A Multiple I/O Delta Network Based Multicast Switching Fabric

Chien-Hung Weng and Wen-Shyen E. Chen
Institute of Computer Science
National Chung-Hsing University
Taichung, Taiwan, ROC
Email: {chweng ,echen} @cs.nchu.edu.tw

摘要

隨著各項多媒體應用在網際網路(Internet)上快速地發展，對網路設備高頻寬、高傳輸率及高處理速度的需求不斷地提昇，其中交換系統(switching system)因為所提供的功能由支援 unicast traffi 進而須處理大量的 multicast 與 broadcast traff i 而日形重要。本論文提出一種以 Delta 網路為基礎的交換核心(switching fabric)，封包(packet 運用 Parallel-In-Parallel-Out 的方式，在 Delta 網路的交換元件(switching element, SE)間以 Multiple Input/Output 路徑傳送，藉以提高交換核心的效能。此外並提出一種有別於傳統 Delta 網路內以參考封包的目的地位址處理遶送(routing)的方法，在封包送入 Delt 網路前將其 unicast / multicas 的目的地位址轉換為 Multicast Tag 後附加於封包前，當封包在交換元件間遶送時，則參考 Multicast Tag 並依據交換元件在 Delt 網路內的位址以建立傳送路徑，而完成封包交換的工作。而且本系統在不外加額外處理系統的狀況下，封包僅須傳送過交換核心一次，就能完成 unicast 與 multicast 的傳送。最後我們根據所提出的架構利用電腦模擬的方式來分析其整體效能。

關鍵詞 ： High-speed Networks, Multicast, Delta Network, Switching

1 簡介

現今的整體寬頻網路(Integrated Broadband Networks)在技術上隨著網際網路(Internet)上各類應用的蓬勃發展而快速的進步，透過網際網路電話(Internet Telephony)可以以遠較長途電話更低廉的價格獲得更多元化且高品質的通話服務；視訊會議(Video Conference)可以讓人們免於長距離的奔波而仍能達到面對面溝的目的；高畫質且高解析度的影音傳送配合隨選視訊(Video-on-Demand)的服務，使得現代人能享受到更高品質的生活。各種網路應用不斷地日新月異也意味著高頻寬、高傳輸率及高處理速度的需求。在過去十年中因為網路技術的進步而大幅提昇了寬頻整體服務數位網路(Broadband Integrated Services Digital Network, B-ISDN)的服務品質與效益，而其中交換系統(switching system)的技術則扮演著關鍵性的角色。針對網路的交換系統，Ahmadi 與 Denze 依內部的架構將交換系統分為六大類[3]。Zegura 則以量化的方法針對不同的交換系統架構分析其成本與效率 [2]。Turner 與 Yamanaka 則以交換系統在 CMOS 積體電路的實現性上將其架構分為三大類，並以每一種架構的成本作分析比較[6]，其中單階層(single stage)的交換系統架構如 crossbar-based 在系統擴充時存在著硬體成平方倍數成長的複雜度($O(n^2)$)，因而導致硬體成本的昇高。而多階層(multistage)交換系統的架構在其硬體的實現上以模組化的方式建構，可達成擴充性的要求，對於每一個節點(node)而言，不須太複雜的硬體設計，甚至在節點內可加入緩衝器(buffer)以降低封包的遺失率(packet loss ratio)；此外，N x N 之類 multistage 的交換架構其硬體的複雜度為 O(NlogN)，當 N 增加時其每一個埠的成本僅微幅上昇而遠較 single stage 架構的 $O(N^2)$來得低，因而適合使用於高擴充性交換系統的設計上。

在 multistage 的交換架構中，Banyan 網路具有簡單的遶送(routing)方式及低硬體成本的特性，其交換元件的任一個輸入與輸出端間均有一條路徑連接；而 Delta 網路為 Banyan 網路的一種，交換元件可以根據接收到封包目的地位址的相對位元決定遶送路徑，亦即具有自我遶送(self-routing)的特性，因而在輸入與輸出間形成一條單點傳送的路徑，對於封包的 multicast 或 broadcast 也可以藉由封包在交換元件內以複製的方式達到多點傳送的目的。

以 Banyan 網路做為交換核心來建構網路時，能夠支援單點對多點(one-to-many)甚至多點對多點(many-to-many)的 multicas 功能是相當重要的一環，在 Banyan 網路的交換系統上有二種方案來實現 multica s 的功能：

(1) 複製網路(copy network, CN)與遶送網路(routin network, RN)的組合[7,8,10,11]
在以 Banyan 網路為架構的遶送網路前放置一個複製網路，封包在傳送前先送入複製網路複製出所需 multicast 的封包，遶送網路在收到這些封包後，再以點對點的傳送方式傳送至其 output port 而完成 multica s 的功能。

(2)在遶送網路上以遞迴(recursive 或 recycling)的方式完成 multicast [4,9,13]
將傳送至遶送網路 output port 的封包再傳回 input port，循環地完成 multicast 的工作，亦即同一個遶送網路除做封包遶送外也間接地產生 multicas 所需的封包。

其中，Turner 在[7,8,10]中以 CN 來複製 multicast 所需之 cell 數量，再經 distribution networ 先避開可能發生擁塞(congestion 的狀況，最後送至 RN 以點對點(point-to-point)方式完成 multicast 傳送工作。Lee 則在[11]中提出 Boolean Interval Splitt i n 的演算法，以解決前述複製的 cell 在 CN 中傳送會發生 blocking[1]的問題。對於運用 recursive 方法處理 multicast，Cusani 與 Sestini 在[9]提出的架構，以 multiplication factor (M)決定在一個 cycle 時每一個 input 所能產生連續位址之 output cel 的數量，使得 multicas 在 M=2 時能在最多為 $\log_2 N$ 的 cycle 下完成。Chen 與 Kumar 在[13]中根據 set-partitioning 的原理提出 cube 的演算法，而得到平均約 3 個 cycle 能完成 multica s 工作。

然而前述兩種交換系統的架構卻有其相對的缺點，對於前者而言：(1)增加額外複製封包的硬體(CN)，也就是需要 2 個 Banyan 網路的硬體。(2)封包的目的地位址須經由 table look u 的方式找出，增加額外的運算時間。(3)須對 multicast routing t a 做寫入處理，增加系統額外負

擔而影響整體效能；於後者的缺點則有：(1) re-cycle 的封包會佔用 Banyan 網路的頻寬而影響交換系統效能。(2) 經由交換核心外部 link 而 re-cycle 的封包在進入 Banyan 網路前可能會與其他的 multicast packe 因為 contention 而有延遲的現象，此種延遲在 multicast rate 高時將更加明顯。

因此，下一章我們將提出另一種 multicas 的架構，以 Delta 網路為基礎的交換核心並利用 output buffer 為緩衝處理 contention，且在交換元件間以 multiple path 傳送封包，以達到高效能的目的。本論文的編排方式為：第 2 章詳述我們所提出在 Delta 網路上封包遞送的方法以及交換系統與內部交換元件的架構，第 3 章為電腦模擬的結果與分析，第 4 章則是結論及未來研究方向。

2 Multiple I/O Delt 網路的交換系統架構

Delta 網路[5] 建構的方式可以視為利用(2×2) crossbar 的交換元件(SE)組成一個二元的解多工樹(2-ary de-multiplexer tree)，每一個(2×2)的 SE 有 2 個輸入端 0 與 1 以及二個輸出端 link 0 及 link 1，封包從任一個輸入端進入 SE 後若目的地位址是 0 則往 link 0 傳送，反之若目的地位址是 1 則往 link 1 傳送，因此一個 stage 的 SE 可以解碼 1 個位元的位址，若輸出端有 N 個時，則須有 $N \log_2 N$ 個 stage 解碼才能讓封包傳送至目的地。此外，在 SE 中也可以加入複製封包的機制，讓封包送入 SE 後同時往 link 0 及 link 傳送，如此很容易地就可以達到 multica s 或 broadcast 的目的。

2.1 Delta 網路的建構方式

一個$(2^n \times 2^n)$的 Delt 網路建構的原則如下[12]：

(1)從 Delta 網路的 input port 開始往 output port 以 0 到 n-1 的整數表示 stage 的編號，即 stage 0 至 stage (n-1)。

(2)由 0 開始，以 n 個 2 為底的一連串位元表示 input port 或 output port 的位址，即 $P_0P_1\cdots P_{n-2}P_{n-1}$，其中 P_0 表示最高位元(MSB)。

(3)每一個 stage 的 SE 其位址由上至下以 n 個 2 為底的位元表示 $E_0E_1\cdots E_{n-2}E_{n-1}$，其中 E_0 為 MSB 並設為 0 (即 $0E_1\cdots E_{n-2}E_{n-1}$)。

(4)每一個 SE 在 Delta 網路的相對位置以<x,y>表示，其中 x 代表第 x 個 stage ($0 \le x \le n-1$)，y 代表由上而下的第 y 個 SE($y=E_0E_1\cdots E_{n-2}E_{n-1}=0E_1\cdots E_{n-2}E_{n-1}$)。

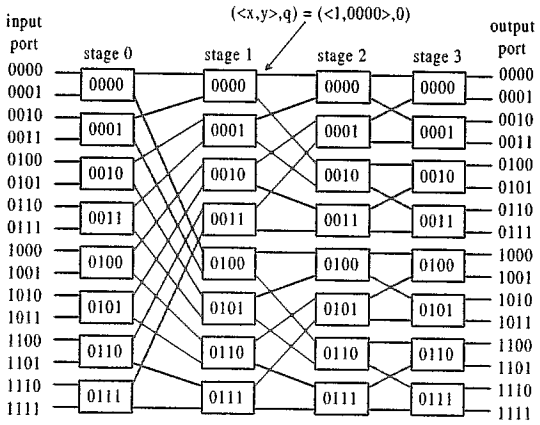


圖 1：一個 $(2^4 \times 2^4)$ Delt 網路的建構方式

(5)每一個 SE 的輸出端以(<x,y>,q)表示，代表第<x,y>個 SE 的第 q 個輸出端(link q，q = 0,1)。

(6)以連接函數(interconnection function) β 表示 Delt 網路中建構每一個 stage SE 的連接方式，

$$\beta(<i, E_0E_1\cdots E_{n-2}E_{n-1}>, q) = <i+1, E_0E_1\cdots E_i q\, E_{i+1}\cdots E_{n-3}E_{n-2}>$$

其中：a. q 表示 SE 的輸出端，q = 0,1。

b. i 代表 stage，$0 \le i \le n-2$。

c. E_0 固定設為 0，E_0 = 0。

因此，在第 i stage SE 的輸出端 0 及 1 由連接函數 β 根據 q 之值與相對第 i+1 stage 二個 SE 的輸入端連接。若以 n = 4 為例，$(2^4 \times 2^4)$的 Delt 網路建構如圖 1 所示。

2.2 以 Delta 網路為基礎的交換系統架構

依據 2.1 所述以$(2^n \times 2^n)$ Delta 網路為基礎建構的交換系統架構如圖 2 所示。封包從 input port 傳入交換系統後，先進入輸入控制模組(Input Control Module, ICM)在封包前附加上 multicast tag (詳述於後)，再傳至$(2^n \times 2^n)$的 Delta 網路完成交換工作，然後送至輸出控制模組(Output Control Module, OCM)移去 multicast tag，最後由 output port 傳出。

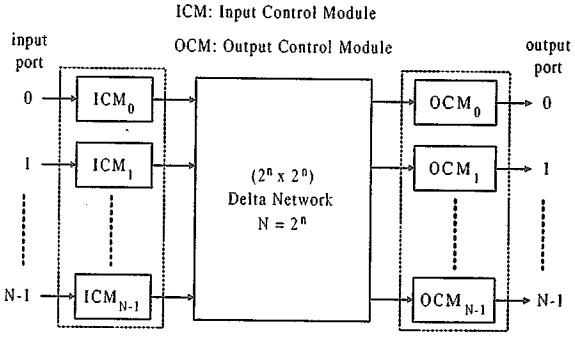


圖 2：以$(2^n \times 2^n)$ Delt 網路為基礎的交換系統架構

2.2.1 輸入控制模組(ICM)與輸出控制模組(OCM)

ICM 主要的工作是：封包標頭的處理、位址的轉換、產生內部的 multicast t a 以及輸入緩衝，其架構如圖 3 所示。

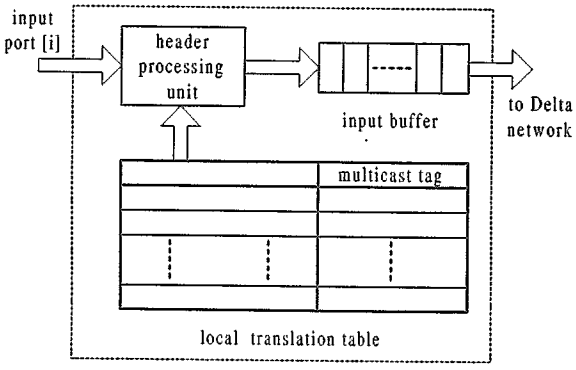


圖 3：輸入控制模組(ICM)架構圖

每一個 ICM 都存有一個 local translation table，封包由第 i 個 input port 傳入交換系統後進入第 i 個 ICM，ICM 內的 header processing unit 先依據封包標頭的位址查出新的位址及 multicast tag，更改位址後並將 multicast t a 附

加在封包之前，最後存入輸入緩衝器(input buffer)內等待傳送至 Delt 網路。

multicast tag 為一 N 個位元($N=2^n$)的資料結構，如圖 4 所示，其中：

(1) N 代表輸出埠(目的地)的位址，由 0 至 N-1。

(2) multicast ta 內的位元 M_i ($0 \le i \le N-1$)代表封包是否要傳送至輸出埠 i，所以 $M_i=1$ 表示封包須傳送至第 i 個輸出埠，$M_i=0$ 則表示不須傳送。

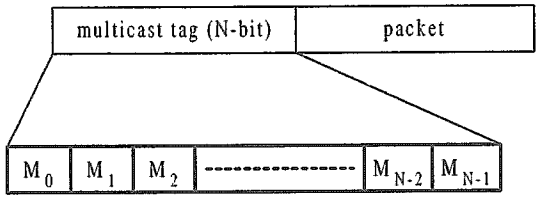


圖 4：multicast tag 的格式

如此根據設定 multicast ta 內相對於 output port 的位元 M_i 之值就可以表示封包是 unicast 或者是需要 multicast 或 broadcast。例如：若 4 個位元 multicast ta 之值為 $M_0M_1M_2M_3=1101$，則表示封包要群播至位址為 0、1 及 3 的 output port。

OCM 的工作主要將 multicast tag 從標頭中移走，並且做為與外部網路溝通的界面。

2.2.2 封包在 Delta 網路中的遞送方式

封包由輸入埠傳入($2^n \times 2^n$) Delta 網路到達輸出埠的路徑可以用 0 至 n-1 階的二元樹(binary tree)表示，其中二元樹的第 0 階代表 Delta 網路的 stage 0，第 1 階代表 stage 1，以此類推；每一個 internal nod 代表 Delta 網路中的 SE；terminal nod 則分別代表輸出埠。因此，若封包由輸入埠 1100 multicas 至輸出埠 0001、0101、1001 及 1110，則所攜帶之 multicast ta 值為 0100010001000010 ($M_1=M_5=M_9=M_{14}=1$)，而封包的傳送路徑以二元樹可以表示如圖 5。

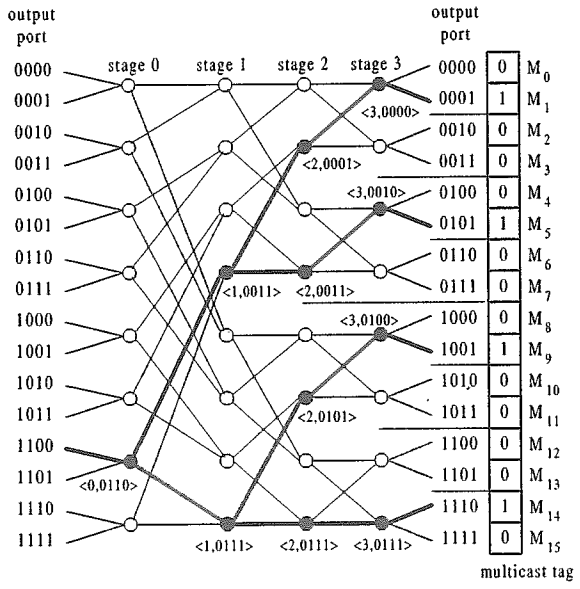


圖 5：以二元樹表示 1100 multicast 至 0001、0101、1001 及 1110 的路徑

因為每一個 SE 在 Delta 網路中的位置是固定且有其各別的位址，由封包傳送路徑的二元樹可知道，每一個 SE 可以根據本身在 Delta 網路中的位置 $<i, l>$，在接收到封包後檢查 multicast ta 中的一段相對位元，用以決定封包將由 SE 之 link 0 或是 link 1 傳送至下一 stage。以下利用圖 6 來說明位於 $<i, l>$ 的 SE 決定遞送路徑之原理：

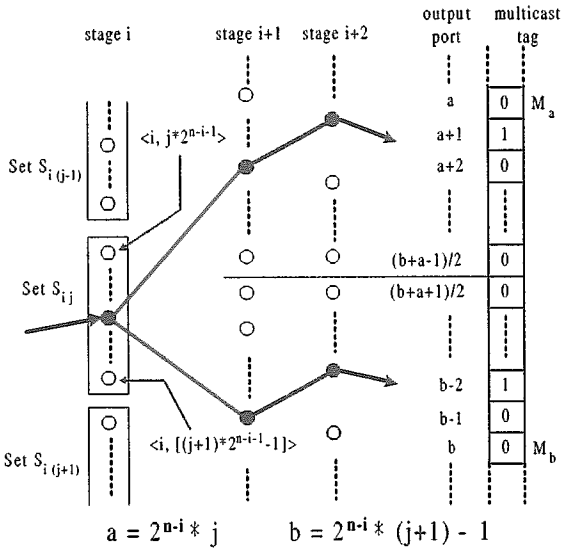


圖 6：S_{ij} 群組的交換元件與 multicast ta之關係

(1)在 stage i 的 SE $<i,l>$ 是否有封包要往 stage $(i+1)$ 傳送(即以 $<i,l>$ 為根(root)，是否有往右延伸的子樹(sub-tree))，是由 M_a 至 M_b 範圍內共 b－a＋1 個位元決定，其中只要有一個位元是 1 就表示封包需往 stage $(i+1)$ 傳送。然而是否要往 link 0 傳送則是由位於 M_a 至 $M_{[(b+a-1)/2]}$ 範圍內共(b－a＋1)／2 個位元決定，若其中有一個位元為 1，表示封包需由 SE 之 link 0 傳送至 stage $(i+1)$，因此可對(b－a＋1)／2 個位元執行 OR 運算得知結果。同理，$M_{[(b+a+1)/2]}$ 至 M_b 範圍內共(b－a＋1)／2 個位元則決定封包是否需由 SE 之 link 傳送至 stage $(i+1)$。

(2) stage 0 的 SE 以位址的 E_0 位元分為不同的群組，對應到 multicast tag 的 $M_0M_1\cdots M_{N-1}$ 共 N 個位元，以決定是否將封包傳送至下一個 stage。而 stage 1 則以 SE 位址的 E_0E_1 位元分為不同的群組，可分為 $E_0E_1=0$ 及 $E_0E_1=01$ 共 2 個群組，屬於 $E_0E_1=00$ 群組的 SE 對應到 multicast tag 中 $M_0M_1\cdots M_{[(N/2)-1)]}$ 的 N／2 個連續位元，執行 OR 運算以決定是否將封包傳送至下一個 stage；同理，屬於 $E_0E_1=1$ 群組的 SE 則對應到 multicast ta 中的 $M_{(N/2)}M_{[(N/2)+1]}\cdots M_{N-1}$ 之 N／2 個連續位元，執行 OR 運算以決定傳送與否。

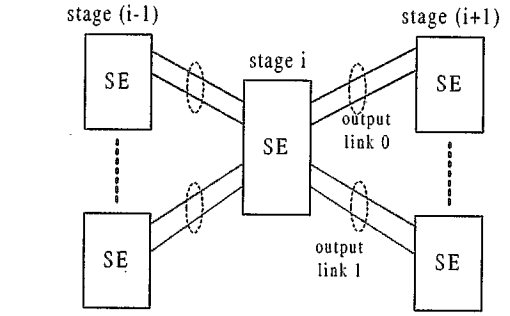
(3) stage i 以 SE 位址的 $E_0E_1\cdots E_i$ 共 $i+1$ 個位元分為不同的群組，若以 S_{ij} 代表在 stage i 的第 j 個 SE 群組(其中 $j=E_0E_1\cdots E_i$ 且 $0 \le j \le 2^i-1$)，則屬於 S_{ij} 群組中的 SE 在 Delta 網路中的位置為 $<i,\ j*2^{n-i-1}>$ 至 $<i,\ [(j+1)*2^{n-i-1}-1]>$ 共 2^{n-i-1} 個。

(4)屬於第 S_{ij} 群組的 SE 檢查 multicast tag $M_{2^{n-i}*j}$ 至 $M_{2^{n-i}*(j+1)-1}$ 範圍內的位元決定是否將封包傳送至 stage $(i+1)$，其中對 multicast ta由 M 至 M 的位元執行 OR 運算，若結果為

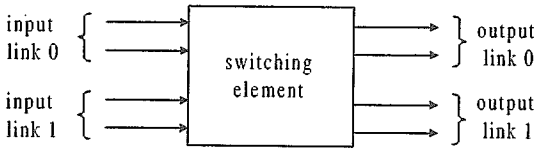
1 表示要將封包送至 link 0，若結果為 0 表示不傳送；同理，對 $M_{2^{n-i-1}*(2j+1)}$ 至 $M_{2^{n-i}*(j+1)-1}$ 的位元執行 OR 運算決定是否要將封包送至 link 1。

2.2.3 Multiple I/O Delt 網路的架構

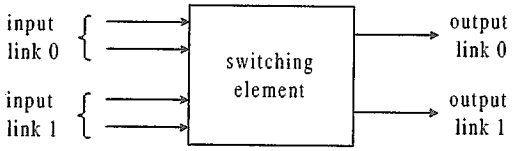
Multiple I/O 的 Delta 網路建構如圖 7 所示，stage 0 至(*n*-2)為 4×4 的 SE，stage (*n*-1)則為 4×2 的 SE，其中 stage *i* 與前一個 stage (*i*-1)以及與後一個 stage (*i*+1)均以 2 個 li 連接。為維持與外部系統的連接，封包均由 stage 0 SE 的其中 1 個 input lin 傳入，另 1 個 input link 則不與外部連接保持開路(open)狀態，最後 stage (*n*-1)的輸出則為 1 個 link，也就是封包在 Delta 網路中的每一個 stage 均由 2 個 lin 傳送。



interconnection between adjacent SEs



switching element of stage 0 to stage (n-2)



switching element of stage (n-1) (last stage)

圖 7：Delt 網路內的交換元件及連接方式

2.2.4 交換元件的內部架構

交換元件內部架構如圖 8 所示，封包進入 SE 後存於 gate 中，邏輯控制單元(LCU)依輸入封包的 multicast t a 根據 2.2.2 所述邏送方法產生輸出控制信號(OC_X)，OC_X=1 時將封包由 gate 傳至 PDU，OC_X=0 時則將封包擋在 gate 中(或是丟棄)不往後傳，送至 PDU 的封包輪流送至 FIFO Buffer Module，再等待 OCU 以 round-robin 的排程方式送至 output link。

2.2.4.1 邏輯控制單元(LCU)與閘道暫存器(Gate)

LCU 主要的工作是利用 multicast tag 並根據 2.2.2 所述的邏送原理經過邏算後產生輸出控制信號給 gate，LCU 的架構如圖 9 所示，其中：

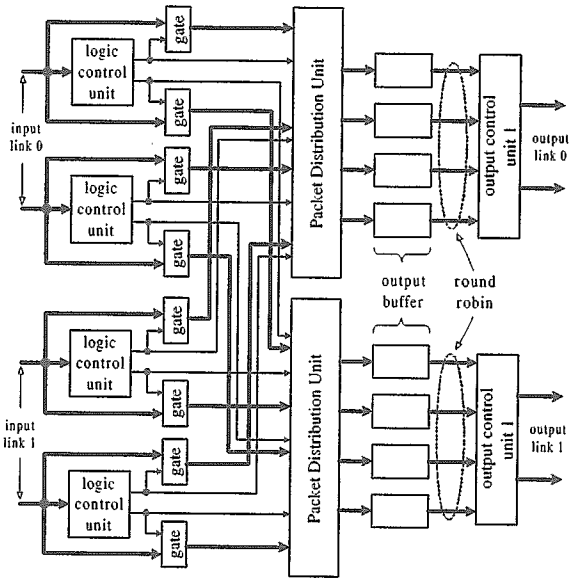


圖 8：交換元件之內部架構圖

(1)filt e 用來取出並暫存附加於封包前的 multicast ta 。
(2)mask register 共有 N 位元($D_0D_1\cdots D_{N-1}$)用來取出 multicast ta中需要執行邏輯運算的位元。
(3) multicast tag 與 mask register 內位元一對一執行 AND 運算就可得到相關於 link 0 的之 $M_{2^{n-i}*j}$ 至 $M_{2^{n-i-1}*(2j+1)-1}$ 位元值，以及相關於 link 1 的 $M_{2^{n-i-1}*(2j+1)}$ 至 $M_{2^{n-i}*(j+1)-1}$ 位元值。
(4)$M_0M_1\cdots M_{N-1}$ 位元執行 OR 運算，結果即可產生決定 gate 是否要將封包送至 PDU 的輸出控制信號。

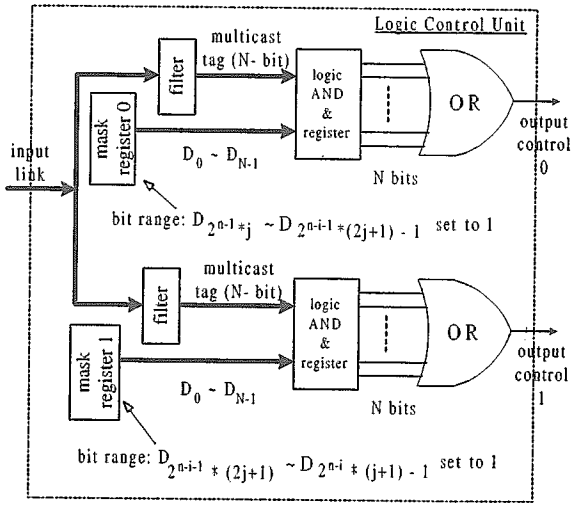


圖 9：邏輯控制單元(LCU)架構圖

2.2.4.2 Packet Distribution Unit (PDU)

因為封包的傳送路徑在 SE 之間都有 2 個 link，而因為每一個 stage 所產生的延遲，故會有很高的機率發生封包傳送順序錯亂(packet miss order)的現象，這種情形會造成 packet retransmi 的現象，而影響網路系統的整體效能，在 ATM 網路中更是不被允許的。因此封包送入 OCM 前先經 PDU 處理，以保證能依序輪流存入 4 個輸出緩衝模組，再由 OCU 依序輪流讀出，圖 10 為 PDU 的架構

圖。

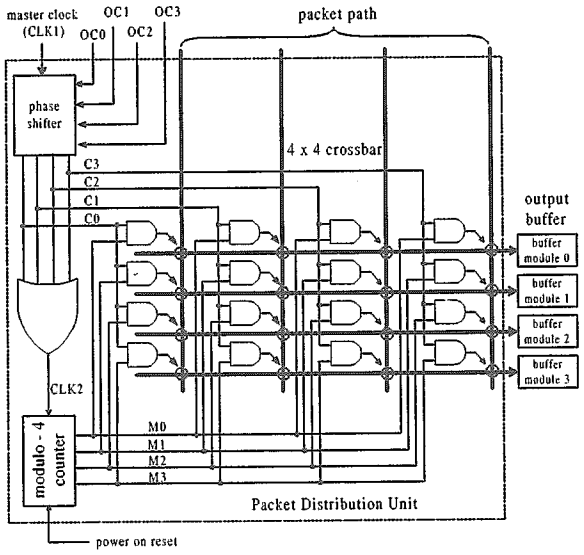


圖 10：Packet Distribution Unit (PDU)之架構圖

2.2.4.3 輸出緩衝模組(Output Buffer Module)與輸出控制單元(OCU)

輸出緩衝模組以 FIFO(first-in-first-out) queue 組成，每一個 output link (2 個 link)有 4 個 module，因此每一個 stage cycle 輸出控制單元以 round-robin 的方式從 2 個模組讀出封包送至 output link 最後的 stage (n-1)則選 1 個 module 讀出封包)。緩衝器之 queue 的大小與交換系統的整體效能有著密切的關係，queue 太小在 traffic 量大時很容易 full 因而發生 packet loss 的現象；queue 太大雖然降低了 packet loss 的機率，但除了硬體成本增加之外，封包留在 queue 裡的時間也增加(封包傳送的延遲時間增長)。

3. 模擬分析

本章以 $2^7 \times 2^7$ 的交換架構實行模擬分析，參數定義如下：

(1)Offered Load (OL)：input traffi 的流量(0.0≤OL≤1.0)。

(2)Multicast Rate (MR)：input tra fic 中屬於 multicast traffic 的比例(0.0≤MR≤1.0)。

(3)Fanout (F)：一個 multicast packet 在 switch 內複製後送至不同 output port 的數量。

(4)Delay (D)：packet 由 input port 傳至 output port 的平均延遲時間。

(5)Packet Loss Ratio (PLR)：SE 因 buffer ful 而丟棄 packet 與輸入 packet 的比率。

(6)Buffer Size (PIPO[X,Y])

X：代表 stage 0 至 stage n-2 的每 1 個 SE 中 queue module 的大小，因此每 1 個 SE 的 output buffer 大小為 4 * X。

Y：代表 stage n-1 的每 1 個 SE 中 queue module 的大小，因此 output buffer 大小為 4 * Y。

(7)Throughput (S)：switch 的輸出效能。

因為 switch 在初始狀態下的 n 個 stage cycle (n 為 stage 數)不會有 packet 送出，因此 Throughput 將小於 1.0 (0 ≤ S < 1.0)。

3.1 Throughpu t的分析

圖 11 說明 multiple I/O switch 在固定的 F=5 與 buffer (PIPO[64,64])下，不同的 OL 與 MR 的 Throughput，因為 multicast 時在 switc 內會複製 packet，因此由圖中可看到，即使 MR 只有 0.2 當 OL 到達 0.6 以後，output port 的 Throughput 就大約接近到 100%。而如果 input traffic 中有 80%是 multicast packet 時(MR=0.8)，雖然 OL 只有 0.3，但卻已達到約 100%的效能，亦即在 multicas 的狀況下，因為 packet 在 switc 內部大量複製，雖然 OL 不大，但只要 MR 越高 switch 的輸出就越容易到達飽和，一旦 switch 的輸出效能到達飽和後 switc 就無法再增加 packet 的輸出量。

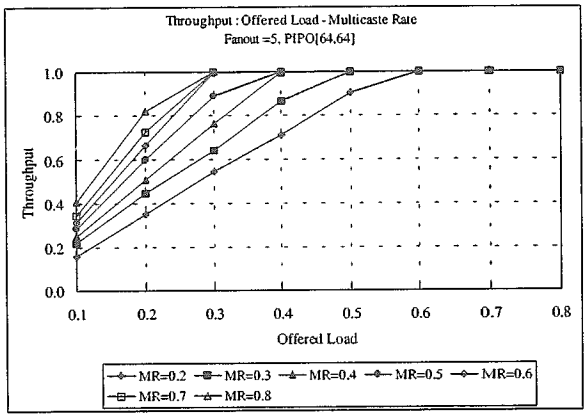


圖 11：不同 OL 與 MR 情況下的 Throughput

3.2 Packet Loss Rati 的分析

圖 12 說明在 buffer 大小為 PIPO[64,64]與 F=5 的條件下，不同的 OL 及 MR 對 packet loss 的影響。以 MR=0.2 為例對照圖 11 來看，當 OL 為 0.6 時 Throughput 接近 100%，packet los 開始大幅增加，亦即輸出效能接近飽和狀態後，packet 因為無法再增加傳送至輸出埠的數量，又因為 buffer 大小只有 Y=64 不足以存放等待輸出的 packet，因而開始有 packet los 的情況出現。

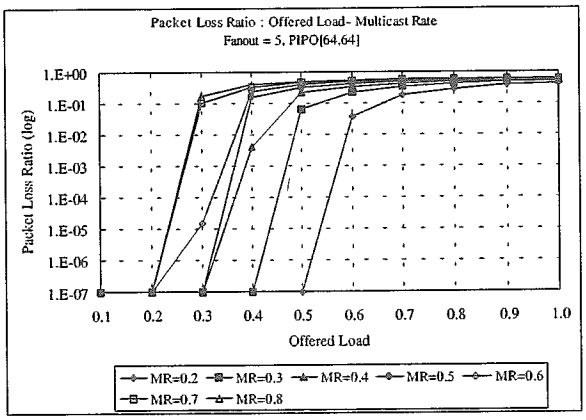


圖 12：不同的 OL 與 MR 對 packet los 的影響

圖 13 以 MR 固定為 0.5 的情況下，說明在不同的 OL 時 switch 內的每一個 stage 發生 packet loss 的數量，配合圖 11 可看出當 OL=0.4 時 Throughput 接近 100%，此時 switch 的最後一個 stage 開始發生 packet loss 現象，而隨

OL 的增加呈現顯著的上昇現象，等到 OL 上昇至 0.7 時，中間 stage 也開始發生 packet loss，顯示有大量的複製 packet 在中間 stage 產生而佔滿 buffer 導致 packet loss。

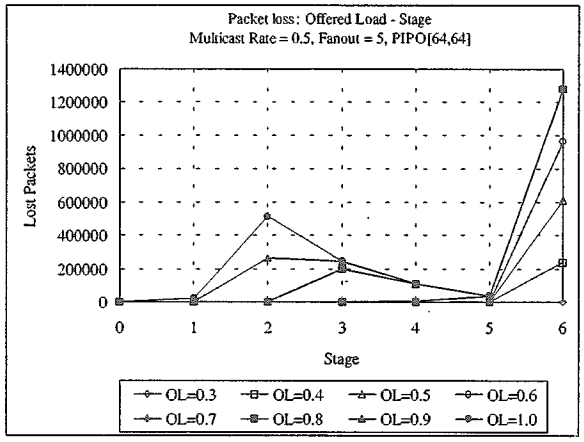


圖 13：不同 Offered Load 的狀況下每一個 stage 的 packet loss 情形

3.3 Delay 的分析

圖 14 以不同的 OL 與 MR 來說明 packet 由輸入傳送到輸出埠之間所發生的延遲現象，同樣地配合圖 11 的 Throughput 觀察發現，switc 的輸出效能尚未到達飽和狀況時，delay 約在 7 至 8 個 stage cycle 之間，因為我們是以 7 個 stage 的 switch 來模擬，因此 packet 在每一個 stage 間幾乎是沒有延遲的現象。而當 Throughput 到達飽和狀態時，delay 上昇至平均約 250 至 260 stage cycle 之間，若配合圖 13 觀察可發現，發生在最後 stage 的 packet loss 首先上昇，表示最後 stage 的 buffer 經常會在 full 的狀態，因而 packet 在最後 stage 的延遲接近 4*64=256 個 stage cycle，因此可說輸出效能開始接近飽和時，packet 的延遲絕大部分發生在最後 stage。此時如果再增加 OL 時，packet 在中間 stage 開始有延遲現象，因而使 switch 的整體 delay 呈現進一步上昇的趨勢。

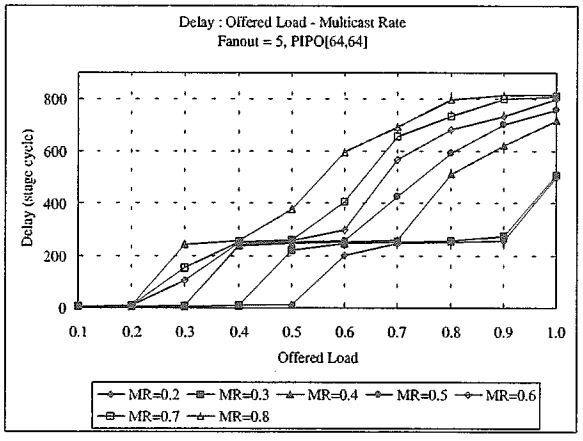


圖 14：不同 OL 與 MR 情況下 packet 的 delay

3.4 Remark

依據利用電腦模擬分析 Packet Loss Rati 、Delay 及 Throughput，可得到以下結論：

(1)即使在 input traffic 不大的狀況下，也會因為 multicast 而在 switc 內部產生大量的複製 packet，而使得 Throughput 接近 1.0 造成輸出效能達到飽和狀態。

(2)在 Throughput 未達到飽和狀態前，switch 的工作形式基本上與[12]架構上運用 uniform traffi 模型對 unicast 所做的分析是類似的。

(3)在 Throughput 接近飽和狀態後，因為 packet 的輸出量無法再增加，使得 packet 會被存放在最後 stage 的 buffer 中等待傳送至輸出埠，因而 packet delay 開始逐漸增加。一旦 buffer 的大小不足以存放進入最後 stage 的 packet 數量時，則開始發生 loss 的現象，使得 switc 的整體 PLR 大幅增加，此時 packet 在 switc 內的 delay 與 loss 完全肇因於最後一個 stage 之上，buffer (Y)越大 delay 越大，buffer (Y)越小則 loss 上昇的幅度越快。

(4) Throughput 進入飽和狀態後，若繼續增加 OL (或提高 MR)，則 packet 停留在中間 stage buffer 的延遲時間也開始增加，因而 switc 的整體 packet delay 呈現再上昇趨勢。同樣地，如果中間 stage 的 buffer (X)大小不足以存放進入的 packet 數量，則中間 stage 也開始發生 packet los 現象。

4. 結論與未來研究方向

本篇論文說明了我們所提出以 Delta 網路為基礎的 multiple I/O multicast swit c 之架構及遞送的方法，其主要特性為：

(1)僅須參考 multicast t a 與 SE 在 switch 內的位址即能決定 packet 的繞送路徑。

(2)在 SE 內以 output buffer 來做為 packet 傳送的緩衝，藉以降低傳送過程所可能產生的 packet loss 及 contention。

(3)unicast 或是 multicast packet 在 switc 內只須傳送一次即可送達目的地。

(4)對於 multicast tag 的處理只做讀取而不做寫入動作，以降低 SE 處理上的負擔而提升 switch 的效率。

(5)以簡單的 PDU 達到類似 Knockout switch 的功能做為 speed-up 的機制。

(6)SE 之間使用 2 個 link 的路徑傳送 packet 以提升內部的傳送量，而模組化的設計可以容易讓 SE 之間使用 2 個以上的 lin 做為傳送路徑。

Delta 網路因為內部 queuing delay 與傳送路徑的特性，應用於交換架構上仍有許多待考慮的議題，例如對於即時(real time)應用系統的支援，交換元件之間的容錯(fault tolerant)處理，支援具優先權的 traffic (QoS)等皆可做為未來進一步研究的議題。

This work was supported in part by the ROC National Science Council under contract number NSC 89-2213-E-005-017. Wen-Shyen E. Chen is the corresponding author.

參考文獻

[1] A. Huang and S. Knauer, "Starlite: A Wideband Digital Switch," in Proceedings of IEEE GLOBECOM '84, pp121-125, 1984.

[2] Ellen Witte Zegura, "Architectures for ATM Switchin System," IEEE Communication Magazine, pp.28 -37, February 1993.

[3] Hamid Ahmadi and Wolfgang E. Denzel, "A Survey of Modern High-Performance Switching Techniques," IEEE Journal on Selected Areas in Communications,

Vol. 7, no.7, pp.1091-1103, September 1989.
[4] Jaehyung Park, Lillykutty Jacob, and Hyunsoo Yoon, "Performance Analysis of a Multicast Switch Based on Multistage Interconnection Networks," in Proceedings of INFOCOM '97.
[5] J. H. Patel, "Performance of Processor-Memory Interconnections for Multi-Processors," IEEE Transactions on Computers, C-30(10): 771-780, October 1981.
[6] Jonathan Turner and Naoaki Yamanaka, "Architectural Choices in Large Scale ATM Switches," IEICE Transactions, 1998.
[7] Jonathan Turner, "Design of a Broadcast Packet Switching Network," IEEE Transactions on Communications, vol. 36, no.6, June 1988.
[8] Jonathan Turner, "Design of an Integrated Service Packet Network," IEEE Journal on Selected Areas in Communications, vol. SAC-4, no.8, November 1986.
[9] R. Cusani and F. Sestini, "A Recursive Multistage Structure for Multicast ATM Switching," in Proceedings of IEEE INFOCOM '91, pp1289-1295, April 1991.
[10] Richard G. Bubenik and Jonathan Turner, "Performance of Broadcast Packet Switch," IEEE Transactions on Communications, vol. 37, no.1, pp. 60-69, January1989.
[11] Tony T. Lee, "Nonblocking Copy Networks for Multicast Packet Switching," IEEE Journal on Selected Areas in Communication, vol. 6, no.9, pp1455-1467, December 1988.
[12] Wen-Shyen E. Chen, "Design and Analysis of Hig -speed Packet Switching Fabrics for Integrated Broadband Networks," Ph.D. Dissertation, Dept. of Computer and Information Science, The Ohio State University, 1991.
[13] Xiaoqiang Chen and Vijay Kumar, "Multicast Routin in Self-routing Multistage Network," in Proceedings of INFOCOM '94, pp. 3a.3.1-3a.3.9, 1994.