學習更多的案例，並藉此調整推論用的相關資料，以達到自我成長的目的。
   (2)、經由網管參數的度量與分析，確定並比較網路正常及異常的行爲模式，做爲定義網路警訊的參考。
   (3)、實驗設計與評估：經由網路管理專家規劃與設計開放與封閉式的網

路環境與資料蒐集方式，以便蒐集網路相關的資訊。

(4)、環境檢查：掌握實驗環境的實際狀況，避免產生不預期的環境因數，以免無法產生預期的節果甚至是產生錯誤的結果。

(5)、數據分析：每一個網路錯誤的發生時間、參數數量及參數變化、持續時間等都要詳細記錄與探討其產生原因。

(6)、警訊與錯誤的關係：依分析所得的結果，將警訊的種類與相關數據以所選則的表示方法記錄至資料庫中。

(7)、重複實驗與資料修正：以同一種網路錯誤在不同拓樸與地點重複實驗，並藉此修正相關資料，以確定最終資料的正確性。

二、案例資料庫的知識表示

案例資料庫中所定義的網路錯誤資料是過去已經發生且經確認過的案例，在經過正規化的方式重複實驗後，確認已正確地解決問題的錯誤種類，過程中須記錄每一種錯誤名稱、導至該錯誤發生時所出現的警訊種類及每一種被觸發警訊的值或是狀態。

Hoare 於 1978 年推導出 CSP(Communication Sequentail Process)，其主要目的是描述平行分散式環境架構。我們利用一種類似 CSP(a→P)規格描述的簡單描述語言，讓網管專家來建立錯誤行爲知識，並描述網路錯誤與警訊間的因果關係。描述方式如下：

F1 ➔ A1A3A5A7(S1[uu]S2[uu]S2[uu]S1[uu])

其中 Fx 爲網路錯誤的代號，x 代表網路錯誤的編號，即 F1 代表編號一的網路錯誤，F2 代表編號二的網路錯誤…。Ay 爲網路警訊的代號，y 代表網路警訊的編號，即 A1 代表編號一的網路警訊，A2 代表編號二的網路警訊…。Sz 爲網路警訊發生的狀態，z 代表網路警訊的編號，即 S0 代表字串式的警訊，S1 代表觸發式的警訊，S2 爲數值式的警訊，記錄 Sz 時也一並將當時的 MIB 值記錄下來，以[uu]表示，其中 uu 代表該 MIB 當時的數值。

F1 ➔ A1A3A5A8 (S1[down]S1[down]S2[320]S2[100]) 代表本案例發生時所

產生的網路錯誤 F1 當時同時出現了 A1，A3，A5， A8 四種警訊，其中 A1 和 A3 是觸發式的警訊，A5 和 A8 是數值形態的警訊。

但是同樣的網路錯誤會因爲拓樸的不同，以及時間因素的關係，可能會有不同的描述方法，也就是同樣的網路錯誤可能會對應兩個甚至兩個以上的警訊組合。例如，同樣是網路拓樸造成迴圈，卻因爲設備是否支援 Spanning Tree 的功能而有不同的結果及警訊產生，支援 Spanning Tree 的設備會繼續工作，但會出現一個網路設備的某個 port 被 blocking 的狀態，不支援 Spanning Tree 的設備則會出現高負載的狀態，導至速度變慢甚至無法使用。

3.3 模糊知識庫

模糊知識庫是模糊推論系統中用以儲存人類知識的寶庫，主要由兩部份構成，其一爲記錄專家經驗法則之模糊規則庫（Fuzzy Rule Base），另一爲定義模糊語意之資料庫（Data Base）。

3.3.1 模糊規則庫

傳統上，知識庫和推論引擎大都採用布林邏輯（Boolean Logic）和明確推理（Crisp Reasoning）來運作。然而在真實世界中，人類有許多思維過程是非常「模糊」的，這與需要精確數據才能計算的電腦，在運作上顯然有著極大的差異。在眾多知識表示法中，我們選用了最符合人類思維模式也是應用最廣的「規則式表示法」，來記錄及表達網路錯誤的產生。

規則庫中存放的是模糊推論機構進行推論時所需要之推論規則，其型式則是仿效人類的思考推理方式，每個推論規則皆以「若（原因或條件）則動作」(IF condition THEN action)來表示。

模糊規則也是以「若 ~ 則 ~」(「IF ~ THEN ~」) 的方式記錄，但在條件及動作中加入模糊的程度，例如一個典型的教學規則可能如下所示：

IF 學生程度「好」 AND 學習能力「強」

THEN 課程進度「快一點」 AND 課程難度「高一點」

其中，「好」、「強」、「快一點」、「高一點」等語意詞，都被視爲是模糊集合的元

素，其對應的歸屬函數則在語意資料庫中加以定義。

若以通式表示，一個模糊規則庫可記作$\boldsymbol{RB} := \bigcup_{j=1}^{N} R_j$，意義是模糊規則庫是由 N 條模糊規則所組成，而每條規則 R_j 可表示爲：

$$R_j : IF\ x_1\ is\ A_{1j}\ \text{and}\ x_2\ is\ A_{2j}\ \text{and}\ \ldots\ x_n\ is\ A_{nj}$$
$$THEN\ y\ \ is\ B_j$$

其中 x_i $(i = 1, 2, ..., n)$ 爲輸入變數，y 爲輸出變數；輸入語意值 A_{ij} 和輸出語意值 B_j 分別是定義於輸入論域 X_i 和輸出論域 Y 的模糊集合之元素。

在本研究中，案例資料庫中每一筆資料的表現方式爲 CSP 的描述方法，再以此爲基準定義出每一種錯誤的模糊規則，網路錯誤 F1 的描述方法如下：

F1 ➔ A1A3A5A7(S1[xx]S2[yy]S2[zz]S1[uu])，則網路錯誤 F1 的模糊規則爲 IF (A1 is S1) and (A3 is S2) and (A5 is S2) and (A7 is S1) then (F1 is Ture)。因同樣的網路錯誤可能會對應兩個甚至兩個以上的警訊組合，同一個的網路錯誤可能會有不同的描述方法，也就是一個網路錯誤會有一個或一個以上的模糊規則。因此，規則庫的結構可歸納如下：

(1) 規則庫是由一條或一條以上的規則所組成

(2) 每條規則分爲輸入的 *IF* 和輸出的 *THEN* 兩部份

(3) 輸入的 *IF* 部份由一個以上的輸入條件合成

(4) 輸入條件是由輸入變數名稱及其對應的輸入語意值構成的

(5) 輸入合成運算爲模糊邏輯 AND

(6) 輸出的 *THEN* 部份只有一個變數

(7) 輸出結論是由輸出變數名稱及其對應的輸出語意值構成的

知識庫中的知識是由專家或是由實驗結果所提供，本研究模糊知識庫的初始建立是由實驗結果提供，並倚賴領域專家（Domain experts）依經驗修正之，並確定知識庫的內容，再由知識工程師將其轉換成對應的模糊規則及歸屬函數。

爲了要讓本研究的模糊推論系統具備學習能力，達到自我維護及學習知識庫

的目標，以免去人工建立之苦，所以每當有新的案例確認後，除了會增加案例數外，也會依據目前已確定的案例調整模糊規則庫的規則，以期達到自我學習（Self-learning）或自我組織（Self-organizing）的目的。

3.3.2 模糊語意資料庫定義

模糊語意資料庫記錄各項輸出入語意變數的範圍及定義，作爲界定規則庫中所使用到的各個模糊量的換算依據。若依上述的規則結構，歸納輸入及輸出部份所應含括的的資料定義爲：

(1) 輸入部份只定義一個輸入變數

(2) 輸入變數的定義包含輸入值域的範圍和輸入語意的歸屬函數

(3) 輸出部份只定義一個輸出變數

(4) 輸出變數的定義爲輸出值域的範圍

值域範圍包含上下限，以最小值 min 及最大值 max 來界定，定義模糊語意資料庫事實上就是定義每一個 MIB 所對應的歸屬函數。在監控的 MIB 中可分爲三類 1. 字串式 (String) 2. 觸發式 (Trigger) 3. 數量式 (Magnitude)。其中字串式爲文字描述，因此沒有對應的歸屬函數，觸發式的 MIB 爲布林值，對應的爲一般的 O 與 1 函數，因此在模糊語意資料庫中，我們直接取其值或文字描述定義此兩類的 MIB。數量式的 MIB 則以左邊梯型及右邊梯型歸屬函數爲表示方式。

數學描述如下

右邊梯型歸屬函數的表示方式如下：

$$\mu_{A}(x)=\begin{cases}0, & x<0\\ \dfrac{x}{a}, & 0\le x<a\\ 1, & x\ge a\end{cases}$$

左邊梯型歸屬函數的表示方式如下：

$$\mu_{A}(x)=\begin{cases}1, & x \le 0 \\ \dfrac{x}{a}, & 0 < x \le a \\ 0, & x > a\end{cases}$$

例如 MIB1 所定義的臨界值爲 100 時，若產生警訊 A1，所偵測到 MIB1 的值爲 65，則依照右邊梯型歸屬函數來找出發生的歸屬度爲 0.65，依左邊梯型歸數函數找出不發生的歸屬度爲 0.35，當 MIB 的值超過 100 發生的歸屬度就是 1，而沒發生的歸屬度就是 0。

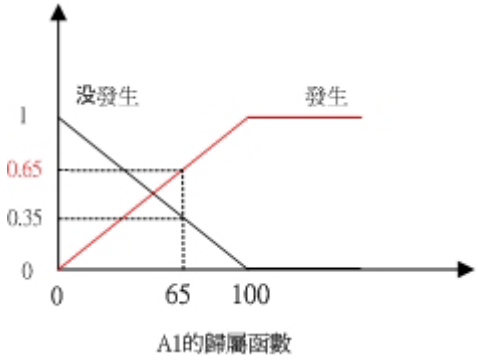

圖 3.3 左邊梯型及右邊梯型歸屬函數

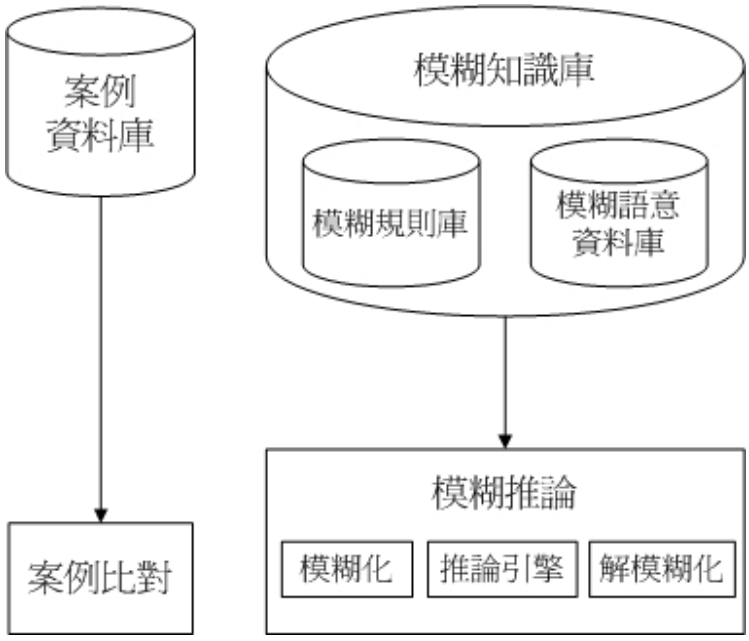

圖3.4 資料庫架構與支援的實驗流程

3.4 案例比對

案例比對的目的在於同一個網路架構下，若因相同因素而發生同樣的網路錯誤時，會產生相同的警訊集合，因此若有與發生過的案例有相同的警訊集合出現時，便判斷該案例比對成功，該項網路錯誤發生。

在特定的網路介面在單位時間內取得網路當時的事件種類與其數值，再與案例資料庫比對，其目的在以過去的案例經驗比對出本次是否有相同的網路錯誤發生，也就是從過去的經驗中學習，其優點是學習與比對的方式不需要花費很多的時間與很多的維護工作，缺點是比對的範疇不大，因爲網路狀況並非一成不變，雖然有相同的網路架構會因爲使用者的多寡、網路使用的特性，有時甚至會因氣候狀況而影響網路運作的狀況，完全比對成功的機率不一定很高，但不失爲一個基本判斷網路問題的方式。

比對流程如下：

a. 在單位時間內取得所有的 MIB 值，並蒐集超出臨界值的 MIB 所對應的所有警訊。
b. 與每一種曾造成網路錯誤 Fj(1≦j≦N)發生時的網路警訊集合逐一比對，找出網路警訊的種類與數目均與本次紀錄警訊集合相符合的 Fj，若有符合的 Fj, 則發出錯誤通知給網管人員。
c. 若沒有相同的狀況，則離開案例比對，進行模糊推論。

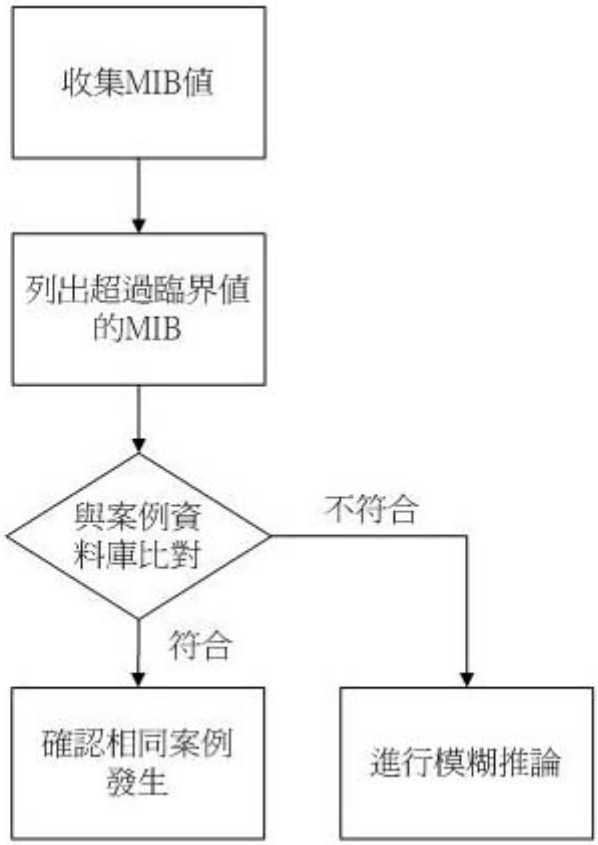

圖 3.5：案例比對流程

例如案例資料庫中有 F1➔ A1A2A5(S1[down]S2[90]S2[1000])30，

F2➔A2A3A6A7(S2[100]S2[900]S2[90]S1[down])30，

在單位時間內偵測到警訊集合 A2[90]A6[100]A8[up]，則無比對成功的案例，在另一時間區段偵側到 A1[down]A2[120]A5[1100] ，與 F1 相符合，則判斷爲 F1 發生，並通知管理人員處理。

3.5 模糊推論與反模糊化

在此之前我們採用案例比對的方式來尋求解答，是假設在網路拓樸及設備沒有出現重大改變時，所出現的網路問題與曾經發生過的網路問題有著完全相同的狀況。但因網路拓樸、網路設備的使用效率及使用者多寡等環境因數的改變，可能會使得同一種網路錯誤發生時所產生的網路警訊組合有數量及種類上的改

變，因此精確的比對方式反而不容易判斷出實際的問題。

而以模糊規則爲基礎的專家系統的精神是將某些客觀事物在中間過渡時所呈現的”亦此亦彼”的差異性做有效的處理，並接受模糊性的存在，其目標是處理條件模糊的網路狀況，並將其數值化進行嚴密的處理，就是希望能夠兼顧以規則爲基的專家系統及模糊推論的優點，也因爲模糊推論的關係，不似以規則爲基礎的專家系統容易被環境及其他因素干擾而影響到正確性。

推論流程如下：

a. 找出案例資料庫中的每一種網路錯誤所對應的模糊規則。

例如： F1➔ A1A2A5(S1[down]S2[90]S2[1000])，所對應的模糊規則爲

IF (A1 發生) & (A2 發生) & (A5 發生) THEN (F1 發生)

b. 將本次蒐集的每一項 MIB 值帶入模糊語意資料庫所對應的歸屬函數中，以取得到每一個網路警訊的觸發強度αi。

例如：記錄到 A1 的 MIB 的值爲 80 ，若 A1 在模糊語意資料庫中的歸屬函數如圖 3.5 所示，則 A1 發生的觸發強度αi爲 0.8。

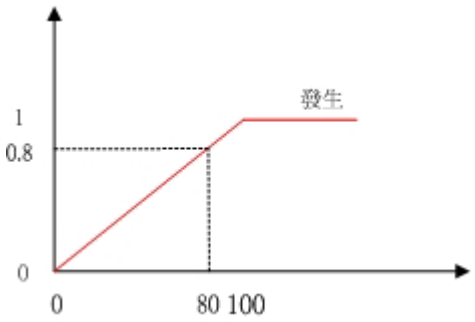

圖 3.6 A1 的歸屬函數

c. 依據每一條模糊規則，找出規則中每一個條件的觸發強度 αi，並取其中的最小值，此最小值即爲該模糊規則發生的α值 ，依據 Zadeh 模糊集合的交集定義，若 C=A∩B，則 $\forall x \in X$ ，$\exists \mu c(x) = \min[\mu_A(x), \mu_B(x)]$。

例如： 案例資料庫中存在 F1➔ A1A2A5(S1[320]S2[98]S2[1100])，模糊語意資料庫中，A1 的臨界值爲 300，A2 的臨界值爲 90，A3 的臨界值爲

1000，若記錄到的 A1 爲 210 ，A2 爲 91，A5 爲 760，則 A1 所對應的 $\alpha 1$ 算法爲 210/300 = 0.7，A2 所對應的 $\alpha 2$ 爲 1 ，A3 所對應的 $\alpha 3$ 爲 0.76，而 $\alpha 1$、$\alpha 2$、$\alpha 3$ 取最小值爲 0.7，則 F1 的模糊規則所產生的觸發強度 α 值 爲 0.7。

d. 若單一種網路錯誤只有一種模糊規則，則該模糊規則所產生的觸發強度即爲模糊推論的 α 值，若單一種網路錯誤，有一條以上的模糊規則時，將所有規則所推論出的 α 值，取其中最大者，該值即爲模糊推論的 α 值。

e. α 值爲模糊推論的觸發強度，必須再經過解模糊化的步驟將觸發強度轉成發生的可能性，解模糊化的歸屬函數我們採用與模糊語意資料庫相同的線性函數，其數學表示方法如下：

$$\alpha_{A}(x)=\begin{cases}0, & x=0\\ \dfrac{x}{100\%}, & 0\le x<100\%\\ 1, & x=100\end{cases}$$

將模糊推論的 α 值對應至發生可能性的歸屬函數之上。使用最左極大法 LoM(Left-of-Maximun)解模糊化計算出發生的可能性，解模糊化後對應至 X 軸的值即爲 Fj 發生的可能性。

f. 推算每一個與本次蒐集的網路事件相符合的 Fj 發生的可能性，從大至小列出，供網路管理人員參考，俾進行錯誤回復的流程。

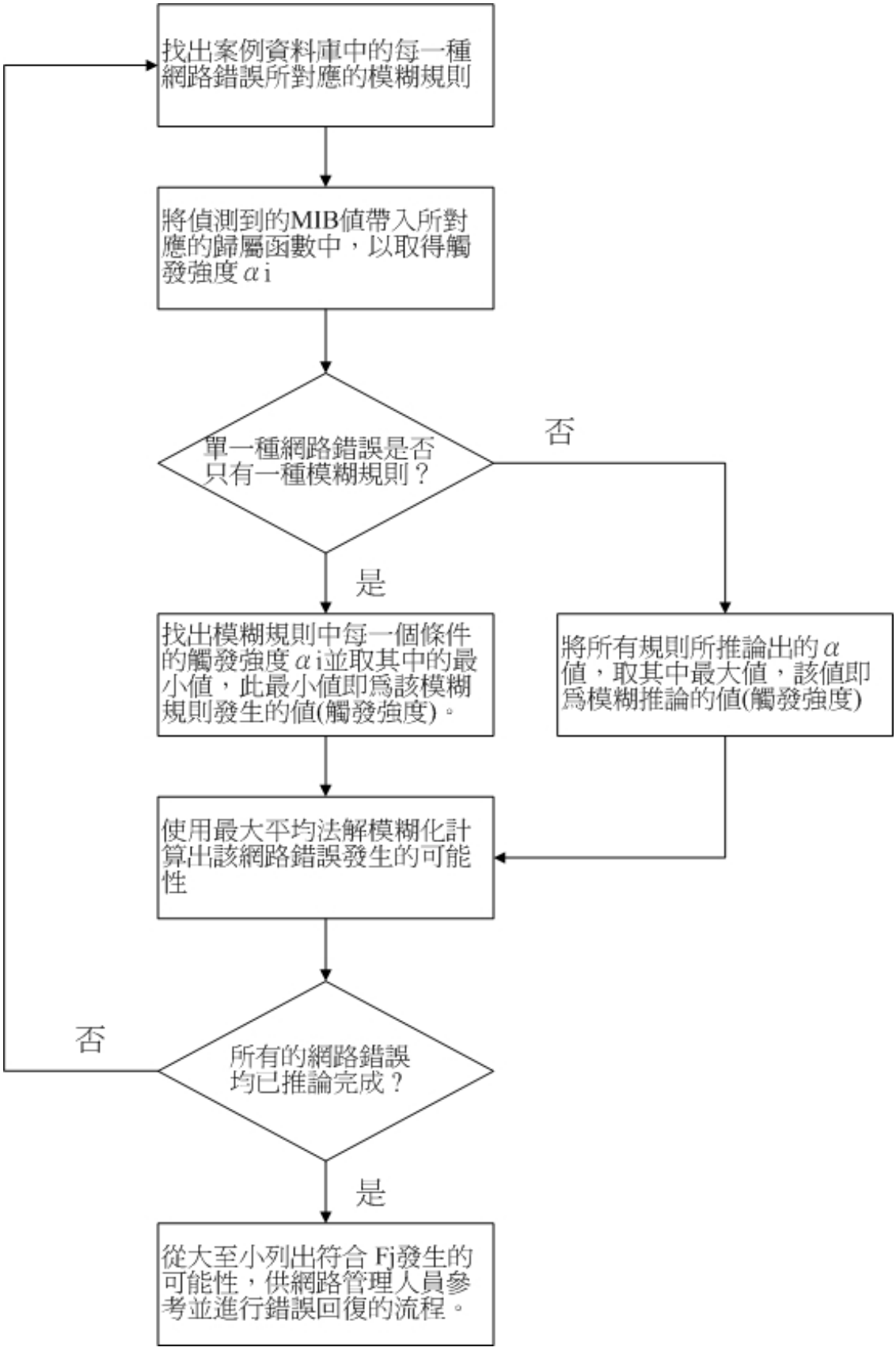
找出案例資料庫中的每一種網路錯誤所對應的模糊規則
將偵測到的MIB值帶入所對應的歸屬函數中，以取得觸發強度αi
單一種網路錯誤是否只有一種模糊規則？
否
是
找出模糊規則中每一個條件的觸發強度αi並取其中的最小值，此最小值即爲該模糊規則發生的值(觸發強度)。
將所有規則所推論出的α值，取其中最大值，該值即爲模糊推論的值(觸發強度)
使用最大平均法解模糊化計算出該網路錯誤發生的可能性
所有的網路錯誤均已推論完成？
否
是
從大至小列出符合 Fj發生的可能性，供網路管理人員參考並進行錯誤回復的流程。

圖 3.7 模糊推論流程圖

範例一：

警訊與其歸屬函數資料庫中

a1，a2，a4，a5 的 threshold 分別爲 96，84，70，62

而案例統計資料庫中,有三種網路錯誤 Fj 的統計資料

F1：a1a2a5 爲曾發生警訊集合的聯集，錯誤成立時有過以下警訊的集合 a1a5 , a2a5 , a1,a2,a5。

F3：a1a2a4a5 爲曾發生警訊集合的聯集，錯誤成立時有過以下警訊的集合 a4a5 , a1a4a5 , a2a4a5 , a1a2a5 。

F5： a1a4a5 爲曾發生警訊集合的聯集，錯誤成立時有過以下警訊的集合 a4a5, a1a4 , a1a4a5 。

偵測網路當時的網路警訊及其數據，a1a2a4a5(80,75,30,85)

步驟一：將本次測量到的 MIB 值帶入所對應的歸數函數，並計算其發生的歸屬度。

a1 的發生歸屬度爲 80/96 = 0.83 沒發生的歸屬度爲 1-0.83=0.17

a2 的發生歸屬度爲 75/84 = 0.89 沒發生的歸屬度爲 1-0.89 = 0.11

a4 的發生歸屬度爲 30/70 = 0.43 沒發生的歸屬度爲 1-0.43 = 0.57

a5 的發生歸屬度爲 85/62 因 85 超過 threshold 62 所以發生的歸屬度取 1

不發生的歸屬度 1 - 1 = 0

步驟二：依 F1 目前已發生過的案例導出模糊規則。

規則一：if a1 發生且 a2 不發生且 a5 發生 則 網路錯誤 F1 發生

規則二：if a1 不發生且 a2 發生 且 a5 發生 則 網路錯誤 F1 發生

規則三：if a1 發生 且 a2 發生 且 a5 發生 則 網路錯誤 F1 發生

步驟三：推論所有規則合成的觸發強度α。

規則一：if a1 發生且 a2 不發生且 a5 發生則網路錯誤 F1 發生

α_1 = min(0.83 ,0.11, 1) = 0.11

規則二：if　a1 不發生且 a2 發生 且 a5 發生 則 網路錯誤 F1 發生

α_2 = min(0.17,0.89 , 1)= 0.17

規則三：if　a1 發生 且 a2 發生 且 a5 發生 則 網路錯誤 F1 發生

α_3 = min(0.83 , 0.89 , 1) = 0.83

α 值(觸發強度) = max (0.11 , 0.17 , 0.83) = 0.83

步驟四：解模糊化。

α 值(觸發強度) = 0.83，解模糊化則 0.83 / 100% = 0.83 ，F1 發生的可能性爲 83 % 。採用最左極大法的理由是反模糊化的結果會在 0 與 1 之間，正好可以當作是錯誤發生的機率，對本研究來說較爲合理，若採用重心法、中間平均法則反模糊化的結果會在 0.5 與 1 之間，雖然運算結果還是會有一樣的排列順序，但發生的可能性不會低於 50% 似乎顯得不合適。

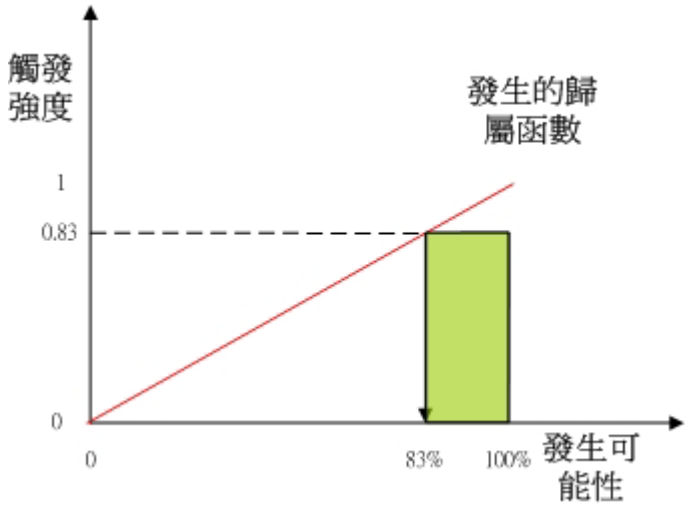

圖 3.8 反模糊化的歸屬函數

步驟五：回到步驟一，計算其他網路錯誤發生的可能性。

F3：發生的案例所產生的規則如下。

規則一：if　a1 不發生且 a2 不發生且 a4 發生且 a5 發生則網路錯誤 F3 發生

α_1 =min(0.17 ,0.11,0.43, 1) = 0.11

規則二：if　a1 發生且 a2 不發生且 a4 發生且 a5 發生則網路錯誤 F3 發生，

α_2 =min(0.83 ,0.11, 0.43 , 1)= 0.11

規則三：if　a1 不發生且 a2 發生且 a4 發生且 a5 發生則網路錯誤 F3 發生，

α_3 =min(0.17 ,0.89, 0.43 , 1)= 0.17

規則四：if　a1 發生且 a2 發生且 a4 不發生且 a5 發生則網路錯誤 F3 發生，

α_4 =min (0.83 , 0.89 , 0.57,1) = 0.57

α 值(觸發強度) = max (0.11 , 0.11 , 0.17,0.57) = 0.57

α 值(觸發強度) = 0.57 則 F3 發生的可能性爲 57 % ，即 F3 發生的可能性爲 57%

F5：發生的案例所產生的規則如下。

規則一:if　a1 不發生且 a2 不發生且 a4 發生且 a5 發生則網路錯誤 F5 發生，

α_1 =min(0.17 ,0.11,0.43, 1) = 0.11

規則二:if　a1 發生且 a2 不發生且 a4 發生且 a5 不發生則網路錯誤 F5 發生，

α_2=min(0.83 ,0.11, 0.43 , 0)= 0

規則三：if　a1 發生且 a2 不發生且 a4 發生且 a5 發生則網路錯誤 F5 發生，

α_3 =min(0.83 ,0.11, 0.43 , 1)= 0.11

α 值(觸發強度) = max (0.11 , 0 , 0.11) = 0.11，即 F5 發生的可能性爲 11 % 。

解模糊化 (11% / 100%) = 11 % ，F1 發生的可能性爲 11%

步驟六:從大至小排序各錯誤之發生機率,供網路管理人員參考並進行錯誤回復。

F1 發生的可能性爲 83 %

F3 發生的可能性爲 57 %

F5 發生的可能性爲 11 %

推論結果：

以 F1 發生的可能性最高，因此建議網路管理人員以 F1 爲排除問題的優先選擇，其次爲 F3，最後爲 F5。

3.6 知識庫修正與案例回饋

在管理者依照案例比對或是模糊推論的結果實地去處理網路問題時，可以確認比對或是推論的結果是否正確。

若比對或是推論的結果正確，則回饋資料以增加案例或是修正知識庫，若是不對，則交由網路專家判斷確切的問題後，再將相關資料輸入案例及知識庫中，其流程見圖 3.8。

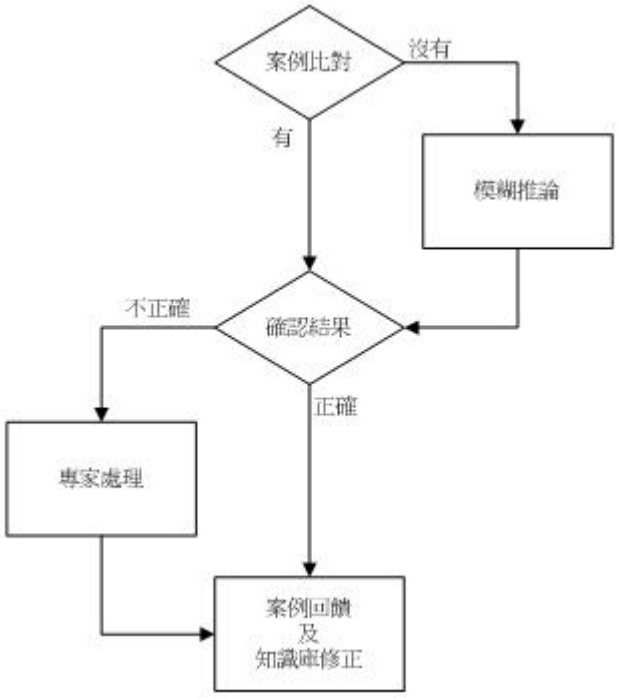

圖 3.9：案例比對及模糊推論的案例回饋及知識庫修正流程

1.案例回饋

案例資料庫的回饋來源分爲兩種，第一種爲案例比對成功，且正確解決網路問題，第二種爲模糊推論結果能正確解決網路問題。若是案例

比對成功，則該案例發生次數增加一次，若爲模糊推論結果正確，則增加案例種類或是增加案例的警訊組合，並調整原有 MIB 的臨界值，MIB 臨界值的調整會在下一段說明，但無論是增加案例種類或是增加案例的警訊組合均會造成模糊規則的增加。

2.知識庫修正

當模糊推論成功時，就必須要修正知識庫，以利後續有狀況再發生時之推論，知識庫修正包含模糊規則庫及模糊語意資料庫的修正。

模糊規則庫的修正在系統自我成長時其實就是增加模糊規則，無論是案例種類或案例的警訊組合增加時均會增加模糊規則庫中的規則，若網路管理人員在發現系統常常有誤判的狀況時，則必須請領域專家來檢視並刪除誤判的案例或是以人爲判斷方式調整警訊組合。

而模糊語意資料庫的修正，就是調整單一 MIB 的歸屬函數，也只有在模糊推論成功地解決網路問題後，才需要調整 MIB 的臨界值，調整方法是先找出模糊推論最後所使用的模糊規則，模糊規則中有超過單一 MIB 臨界值的不處理，沒超過原有 MIB 臨界值的則調整 MIB 的臨界值，MIB 的新的臨界值則以下列公式產生：

(原臨界值*以前發生次數 + 本次 MIB 值) / (以前發生次數+1)

第四章：實驗設計與實作

4.1 實驗規劃與設計

網路實體架構依據特性區分爲被監控設備、網管資料蒐集與推論系統三大部份，所有的網路設備與網路主機均爲被監控的設備，監控主機本身負責蒐集網管資訊與問題推論。

本實驗的網路設備有 Router、Switch、 Hub，所有的設備均須支援網管功能，硬體設備清單如下：

設備編號	設備廠牌、型號	網址	Community string
路由器 1	Cisco 7505	192.168.3.1	public
路由器 2	Cisco 2500	192.168.1.2	public
路由器 3	Cisco 2500 pro	192.168.2.3	public
交換器 1	D-Link 3225G	192.168.5.11	public
交換器 2	D-Link 3226	192.168.4.11	public
交換器 3	Cisco 3524	192.168.1.11	public
交換器 4	D-Link 3224	192.168.1.13	public
集線器 1	D-Link 1824is	192.168.5.12	public
集線器 2	D-Link 1824is	192.168.5.13	public
集線器 3	D-Link 1824is	192.168.5.14	public
集線器 4	D-Link 1824is	192.168.1.12	public
集線器 5	D-Link 1824is	192.168.1.14	public
光電轉換器	10MB * 1		
光電轉換器	100MB * 2		
光纖	5M*2		
UTP 線	10M * 20		
服務主機	PC Server * 2	192.168.4.101-102	public

監控、推論主機	PC * 1	192.168.3.100
User 工作站	PC * 2	依任務而定
Sniffer 工作站	Notebook * 1	依任務而定

在軟體的使用方面，服務主機與監控、推論主機的作業系統均是 Linux RedHat 9.0，監控、推論主機使用 MySQL 資料庫，程式開發則是使用 Java，用來模擬網路使用者的流量產生器是 NLANR 公司的 Iperf，該軟體可依需求同時或單獨產生 TCP、UDP 及 ICMP 的封包，也可模擬單一或多數使用者的連線使用狀況，主要的目的是模擬各式網路流量，並測量網路的頻寬與效能。

另外使用了 AdvenNet 公司的 MIB Browser 隨時觀察實驗中的各項 MIB 變化情形及了解各 MIB 的初始狀況是否正常，在實驗過程中除了用自行開發的軟體抓取網管資訊，也使用 Network Associates 的 Sniffer pro 來協助了解網路的各種狀況，同時比對資料庫中的資料與 Sniffer 的資料是否相符合，藉以確定實驗時各項資料的正確性。

被監控的設備部份是依據網路實驗的內容而特別規劃的，目的在於能突顯特定的網路狀況且容易蒐集與分析資料，實驗網路架構圖 4.1 所示：

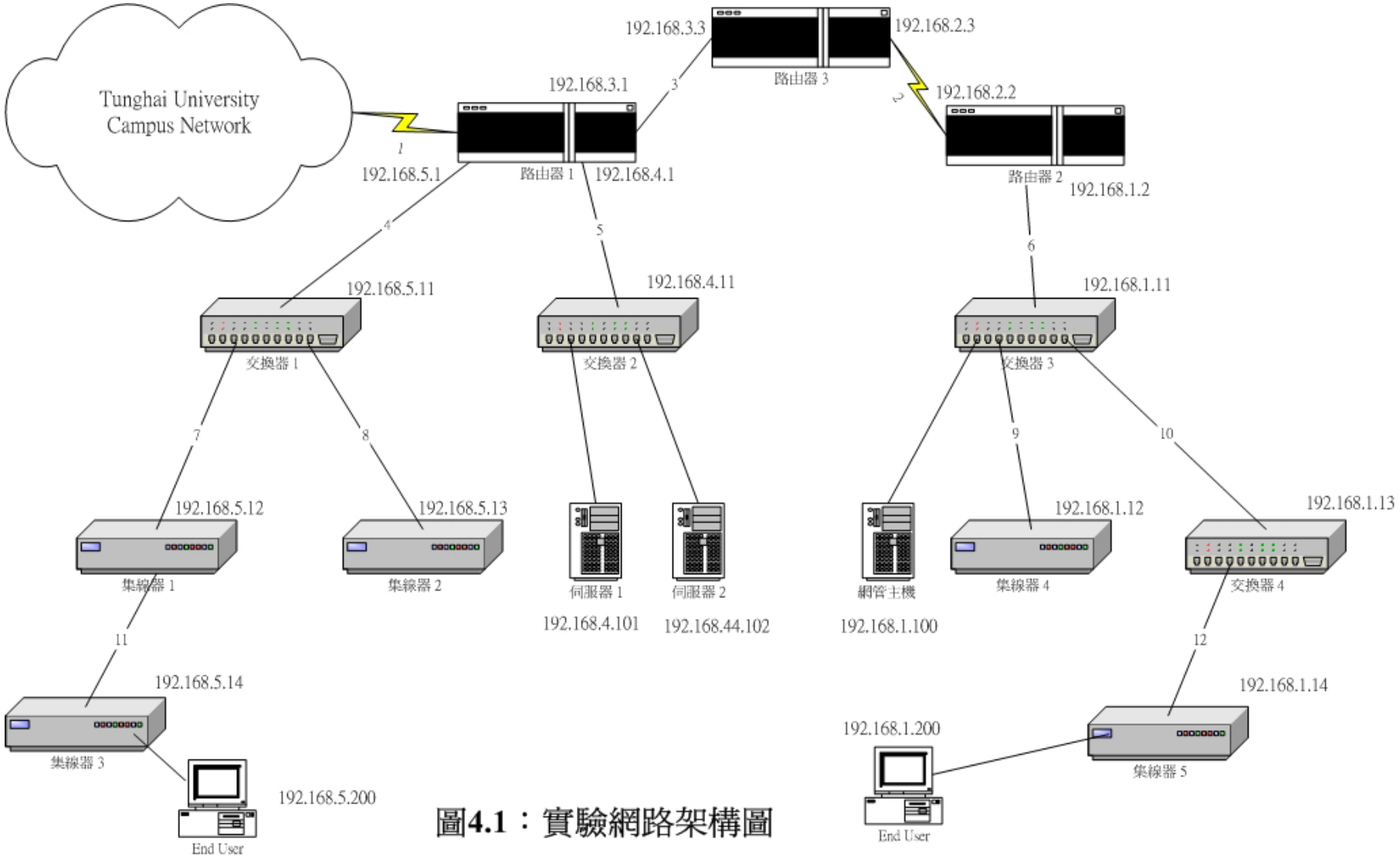
Tunghai University
Campus Network
192.168.3.3
192.168.2.3
路由器 3
192.168.3.1
192.168.2.2
路由器 1
路由器 2
192.168.5.1
192.168.4.1
192.168.1.2
192.168.5.11
192.168.4.11
192.168.1.11
交換器 1
交換器 2
交換器 3
192.168.5.12
192.168.5.13
192.168.1.12
192.168.1.13
集線器 1
集線器 2
集線器 4
交換器 4
伺服器 1
伺服器 2
網管主機
192.168.4.101
192.168.44.102
192.168.1.100
192.168.5.14
192.168.1.14
集線器 3
集線器 5
192.168.1.200
192.168.5.200
End User
End User

圖4.1：實驗網路架構圖

4.2 實驗網路架構說明

網路架構功能區分爲三大部份，見圖 4.2：

1.網路主機區：圖中的橢圓型區塊，主要在提供特定的網路服務，模擬一般公司或學校主機房或是 Server Farm 的網路運作狀況，其網路特性爲獨立一區，頻寬大，一般是銜接在網路主幹上，爲主要網路資源的提供地點，但屬於容易受網路攻擊或駭客入侵的區域。在本實驗中網路主機也是被監控的設備，讓管理者能隨時了解網路主機的服務狀況。

2.使用者區：圖中的方型區塊，包含網路中繼設備與一般終端使用者銜接網路的區段，其網路特性爲其架構會依使用者需求而不同，網路主幹常發生的網路狀況通常是起因於設備因素，而終端網路設備通常是人爲因素造成。一般網路使用者是接在這些終端的網路設備上，所以純網路中繼及銜接使用者的終端的網路設備是網路管理資訊最重要也是最直接的來源。

3.監控主機：圖中的三角型區塊，負責蒐集與分析網路管理相關資料。所有的網路管理資訊會集中到監控主機，所以規劃時必須考慮網路資料蒐集對網路的負擔、網路的穩定與其安全性。

圖 4.2 中的 4 號線路及 10 號線路是以光纖與光電轉換器來連結兩個網路設備，這種銜接方式在現實網路中通常是運用在建築物與建築物間，或是網路銜接距離超過 UTP 網路線的有效距離時使用，路由器 2 與路由器 3 是以 WAN 介面銜接，網路速度爲 2Mb，模擬長距離的網路狀況，一般公司或學校的網路瓶頸也會在 WAN 的地方，連接東海大學的校園網路則是路由器 1 負責。

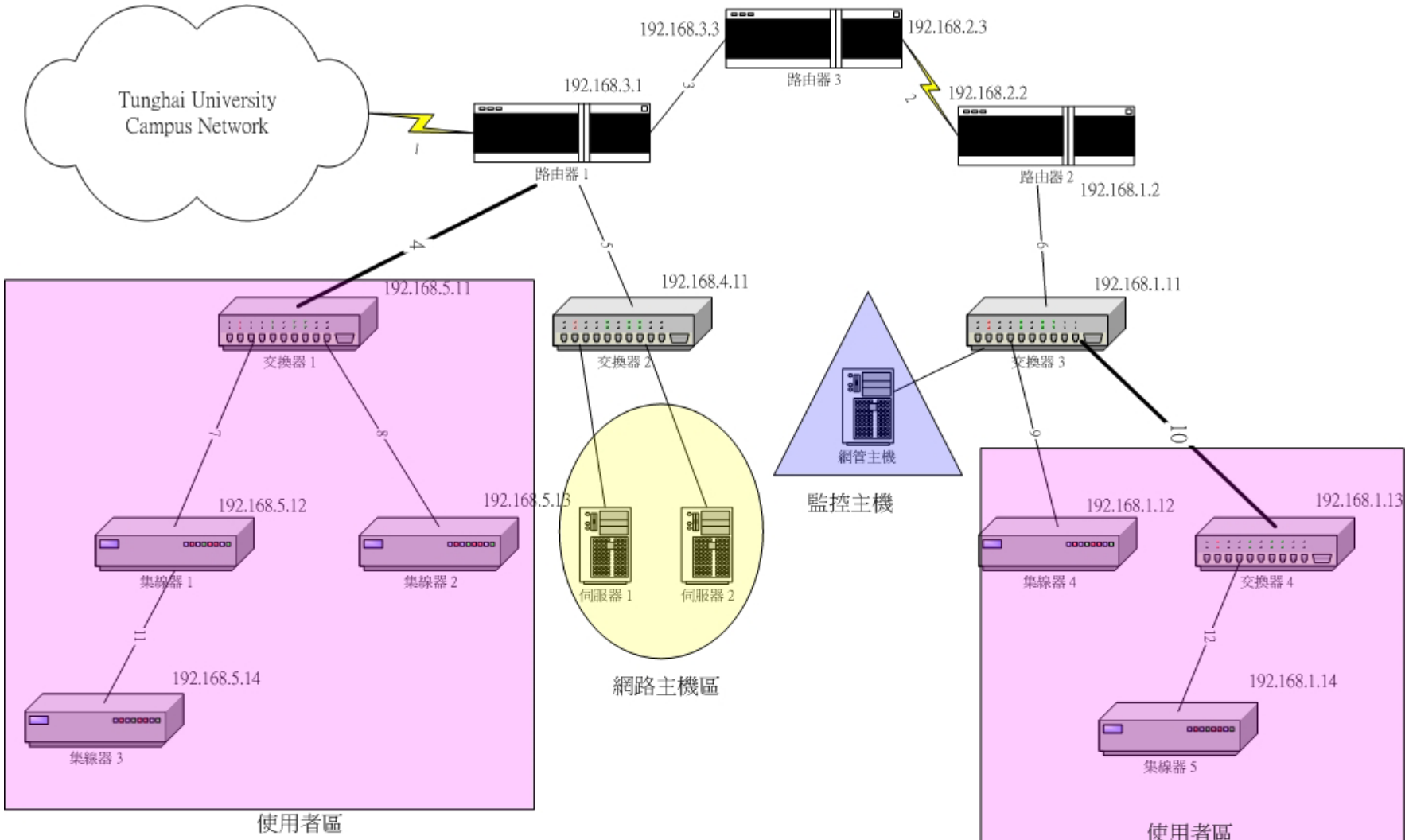

圖4.2：實驗網路功能區分圖

4.3 實驗流程

實驗流程

由於錯誤實驗的目的是要分析出網路錯誤發生時所產生的警訊與錯誤本身的因果關係，爲了能以有效率的方式進行資料蒐集，我們必須掌控環境變數，反複實驗以確切實驗出正確的結果，也就是在封閉型的網路狀況下，先找出錯誤與警訊的關係，然後在開放型的網路狀況下，重複相同的實驗，藉以調整與建立案例資料庫與模糊推論用的相關規則，見圖 4.3。

最後則是在開放式的網路狀況下，利用之前實驗所建立的案例資料庫與模糊規則庫，再次重複相同實驗，藉以驗證以案例比對或是模糊推論的方式是否能夠正確推論或是比對出正確的網路錯誤。

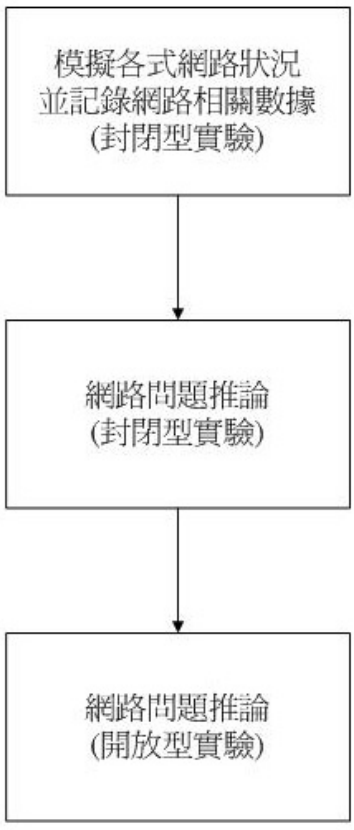

圖 4.3：實驗流程

4.4 實驗項目與結果

網路問題 1.

實驗名稱：網路斷線 (Dropped line)

實驗目的：模擬因設備損壞或是線路損壞造成的線路中斷

實驗說明：網路斷線在實際狀況中是最常發生的一種，造成的原因也相當多，例如，線路因外力破壞而中斷，人為拔除，線路接頭損壞等原因。

實驗步驟：此實驗採用的作法為將所觀察的網路設備間的連線拔除，然後記錄當時網路設備各項 MIB 的情況，本實驗在交換器 1 與集線器 1 間進行。

實驗結果：

網路斷線時警訊 a0103(ifOperStatus)、a0202(ifInOctets)、a0204(ifOutOctets)，同時間發生變化，請分別參考表 4.1，圖 4.4，圖 4.5 和圖 4.6，網路介面從 up => down，網路進、出的流量狀況變為 0 並一直持續沒有變化。

表 4.1：網路斷線的警訊集合

網路錯誤名稱	警訊集合	備註
網路斷線 (Dropped line)	A0103 A0202 A0204	

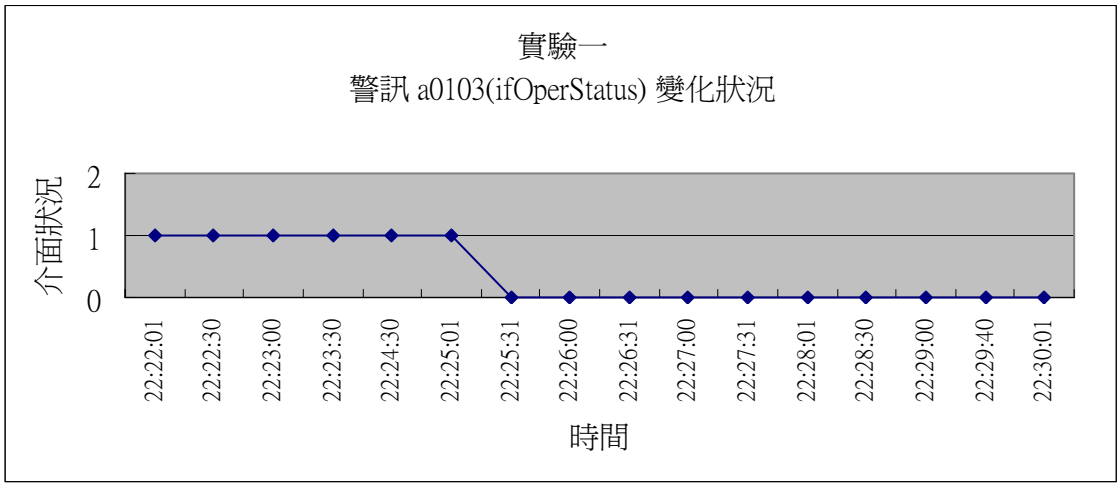

圖 4.4：實驗一的警訊 a0103(ifOperStatus)數據變化圖

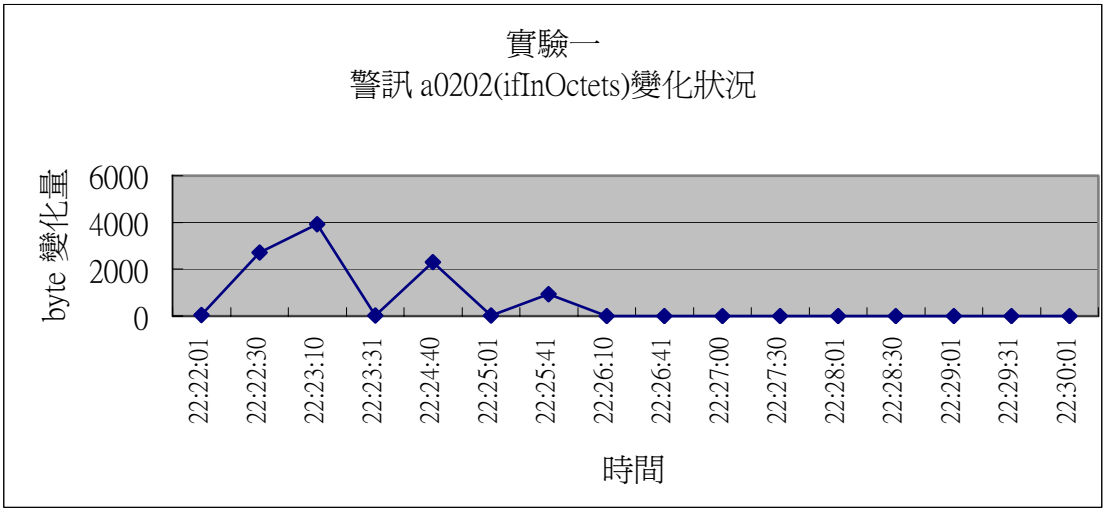

圖 4.5：實驗一的警訊 a0202(ifInOctets)數據變化圖

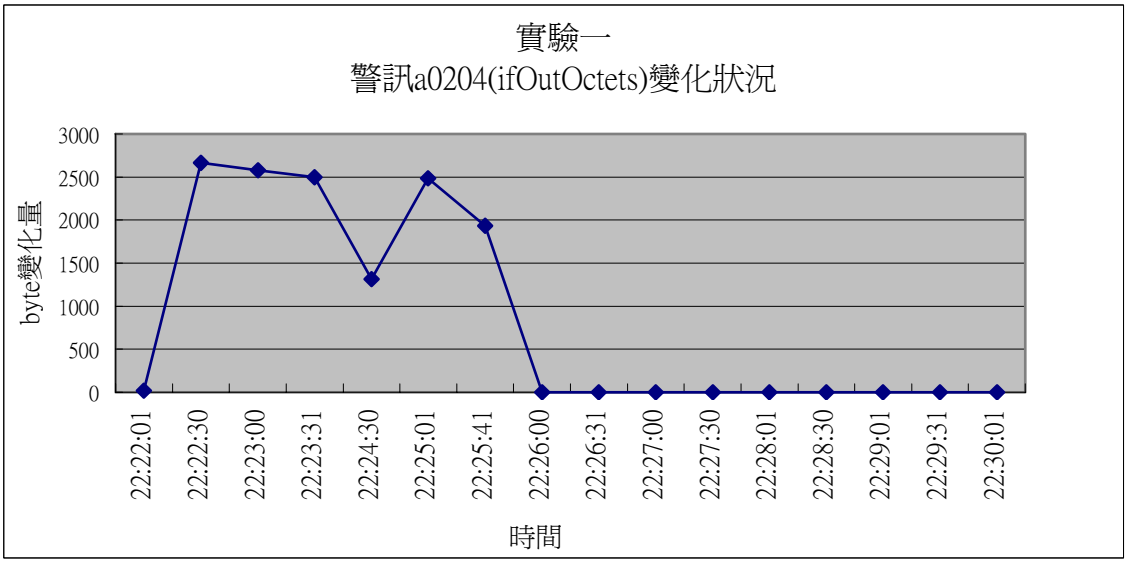

圖 4.6：實驗一的警訊 a0204(ifOutOctets)數據變化圖

實驗一在 22:25:30 將連線拔除後，警訊 a0103 在 22:25:31 就從 up 狀態變成 down，警訊 a0202 與 a0204 則在 22:25:26 左右流量均趨近於 0。

網路問題 2.

實驗名稱：主機當機 (Server Crash)

a.開機不成功或當機

b.主機負載過高

實驗目的：模擬網路主機因開機不成功或因不明原因負載過高造成的不正常運作。

實驗說明：本實驗著重在監控主機的運作狀態，區分爲兩個小實驗，第一種是無法開機成功或當機狀況，第二種是主機遭受某種攻擊或不正常運作而無法正常的提供服務。

實驗步驟：

實驗 a ➔ 將網路主機關機重開，主機在重開的過程中失敗

實驗 b ➔ 編寫一隻無限迴圈，耗盡主機資源的程式，讓主機負載過高或當機。

實驗結果：

開機不成功或當機時的網路特徵爲網路連線狀態正常，但網路流量接近零，主機狀態完全沒有資料。

主機負載過高時特徵爲網路連線狀態正常，但網路流量正常或接近零，主機狀態會時有時無，主機 cpu 負載過高。在本實驗中，CPU 負載在短時間內急速升高，且在最後兩個時間點讀不到 MIB 值，代表主機負載已經過高，主機狀況已經接近當機，網路流量有減少但並沒有觸發警訊。請參考 表 4.2，實驗二 a 參考圖 4.7 – 圖 4.9，實驗二 b 參考圖 4.10 – 圖 4.12。

表 4.2：主機當機的警訊集合

網路錯誤名稱	警訊集合	備註
a.開機不成功或當機	設備端: a0202，a0204	

主機端:

c0101

b.主機負載過高　　主機端:

b0301，b0302，b0303，c0101

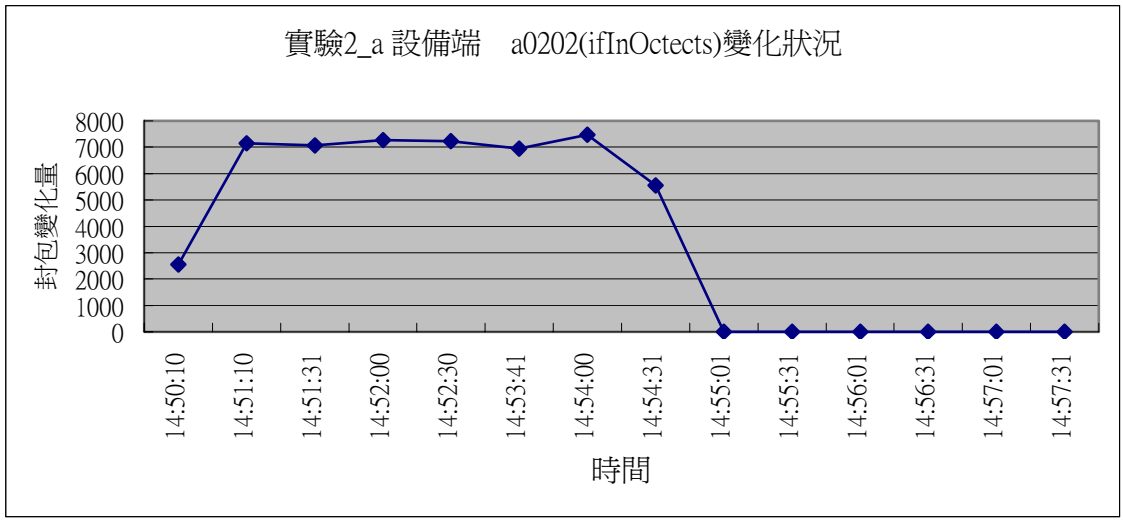

圖 4.7：實驗 2_a 設備端　a0202(ifInOctects)數據變化圖

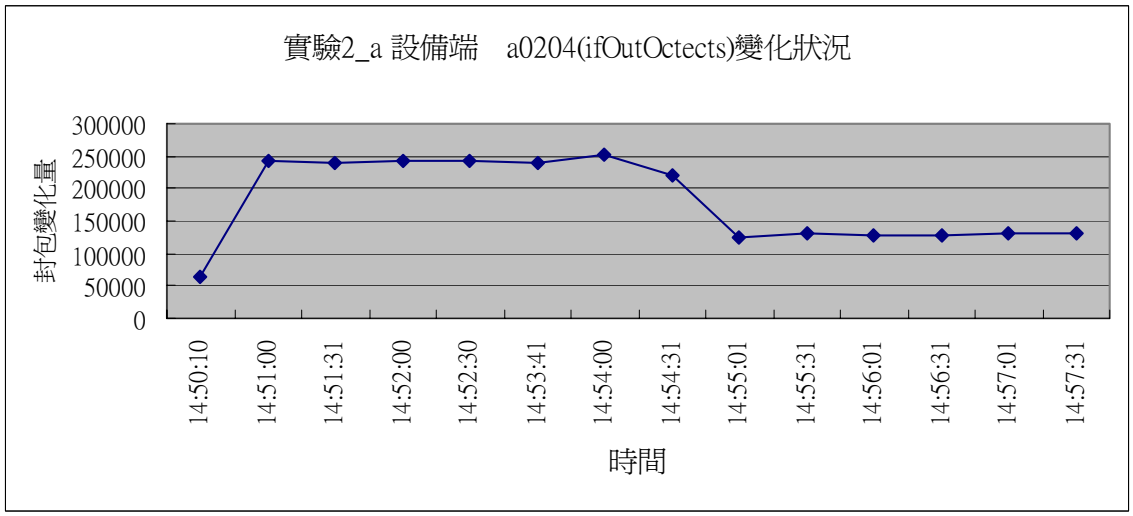

圖 4.8：實驗 2_a 設備端　a0204(ifOutOctects)數據變化圖

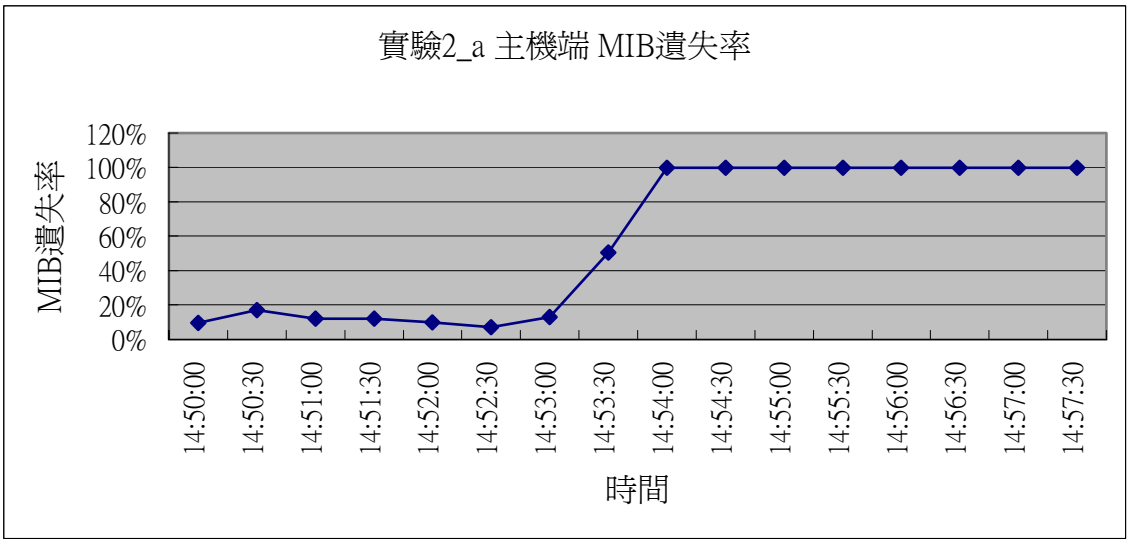

圖 4.9：實驗 2_a 主機端 c0101(MIB 接收遺失率)數據變化圖

實驗 2_a 主機當機後，主機端 MIB 接收遺失率從原本的 15% 以下，突增至 100% 且一持續著，網路流進與流出的量也突然接近零，唯本實驗以一半的 TCP 與一半的 UDP 封包當作背景流量，所以設備端還有一半的流量，爲 UDP 封包的流量。

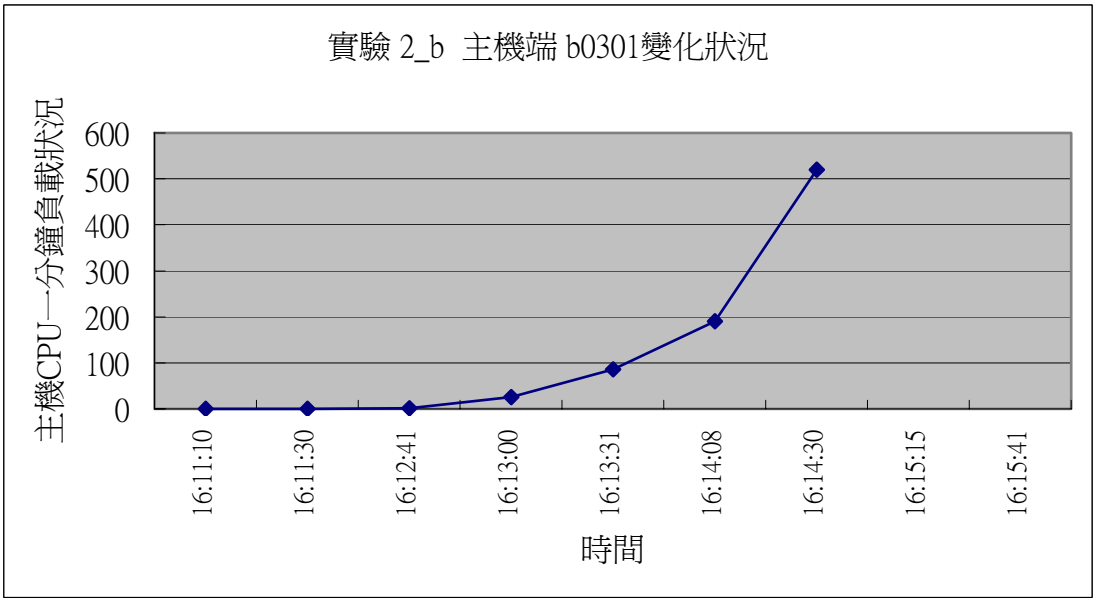

圖 4.10：實驗 2_b 主機端　警訊 b0301 數據變化圖

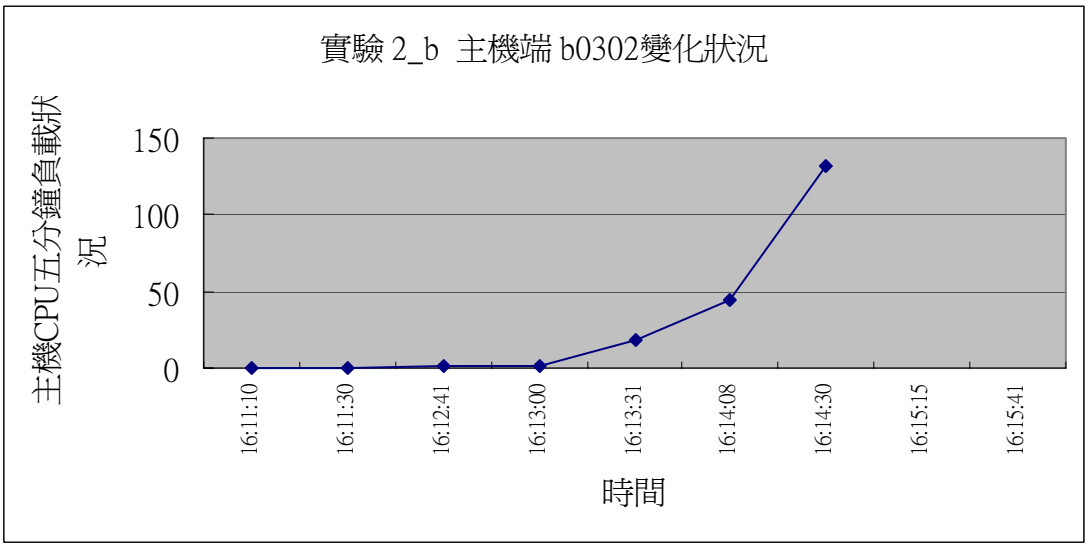

圖 4.11：實驗 2_b 主機端　警訊 b0302 數據變化圖

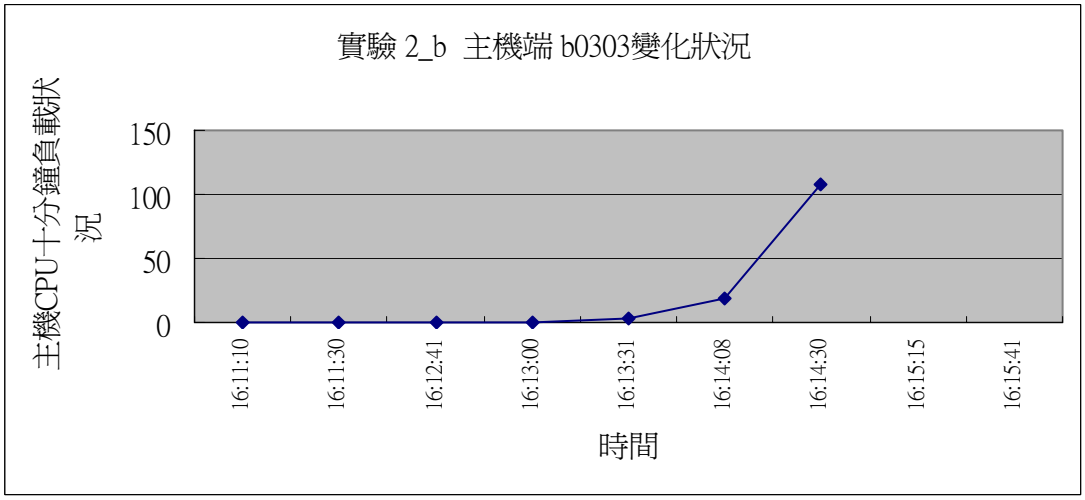

圖 4.12：實驗 2_b 主機端　警訊 b0303 數據變化圖

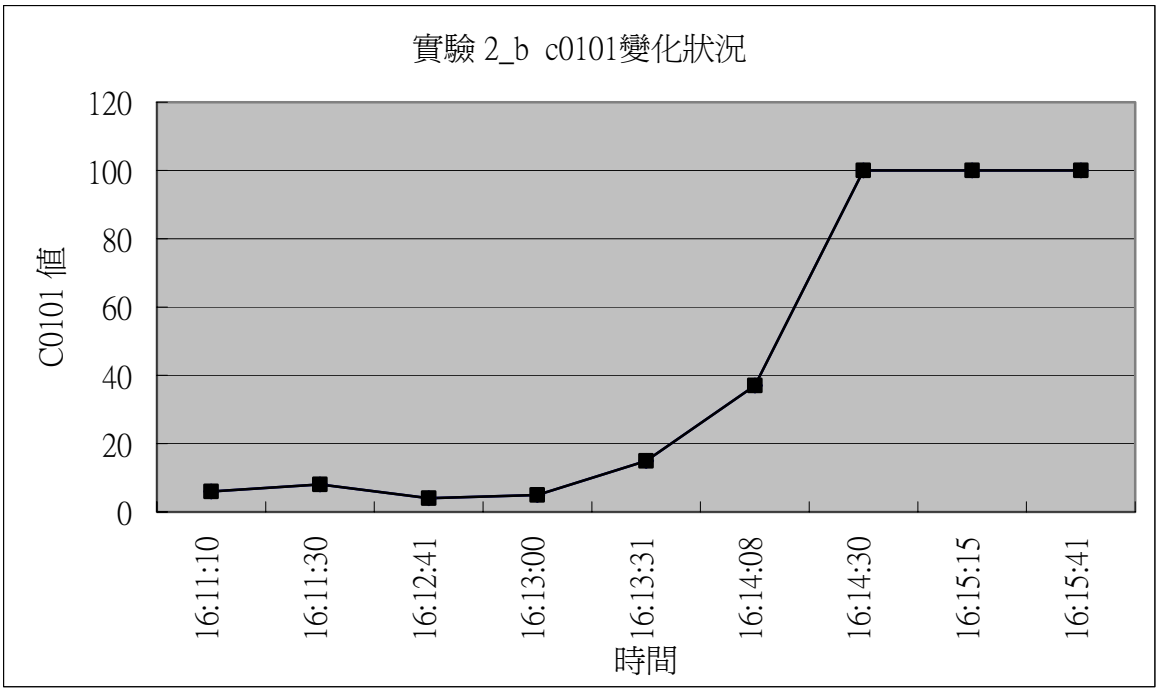

圖 4.13：實驗 2_b 主機端　警訊 c0101 數據變化圖

實驗 2_b 因特定原因導致主機負載過重，b0301、b0302 及 b0303 均在短時間內一直升高直到 16:14:30 就讀不到主機的 mib，而 C0101(MIB 遺失率) 也在同時間達到 100%，代表已經完全讀不到主機的任何 MIB。

網路問題 3.

實驗名稱：主機重開 (Server Reboot)

實驗目的：模擬主機重開機的狀況

實驗說明：網路主機通常是 24 小時提供服務，無論正不正常重開機均是一種相當嚴重的網路行爲，主機重開可能是管理者或是不當行爲所造成。

實驗步驟：此實驗採用的作法是將主機重新啓動，並記錄啓動過程中各網路相關數據。

實驗結果：

主機重開時，”系統開機時間”會復歸回零重新增加，其他如網路流量及網路設備連線狀況變化不足以當作必要條件。請參考表 4.3 及圖 4.14。

表 4.3：主機重開的警訊集合

網路錯誤名稱	警訊集合	備註
主機重開 (Server Reboot)	a0108	

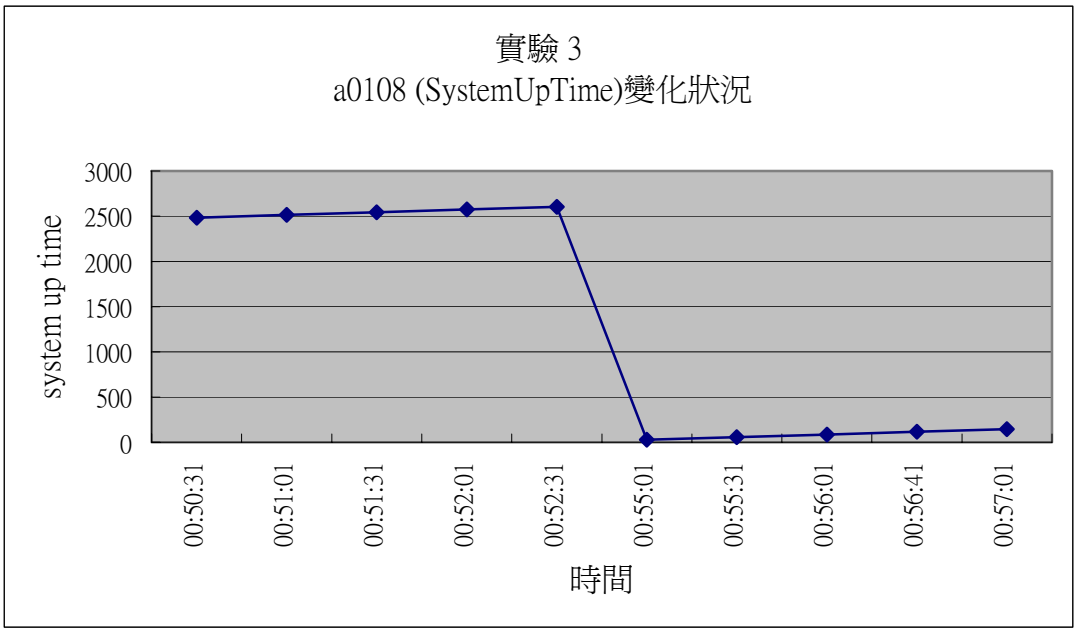

圖 4.14：實驗 3 警訊 a0108 數據變化圖

網路問題 4.

實驗名稱：網路訊號遺失

a. 光電轉換器損壞或光纖受損

b. 無包覆雙絞線(UTP)內部線路受損

實驗目的： 模擬網路線因外力受損，造成訊號傳送不完全的狀況

實驗說明： 本實驗分爲兩部份，分別模擬光纖線路受損及一般 UTP 網路線受損。

實驗步驟：

a. 將實驗線路 4 的光纖拔掉一條，模擬光電轉換器受損或光纖受損的狀況，做法是先拔掉 Rx(收送端)測試完畢後，再拔掉 Tx(發送端)進行測試。
b. 將實驗線路 6 的 UTP 線的第一條及第二條線路(Tx)剪斷進行實驗，再將 UTP 線的第三條及第六條線路(Rx)剪斷進行實驗。

實驗結果：

光纖線路受損與 UTP 線路受損的實驗結果是相同的，並不會因爲線材的不同而產生不同的結果，當線路的 Rx 端受損，光纖訊號消失，或是 UTP 的第一或第二條線路受損時，網路狀態均與斷線時相同；當 Tx 端的線路受損時，光纖訊號消失或是 UTP 的第三或第六條線路受損時，網路狀態沒有改變，但資料流進量會變成零。請參考表 4.4 、圖 4.15 – 圖 4.17 。

表 4.4：網路訊號遺失的警訊集合

網路錯誤名稱	警訊集合	備註
網路訊號遺失(Rx 斷線)	a0103，a0202，a0204	視同網路斷線
網路訊號遺失(Tx 斷線)	a0202	

圖 4.15 – 圖 4.17 說明網路訊號遺失時 a0103、a0202 與 a0204 的變化狀況，在 01:56:00 至 01:58:00 是 Tx 斷線的狀況，在 02:01:00 至 02:03:30 是 Rx 斷線的狀況。

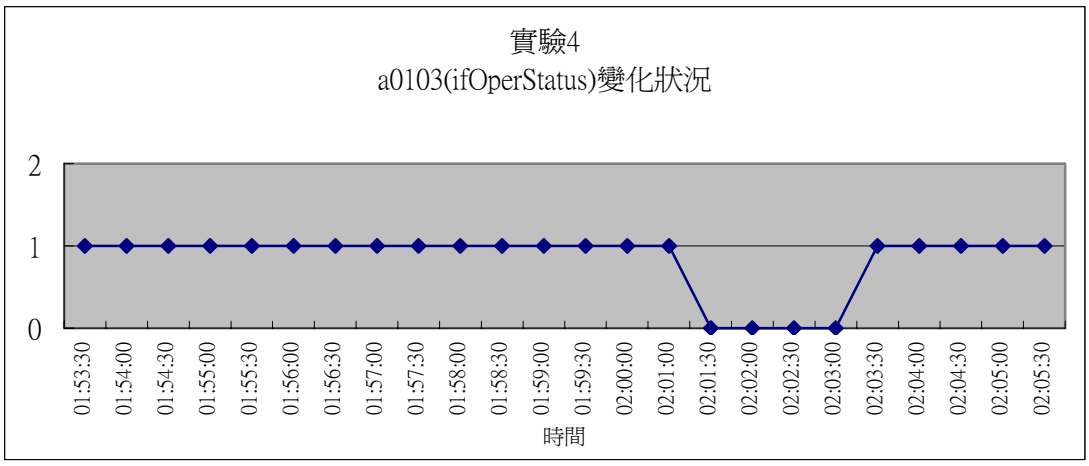

圖 4.15：實驗 4 警訊 a0103 數據變化圖

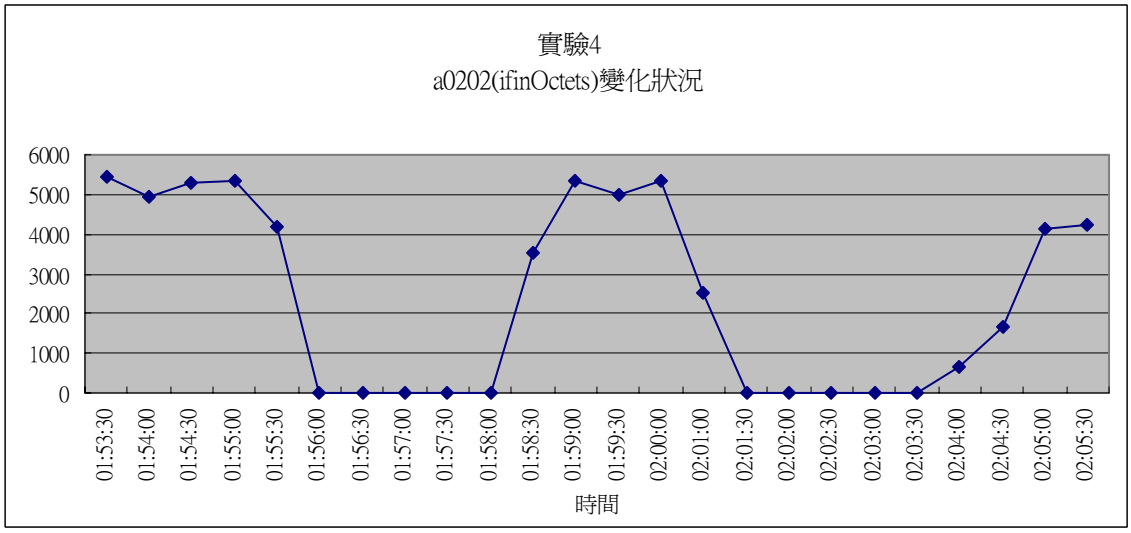

圖 4.16：實驗 4 警訊 a0202 數據變化圖

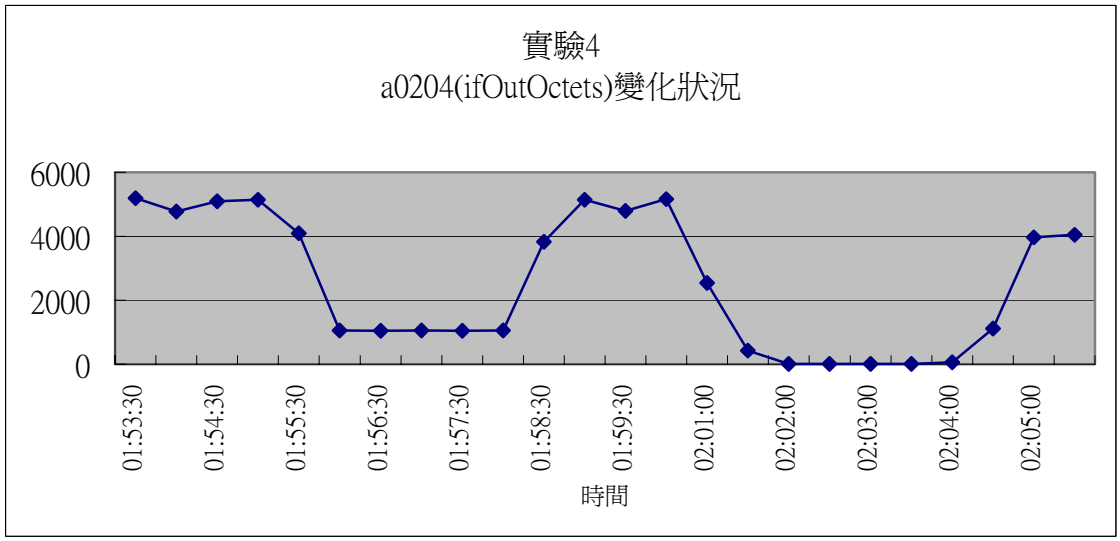

圖 4.17：實驗 4 警訊 a0204 數據變化圖

網路問題 5.

實驗名稱：ip address 重複

實驗目的：第一時間內發現 ip address 重複

實驗說明：模擬相同的 ip address 在同一網段上出現的狀況。

實驗步驟：在相同的網段上先確認 Server1 係正常運作，再另外設定一台未上網的機器與 Server1 有相同的 ip address，設定完畢後再接上網路，模擬同網段 ip address 重複的狀況。使用不同廠牌的網路卡與使用不同的作業系統重複相同的實驗，記錄並比較結果。

實驗結果：當發生 ip address 重複時，只能在第三層的網路設備判斷 ip address 原本對應的 mac address 更動了，而且在短時間內又重複發生。本實驗會因爲廠牌及效能不同的網路卡以及作業系統不同而產生不同的結果，因爲在相同的 ip 下效能好的網路卡或作業系統會搶得較多的封包，或是兩台機器互相搶奪 ip 的狀況出現，造成 IpNetToMediaPhysAddress 警訊會有變動情況，但變動的狀況卻不相同的情況出現。請參考表 4.5、表 4.6。

表 4.5：ip address 重複 的警訊集合

網路錯誤名稱	警訊集合	備註
ip address 重複	A1201	IpNetToMediaPhysAddress

表 4.6：a1201 警訊的變化狀況

Router 位置	警訊號碼	MAC Address	時間	警訊啓動
192.168.1.2	a1201	00 80 c8 e9 32 12	00:03:30	0
192.168.1.2	a1201	00 80 c8 59 aa 48	00:04:00	1

192.168.1.2	a1201	00 80 c8 59 aa 48	00:04:30	0
192.168.1.2	a1201	00 80 c8 e9 32 12	00:05:00	1
192.168.1.2	a1201	00 80 c8 59 aa 48	00:05:30	1
192.168.1.2	a1201	00 80 c8 e9 32 12	00:06:00	1
192.168.1.2	a1201	00 80 c8 e9 32 12	00:06:30	0
192.168.1.2	a1201	00 80 c8 59 aa 48	00:07:00	1
192.168.1.2	a1201	00 80 c8 e9 32 12	00:07:30	1

網路問題 6.

實驗名稱：網路擁塞 (utility high)

實驗目的：利用 Iperf 製造網路流量，模擬大量使用者使用網路的狀況

實驗說明：網路擁塞是最常見的網路問題之一，雖然網路尚可使用，但效率低落，造成的原因相當的多，通常發生在主機群的對外線路及一個單位的對外線路上。

實驗步驟：利用 Iperf 製造網路流量，模擬大量使用者使用網路的狀況，爲避免流量太大造成瞬間網路癱瘓，製造網路流量時是從小流量逐步增加至大流量，一開始模擬 5 位使用者，增加爲 10 位再增加到 25 位。

實驗結果： 正常流量時，進出的網路封包數均在正常範圍，當流量過大錯誤封包及封包重送的比率也會增加。警訊集合也會因爲封包大小的不同而有所改變。主要的警訊集合爲 a0201 與 a0203，會因爲封包大小，流量大小而產生多種變化。請參考表 4.7、圖 4.18 及圖 4.19。

表 4.7：網路擁塞的警訊集合

網路錯誤名稱	警訊集合	備註
網路擁塞 (utility high)	a. a0201,a0203,a0302 b. a0107,0203	

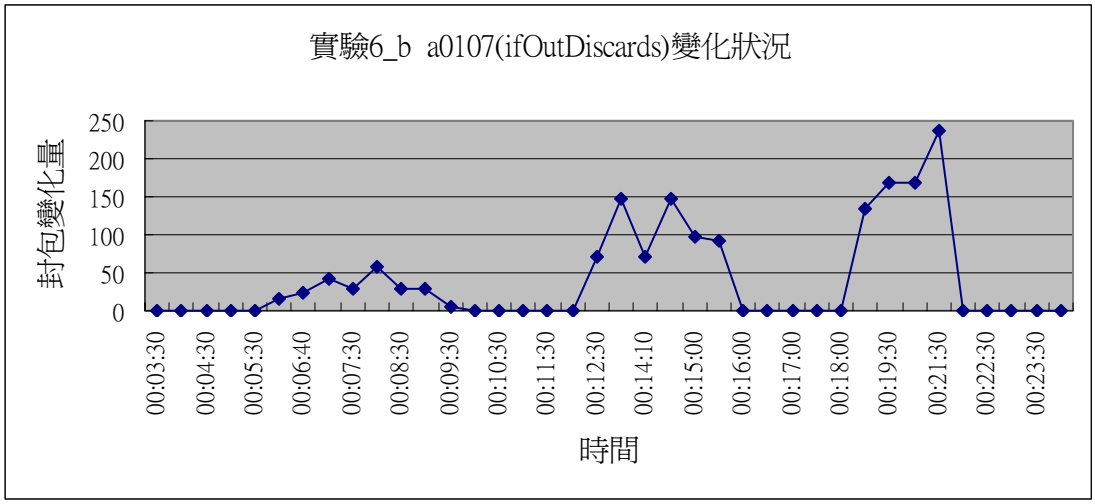

圖 4.18：實驗 6 警訊 a0107 數據變化圖

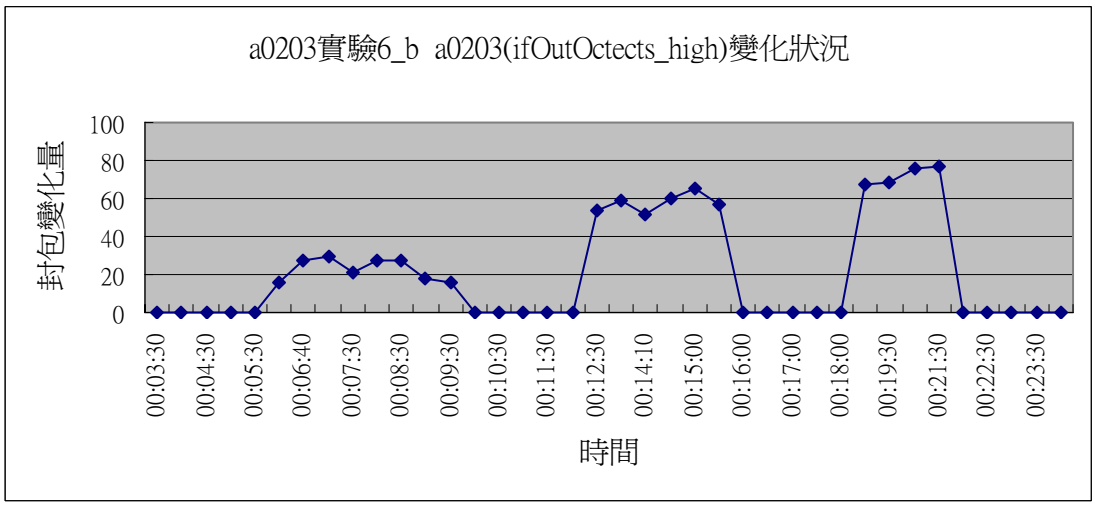

圖 4.19：實驗 6 警訊 a0203 數據變化圖

網路問題 7.

實驗名稱：迴圈 / Broadcast Storm

實驗目的： 網路迴圈形成後，網路設備若無 Spanning Tree 的功能，將會導至資料在迴圈內重覆傳送，若有廣播封包更會因此造成廣播風暴，了解不同大小的迴圈及封包種類在迴圈的狀況下對網路設備造成影響。

實驗說明：網路拓樸形成迴圈，通常是使用者誤接線路所造成，該種錯誤常常致使網路效能低落，是一種會讓網路無法使用的錯誤，且網路設備的效能越好，造成的嚴重性與範圍越大。

實驗步驟：1. 在單台 Switch 製造迴圈，並啓用網路芳鄰，造成 Broadcast Storm，並在 spanning tree 沒有啓動及啓動的狀況下觀察網路運作狀況。

2. 在兩台 Switch 間製造迴圈並使用 ping 指令對廣播位置發出 ICMP 封包，以製造廣播風暴，並在 spanning tree 沒有啓動及啓動的狀況下觀察網路運作狀況。

3. 在單台 Switch 及 兩台 Switch 間製造迴圈，利用 Iperf 模擬一般的網路流量，觀察並記錄網路運作狀況。

實驗結果：

使用網路芳鄰與 ping 指令均能在迴圈的狀況下產生廣播風暴對網路造成相當的負荷，硬體上產生迴圈的範圍越小或是網路設備越快所造成網路不正常運作的速度也越快。網路設備若有提供 spanning tree 的功能，當迴圈產生時，會自動將產生迴圈的其中一個介面 Blocking 住，其他介面的運作則不受影響，被 Blocking 的介面會啓動 a1105、a0202 及 a0204 的警訊。網路設備若沒有提供 spanning tree 的功能，則會因爲網路設備負載過高，所有的 MIB 接無法讀取，形同當機。

a. spanning tree 啓動：銜接迴圈的兩個 interface 其中會有一個 interface 的 a1105 會被啓動，也就是 port state 會從 Forwarding 變成 Blocking 。

b. spanning tree 沒有啓動：網路設備當機。

請參考表 4.8、圖 4.20、圖 4.21 及圖 4.22。

表 4.8：迴圈的警訊集合

網路錯誤名稱	警訊集合	備註
迴圈 / Broadcast Storm	a1105,a0202,a0204	需設備支援

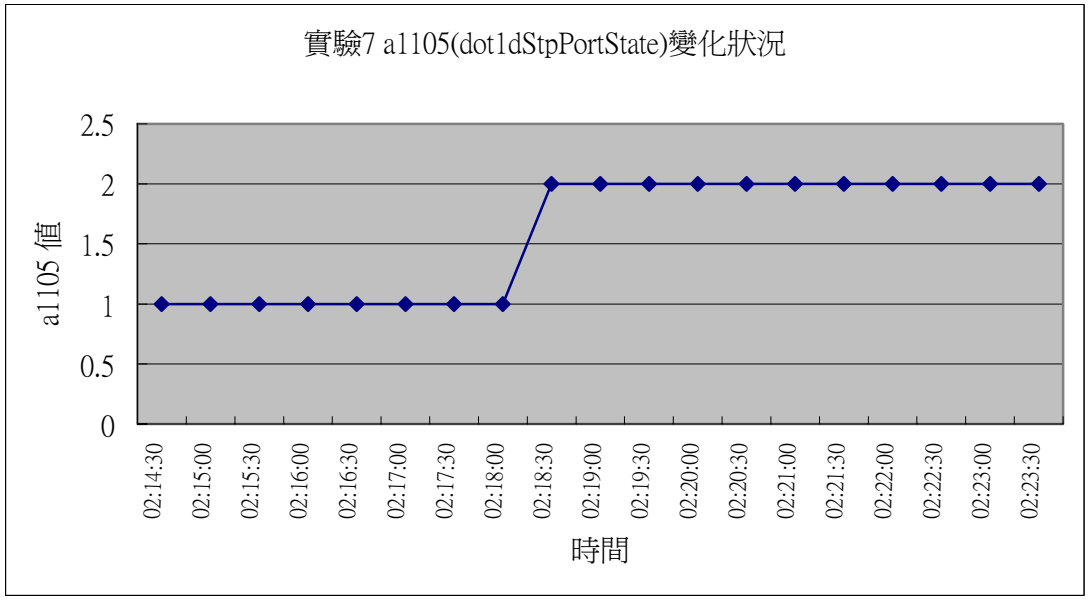

圖 4.20：實驗 7 警訊 a1105 數據變化圖

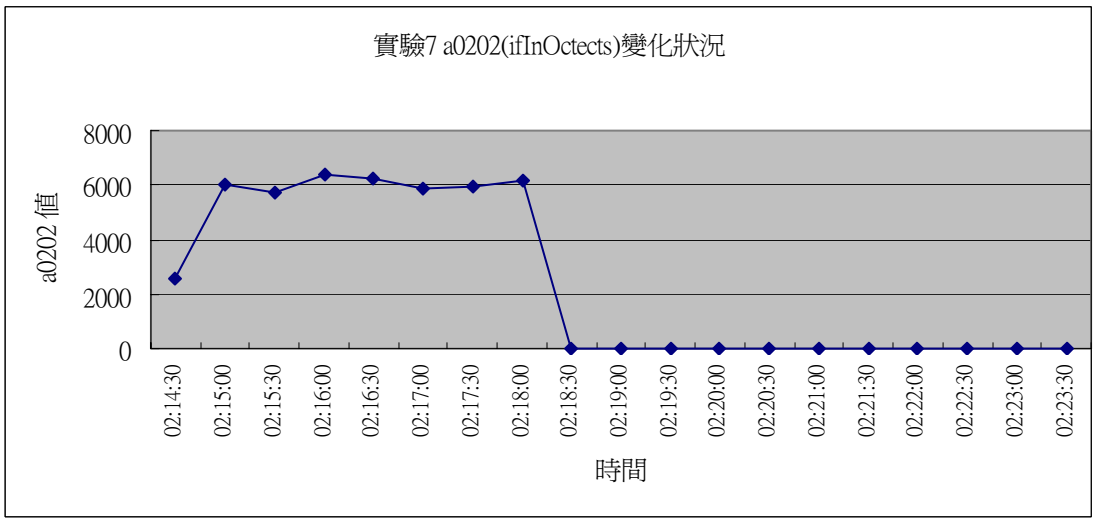

圖 4.21：實驗 7 警訊 a0202 數據變化圖

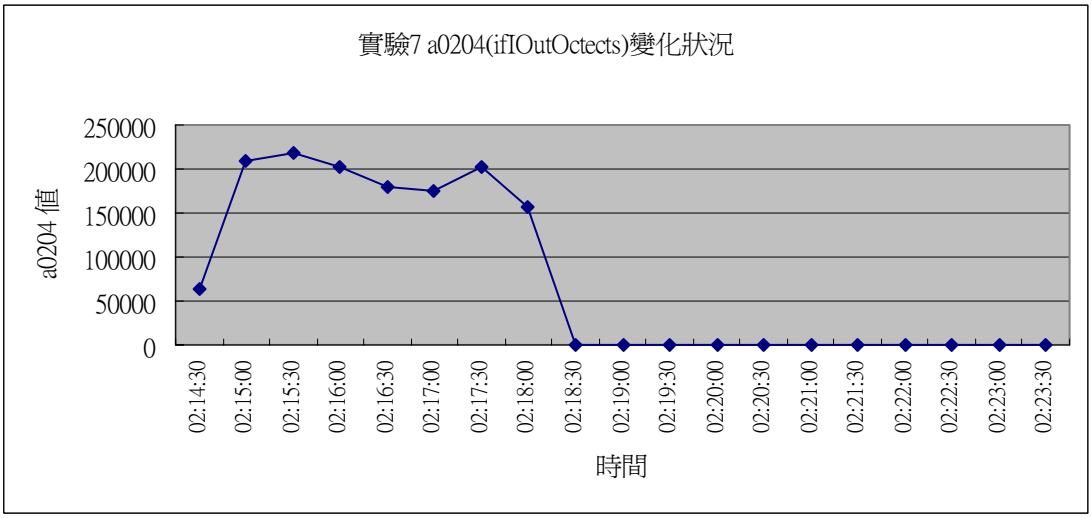

圖 4.22：實驗 7 警訊 a0204 數據變化圖

網路問題 8.

實驗名稱：網路設備間全／半雙工不協調

實驗目的：網路設備若因全／半雙工不一致，會導致資料傳送速度變慢

實驗說明：因爲 hub 只支援半雙工，我們選擇在 Switch 或 router 間做實驗，主要原因在於該兩種設備均能調整全／半雙工的傳輸模式。

實驗步驟：將 Switch1 調整傳輸模式爲全雙工， Router1 調整爲半雙工，先在一般狀態下觀察，經過 3-5 分鐘，使用 IPerf 製造網路流量，再經過 3-5 分鐘結束實驗，期間觀察並記錄網路運作狀況。

實驗結果：若互相連接的網路設備全半雙工不協調，會因爲資料 Collection 造成網路效能低落，其中封包 collection 的狀況最爲明顯，且網路資料傳輸量越大，所造成的影響也越大。請參考 表 4.9、圖 4.23 及圖 4.24。

表 4.9：網路設備間全半雙工不協調的警訊集合

網路錯誤名稱	警訊集合	備註
網路設備間全半雙工不協調	a1307,a0302	

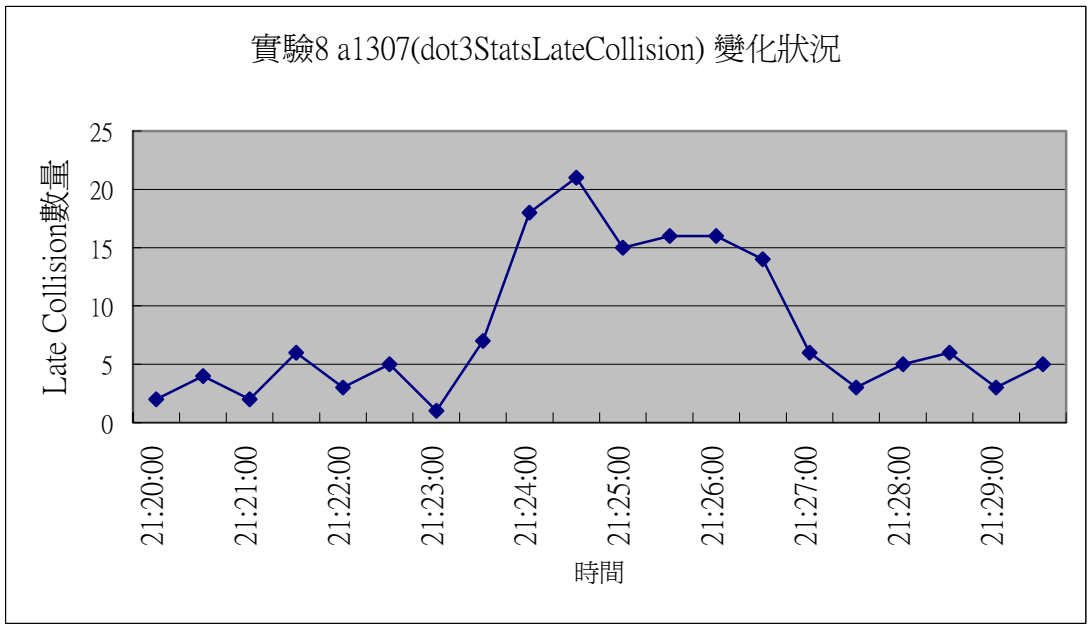

圖 4.23：實驗 8 警訊 a1307 數據變化圖

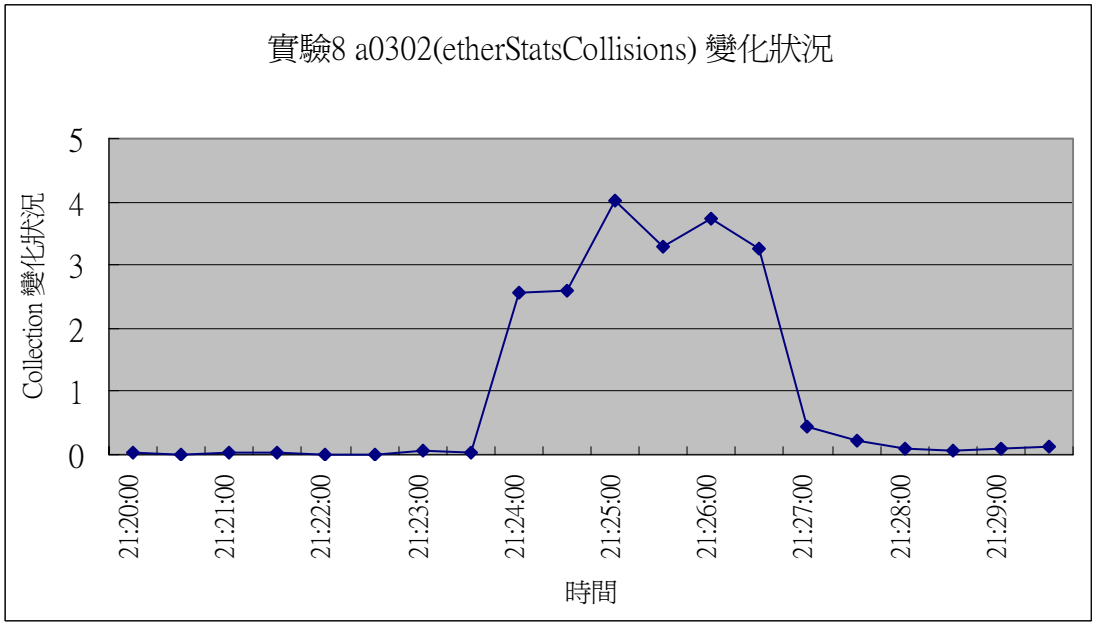

圖 4.24：實驗 8 警訊 a0302 數據變化圖

4.5 模糊推論實驗與結果

依據前一節實驗結果建立的案例資料庫如下：

表 4.10：經實驗建立的案例資料庫

編號	網路錯誤的表示法	說明
1	F1 ➔ a0103,a0202,a0204	網路斷線 (Dropped line)
2	F2 ➔ a0202,a0204 F2 ➔ c0101	主機當機 (Server Crash)
3	F2 ➔ a0202,a0204 F2 ➔ b0301,b0302,b0303,b0402,c0101	
4	F3 ➔a0108	主機重開 (Server Reboot)
5	F4 ➔a0202	網路訊號遺失
6	F5 ➔a1201	ip address 重複
7	F6 ➔ a0201,a0203,a0302	網路擁塞 (utility high)
8	F6 ➔ a0107,0203	
9	F7 ➔ a1105,a0202,a0204	迴圈 (loop) / 廣播風暴 (Broadcast Storm)
10	F8 ➔ a1307,a0302	網路設備間全／半雙工不協調

4.5 模糊理論爲基礎的網路錯誤推論系統

至此我們初步建立了案例資料庫、模糊規則庫與模糊語意資料庫，我們選擇了四個網路錯誤來進行實驗，並以模糊理論爲基礎的網路錯誤推論系統來判斷

網路問題，實驗是在開放式的網路架構下進行，爲了實驗能夠貼近現實的狀況，我們選擇凌晨的時段在東海大學主幹上銜接我們的設備進行實驗，將影響層面降至最低，造成錯誤的方法也與先前實驗使用不同的流量或封包大小，藉此驗證模糊推論的準確性。

模糊推論實驗一：

實驗名稱：網路斷線

實驗步驟：將所觀察的網路設備間的連線拔除。

實驗結果：a0103,a0204

案例比對：沒有符合案例

推論過程：

編號	網路錯誤的表示法	推論結果
1	F1 ➔ a0103,a0202,a0204 α = min (1，0.8，1) = 0.8 反模糊化 0.8 / 100% = 80%	80%
2	F2 ➔ a0202,a0204 F2 ➔ c0101 α = min (1，1，0.15) = 0.15 反模糊化 0.15 / 100% = 15%	15%
3	F2 ➔ a0202,a0204 F2 ➔ b0301,b0302,b0303,b0402 α = min (1，1，0，0，0，0) = 0	0
4	F3 ➔a0108	0
5	F4 ➔a0202	0
6	F5 ➔a1201	0
7	F6 ➔ a0201,a0203,a0302	0
8	F6 ➔ a0107,0203	

9	F7 ➔ a1105,a0202,a0204	0
	α = min (0，1，1) = 0	
10	F8 ➔ a1307,a0302	0

推論結果：推論網路斷線有 80% 發生可能性

案例回饋：新增 F1 ➔ a0103,a0204 的警訊組合。此錯誤發生時，a0103，a0204 均已超過臨界值，不用修正模糊語意資料庫。

模糊推論實驗二：

實驗名稱：主機重開

實驗步驟：將所觀察的主機關機後重新啓動。

實驗結果：a0108

案例比對：案例比對成功

案例回饋：本案例發生的次數增加一次。

模糊推論實驗三：

實驗名稱：網路擁塞

實驗步驟：利用 Iperf 使用小封包製造網路流量，模擬大量使用者使用網路的狀況。

實驗結果： a0203，a0302，a0310

案例比對：沒有符合案例

推論過程：

編號	網路錯誤的表示法	推論結果
1	F1 ➔ a0103,a0202,a0204	0
	F1 ➔ a0103,a0204	0
2	F2 ➔ a0202,a0204	0
	F2 ➔ c0101	

3	F2 ➔ a0202,a0204 F2 ➔ b0301,b0302,b0303,b0402	0
4	F3 ➔a0108	0
5	F4 ➔a0202	0
6	F5 ➔a1201	0
7	F6 ➔ a0201,a0203,a0302 α = min (0.64，1，1) = 0.64 反模糊化 0.64 / 100% = 64%	64%
8	F6 ➔ a0107,0203 α = min (0.03，1) = 0.03 反模糊化 0.03 / 100% = 3%	3%
9	F7 ➔ a1105,a0202,a0204	0
10	F8 ➔ a1307,a0302 α = min (0，1) = 0	0

推論結果：推論網路擁塞有 64% 發生可能性，實際上亦是網路擁塞。

案例回饋：新增 F6 ➔ a0203，a0302，a0310 的警訊組合。

模糊推論實驗四：

實驗名稱：使用大量 ICMP 封包攻擊網路主機

實驗步驟： 使用多台工作站，用 PING 指令及 Iperf 發生大量 ICMP 封包攻擊特定主機。

實驗結果：a0203，a0205，a0306，a0309

案例比對：沒有符合案例

推論過程：

編號	網路錯誤的表示法	推論結果
1	F1 ➔ a0103,a0202,a0204	0

2	F2 ➔ a0202,a0204	0
	F2 ➔ c0101	
3	F2 ➔ a0202,a0204	0
	F2 ➔ b0301,b0302,b0303,b0402	
4	F3 ➔a0108	0
5	F4 ➔a0202	0
6	F5 ➔a1201	0
7	F6 ➔ a0201,a0203,a0302	42%
	α = min (0.42，1，0.63) = 0.42	
	反模糊化 0.42 / 100% = 42%	
8	F6 ➔ a0107,0203	35%
	α = min (0.35，1) = 0.35	
	反模糊化 0.35 / 100% = 35%	
9	F7 ➔ a1105(),a0202,a0204	0
10	F8 ➔ a1307,a0302	0

推論結果：推論出 42% 及 35% 均是符合網路擁塞

案例回饋：新增 F6 ➔ a0203，a0205，a0306，a0309 的警訊組合。

或是新增網路錯誤 F9 ICMP 封包攻擊，F9 ➔ a0203，a0205，a0306，a0309

本實驗案例回饋方式有兩種，兩種回饋方式均須經過網管人員實際確認後推論結果始可將資料回饋至資料庫中。第一種爲案例比對成功，且正確解決網路問題，模糊推論實驗二即屬於第一種，第二種爲模糊推論結果能正確解決網路問題，模糊推論實驗一、三、四即屬於第二種。若是案例比對成功，則該案例發生次數增加一次，若爲模糊推論結果正確，則增加案例種類或是增加案例的警訊組合，並調整原有 MIB 的臨界值。

第五章、結論與未來發展

5.1 結論

本研究的目的是希望能從主動監控網路設備的角度，自動推論網路錯誤，在第一時間告知網路管理人員在何時(when)、何處(where)、發生何種可能狀況(what)的相關資訊，在網管人員確認網路問題後，能夠順利解決問題並能將資料回饋給案例資料庫及模糊規則庫讓系統能自我成長，本研究除了協助網管人員判斷網路問題外，也進一步希望提供網路狀況供管理人員規劃與設定網路政策，讓網路與主機設備不但在出問題時，能在最短時間復原，更期望能夠讓網路設備能以最經濟的規模達到最大的效能。

本研究與其他網路偵測系統最大的不同，在於其他的系統只是偵測及比對，重點在於偵測及比對的方法改善，若是警訊不同便無法順利找出結果，我們的重點則在於推論知識的建立與推論過程，能以較少的資訊找出最接近的答案。

5.2 未來發展

未來的發展如下：

一、 增加案例及資料庫內容：

繼續蒐集與分析網路各故障資訊，增加新的 MIB 以補充更多的網路管理資訊，進一步增加網路錯誤推論引擎的範圍及效能。

二、 歸屬函數的合適性：

在本文中 MIB 的歸屬函數及反模糊化時，所採用的規屬函數均採用線性的 01 函數，雖然在整體的推論過程中不至影響推論結果的次序，但就推論過程與最後反模糊化時的數據還有進一步改進的地方。

三、 特定警訊增加權重：

在推論過程中發現每一種錯誤雖然會有不同的警訊組合，但會有特定一個或數個警訊是一定會出現的，若是能以權重或是其他方式來加重特定警訊的份量，相信更能提高推論的結果。

四、 網路拓樸及其他網路工具的搭配：

原本希望能以最簡單的方式來表達網路錯誤，在實驗過程中發現，若是搭配網路拓樸及簡單網管工具，可以更清楚的表達出更多的網路錯誤。

值得注意及加強的是在蒐集所有的網管訊號的同時，是否也造成網路設備及網路本身過大的負擔，若是網管的流量與所造成的系統負載常常維持在原有流量及負載的十分之一以上時，就必須留意系統是否常有瓶頸現象，若有則網路系統中有部分設備或線路必須更換或升級，否則，因網管而造成網路瓶頸，就失去網路管理的意義了。

參考文獻

[1]. A Fuzzy expert system for network fault management: Jiann-Liang Chen / Pei-Hwa Huang

[2]. A.T. Bouloutas, S. Calo and A. Finkel, “Alarm correlation and fault identification in communication networks,” Communications, IEEE Transactions on , Volume: 42 Issue: 2 Part: 1 , Feb-Apr 1994, pp. 523 –533.

[3].Cisco QoS Device Manager Release 2.0, http://www.cisco.com/warp/public/cc/pd/nemnsw/qodvmn/prodlit/qdm_ds.htm，2004/01/03。

[4]. Donald A. Waterman ：A Guide to Expert System， ADDISON-WESLEY PUBLISHING COMPANY

[5]. Kickert, Walter J. M. /Martinus Nijhoff ：Fuzzy theories on decision-making : a critical review，Social Sciences Division ,1978;

[6].L. A. Steinberg, Troubleshooting with SNMP & Analyzing MIBs , McGraw-Hill,2000.

[7]. R.D. Gardner and D.A. Harle, “Fault resolution and alarm correlation in high-speed networks using database mining techniques” 1997. ICICS., Proceedings of 1997 International Conference on Information, Communications and Signal Processing, Volume: 3 , 1997, pp. 1423 –1427.

[8]. R.N. Cronk, P.H. Callahan and L. Bernstein, “Rule-based expert systems for network management and operations: an introduction,” IEEE Network , Volume: 2 Issue: 5 , Sept. 1988, pp. 7 –21.

[9]. Rule-based expert systems for network management and operations: an introduction :*Cronk, R.N.; Callahan, P.H.; Bernstein, L.* ,IEEE Network , Volume: 2 Issue: 5 , Sept. 1988 Page(s): 7 –21.

[10]. Terano, Toshir,/Asai, Kiyoji,：Fuzzy systems theory and its applications ， Academic Press ,1992

[11]. T.K. Apostolopoulos and V.C. Daskalou, **“**A model for SNMP based performance management services,” Networks, 1995. Theme: Electrotechnology 2000: Communications and Networks. [in conjunction with the] International Conference on Information Engineering., Proceedings of IEEE Singapore International Conference on , 1995, Pp. 269 –273.

[12].井上洋/天笠美知夫原著 陳耀茂譯：模糊理論 / fuzzy theory，五南圖書出版公司.

[13].余禎祥, “以實驗爲基礎之區域網路錯誤行爲知識庫” 逢甲大學 資訊工程學系,中華民國八十八年六月碩士論文。

[14].李成泰, “儲存爲基的 TCP/IP 區域網路拓樸探索與管理系統之建立”輔仁大學 資訊管理學系 中華民國八十八年碩士論文

[15].李信宏, “以時間爲考量之網路錯誤推理診斷” 逢甲大學 資訊工程學系 中華民國八十九年碩士論文

[16].秉昱科技編譯：模糊邏輯與類神經模糊在商業和財政的應用，儒林圖書有限公司。

[17].陳英材, “知識爲基網路錯誤管理系統之研究與建立” 輔仁大學 資訊管理學系 中華民國八十八年碩士論文

[18]. 楊英魁/孫宗榮/鄭魁香/林建德/蔣旭堂 編著：模糊控制理論與技術，金華科技圖書股份有限公司

[19].謝瑞宏, “分散式多重智慧型代理軟體爲基之網路流量擷取系統之研究” 輔仁大學 資訊管理學系, 中華民國八十九年六月碩士論文。

附錄 A、警訊對應的 OID 及 Threshold

警訊號碼	MIB 物件 ID	比對方式	臨界值	資料形態
a0001	.1.3.6.1.2.1.2.2.1.5		0	none
a0101	.1.3.6.1.2.1.2.2.1.6	<>	0	delta
a0102	.1.3.6.1.2.1.2.2.1.7	<>	0	none
a0103	.1.3.6.1.2.1.2.2.1.8	<>	0	none
a0104	.1.3.6.1.2.1.2.2.1.13	>	1	rate
a0105	.1.3.6.1.2.1.2.2.1.14	>	1	rate
a0106	.1.3.6.1.2.1.2.2.1.15	>	1	rate
a0107	.1.3.6.1.2.1.2.2.1.19	>	1	rate
a0108	.1.3.6.1.2.1.1.3	<	0	delta
a0109	.1.3.6.1.2.1.2.2.1.20	>	1	rate
a0201	.1.3.6.1.2.1.2.2.1.10	>	70	percent
a0202	.1.3.6.1.2.1.2.2.1.10	<	32	percent
a0203	.1.3.6.1.2.1.2.2.1.16	>	70	percent
a0204	.1.3.6.1.2.1.2.2.1.16	<	32	percent
a0205	.1.3.6.1.2.1.2.2.1.11	>	90	packrate
a0206	.1.3.6.1.2.1.2.2.1.17	>	90	packrate
a0207	.1.3.6.1.2.1.2.2.1.12	>	25	packrate
a0208	.1.3.6.1.2.1.2.2.1.18	>	25	packrate
a0301	.1.3.6.1.2.1.16.1.1.1.8	>	1	rate
a0302	.1.3.6.1.2.1.16.1.1.1.13	>	1	rate
a0303	.1.3.6.1.2.1.16.1.1.1.11	>	1	rate
a0304	.1.3.6.1.2.1.16.1.1.1.4	>	60	percent
a0305	.1.3.6.1.2.1.16.1.1.1.10	>	1	rate
a0306	.1.3.6.1.2.1.16.1.1.1.5	>	500	rate
a0307	.1.3.6.1.2.1.16.1.1.1.9	>	1	rate
a0308	.1.3.6.1.2.1.16.1.1.1.12	>	1	rate
a0309	.1.3.6.1.2.1.16.1.1.1.14	>	1000	rate
a0310	.1.3.6.1.2.1.16.1.1.1.15	>	1000	rate
a0311	.1.3.6.1.2.1.16.1.1.1.16	>	1000	rate
a0312	.1.3.6.1.2.1.16.1.1.1.17	>	1000	rate
a0313	.1.3.6.1.2.1.16.1.1.1.18	>	1000	rate
a0314	.1.3.6.1.2.1.16.1.1.1.19	>	1000	rate
a0401	.1.3.6.1.4.1.9.2.1.57	>	50	none

a0402	.1.3.6.1.4.1.9.2.1.58	>	40	none
a0501	.1.3.6.1.4.1.9.2.1.46	>	15	delta
a0502	.1.3.6.1.4.1.9.2.1.47	>	35	delta
a0701	.1.3.6.1.2.1.4.4	>	100	rate
a0702	.1.3.6.1.2.1.4.5	>	10	rate
a0703	.1.3.6.1.2.1.4.7	>	1	rate
a0704	.1.3.6.1.2.1.4.8	>	10	rate
a0705	.1.3.6.1.2.1.4.11	>	10	rate
a0706	.1.3.6.1.2.1.4.12	>	100	rate
a0707	.1.3.6.1.2.1.4.16	>	10	rate
a0708	.1.3.6.1.2.1.4.17	>	100	rate
a0709	.1.3.6.1.2.1.4.18	>	10	rate
a0710	.1.3.6.1.2.1.4.23	>	50	rate
a0801	.1.3.6.1.2.1.6.7	>	5	rate
a0802	.1.3.6.1.2.1.6.8	>	5	rate
a0803	.1.3.6.1.2.1.6.12	>	5	rate
a0804	.1.3.6.1.2.1.6.14	>	5	rate
a0805	.1.3.6.1.2.1.6.15	>	5	rate
a0806	.1.3.6.1.2.1.7.2	>	50	rate
a0807	.1.3.6.1.2.1.7.3	>	50	rate
a0901	.1.3.6.1.2.1.11.4	>	5	rate
a0902	.1.3.6.1.2.1.11.6	>	5	rate
a1001	.1.3.6.1.2.1.5.1	>	10	rate
a1002	.1.3.6.1.2.1.5.2	>	10	rate
a1003	.1.3.6.1.2.1.5.3	>	10	rate
a1004	.1.3.6.1.2.1.5.4	>	10	rate
a1005	.1.3.6.1.2.1.5.6	>	10	rate
a1006	.1.3.6.1.2.1.5.7	>	10	rate
a1007	.1.3.6.1.2.1.5.14	>	10	rate
a1008	.1.3.6.1.2.1.5.15	>	10	rate
a1009	.1.3.6.1.2.1.5.16	>	10	rate
a1010	.1.3.6.1.2.1.5.17	>	10	rate
a1011	.1.3.6.1.2.1.5.19	>	10	rate
a1012	.1.3.6.1.2.1.5.20	>	10	rate
a1101	.1.3.6.1.2.1.17.2.4	>	5	rate
a1102	.1.3.6.1.2.1.17.4.1	>	10	rate

a1103	.1.3.6.1.2.1.17.1.4.1.4	>	10	rate
a1104	.1.3.6.1.2.1.17.1.4.1.5	>	10	rate
a1105	.1.3.6.1.2.1.17.2.15.1.3	=	2	none
a1106	.1.3.6.1.2.1.17.2.15.1.10	<>	0	delta
a1201		>	1	duplicate
a1301	.1.3.6.1.2.1.10.7.2.1.2		0	data
a1302	.1.3.6.1.2.1.10.7.2.1.3		0	data
a1303	.1.3.6.1.2.1.10.7.2.1.4		0	data
a1304	.1.3.6.1.2.1.10.7.2.1.5		0	data
a1305	.1.3.6.1.2.1.10.7.2.1.6		0	data
a1306	.1.3.6.1.2.1.10.7.2.1.7		0	data
a1307	.1.3.6.1.2.1.10.7.2.1.8		0	data
a1308	.1.3.6.1.2.1.10.7.2.1.9		0	data
a1309	.1.3.6.1.2.1.10.7.2.1.10		0	data
a1310	.1.3.6.1.2.1.10.7.2.1.11		0	data
a1311	.1.3.6.1.2.1.10.7.2.1.13		0	data
a1312	.1.3.6.1.2.1.10.7.2.1.16		0	data
a1313	.1.3.6.1.2.1.10.7.2.1.17		0	data
b0301	.1.3.6.1.4.1.2021.10.1.3.1	>	5	none
b0302	.1.3.6.1.4.1.2021.10.1.3.2	>	4	none
b0303	.1.3.6.1.4.1.2021.10.1.3.3	>	3.5	none
b0304	.1.3.6.1.4.1.2021.11.9.0		0	none
b0305	.1.3.6.1.4.1.2021.11.10.0		0	none
b0306	.1.3.6.1.4.1.2021.11.11.0		0	none
b0401	.1.3.6.1.4.1.2021.4.5.0		0	none
b0402	.1.3.6.1.4.1.2021.4.11.0		0	none
b0403	.1.3.6.1.4.1.2021.4.6.0		0	none
b0201	.1.3.6.1.2.1.2.2.1.10.2			percent
b0202	.1.3.6.1.2.1.2.2.1.10.2			percent
b0203	.1.3.6.1.2.1.2.2.1.16.2			percent
b0204	.1.3.6.1.2.1.2.2.1.16.2			percent
c0101		>	0.3	lostratio

附錄 B、警訊號碼種類特性說明

警訊號碼	警訊名稱	特性說明	Type
a0001	portspeed	網路介面速度	none
a0101	ifPhysAddress	網路介面 MAC Address	delta
a0102	ifAdminStatus	管理者操作網路介面狀況	none
a0103	ifOperStatus	網路介面運作狀況	none
a0104	ifInDiscards	流進網路介面被捨棄封包數	rate
a0105	ifInErrors	因有錯誤被放棄流進網路介面的封包數	rate
a0106	ifInUnknownProtos	不明協定流進網路介面被捨棄的封包數	rate
a0107	ifOutDiscards	流出網路介面被捨棄的封包數	rate
a0108	sysUpTime	系統開機時間	delta
a0109	ifOutErrors	因有錯誤被放棄流出網路介面的封包	rate
a0201	ifInOctets	流進網路介面的 byte 數(過多)	percent
a0202	ifInOctets	流進網路介面的 byte 數(過少)	percent
a0203	ifOutOctets	流出網路介面的 byte 數(過多)	percent
a0204	ifOutOctets	流出網路介面的 byte 數(過少)	percent
a0205	ifInUcastPkts	流進網路介面的 Unicast 封包數量	packrate
a0206	ifOutUcastPkts	流出網路介面的 Unicast 封包數量	packrate
a0207	ifInNUcastPkts	流進網路介面的非 Unicast 封包數量	packrate
a0208	ifOutNUcastPkts	流出網路介面的非 Unicast 封包數量	packrate
a0301	etherStatsCRCAlignErrors	CRC 錯誤的封包統計數量	rate
a0302	etherStatsCollisions	碰撞的封包統計數量	rate
a0303	etherStatsFragments	Fragment 的封包統計數量	rate
a0304	etherStatsOctets	Byte 總數的統計數量	percent
a0305	etherStatsOversizePkts	超大封包的統計數量	rate
a0306	etherStatsPkts	封包總數的統計數量	rate
a0307	etherStatsUndersizePkts	過小封包的統計數量	rate
a0308	etherStatsJabbers	破碎封包的統計數量	rate
a0309	etherStatsPkts64Octets	64 byte 以下的封包數	rate
a0310	etherStatsPkts65to127Octets	65 – 127 byte 的封包數	rate
a0311	etherStatsPkts128to255Octets	128 – 255 byte 的封包數	rate
a0312	etherStatsPkts256to511Octets	255 – 511 byte 的封包數	rate
a0313	etherStatsPkts512to1023Octets	512 – 1023 byte 的封包數	rate

a0314	etherStatsPkts1024to1518Octets	1023 – 15180 byte 的封包數	rate
a0401	avgBusy1	Cisco Cpu 負載一分鐘平均數	none
a0402	avgBusy5	Cisco Cpu 負載五分鐘平均數	none
a0501	bufferFail	Cisco 存取 Buffer 失敗之次數	delta
a0502	bufferNoMem	Cisco 設備 Buffer 全滿之次數	delta
a0701	ipInHdrErrors	因表頭錯誤被捨棄的 datagram 數目	rate
a0702	ipInAddrErrors	因錯誤網址被捨棄的 datagram 數目	rate
a0703	ipInUnknownProtos	因不明協定被捨棄的 datagram 數目	rate
a0704	ipInDiscards	因缺少緩衝空間而被捨棄的已收 datagram 數目	rate
a0705	ipOutDiscards	因缺少緩衝空間而被捨棄的輸出 datagram 數目	rate
a0706	ipOutNoRoutes	因找不到路由而被捨棄的輸出 datagram 數目	rate
a0707	ipReasmFails	由 ip 重組的演算法算出的失敗數目	rate
a0708	ipFragOks	被成功分段的 datagram 數目	rate
a0709	ipFragFails	因無分段旗標，不能分段的 datagram 數目	rate
a0710	ipRoutingDiscards	被捨棄的路由紀錄數目	rate
a0801	tcpAttmptFails	未能連結成功的次數	rate
a0802	tcpEstabResets	連接重新啓動的次術	rate
a0803	tcpRetransSegs	錯誤重送的資料數目	rate
a0804	tcpInErrs	因格式錯誤而捨棄的資料數目	rate
a0805	tcpOutRsts	重新啓動次數	rate
a0806	udpNoPorts	收到目的埠沒有應用行程的的 UDP dadagram 數	rate
a0807	udpInErrors	無法傳送的 UDP datagram	rate
a0901	SnmpInBadCommunityNam	使用錯誤 SNMP community 名稱存取設備之數量	rate
a0902	SnmpInASNParseErrs	SNMP 解析 ASN.1 錯誤的數目	rate
a1001	icmpInMsgs	收到 ICMP 的訊息數量	rate
a1002	icmpInErrors	收到有錯誤的 ICMP 的訊息數量	rate
a1003	icmpInDestUnreachs	收到無法傳送至目的 ICMP 的訊息數量	rate
a1004	icmpInTimeExcds	收到逾時 ICMP 的訊息數量	rate
a1005	icmpInSrcQuenchs	收到的 ICMP 來源消失訊的息數量	rate
a1006	icmpInRedirects	收到的 ICMP 重新導向的訊息數量	rate

a1007	icmpOutMsgs	送出 ICMP 的訊息數量	rate
a1008	icmpOutErrors	送出有錯誤的 ICMP 的訊息數量	rate
a1009	icmpOutDestUnreachs	送出無法傳送至目的 ICMP 的訊息數量	rate
a1010	icmpOutTimeExcds	送出逾時 ICMP 的訊息數量	rate
a1011	icmpOutSrcQuenchs	送出的 ICMP 來源消失訊的息數量	rate
a1012	icmpOutRedirects	送出的 ICMP 重新導向的訊息數量	rate
a1101	dot1dStpTopChanges	STP 協定拓樸變動的數量	rate
a1102	dot1dTpLearnedEntryDiscar	STP Forwarding 資料庫封包被捨棄之數量	rate
a1103	dot1dBasePortDelayExceede	STP 協定封包傳輸延遲逾時被丟棄的封包數	rate
a1104	dot1dBasePortMtuExceededD	STP 協定封包超過大小被丟棄的封包數	rate
a1105	dot1dStpPortState	STP 協定 Port 之狀態	none
a1106	dot1dStpPortForwardTransi	STP 協定 Port 從 Learning 狀態變成 Forwarding 狀態之次數	delta
a1201	ARP_table		duplicate
a1301	dot3StatsAlignmentErrors		data
a1302	dot3StatsFCSErrors		data
a1303	dot3StatsSingleCollisionFrames		data
a1304	dot3StatsMultipleCollisionFrames		data
a1305	dot3StatsSQETestErrors		data
a1306	dot3StatsDeferredTransmis		data
a1307	dot3StatsLateCollisions		data
a1308	dot3StatsExcessiveCollisions		data
a1309	dot3StatsInternalMacTrans		data
a1310	dot3StatsCarrierSenseErrors		data
a1311	dot3StatsFrameTooLongs		data
a1312	dot3StatsInternalMacReceiveErrors		data
a1313	dot3StatsEtherChipSet		data
b0301	C_load1	主機 CPU 負載一分鐘平均值	none
b0302	C_load5	主機 CPU 負載五分鐘平均值	none
b0303	C_load10	主機 CPU 負載十分鐘平均值	none

b0304	C_user	主機 cpu 使用者使用比率	none
b0305	C_sys	主機 cpu 系統使用比率	none
b0306	C_idle	主機 cpu 閒置比率	none
b0401	M_total	主機記憶體總量	none
b0402	M_used	主機記憶體被使用總量	none
b0403	M_free	主機記憶體未被使用總量	none
b0201	Tra_in	主機流入資料量比率過高	percent
b0202	Tra_in	主機流入資料量比率過低	percent
b0203	Tra_out	主機流出資料量比率過高	percent
b0204	Tra_out	主機流出資料量比率過低	percent
c0101	MibLostRatio	MIB 接收遺失率	lostratio