Multicast ATM Switches: Survey and Performance Evaluation

Ming-Huang Guo and Ruay-Shiung Chang

Department of Information Management
National Taiwan University of Science and Technology
Taipei, Taiwan, R.O.C.,
email: rschang@cs.ntust.edu.tw

Abstract

Computer networks are undergoing a remarkable transformation. The widespread use of optical fiber to transmit data has made tremendous increases in network bandwidth. Furthermore, greater CPU power, increasing disk capacity, and support for digital audio and video are creating demand for a new class of network services. For example, video-on-demand, distant learning, distant diagnosis, video conferences, and many others applications have popped up one after another in recent years. Many of these services have one thing in common. They all require that the same piece of data be sent to multiple recipients. Even in traditional networks, this operation, called multicasting, can not be handled easily and cheaply. When scaled up to high speed ATM-based networks, the situation could be worse. Multiple streams of data travel around ATM networks. Each tries to send to many different destinations simultaneously. Therefore, designing economical ATM network switches which can support multicasting operations easily is very important in the future generation high speed networks. In the past twelve years or so, many designs for multicasting ATM switches are proposed. It seems about time to do a historical survey. It is hoped that by learning from the wisdom of the previous authors, new spark or angle can be found and exploited to design new multicasting ATM switches. Without easy and inexpensive multicasting, all the exciting services may become unaffordable. This will in turn lead to the diminishing of customer bases and finally will hinder the full-scale deployment of high speed networks.

CONTENTS

1. Introduction
2. Survey of multicast ATM switches
   2.1 Preview
   2.2 Multicast switches with copy networks
      2.2.1 Starlite switch
      2.2.2 Knockout switch
      2.2.3 Turner's broadcast switch network
      2.2.4 Shared concentration and output queueing multicast switch
      2.2.5 Multicast switches based on link-grouped multistage interconnection network
      2.2.6 Multinet switch
   2.3 Multicast switches without copy networks
      2.3.1 Knockout switch
      2.3.2 Gauss ATM switching element (ASE)
      2.3.3 Sunshine switch
      2.3.4 Lee's multicast switch
      2.3.5 A recursive multistage structure
      2.3.6 Multinet switch
      2.3.7 Growable packet switch
      2.3.8 Broadcast ring sandwich network
      2.3.9 Multicast output buffered ATM switch (MOBAS)
      2.3.10 Abacus switch
      2.3.11 Dilated network and prioritized services, internally nonblocking, internally unbuffered multicast switch (PINIUM)
   2.4 Prescheduling before copying
      2.4.1 Guo and Chang’s switch
      2.4.2 TATRA scheduling algorithm
3. Performance Evaluation
   3.1 Simulation assumptions
   3.2 Simulation results
4. Conclusions and future research directions

1. Introduction

Asynchronous Transfer Mode, commonly known by the slightly unfortunate acronym of ATM (Automatic Teller Machines or Another Technical Mistake), is the most widely studied and implemented form of high speed networks. Its standards are defined by ITU-T, formerly CCITT, and some interim standards are being developed by a user and vendor group known as the ATM Forum. ATM is the underlying transmission system for CCITT's next-generation ISDN, Broadband ISDN. BISDN is designed to provide subscriber communication services over a wide range of bit rates from a few megabits to several gigabits. Among the services emerged are video-on-demand, distant learning, video conference, and so on. Many of these services are characterized by point-to-multipoint communication. To support this capability in an ATM network, the ATM switch must be able to transmit copies of an incoming cell to different outlets. In doing so, there are many problems to be solved. For example, where is the cell copying to be done? How to handle the output port conflicts resulting from multicasting cells? How can cell ordering sequence be kept? The solutions to all the above problems require nontrivial modification to the original point-to-point ATM switches.

In principle, an ATM switch shall perform two basic functions: switching and queueing. Switching is to transport the information from an incoming logical ATM channel to an outgoing logical channel ATM channel. This logical channel is characterized by:

- A physical inlet/outlet pair indicated by physical port number
- A logical channel on the physical port, characterized by a virtual channel identifier (VCI) and/or a virtual path identifier (VPI).

When more logical channels contend for the same link or switching element in the internal of an ATM switch, blocking would occur. Since it is usually unwise to discard cells randomly, queueing, the second basic function of an ATM switch, is necessary. According to where the queues are provided in a switch, three types of switches can be identified: input queueing switches with buffers at the input ports; output queueing switches with the buffers at the output ports; and shared buffer switches with buffers inside a switch to be shared by the entire traffic. Examples of switch designs are abundant in the literature. Please refer to [1] for an extensive survey. In this paper, we only focus on the multicast ATM switches based on space-division, and the discussion of the prescheduling algorithms for the multicasting cells. Switches based on time-division are not considered in the paper.

Multicasting for ATM switches means that an incoming cell is destined for more than one output port in the switch. If an ATM switch does not include the cell replicating function, it is not a multicast switch - since it can not do multicasting. Therefor, in studying multicast ATM switches, cell replicating is a very important operation that needs special attention.

In cascaded multicast switch architectures, the interconnection networks of the switch fabric can be decomposed into several subfunctions, and each subfunction is accomplished by one subnetwork component. A number of cascading design techniques for multicast connectivity has been previously described in the literature [2, 4, 5, 7, 9-22, 24]. Fig 1 shows the basic architecture for a multicast switch. It is usually a serial combination of a copy network followed by a point-to-point (unicast) switch. The copy network replicates input cells from various sources simultaneously, and then copies of multicast cells are routed to their final destination by the point-to-point switch.

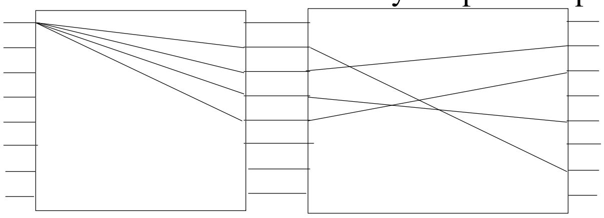


Fig.1: A multicast packet switch consists of a copy network and an unicast network.

Copy network, as it is named, is a special kind of functional networks for replicating cells. It is used to duplicate a multicast cell into many corresponding unicast cells. If there is a copy network doing cell replicating in a ATM switch, then the switch can be regarded as a multicast switch.

However, the copy network is not absolutely necessary for a multicast switch. If a switch does not have a copy network but somehow manages for the cell replicating operation in itself, it is still a multicast switch. Often in this kind of switches there is no particular switch element which is in charge of cell replicating. More or less, the cell duplication is just a sub-function in some components in the switch such as routing elements.

From Starlite in 1984 [13], various multicast ATM switches are explained and examined in this paper. The purpose is to take a snapshot of the current status quo. Since multicasting is expected to be more important and common, it is our sincere hope that this paper will arouse new enthusiasm and entice new design directions. The remainder of this paper is organized as follows. Section 2 surveys various switches for multicasting. Performance evaluations are conducted on some of the switches on Section 3. Finally, conclusions and future research directions are given in Section 4.

2. Survey of multicast ATM switches

2.1 Preview

Before dwelling on details, historical developments of some multicast switches are shown in Fig. 2, and Fig. 3 is the relationship between some multicast ATM switches. In this graph, two switches have an edge between them if the lower one is derived from the upper one. Please note that some switches are derived from more than one switch.

In general, multicast ATM switches can be built from three basic fabrics: Crossbar network, Banyan network, and Clos network. Crossbar network is based on mesh architecture and there is no internal blocking inside the switch. By allowing many input ports to send their cells to the same output port at the same time, head of line (HOL) blocking will not happen in the input ports either. The price paid is high hardware complexity and output port congestion.

Switch Proposed Year	Based on Knockout switch	Based on Banyan network		Based on Clos network	Based on Knockout switch
1984		Starlite switch[13]			
1988	Multicast Knockout switch[10]	Turner's broadcast switch[20]	Lee's multicast switch[16]		
1990	Gauss ASE [21]				
1991		Sunshine switch[11]	Recursive multistage structure [9]	Growable packet switch[17]	
1994	SCOQ switch[7]	LGMIN[24]	Multinet switch[14]		
1995	MOBAS[4]			Ring sandwich network[22]	
1996		Dilated network [19]			
1997	Abacus switch[5]		Guo & Chang's switch[12]		TATRA algorithm [18] PINIUM switch[15]

Fig. 2: The relationships in time and basic architecture of some ATM switches.

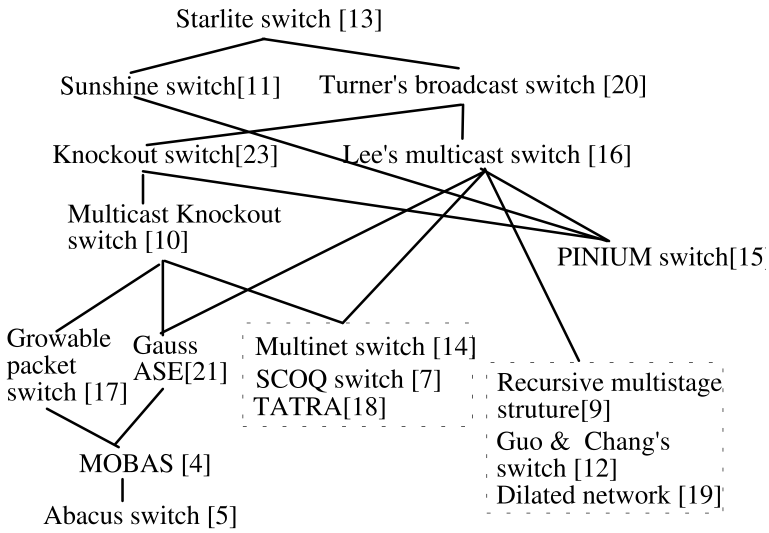


Fig. 3: The relationships between multicast ATM switches

In the history of ATM switches, Knockout switch [23] is the first one to build from Crossbar network. It uses a Knockout concentrator in every output port to ease the output congestion occurred in Crossbar network. Hence, Knockout switch is regarded as having the best performance among unicast switches. The multicast version of Knockout switch is proposed in 1988 [10]. After its publication, many switches based on Knockout switch have been proposed, such as Gauss ASE [21], SCOQ switch[7], MOBAS [4], Abacus switch [5], and PINIUM switch [15]. As a result, Knockout switch will be considered as a basic multicast switch fabric instead of Crossbar network in the following discussion.

Banyan network is one kind of multistage interconnection networks (MINs). It has less hardware complexity than Crossbar network and adopts the simple self-routing algorithm. Self-routing algorithm provides one-to-one path from one input port to one output port. Internal path conflicts may occur inside the Banyan network. Besides, no more than one cell is allowed to reach to the same output port at one time. Hence, HOL blocking will occur in input ports of Banyan network. Examples of multicast ATM switches based on Banyan network are Starlite switch [13], Turner's broadcast switch [20], Lee's multicast switch [16], Sunshine switch [11], the recursive multistage structure [9], LGMIN switch [24], Multinet switch [14], Guo and Chang's multicast switch [12], dilated network [19], and PINIUM switch [15].

Clos network is also one MIN but with only three stages [8]. The first stage distributes the incoming cells into the second stage so that there will be no hot spot phenomenon among the inputs of the second stage. The second stage routes cells to the proper input port of the third stage. In the third stage, cells will be concentrated into the requested output port. Since Clos network provides more than one path from one input port to one output port, there are less internal conflicts inside Clos network than inside Banyan network. Clos network has better performance than Banyan network but it has higher hardware complexity. Furthermore, since each output port in Clos network accepts only one cell at a time, as in Banyan network, HOL blocking is also a problem for Clos network. Examples of multicast ATM switches based on Clos network are growable packet switch [17], and ring sandwich network [22].

In the ATM multicast switches history, Starlite switch [13] is the earliest architecture which discussed the multicasting traffic. It builds on the combination of two simple switch architecture, crossbar network and Batcher-Banyan network. Sunshine switch [11] is a modified version of Starlite switch to improve the performance. Knockout switch [23] is built only on crossbar network. It is an output queueing switch and is regarded as having the best performance. Gauss ATM Switching Element (ASE) [24] is a switch which is very similar to Knockout switch and which is output queueing also. In 1991, Chao proposed a new idea of grouping principle [3] which is motivated from the Knockout concentrated. He used the idea to build the MOBAS switch [4] and the Abacus switch [5].

Turner proposed the first broadcast switch based solely on Batcher-Banyan network in 1988 [20]. At the same year, Lee presented another switch for multicasting traffic [16]. After these harbingering works, many new designs for multicast switches followed.

Shared Concentration and Output Queueing Multicast (SCOQ) switch [7] is a switch with shared concentration and output queueing. It is modified from Knockout switch. The recursive multistage structure [9], a switch based on link-grouped multistage interconnection network (LGMIN) [24], and multinet switch [14] are both improved from Turner's or Lee's switch.

Besides the above switches, another switches design directions are based on the Clos network. Growable Packet Switch [17] and Ring Sandwich Network [22] are examples of this kind of switches.

Guo and Chang developed an ATM multicast switch from another point of view [12]. They tried to make the total switch cycles needed as few as possible to transmit the incoming cells (including unicast cells and multicast cells). The way adopted is prescheduling the incoming cells so that the cells going through the followed multicast switch at the same "time" (switch cycle) will need no recirculation and will cause no internal or output port conflicts in the switch.

In 1997, McKeown et al. proposed another scheduling for multicasting cells at input port by focusing on the distribution of the residue [18]. The residue is the set of all cells that can not be transmitted and remain at the head of line of the input ports at the end of each transmission cycle. The algorithm is based on the crossbar network, and by this algorithm, the set of multicasting cells that can be transmitted through the switch in the same switch cycle can be easily found.

In the following, various multicast ATM switches are examined in detail. The switches are classified according to : (1) with copy networks, (2) without copy networks, and (3) with prescheduling.

2.2. Multicast switches with copy networks

2.2.1 Starlite switch

The outline of the Starlite Switch [13] is shown in Fig. 4. It does multicasting by a two-stage copy network placed above the concentrator, as shown in Fig. 5. The first stage is a sorting network. New cells entering the switching fabric are put into input ports on the left. Multicasting cells are put into the input ports on the right. Multicasting cells are special cells that contain the channel id of the cells they want to copy and the destination port to which the copy is to be sent. The sorting network sorts the cells based on their source addresses (channel id), so that each new cell appears next to multicasting cells that want to copy it. The copy network then duplicates each new cell into all of its copies and injects the cells into the concentrator.

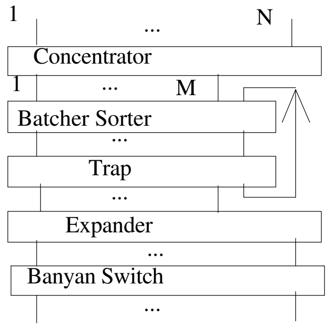


Fig. 4: Starlite Switch

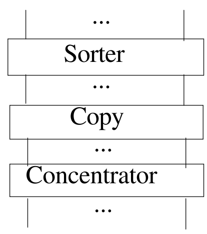


Fig. 5: Multicast In Starlite switch

In the previous cell replication process, Starlite switch assumes the synchronization of the source and destinations, and an "empty packet setup procedure" is also required. But it is not feasible to implement this approach in a broadband packet network where packets may usually experience delay variation due to buffering, multiplexing, and switching in a multiple-hop connection.

2.2.2 Knockout switch

The original Knockout switch [23] does not support multicasting. But after some simple modifications, multicasting can be added. To support multicasting, in addition to the N bus interfaces, the input buses are attached onto, as needed, up to M multicast modules [10] specially designed to handle multicast packets. As shown in Fig. 6, each multicast module has N inputs and one output. The inputs are the signals from the input interface modules. The output drives one of M bus wires as part of the arrangement for broadcasting to all the N bus interfaces.

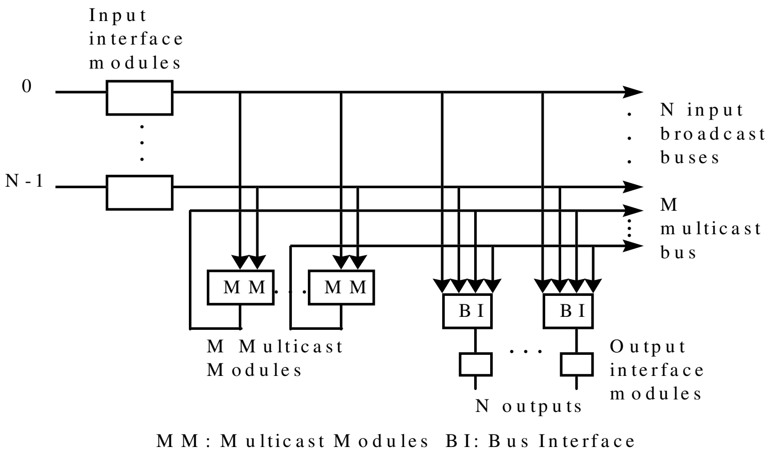


Fig. 6: A Knockout switch with multicast modules.

There are two proposed approaches to implement the multicast module. The first approach is based on the notation of cell duplication. The second approach is based on a fast cell filter. The latter approach belongs to the second multicast switch category, so it will be explained later. The following are the descriptions of how multicasting is done by cell duplication.

A block diagram of the multicast module based on cell duplication is illustrated in Fig. 7. The incoming cells are selected through the cell filters which are set to accept multicasting cells only. The selecting principle adopted is the same as in the original Knockout switch: an $N:L$ ($L<<N$) Knockout concentrator is used and L "winners" from the $N:L$ concentrator are stored in an L-input, one-output FIFO buffer after proper shifting. The retrieval of the cells from the buffer is done according to the FIFO discipline, and the cell sequence is thus preserved. Upon exit from the buffer, a multicasting cell enters into the cell duplicator to duplicate cells with different destination addresses in the header. The duplicated cells are sent along the broadcast bus to the required bus interfaces. In this scheme, the various destination addresses of replicated cells are obtained by table looking-up and a multicast cell transported on different links is not restricted to carry the same virtual circuit number throughout.

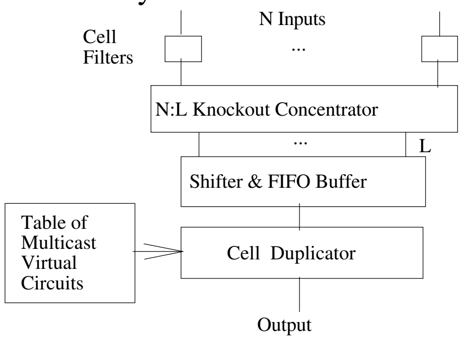


Fig. 7: Multicast module with a cell duplicator.

2.2.3 Turner's broadcast switching network

The switch fabric architecture of Turner's broadcast switching network [20] is shown in Fig. 8.

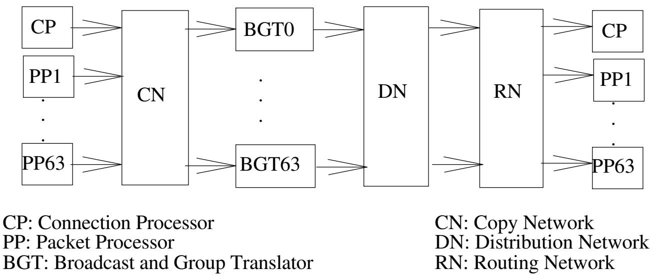


Fig. 8: The switch fabric architecture of Turner’s broadcast switching network.

When a broadcast cell having K destinations passes through the copy network, it is replicated so that K copies of that cell emerge from the copy network. For unicasting cells, they pass through the copy network without change. After that, the broadcast and group translators will determine the routing information for each cell in the rest of the switch. By these translated routing information, the distribution and routing networks move the cells to the proper outgoing packet processor.

The most novel aspect of Turner’s switch is the flexible broadcast capability, which is suitable for a range of services including broadcast, television, and conferencing. However, there is a serious problem in the Turner's switch - routing collision in the routing network. When two cells arrival at the same switching element in the routing network, routing collision is occurred if these two cells attempt to leave on the same output link. This makes Turner's switch become a blocking one. Therefore buffers are required for every internal node in the routing network to prevent packet loss and cell collision.

2.2.4 Shared concentration and output queueing (SCOQ) multicast switch

Fig. 9 shows the multicasting SCOQ switch [6]. It is modified from the original one [7], with a copy network in the feedback loop. In a multicasting SCOQ switch, there is a sorting network, L switching modules, a copy network, and N input buffers. The sorting network and the switching modules operate in the same way as in the original SCOQ switch. While transmitting the multicast and broadcast cells, these cells will be fed back through the copy network. In copy network, cells will be duplicated according to their requests, and the destination addresses of replicated cells will be assigned by the trunk number translators inside the copy network. After the copy network, a multicasting cell will become some unicasting cells.

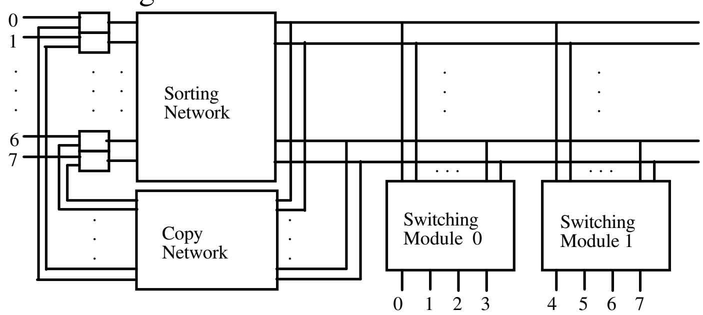


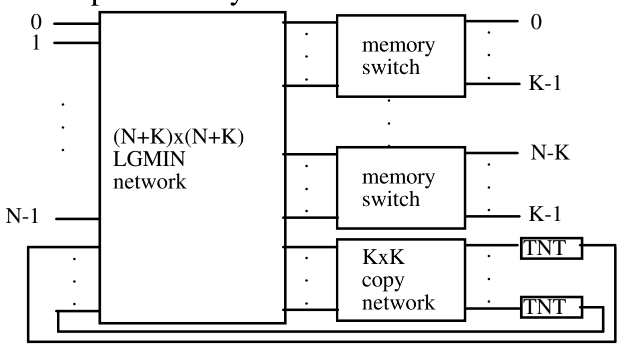
Fig. 9: The SCOQ multicast switch with $N = 8$, $L = 2$, and $K = 4$.

Cell duplication in the multicasting SCOQ switch is performed in the feedback configuration. The advantages of this method are: (1) there is no interference between unicasting cells and the duplication of multicasting cells, (2) the copy network is nonblocking even without an extra selection or arbitration mechanism and (3) the buffers at the input ports operate independently and without central control.

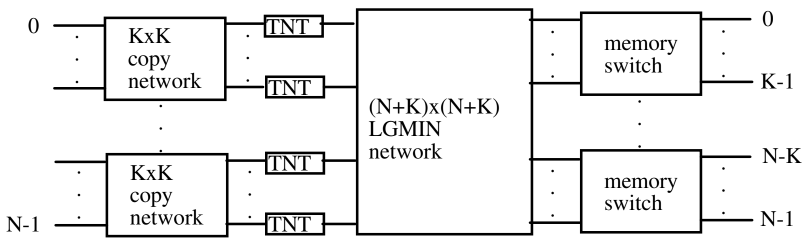
Although the feedback duplication has the above advantages, there is a serious problem in itself: the copy network might make less and less replication when there are more and more cells request replication at the same switch cycle. This will make the switch performance degrade quickly and make the switch performance unstable. We will show this phenomena in the later simulation.

2.2.5 Multicast switches based on link-grouped multistage interconnection network

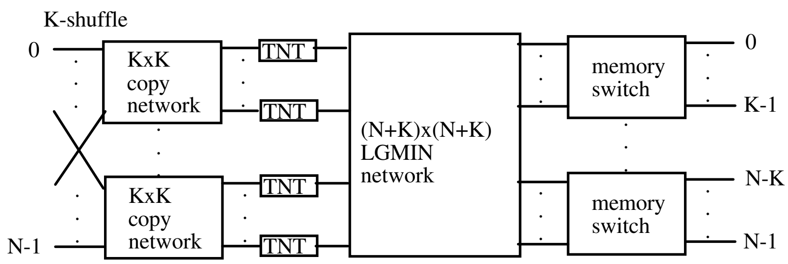
Fig. 10 shows several multicast switch architectures based on link-grouped multistage interconnection network (LGMIN) [24]. These architectures consist of multiple shared-buffer copy network modules of adequate size suitable for fabrication on a single chip, and small output memory switch modules.



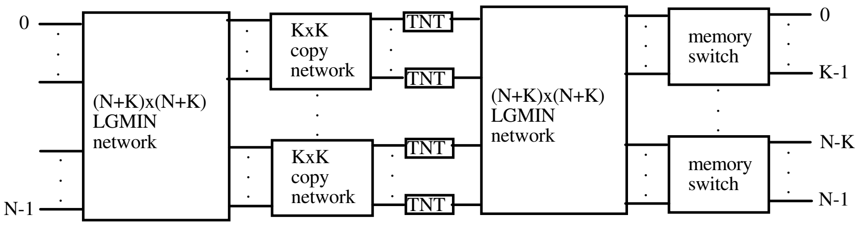
(a) scheme 1



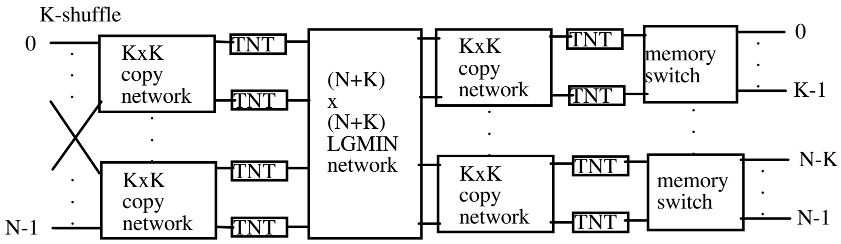
(b) scheme 2



(c) scheme 3



(d) scheme 4



(e) scheme 5

TNT: Trunk Nmber Translator
LGMIN: Link-Grouped Multistage Interconnection Network

Fig. 10: Various modular multicast ATM switch architectures based on LGMIN.

Each architecture depicted in Fig. 10 is applicable to a different traffic situation. The first scheme, Fig. 10-(a), is for the case when there is much less multicast traffic than unicast traffic. The next scheme, Fig. 10-(b), is aimed at heavy multicast traffic that is uniform over all of the copy network modules. That is, the multicast traffic is not concentrated on a few of the copy network modules.

The third scheme, Fig. 10-(c), is basically similar to the second scheme, except that the input ports of the whole switch are interconnected with the inputs of the copy network modules through a ***K***-shuffle interconnection. This ***K***-shuffle interconnection allows heavy multicast traffic concentrated over various consecutive input ports to be distributed to different copy network modules in a static fashion. This scheme cannot deal with every kind of non-uniform multicast traffic, because the ***K***-shuffle interconnection only statistically distributes the incoming traffic. By replacing the ***K***-shuffle interconnection with an LGMIN network, we derive the fourth scheme, as in Fig. 10-(d). In this scheme, the additional LGMIN network that precedes the copy network modules dynamically distributes the incoming traffic, on the basis of connection level, to the copy network modules as uniformly as possible. The fifth scheme, Fig. 10-(e), is intended to deal with multicast connections that require a bandwidth that a single copy network module cannot provide.

The shared-buffer copy networks in the various multicast architectures can be divided into two categories from the viewpoint of cell replication mechanism: (1) those in which copies are generated recursively, i.e., some of the copies are made by feeding some copies cells back to the input side of the network, and (2) those in which copies are generated by a Broadcast Banyan network (BBN) whose output ports are reserved before replication. The former is called the recursive copy network (RCN), and the latter is called the output-reserved copy network (ORCN). BBN is used in the next Lee's multicast switch [16] and ORCN is showed in Fig. 11.

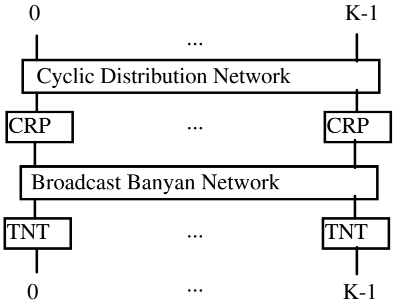


Fig. 11: A K by K output-reserved copy network (OCRN) including TNT.

ORCN consists of a cyclic distribution network (CDN), a set of contention resolution processors (CRP), a BBN, and a set of Trunk Number Translators (TNTs). The CRPs are coordinated through a token ring. The objective of the CDN is to distribute the master multicasting cells to the CRPs cyclically. This will ensure that all CRPs are shared uniformly. Furthermore, by making the active incoming master multicasting cells cyclically concentrated and the corresponding outputs sequence of the master multicasting cells monotonic, cells will not block in BBN.

The CDN consists of a running adder network and a reverse Banyan network. The main functions of the CRPs are (1) to store the master multicasting cells distributed by the CDN and process them in FIFO order and (2) to update the header of the master multicasting cell in order to reserve as many consecutive outputs of the BBN as the number of copies requested. The combination of the CDN and a token-ring reservation scheme ensures the cyclically nonblocking property of the BBN.

In Fig. 10, these fabrics are constructed for different multicasting traffic types, and they can be easily expanded. But by inspecting from Fig. 10-(a) to Fig. 10-(e), the hardware complexity rises drastically when the switch architecture is expanded. Fig. 12 shows an LGMIN network constructed from $2K \times 2K$ building units. As the previous conclusion, once K increases, the hardware complexity of the switch structure will increase.

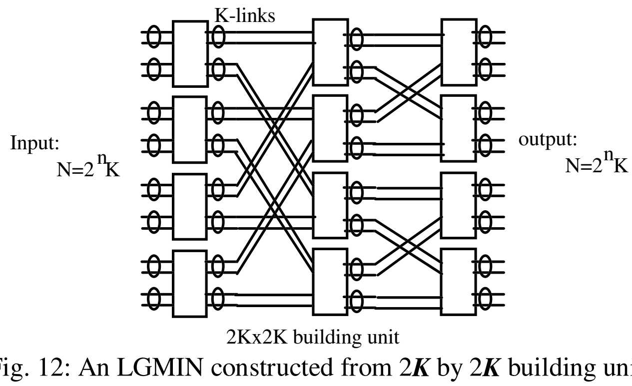


Fig. 12: An LGMIN constructed from 2K by 2K building units.

2.2.6 Multinet switch

Multinet switch [14] is a self-routing multistage switch. It consists of virtual FIFO buffers located between stages and in the output ports. Although it provides incoming ATM cells with multiple paths, the cell sequence is maintained throughout the switch fabric. Cells contending for the same output addresses are buffered internally according to a partially shared queueing discipline. Fig. 13 shows the block diagram of an 8 by 8 Multinet switch.

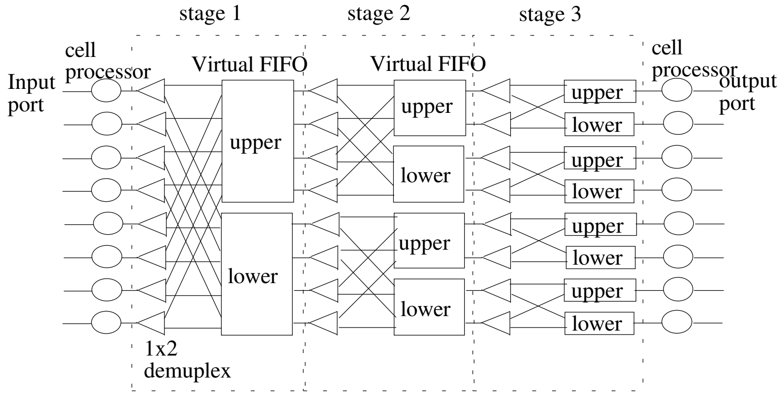


Fig. 13: Overall block diagram of 8 by 8 Multinet switch.

The routing of cells in the Multinet switch is similar to that of the self-routing banyan networks. However, the differences are that the Multinet switch provides multiple paths through the switch fabric, unlike the banyan network in which only a single path exists, and that it shares FIFO buffers in the concentrators. Fig. 14 illustrates how an arriving cell might go through multiple paths to reach the same output port. The total number of possible paths through the switch for an input-output connection pair can be easily obtained as $\prod_{i=1}^{\log N} 2^{n-i} = 2^{(n-1)/2}$, where N is the switch size, and $n = \log N$, the number of stages in the switch.

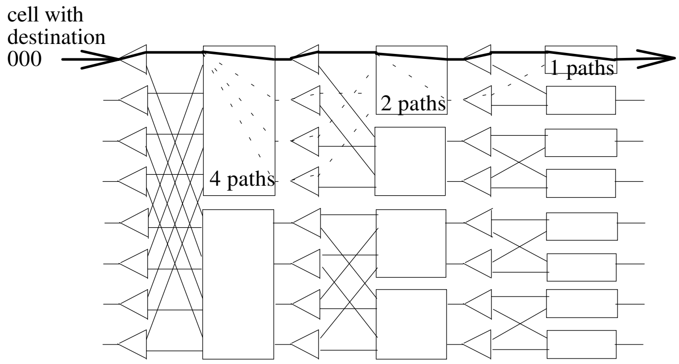


Fig. 14: Multiple paths through the Multinet switch.

The routing in the switching planes does not require a centralized control. It is done by each demultiplexer simply checking a bit of destination address. Let $d_1d_2 \ldots d_i \ldots d_n$ be the binary destination address of the switch size N and $n = \log N$. Each demultiplexer in the ith stage checks the ith bit and sends the cell to the upper virtual FIFO or lower virtual FIFO if the ith bit is "0" or "1," respectively. Fig. 15 shows the routing operation of the Multinet switch. The input port 0 has a cell destined to output port 7. The destination address 7 is "111" in binary form. In the first stage, the cell is sent to the lower virtual FIFO. The cell passes through the virtual FIFO and it could be placed in any of the demultiplexers in the next stage depending on the number of cells present in the virtual FIFO. Assume that the cell goes to input port 4 in the second stage. Then, the cells is sent to the lower virtual FIFO since its 2nd bit is "1." Again assume that the cell is placed in input port 6 in the third stage. Then, the cell is sent to the lower virtual FIFO since its third bit is "1" and it finally reaches the destination output port. The routing is done in such a way that no two cells will be contending for the same resources.

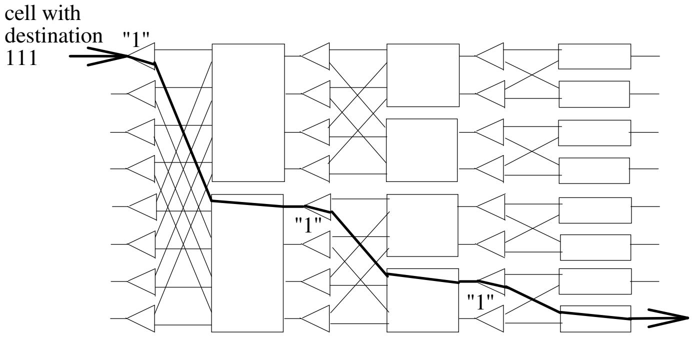


Fig. 15: Routing operation of the Multinet switch.

There are two ways for the Multinet switch to deal with the multicasting traffic. The distinction is the same as before, one with explicit copy networks, one without. The following introduces the one with the copy networks and the other will be left to the next section.

The first method is to place a copy network before the switch, as descried in previous switches. The replicated cells are then distributed evenly across the input ports of the switch so that no congestion arises in the input ports. After entering the switch, these cells will be treated the same as for the point-to-point cells.

The Multinet switch is a self-routing multistage switch and able to maintain the cell sequence while transmitting cells through the switch fabric. Cells contending for the same output addresses are buffered internally according to a partially shared queueing discipline. Moreover, the Multinet switch is robust to the nonuniform traffic pattern and takes advantage of statistical multiplexing gain through a limited buffer sharing. But for these advantages, it has to use additional $N\log N$ demultiplexers, where N is the switch size. Also, to provide multiple paths while transmitting, the switching element in each stage will be an k by k interconnection network, where $k = 2^{n-i}$, n is the number of stages in the switch, and i is the stage number at which the switching element. This will introduce a much hardware complexity than that in the traditional multistage interconnection network.

2.3 Multicast switch without copy networks

2.3.1 Knockout switch

This is the second approach to implement multicasting in the Knockout switch [23]. This approach does not involve any cell duplication. It is based on the notion of a fast cell filter capable of determining whether an incoming multicast packet is or is not destined for a output within a few bit intervals. The following are the descriptions for this approach.

The multicast module is shown in Fig. 16, which is identical to Fig. 6 except that the cell duplicator has been removed. Therefore, once a multicast packet emerges from the $N{:}L$ Knockout concentrator as a winner, it is buffered and then broadcast directly to all the bus interfaces without any alteration. It is up to the cell filter in each bus interface to determine whether each arriving packet is destined for its output, and this has to be done in a few bit intervals so as not to incur excessive delay. A list of multicast virtual circuits, each of which has this bus interface as a member of its multicast group, can be stored in the bus interface. The job is now to compare the multicast virtual circuit number in the header of an incoming cell to this list. If there is a match, the packet is accepted. Otherwise the packet is discarded. With a 10-bit multicast address word, there may be a maximum of 1024 active multicast virtual circuits, and a straightforward comparison would be hopelessly time consuming. Therefore the use of a special fast cell filter capable of doing these comparisons in a few bit intervals is proposed.

The fast cell filter uses a very simple technique to perform fast table loop-up. An exemplary implementation is shown in Fig. 17. Although the fast cell filter is somewhat more complicated than the others, only $\boldsymbol{M}$ ($\boldsymbol{M} \ll \boldsymbol{N}$) of them are needed in each bus interface as the multicast packets are all funneled onto a small number of buses by the multicast modules. The overall system complexity using either this approach or the cell duplicator method is estimated to be about the same. One characteristic of the fast cell filter approach is that a multicast packet broadcast to multiple outputs on the switch is forced to carry the same virtual circuit number. This problem can easily be overcome by allowing a simple address translation in the output interface module before the packet is transmitted on the outgoing link.

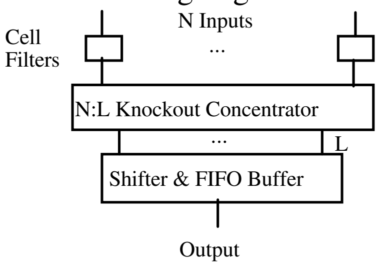


Fig. 16: Multicast module without cell duplicator.

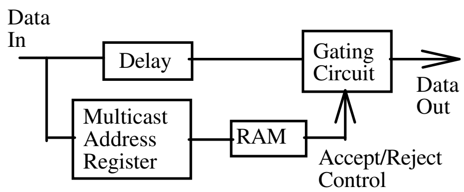


Fig. 17: A fast cell filter.

The Knockout switch provides direct interconnection paths from the switch inputs to the outputs. It greatly simplifies the buffer design and achieves a more efficient switch. The lost packet rate of the Knockout switch can be made as small as desired and the latency is the smallest achievable by any switch. Moreover, the interconnect allows for easy modular growth, simple maintenance procedures, and a design that can be made fault tolerant.

Both the cell duplicator and the fast cell filter methods can be implemented in a modular fashion and conform well to the original Knockout switch architecture. Each guarantees the FIFO cell sequence for the multicasting cells. As far as delay is concerned, the fast cell filter technique is clearly better, particularly so in cases where a multicast packet has many destinations. Therefore, the fast cell filter is definitely favored in applications where the multicast traffic is heavy and involves a large number of simultaneous destinations for each packet. However, the cell duplicator method does not require distributed circuit tables and is somewhat easier to manage in terms of circuit updates. It thus seems to be more appropriate in situations where heavy multicast traffic is not demanded.

Besides that, since it is based on the cross-bar network, the hardware complexity of Knockout switch is very high. It pays in being regarded as having the best performance. Hence there is a tradeoff here - to decrease the hardware complexity, or to have the best performance.

2.3.2 Gauss ATM switching element (ASE)

The ASE (ATM Switching Element) architecture is called 'Gauss', after the main principle of the switching element, namely Grab Any UnuSed Slot. The overall architecture of the Gauss ASE [21] is shown in Fig. 18. It has the same basic architecture as the original Knockout switch but contains different output modules. Fig. 19 depicts the architecture of the output modules in Gauss ASE. By using broadcast buses, Gauss ASE provides a favorable broadcast and non-blocking property.

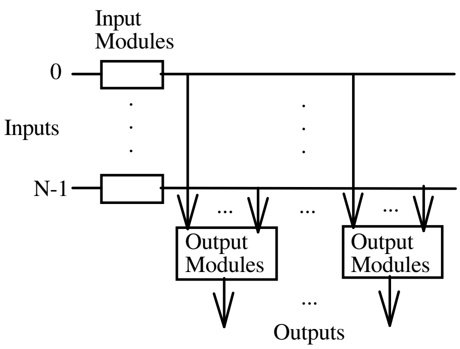


Fig. 18: Overall architecture of Gauss ASE.

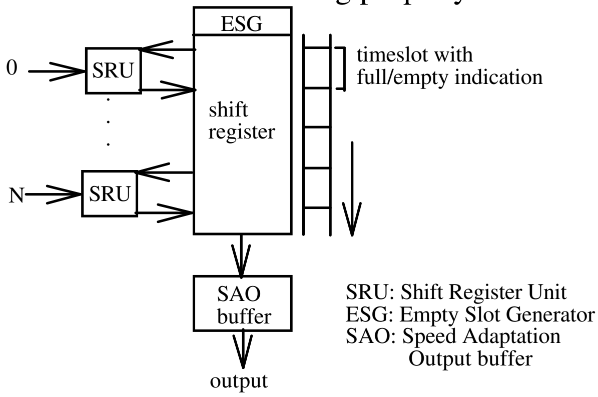


Fig. 19: Output module architecture of Gauss ASE.

To transmit a multicasting cell, Gauss ASE adopts the same idea used in the previous section, but Gauss ASE uses different output modules. When a multicasting cell arrives at an output module, the shift register unit (SRU) checks the cell. If the cell required to be sent to the output module $\boldsymbol{i}$, the SRU in output module $\boldsymbol{i}$ will keep the cell; otherwise, it will discard it. To pass the cell, the SRU has to reserve one timeslot first. At this step, Gauss ASE adopts a shift register which acts like a serial bus to carry the cells in each SRU out of the switch. In each timeslot, the SRU which has reserved the timeslot is allowed to put the cell on the shift register. At most one cell is able to be transmitted in every timeslot. If there is no priority in cells, no preemption is allowed in reserving and transmitting.

Gauss ASE is very similar to Knockout switch, so it carries the same advantages and disadvantages with it - having much higher hardware complexity but with good performance. There is another problem in the Gauss ASE. Due to the sequential property of the empty slot generator (ESG) in the Gauss ASE, the time of delay caused by ESG will be a problem. Also if the incoming cell rate is much higher than the ESG generating rate, than the waiting time of the cells in every SRU might be long. Besides, if there is no proper mechanism to control the timeslot reservations among the SRUs, the SRUs which is closed to ESG will have more chance to reserve the timeslot than one which is far away. This will cause unfairness problem in timeslot reservations.

2.3.3 Sunshine switch

Fig. 20 shows the design of Sunshine switch [11]. The similarities to the Starlite should be clear - a Batcher sorting network with a trap network and recirculator lines. But the Sunshine switch uses a set of $\boldsymbol{K}$ banyans in parallel to route the cells and changes the location of the concentrator. It also replaced the expander network with a selector network. Other import features of the Sunshine switch are that it supports multiple cell priority levels, and that each output port can accept up to $\boldsymbol{K}$ cells in a switch cycle.

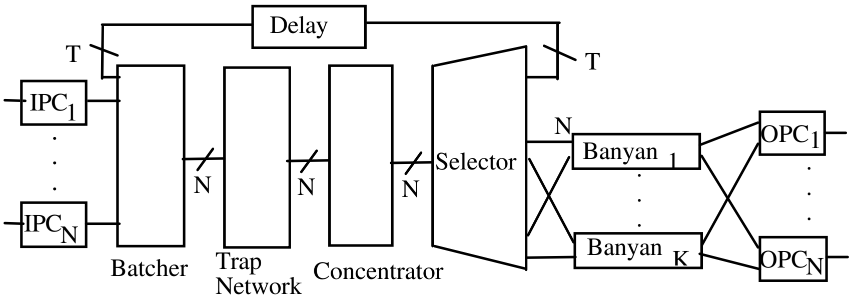


Fig. 20: The architecture of Sunshine switch.

If more than ***K*** cells request a particular address during a timeslot, then the excess cells will overflow into the shared recirculating queue and cells will be shifted to the following timeslot to resubmit into the switch. The recirculating queue consists of the ***T*** parallel loops and the ***T*** dedicated inputs on the Batcher sorting network. A delay block inserted within these loops aligns recirculated cells with the new arrivals submitted by the input port controllers at the start of the following time slot. Cells then enter the Batcher network and continue the trip in the switch.

The Sunshine switch supports multicasting in each input port rather than by building a special copy network. Each input port contains a processor that recognizes cells from multicast ports and makes the appropriate number of copies and sends these copies into the switch as many unicasting cells.

The advantages of the Sunshine switch are: (1) it is a self-routing switch architecture. (2) it is built based on Batcher-banyan networks and combines a recirculating queue with output queues in a single architecture. This queueing strategy results in an extremely robust and efficient architecture. (3) The output queues discipline enables the architecture to achieve the extremely low cell loss probabilities. (4) The parallel banyan networks and output queues can effectively handle the traffic in a bursty environment where output overloads are likely. (5) The combined queueing strategy used in Sunshine switch reduces the internal buffer requirements, simplifies the interconnection network, and improves the efficiency over architectures using either queueing technique individually.

By providing the above advantages, the Sunshine switch has much higher hardware complexity. Besides, it combines the recirculating lines inside the switch fabric. The recirculating lines have the delay problem, and the cell sequence order keeping problem.

2.3.4 Lee's multicast switch

A broadcast Banyan network is a Banyan network with switch nodes capable of cell replications. Fig. 21 shows a generalized self-routing in Banyan network to multiple addresses.

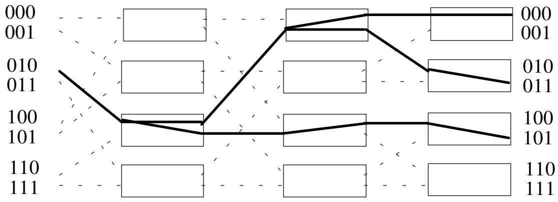


Fig. 21: An input-output tree generated by self-routing in Banyan network.

The following describes how the multicasting tree expanded in a broadcast Banyan network. When a multicasting cell contains a set of arbitrary $\boldsymbol{n}$-bit destination addresses and arrives at a node in stage $\boldsymbol{k}$, the cell routing and replication are determined by the $\boldsymbol{k}$th bit of all the destination addresses in the header. If they are all "$\boldsymbol{0}$" or all "$\boldsymbol{1}$", then the cell will be sent out to upper link or lower link, respectively. Otherwise, the cell and its replica are sent out on both links with the following modified header information: the header of the cell sent out to upper link or lower link, contains these addresses in the original header with their $\boldsymbol{k}$th bit equal to $\boldsymbol{0}$ or $\boldsymbol{1}$, respectively. The modification of cell headers is performed by the node whenever the cell is replicated. By this way, the set of paths from any input to a set of outputs forms a (binary) tree embedded in the network, and it will be called an input-output tree or multicasting tree. Fig. 22 shows the corresponding multicasting tree of the example in Fig. 21.

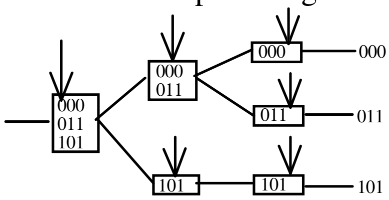


Fig. 22: The multicasting tree for the example in Fig. 21.

The actual Lee's multicast switch [16] is shown in Fig. 23. When the multicasting cells are received at the running adder network, the number of copies specified in the cell headers is added up recursively. The dummy address encoders form new headers consisting of two fields: a dummy address interval and an index reference. The dummy address interval, formed by adjacent running sums, is represented by two binary numbers, namely, the minimum and maximum. The index reference is equal to the minimum of the address interval. It is used later by the trunk number translators to determine the copy index. The broadcast Banyan network replicates cells as showed in Fig. 21. When copies finally appear at the outputs, the trunk number translators compute the copy index for each copy from the output address and index reference. The broadcast channel number together with copy index forms a unique identifier for each copy. The trunk number translators then translate this identifier into a trunk number, which is added to the cell header and used by the switch to route the cell to its final destination.

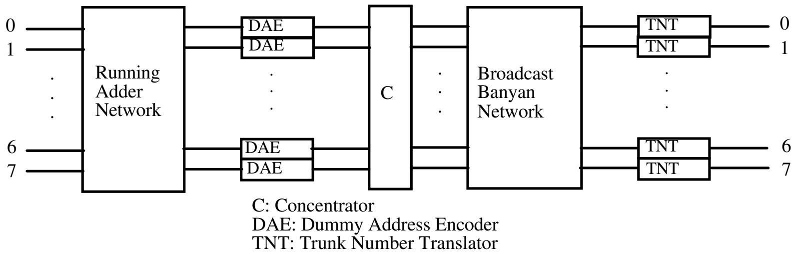
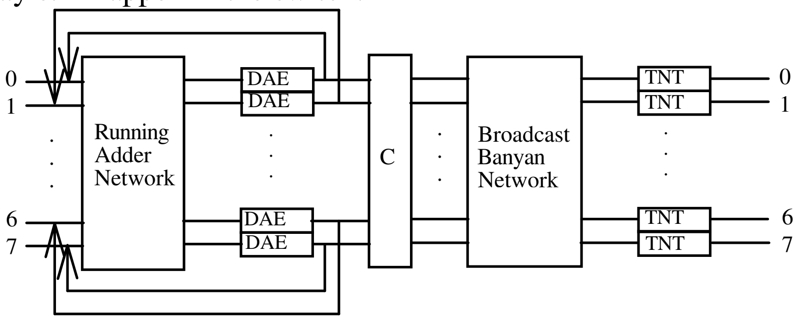


Fig. 23: Lee's multicast switch with $N = 8$.

There are two problems inherent in the design of a space-division-based multicast switch [16]. One is the overflows of the copy network capacity. It means the total number of copy requests exceeds the number of outputs in the copy network. The other is output port conflicts in the routing network. It occurs when multiple cells request the same output port concurrently. To improve the above switch fabric to avoid the first problem, Lee puts some modifications into his original proposed switch in Fig. 23. The result is shown in Fig. 24 [2]. A cyclic running adder network provides a complete solution to the performance degradation caused by overflows of the copy network capacity and the input fairness problem in the broadcast Banyan network. Although the modified switch can prevent the overflows problem in the copy network, output port conflicts may still happen in the switch.



C: Concentrator DAE: Dummy address Encoder TNT: Trunk Number Translator

Fig. 24: Lee's multicast switch with cyclic line and switch size is 8.

Beyond the previous discussion, there is also a serious problem inside broadcast Banyan network. When there are more than one multicasting tree in the broadcast Banyan network, it might occur that a internal link is used by more than one tree. This phenomenon is called as internal conflict or internal blocking. This phenomenon does not only degrade the switch performance, but also increases the cell loss probability in the switch.

2.3.5 A recursive multistage structure for multicast ATM switching

This switch provides self-routing switching nodes with multicast facility, based on a buffered N by N multistage interconnection network with external links connecting outlets to inlets [9]. The structure of this multicast switch is shown in Fig. 25. Such a network is able to route a cell to the addressed output for transmission and to generate M copies of the same cell (with $M \ll N$) on M pre-defined adjacent outlets. M is the "multiplication factor" of this multicast connection network (MCN). In order to reach more than M outputs, some of the M cells generated in the first crossing of the network are recycled back to the corresponding inputs, and each of them again generates other cells until the requested number of copies is obtained.

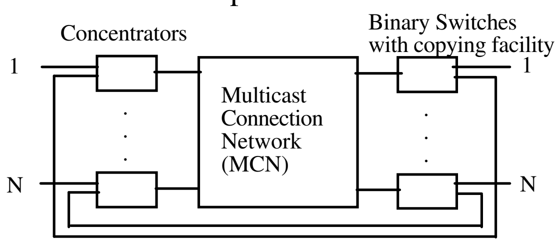


Fig. 25: The architecture for a multicast connection network with external links from inputs to outputs.

For a MCN with general multiplication factor M, if a "copy bit" in the routing tag is set to "$\boldsymbol{0}$" (unicast cell), the input cell is addressed to a single output line; if the "copy" bit is set to "$\boldsymbol{1}$" (multicast cell), C copies (with $2 \leqq C \leqq M$) are simultaneously generated on C consecutive outlets, starting from the addressed output. The number C is specified in the routing tag.

If the requested copy number B (with $2 \leqq B \leqq N$) is less or equal to M, all copies can be generated in just one crossing of the network. If $B>M$, more crossings are necessary. The concentrators at the inlets merge input and recycling cells, while the binary switches on the outlets manage forwarding and recycling by routing the cells respectively toward their upper and/or lower output line.

Intermediate copies appearing at valid final outputs are sent out while at the same time they can recycle back, if required, to generate other copies. For this purpose the output switches are also able to spread an input cell on both output lines. During the iterations some cells may have to recycle back to the inputs to be routed toward their final destinations.

Although the proposed recursive mechanism seems very simple, the switching elements inside the switch architecture have to do more operations than routing. This will make the switching elements more complex and the hardware complexity will be high. Besides, recycle lines will introduce delay to the switch, and cells leaving the switch might be out of the sequence.

2.3.6 Multinet switch

The following describe the second method for the Multinet switch [14] to deal with multicasting traffic. Recalling from Fig. 13 and 14, the header of the cell contains addresses of the destination output ports. The demultiplexer handles the cell in the same fashion as before but now a cell can be sent to both upper and lower virtual FIFO's at the same time. First consider the case of broadcast routing where a cell in the input port is sent to all the output ports. A cell has an extra bit "1" or "0" in the header indicating that the cell is to be broadcast or to be sent to the output ports according to its destination addresses respectively. If the broadcast bit is "0", then the cell is routed according to the unicasting routing or multicasting routing depending on the number of destination addresses in the header. When a demultiplexer detects the broadcast bit to be "1," it sends the cell to both upper and lower virtual FIFO at the same time. Thus, the cell is replicated by the demultiplexer at each stage and by the time it reaches the last stage, there are N replicated cells to be delivered to every output port.

Now look at the multicast routing where a cell in the input port is sent to a particular set of output ports rather than to all output ports. The multicast routing requires additional hardware which is needed to replicate and to route cells to the appropriate output ports. For example, suppose a cell in the input port "0" needs to multicast to the output ports "0," "4," and "6." The cell header contains three destination addresses (000, 100, 110) and the counter for the number of destination addresses. The demultiplexer in the first stage checks the header to determine whether the cell has to be broadcast or not. The counter is then examined to see if there is more than one destination address. An exclusive-OR operation is used to determine the replication function at each stage. Other multicast approaches using embedded routing tables are proposed in [14] and [16].

Although the routing is simple with additional hardware for a small number of replications, the structure becomes quite complex as the number of multicast destination addresses increases. Moreover, the header occupies a large proportion of the cell length if many addresses are to be added into the header.

2.3.7 Growable packet switch

The Growable Packet Switch (GPS) [17] is a three-stage, Clos like, structure as shown in Fig. 26. The first two stages form a memoryless and self-routing network, which is used to route packets from an input line to the appropriate output section. These first two stages induce only one or two cell delay. The output sections that make up the third stage are small packet switches. The switches route the cells they receive to the appropriate output lines. Another appealing characteristic of the GPS architecture is that nearly any type of fast packet switch may be used as an output section. In addition, the modularity of the GPS architecture makes it naturally growable.

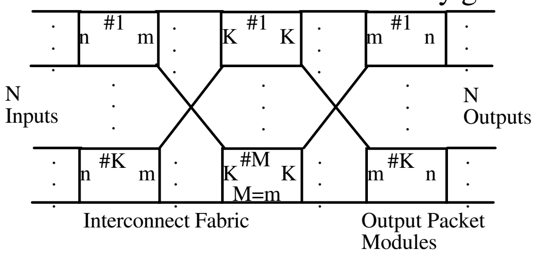


Fig. 26: The architecture of a growable switch.

As described in [15], the GPS switch performs the cell duplication as close to the switch's output ports as possible. The reasoning behind this approach is that the later a cell is duplicated, the less of the switch's resources the multicast will require. Since this approach duplicates cells as close to the output as possible, no duplication is performed in the first stage. The second stage duplicates and sends one copy to each necessary third stage module. The third stage modules, if necessary, duplicate the cell first and then send to the appropriate output lines. Fig. 27 shows an example of multicast routing with cell replications performed as close to outputs as possible.

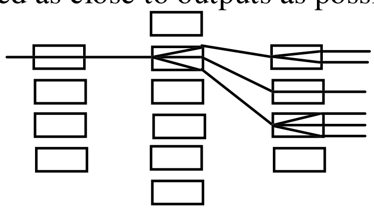


Fig. 27: Multicast routing with cell duplications performed as close to the outputs as possible.

The GPS architecture can be used to grow, in a modular fashion, large packet switches out of nearly any type of small packet switches. It provides the growability while only adding a delay of one to two cell times to that of the smaller packet switches. To add the multicasting capability into the growable packet switch, every switching element needs to have the replication function in itself. This surely will increase the complexity of the switch elements and the whole switch. Moreover, because the GPS adopts broadcasting tree as its replication scheme, the internal blocking phenomenon, as described in 2.3.2, in the last stage is a serious problem that will degrade the performance of the switch.

2.3.8 Broadcast ring sandwich network

An (N_1, M, N_2) ring sandwich network [22] has N_1 input ports, N_2 output ports and M broadcast cells. The (N_1, M, N_2) ring sandwich structure is based on the cascading of three subnetworks, each contributing a different interconnection function. The three component subnetworks of a broadcast ring sandwich structure are called: the presentation network, the broadcast ring, and the distribution network respectively. Each of the N_1 input ports of the presentation network is interfaced to a sender, and each of the N_2 output ports of the distribution network is interfaced to a receiver. A schematic of an (N_1, M, N_2) ring sandwich network is shown in Fig. 28. The sandwiching of the broadcast ring between the presentation network and the distribution network has obviously motivated the name for this interconnection structure.

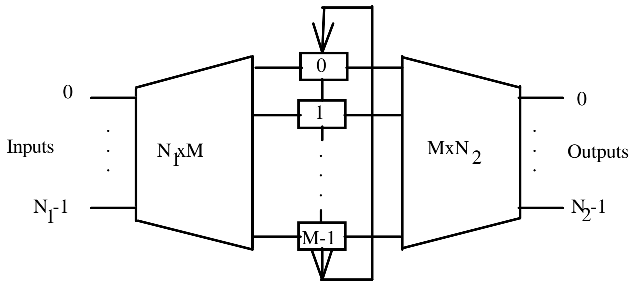


Fig. 28: The switch fabric of a ring sandwich broadcast network.

A broadcast connection in an (N_1, M, N_2) ring sandwich network from sender i to a set of k receivers is realized in a straightforward manner in a ring sandwich network. A single connection path from sender i to the broadcast ring is established through the presentation network. This single connection path from sender i is then fanned-out within the broadcast ring to k outputs of the ring. Finally, k disjoint connection paths are established through the distribution network to the k receivers.

According to [22], the ring sandwich network is quite simple and easy to implement. But the key component, the broadcast ring, might become a problem in the whole switch. The relative size of the broadcast ring to the number of network input ports and network output ports critically influences the realization of broadcast connections. As the number of broadcast cells M grows larger, the need to rearrange currently fanned-out connections on the broadcast ring in order to satisfy a connection request will decrease. However, as M increases, the total cost of switching hardware in the ring sandwich network also increases. Moreover, the inter-connection method in the presentation network and distribution network is also an important point to consider. If these two networks are built on cross-bar networks, the hardware complexity will be high. But if they are based on Batcher-banyan network, the time complexity will need to be taken into consideration.

2.3.9 Multicast output buffered ATM switch (MOBAS)

In the original Knockout switch [23], it has been showed that when the arrival traffic pattern is uncorrelated among input ports and uniformly distributed to all output ports, and L is greater than 12, the probability of more than L cells destined to any particular output port in each transmission cycle is very low, e.g., less than 10^{-10}. According to this, L can be used instead of N as the number of internal links connected to each output port in the original Knockout switch. The hardware complexity of the switch therefore can be reduced to O(LN) from O(N^2). In 1991, Chao used this idea to construct a switch architecture called grouping network (GN) [3]. Fig. 29 shows the architecture of an unicasting network consisting of three stages of GNs. In Fig. 30, the modular structure for the GN at the first stage in Fig. 29 is depicted.

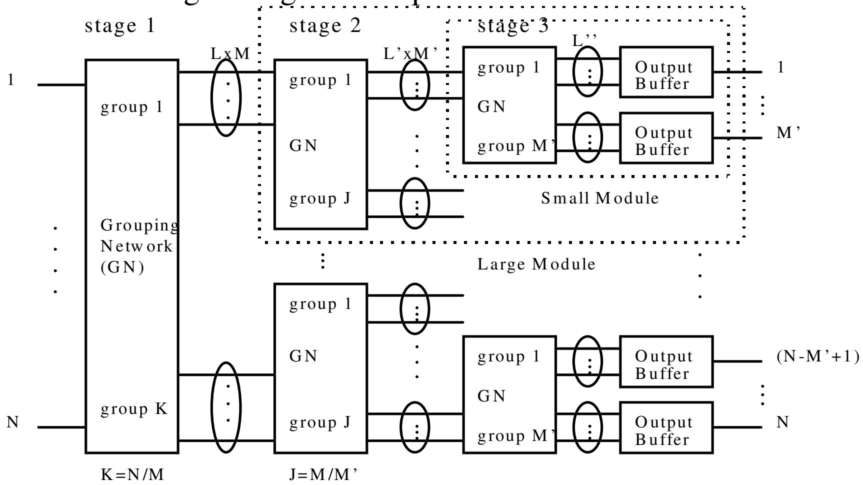


Fig. 29: The architecture of an unicasting network consisting of three stages of GNs.

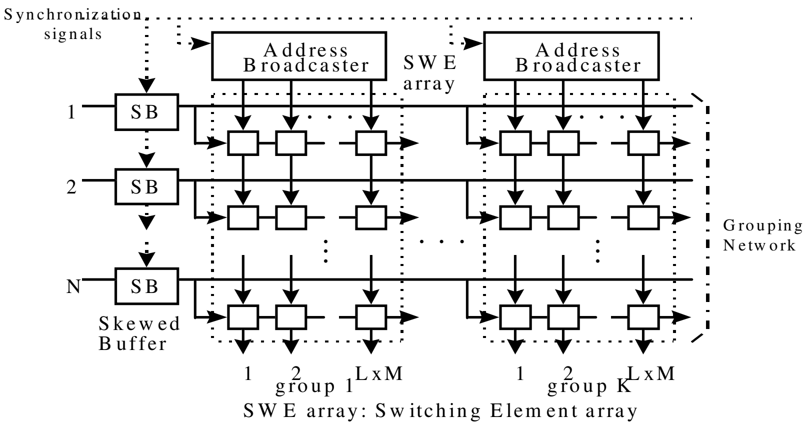


Fig. 30: The modular structure for the GN at the first stage in Fig. 29.

In 1995, Chao improved the GN into the multicast grouping network (MGN) and used it to build the multicast output buffered ATM switch (MOBAS) [4]. The architectures of the MGN and MOBAS are showed in Fig. 31 and Fig. 32, respectively.

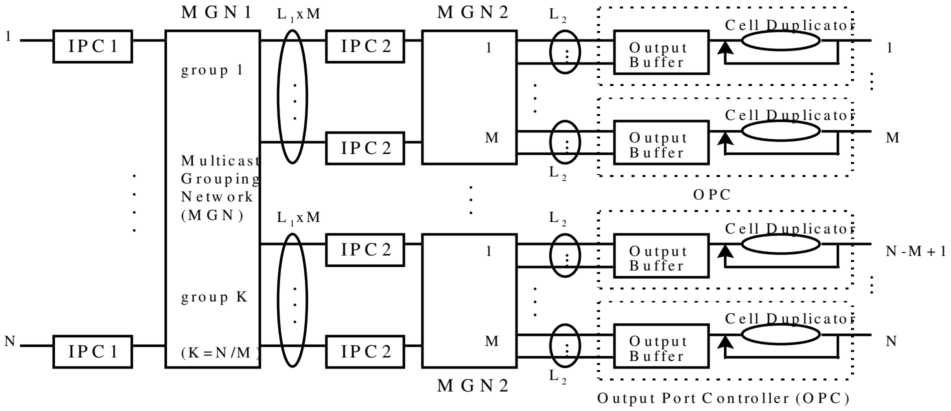


Fig. 31: The architecture of the multicast output buffered ATM switch (MOBAS)

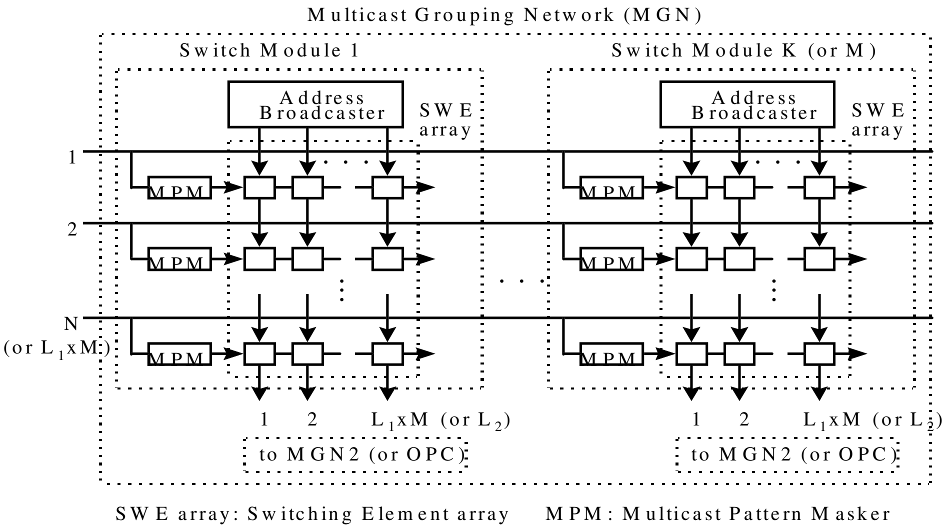


Fig. 32: The architecture of the multicast group network.

The MOBAS performs cell routing and replicating in multicast transmitting by combing the multicasting tree and the broadcast buses, as described above. In Fig. 33, when a multicasting cell is transmitted through the MOBAS, the routing links are forms a multicasting tree, but inside each MGN, the cell is routed by the broadcast buses.

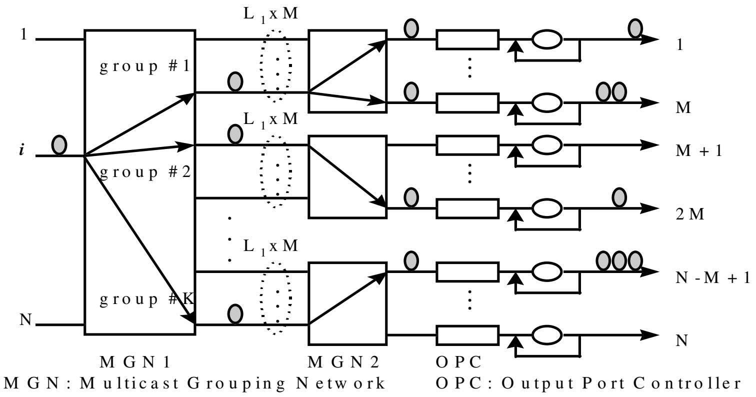


Fig. 33: An example of replicating multiple cells for a multicast call in the MOBAS.

There is a special component in the output port controller (OPC) for each output port in the MOBAS: the cell duplicator (CD). The CD duplicates each arrival cell to multiple copies as needed. In this way, when a multicasting cell requests multiple copies to the same output port, only one copy of the multicasting cell is stored at the required output buffer instead of storing multiple copies.

The MOBAS is a remarkable architecture for the multicast ATM switch, but there are still some problems. First, because the internal routing links form a broadcasting tree, as showed in Fig. 21, the internal blocking phenomenon mentioned in 2.3.2 will occur. Second, the general architecture of the MOBAS is an MIN, and the switching elements in each stage are constructed by MGNs. Although the grouping principle is adopted, the MGNs are built based on broadcast buses, and will make the switching elements more complex than those in the original MIN. Moreover, because the Knockout concentrators are applied in each switching element of the switch, the queueing delay will be another serious problem to the MOBAS.

2.3.10 Abacus switch

In 2.3.9, Chao uses the grouping principle to build two ATM switches [5]. As showed in Fig. 29 and 31, one of them is for unicasting, and the other is for multicasting. However, there is another problem to be solved. When the number of routing links is less than the number of incoming cells destined for the same output port (or output group), cells might be discarded. Therefore, in 1997, Chao proposed another new architecture which will eliminate the possibility of cells being discarded due to the loss of contention in the switch fabric. The new scalable multicast ATM switch is called Abacus switch, and its architecture is showed in Fig. 34.

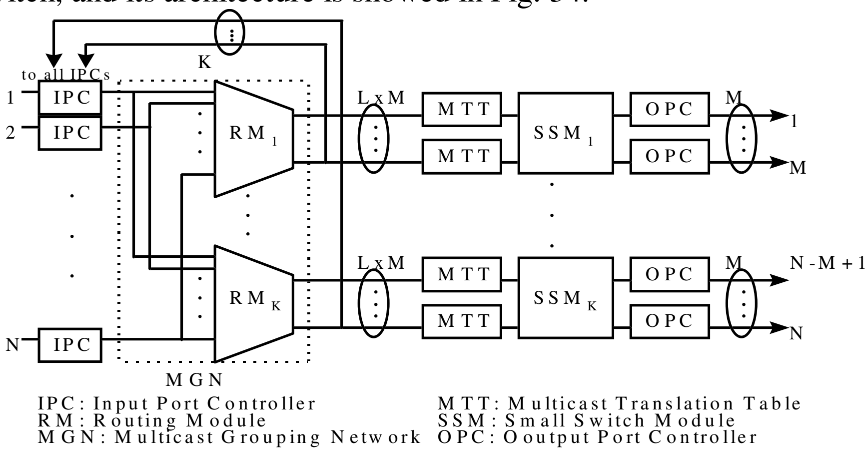


Fig. 34: The architecture of Abacus switch.

The Abacus switch consists of a nonblocking switch fabric followed by small switch modules at the output ports. In the switch, the cell replication and cell routing are performed simultaneously. Cells replication is done by broadcasting incoming cells to all routing modules (RMs). RMs then selectively route cells to their required output ports. Cells routing is done by the switch elements array (SWE). Fig. 35 depicts the architecture of multicasting grouping network and the switch elements array.

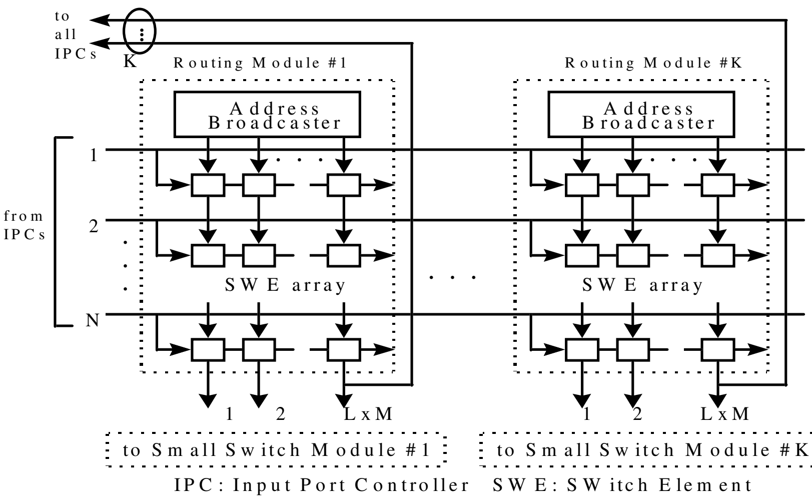


Fig. 35: The architecture of multicast grouping network (MGN).

The architecture in Fig. 35 is very similar to Fig. 32, but with additional ***K*** feedback lines to all input port controllers (IPCs). These feedback lines pass the acknowledgements from ***K*** small switch module back to all IPCs. Because there are buffers in the IPCs, every IPC can decide whether its cell is successfully transmitted or not according to these feedback acknowledgements. For the routing of a multicasting cell in the Abacus switch, it is the same as in MOBAS, and we omit its descriptions here. Fig. 36 is an example of routing a multicasting cell in the Abacus switch.

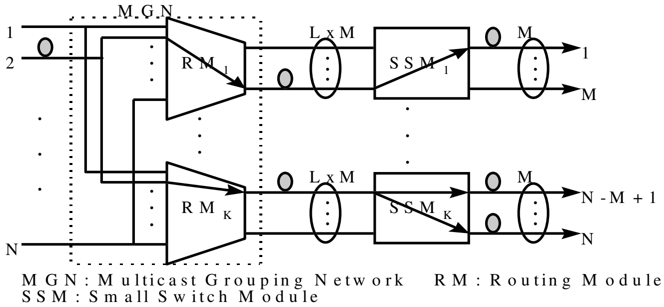


Fig. 36: The example of routing a multicasting cell in the Abacus switch.

The MOBAS switch and the Abacus switch are built from the same principle. The problems in the switches are all the same: the internal blocking problem, the hardware complexity problem in each switching elements, and the queueing delay of the whole switch. For scaling up the switch fabric, Fig. 37 shows the architecture for a large-scale Abacus switch. Although the switch is very scaleable, there are O($\boldsymbol{LMK_2K_1}$) wires interconnected between the stage of multicast grouping networks (MGNs) and the stage of concentration modules (CMs). According to Fig. 36, O($\boldsymbol{LMK_2K_1}$) can be replaced by O($\boldsymbol{LN^2/n}$), and it will be larger than the hardware complexity of the original Knockout switch O($\boldsymbol{N^2}$), if $\boldsymbol{L/n}$ is greater than 1. Besides, the external stages of CMs and MGNs will cause more delay to the cell transmitting in the switch.

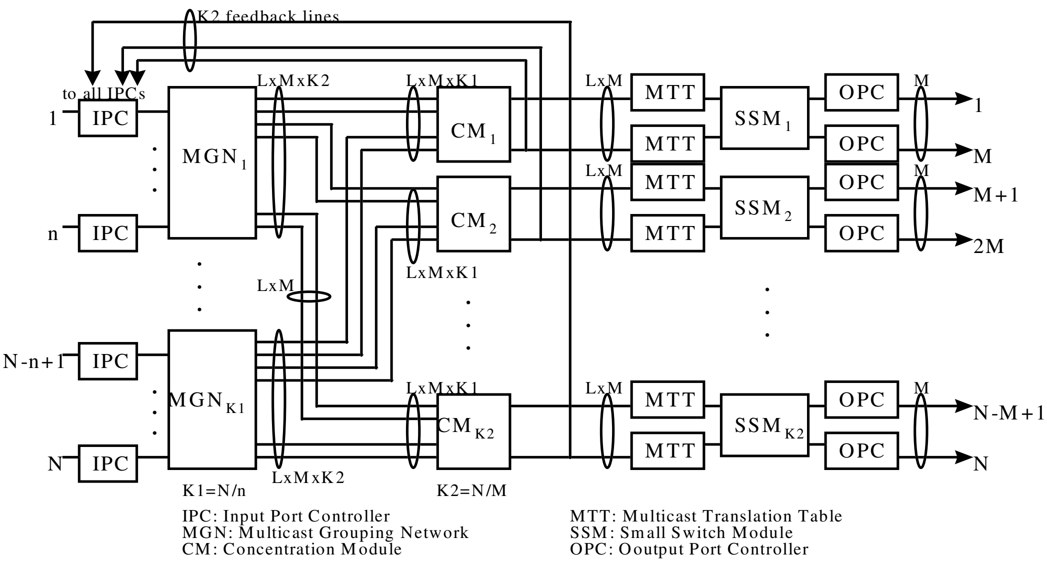


Fig. 37: The architecture for a large-scale Abacus switch.

2.3.11 Dilated network and prioritized services, internally nonblocking, internally unbuffered multicast switch (PINIUM)

In the previous switches which apply the broadcasting tree as the multicasting routing mechanism, there is a serious problem: the internal path conflicts in the routing module. To reduce or prevent the conflict phenomenon, there are two ways: one is increasing the link dilation in the routing module [19], as showed in Fig. 38, and the other is using multiple routing plane and each plane only supports one broadcasting tree [15], as showed in Fig. 39. The architecture in Fig. 38 is a dilated network, and the architecture in Fig. 39 is called prioritized services, internally nonblocking, internally unbuffered multicast switch (PINIUM).

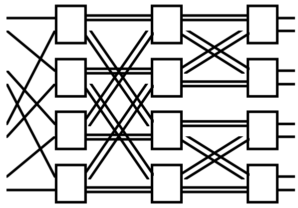


Fig. 38: The architecture of Banyan network with dilation 2 in the internal links.

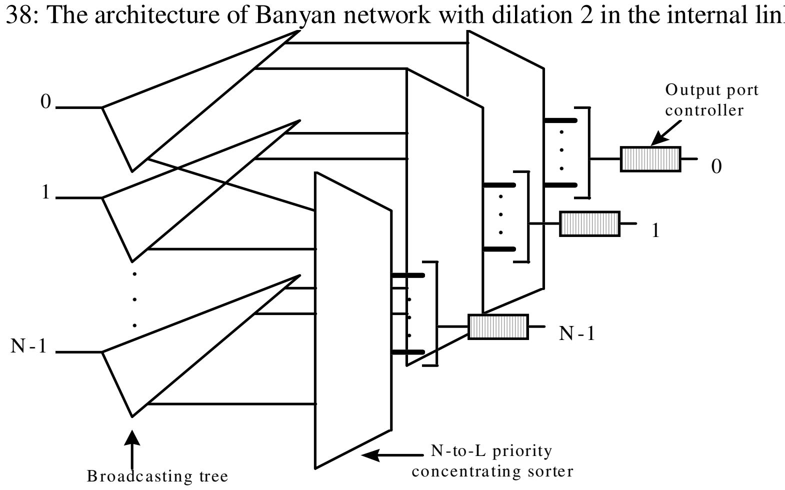


Fig. 39: The architecture of a multicast switch with multiple routing planes.

Although these two schemes did provide more alternatives for the routing module to deliver the cells and eliminate the possibility of internal path conflicts, they still have some drawbacks. First, the architecture in Fig. 39 has no internal path conflicts, but the price is too high. It uses one routing plane for each broadcasting tree. If there are less than ***N*** broadcasting trees in the same transmission cycle, some planes will be idle and wasted. Second, if the architecture in Fig. 38 provides no internal path conflicts, the dilation of the internal links will be very unappreciable.

Moreover, increasing the dilation of the internal links will increase the hardware complexity of the switching elements and the entire switch. Besides, if the routing module can delivered more than one broadcasting tree at one transmission cycle, output port of the switch will receive more than one cell in the same transmission cycle. If the output port could not provide sufficient capability to deal with more and more incoming cells, although there will be no internal path conflicts inside the routing module, the conflicts at the output port will be serious. Furthermore, more cells that each output port can receive, more hardware cost is needed for the output ports, and the entire switch.

Therefore, to achieve the best performance of the switch, the trade-off between the hardware complexity and the dilation of the internal paths or the number of routing planes needs to be considered carefully.

2.4 Prescheduling Before Copying

2.4.1 Guo and Chang's switch

As mentioned in [16], there are two major problems inherent in the design of a space-division-based multicast packet switch. One is the occurrence of overflows of the copy network capacity, and the other is output port conflicts in the routing network. The first problem can be resolved by adding the recirculating lines into the copy or whole switch architecture, such as Starlite switch [13], SCOQ switch [7], Sunshine switch [11], Lee's modified switch [2] or the recursive multistage structure 9]. Another way to overcome this problem is providing more switching paths so that more cells can be transmitted through the switch at same time, such as Turner's switch [20], LGMIN network [24], Multinet switch[14], growable packet switch [17], ring sandwich network [22], MOBAS [4], Abacus switch [5] or dilated network [19]. But this approach might not solve the problem completely especially when more multicasting cells than provided paths are injected into the switch. As for the second problem, all switches mentioned above do not provide a method to prevent its happening. Some of them provide output buffering to resolve this output conflicts problem such as multicast Knockout switch [10], Gauss ASE [21], and SCOQ switch [7].

In [12], Guo and Chang proposed another design method for ATM multicast switch. Their switch can prevent both the overflows of the total number of copy requests and the output ports conflicts and internal path conflicts. Fig. 40 is the overall architecture of Guo and Chang's multicast switch.

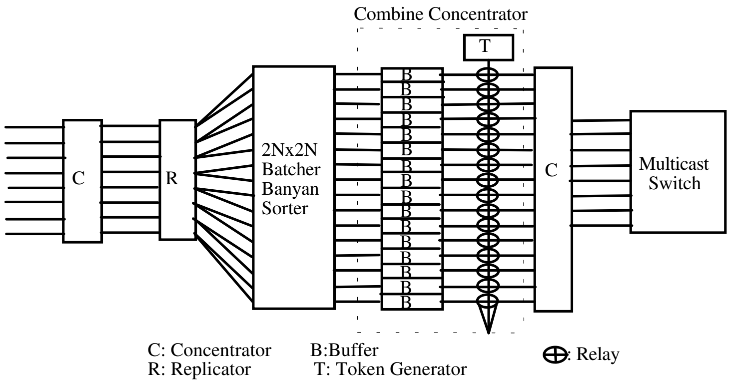


Fig. 40: Guo and Chang's multicast switch.

The cell format in Guo and Chang's switch is shown in Fig. 41. ***A*** is the activity bit which is 1 if the input is active (nonempty), and 0 if it is idle (empty). ***OPN*** is ***n*** bits output port numbers in unary form. ***IPN*** is ***logn*** bits input port number in binary form. ***SK*** is ***logn*** bits ***sort_key*** and is used in the Batcher Banyan Sorter.

A	OPN	IPN	SK	DATA

Fig. 41: The cell format

In Fig. 40, the first concentrator sorts those input cells into two parts: the empty cells and the nonempty cells. The empty cells will be sorted out and the latter will be injected to the next unit, the Replicator. The Replicator duplicates the input cells and fills the

sort_key field. The value of the ***sort_key*** is the maximum output port number for the original cell and the minimum output port number for the duplicate. After passing through the Replicator, the ***2N*** cells are fed to the next unit, the Batcher Sorter. The Batcher Sorter sorts the cells according to the field ***sort_key*** into nonincreasing order from top to bottom, and then sends them into the buffers in the following Combine Concentrator. The combine concentrator resembles the output port concentrator in the Gauss ASE. Each buffer need only have the capability to hold one cell. The Combine Concentrator will combine those multicast cells into monotone subsequences. A monotone sequence is a sequence of strictly increasing or decreasing integers. If the output ports sequence is monotone, then it will not block in the switch.

This switch uses prescheduling concept to prevent the overflows of the total number of copy requests. Also the result output port sequence is output ports conflict-free and internal path conflict-free.

The major problem in Guo and Chang switch is the time it takes for pre-scheduling. Because of the combine concentrator, the pre-scheduling time might not be negligible. Besides that, the hardware complexity costs for pre-scheduling seems too large. The $2N \times 2N$ Batcher Banyan sorter is the key component in its hardware complexity.

2.4.2 TATRA scheduling algorithm

In 1997, McKeown et al. proposed a new idea to schedule the multicasting cells at the input ports [18]. Compared to the above pre-scheduling mechanism, they focused on the distribution of the residue among all the input ports after each transmission cycle. The residue is the set of all cells that can not be transmitted, and remain at the head of line of the input ports at the end of each transmission cycle. The routing module applied in the paper is the crossbar switch, as showed in Fig. 42, and a multicasting cell is successfully transmitted until all its requested output ports received its copies.

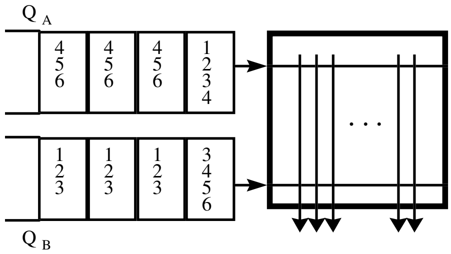


Fig. 42: The 2 by N multicast crossbar switch with single FIFO queue at each input.

By these definitions, there are two simplest schemes to deal with the residue at the end of each transmission cycle: one is centralized algorithm and the other is the distributed algorithm. In these two algorithms, the centralized algorithm provides the highest throughput and low delay. However, it has two drawbacks that make it unsuitable for applying in the physical network switch: it can starve input ports indefinitely, and it is difficult to implement in hardware. But the results of centralized algorithm can be treated as the upper bound on throughput for comparing other algorithms. McKeown et al. proposed the TATRA scheduling algorithm that approximates the throughput of the centralized algorithm. In the following, we will describe the TATRA scheduling algorithm briefly.

The TATRA scheduling algorithm is motivated by Tetris, the popular block-packing game. As showed in Fig. 43, all cells will be represented as an Tetris matrix while scheduling. The expressions of labeling in the figure are described as following: each small box represents a copy of the original cell, and the number labeled on each cell is the input port number where the corresponding original cell comes in. The bottom row of the box at columns 1, 3, 5 are all identical copies of a cell from input port 1 destined to output ports 1, 3, 5. Similarly, according to the figure, the cell at the head of line of input port 2 will be transmitted to output ports 2, 3, 4, and 5.

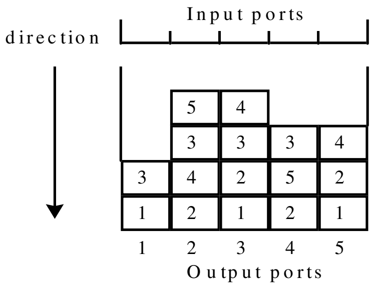


Fig. 43: An example that cells are represented as a Tetris matrix.

In the following, we will only explain the TATRA scheduling algorithm with examples. In Fig. 43, each row represents the cells which should be transmitted in the same cycle. After TATRA scheduling algorithm, the matrix in Fig. 43 can be represented as that in Fig. 44-(a). The TATRA scheduling algorithm arranges the cells so that the routing module can deliver most multicasting cells in each transmission cycle. In the third row of the Fig. 44-(a), because there is no multicasting cells able to be transmitted with cells from input port 2, an “NULL” box with label “N” is used to fill the empty place. Similarly, other boxes with label “N” are in the same situation. Fig 44-(b) shows that after one transmission cycle in Fig. 44-(a), new cells from input ports 1 and 5 are added to the matrix.

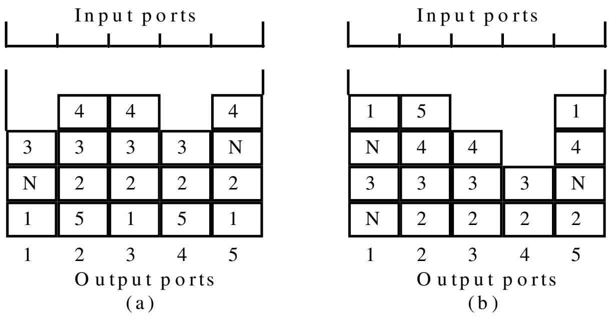


Fig. 44: TATRA schedules (a) cells of Fig. 41, (b) new cells from input ports 1 and 5 at time 2.

3. Performance evaluation

3.1 Simulation assumptions

In this section, simulations are conducted to test three multicast switches. They are the SCOQ switch, the Lee's switch, and Guo and Chang's switch. Let ***A*** represent the event that an incoming cell is active or inactive, and let ***B*** represent the event that an active cell requests unicasting or multicasting transmitting. In the simulation, assume ***Pro(A = active) = p***, ***Pro(B = multicasting transmitting) = q*** and these two events are all

Bernoulli processes. Fig. 45 depicts the relationship of these two events and the decision tree used in the simulation.

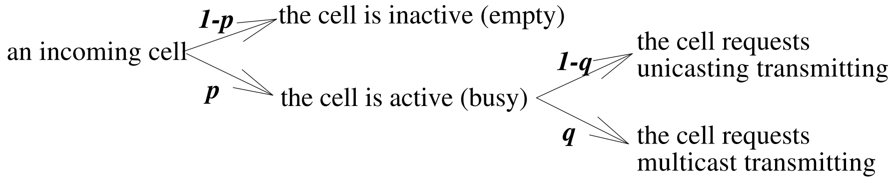


Fig. 45: The decision tree used in the simulation.

Furthermore, assume that every output port in the switch has the same probability of being chosen.

3.2 Simulation results

By using p = 1 (every input port is active), Fig. 46 and Fig. 47 are the results of total cycles needed for different q among three switches in two different switch sizes. Some conclusions from these two figures: (1) When there are not many multicasting patterns in the switch traffic (for example, $q <= 0.5$), the SCOQ switch works better than the other two switches. But when there are more and more multicasting patterns in the switch traffic ($q >= 0.7$), the performance of SCOQ degrades more than the other two switches. The reason is because of the restriction of the copy network in the SCOQ switch. The copy network will replicate less and less cells if there are more and more multicasting patterns in the traffic. (2) Guo and Chang's switch model needs less cycles than Lee's switch. It indicates that prescheduling multicast cells before copying indeed works. Besides that, the performance degradation of Lee's switch and Guo and Chang's switch is not so unstable as SCOQ switch when there are more and more multicasting patterns in the traffic.

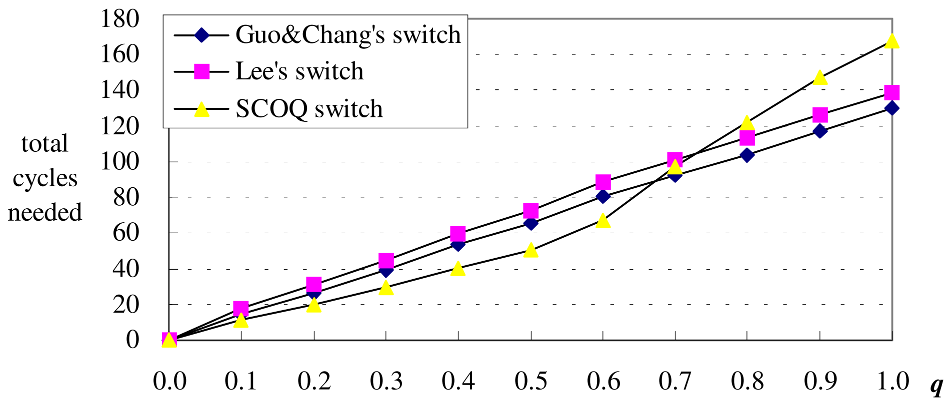


Fig. 46: Comparison of total cycles needed for different q among three switches with N = 256 and p = 1.

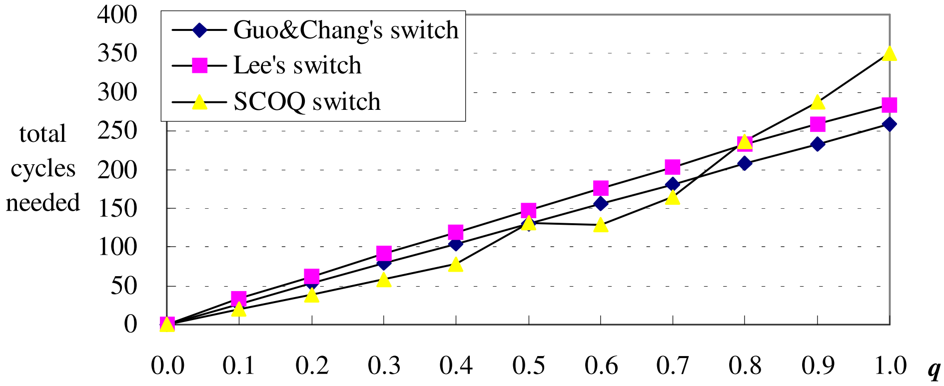


Fig. 47: Comparison of total cycles needed for different q among three switches with $N = 512$ and $p = 1$.

To further test the prescheduling concept, Fig. 48 and Fig. 49 show the total cycles needed for the Guo and Chang's switch model under three types of output port selection patterns with $N = 512$ and $N = 1024$ respectively. Fig. 50 and Fig. 51 show the same testing for Lee's switch and SCOQ switch respectively. The three types of patterns are: (1) All output ports can be chosen as the destination. (2) Only upper half of the output ports can be chosen. (3) Only odd numbered output ports can be chosen.

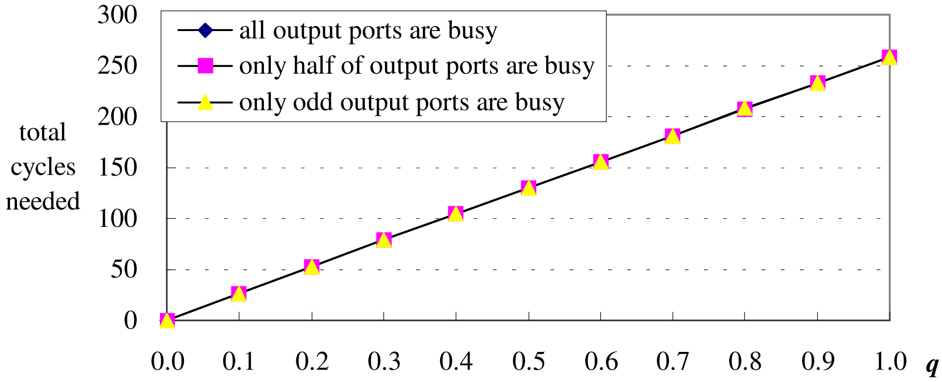


Fig. 48: Total cycles needed of Guo and Chang's switch under three types of traffic patterns for different q with $N = 512$ and $p = 1$.

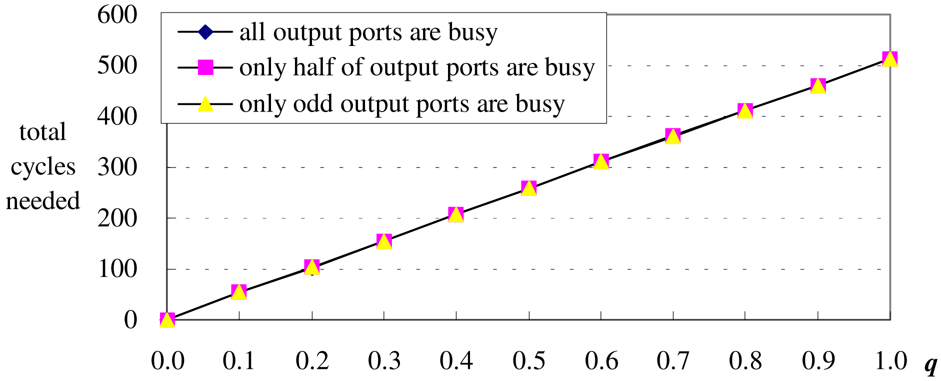


Fig. 49: Total cycles needed of Guo and Chang's switch under three types of traffic patterns for different q with $N = 1024$ and $p = 1$.

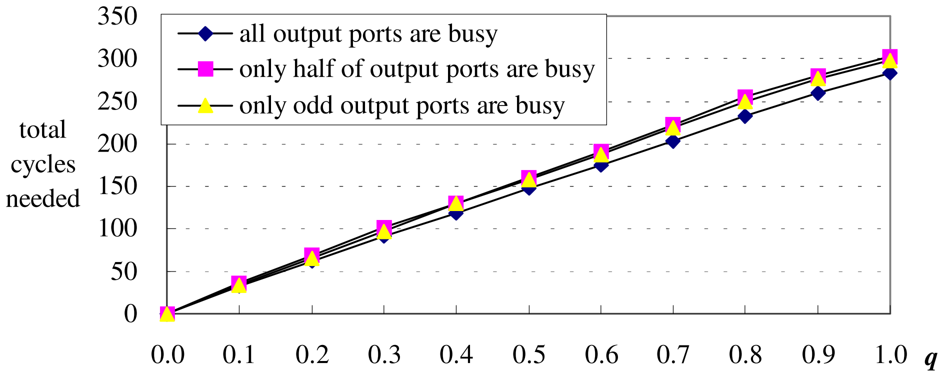


Fig. 50: Total cycles needed of Lee's switch under three types of traffic patterns for different $\boldsymbol{q}$ with $\boldsymbol{N}$ = 512 and $\boldsymbol{p}$ = 1.

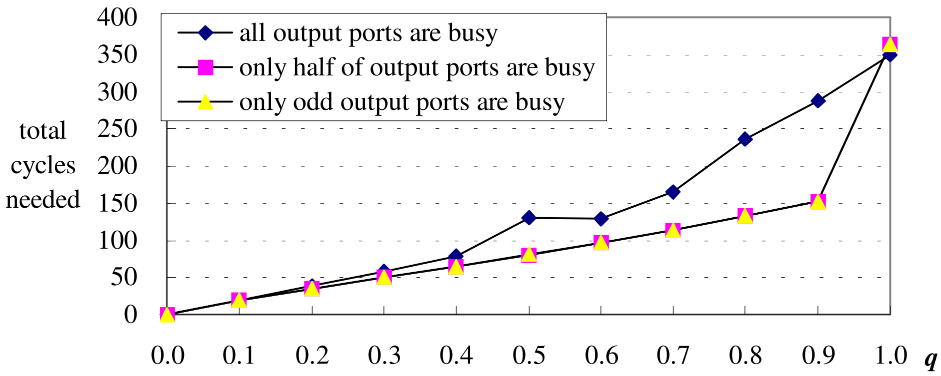


Fig. 51: Total cycles needed of SCOQ switch under three types of traffic patterns for different $\boldsymbol{q}$ with $\boldsymbol{N}$ = 512 and $\boldsymbol{p}$ = 1.

According to these figures the following conclusions can be drawn: (1) Guo and Chang's switch model is robust with respect to output pattern variations. (2) Guo and Chang's switch is the most stable among the three switches.

4. Conclusion and future research directions

In this paper, various multicast ATM switches based on space-division and designed from 1984 to 1997 are examined. Performance evaluations are conducted to test some of the switches. The results seem to indicate that the concept of scheduling before copying is not a bad idea. But, all of the switches have yet to be tested in a real multicasting service environment. The multicast ATM switches based on time-division, especially the architecture built on shared memory are not discussed in this paper. To make this survey more complete, they will be included in the future. Moreover, the comparing of multicast ATM switches based on space-division and time-division is another important issue. This is also will be done in the future research.

Computer networks are changing at a daunting speed. The killer application WWW (World Wide Web) has made the network (Internet) interesting and easily accessible. This will make the deployment of next generation high speed networks become more necessary and urgent. The users can not wait (WWW = World Wide Wait). So designing new, efficient, and cheap high speed network elements will be crucial to the development of networks. From the above surveys, several research directions can be identified:

・Combine the concept of prescheduling with the best features of other switches to design a new multicast switch.

・The scheduling algorithms are not optimal. Is further improvement possible?

・Benchmarks for testing ATM switches in general (multicast switches in particular) have to be developed and standardized.

・Compatibility and interoperability problems between multicast ATM switches have to be addressed.

Acknowledgements

This research is supported in part by NSC under contract numbers NSC85-2622-E-007-009 and NSC85-2221-E-001-003.

Reference

[1]. R. Y. Awdeh and H. T. Mouftah, "Survey of ATM switch architectures," *Computer Networks and ISDN Systems*, vol. 27, pp. 1567-1613, 1995.

[2]. Jae W. Byun and T. T. Lee, "The Design and Analysis of an ATM Multicast Switch with Adaptive Traffic Controller," *IEEE/ACM Trans. Networking*, vol. 2, pp.288-298, Jun. 1994.

[3]. H. J. Chao, “A recursive modular Terabit/second ATM switch,” *IEEE J. Selected Areas Commun.*, vol. 9, no.8, pp. 1161-1172, Oct. 1991.

[4]. H. J. Chao and B. S. Choe, “Design and analysis of a large-scale multicast output buffered ATM switch,” *IEEE/ACM Trans. Networking*, vol. 3, pp. 126-138, Apr. 1995.

[5]. H. J. Chao, B. S. Choe, J. S. Park, and N. Uzun, “Design and implementation of Abacus switch: a scalable multicast ATM switch,” *IEEE J. Selected Areas Commun.*, vol. 15, no. 5, pp. 830-843, 1997.

[6]. D. X. Chen and J. W. Mark, "SCOQ: A fast Packet Switch with Shared Concentration and Output Queueing," *IEEE/ACM Trans. Networking*, vol. 1, pp. 142-151, 1993.

[7]. D. X. Chen and J. W. Mark, "Multicasting in the SCOQ Switch," *INFOCOM'94*, pp.290-297, 1994.

[8]. C. Clos, “A study of non-blocking switching networks,” *Bell Syst. Tech. J.*, vol. 32, 406-424, 1953.

[9]. R. Cusani and F. Sestini, "A Recursive Multistage Structure for Multicast ATM Switching," *INFOCOM'91*, pp.1289-1295, 1991.

[10].K. Y. Eng, M. G. Hluchyj and Y. S. Yeh, "Multicast and Broadcast Services in a Knockout Packet Switch," *INFOCOM'88*, pp.29-34, 1988.

[11].J. N. Giacopelli, J. J. Hickey, W. S. Marcus, W. D. Sincoskie, and M. Littlewood, "Sunshine: A High-Performance Self-Routing Broadband Packet Switch Architecture," *IEEE, J. Select. Areas Commun.*, vol. 9, , Oct., 1991.

[12].M. H. Guo and R. S. Chang, "Multicast ATM switches based on input cells scheduling," *APCC*, pp.1629-1632, 1997.

[13].A. Huang and S. Knauer, "Starlite: A wideband digital switch," *Proc. IEEE GLOBECOM'84*, pp. 121-125.

[14].H. S. Kim, "Design and Performance of Multinet Switch: A Multistage ATM Switch Architecture with Partially Shared Buffers," *IEEE/ACM Trans. Networking*, vol. 2, pp.571-580, Dec., 1994.
[15].K. L. E. Law and A. Leon-Garcia, “A large scalable ATM multicast switch,” *IEEE J. Selected Areas Commun.*, vol. 15, no. 5, pp. 844-854, 1997.
[16].T. T. Lee, "Nonblocking Copy Networks for Multicast Packet Switching," *IEEE J. Select. Areas Commun.*, vol. 6, pp.1445-1467, Dec, 1988
[17].D. J. Marchok, C. E. Rohrs, R. M. Schafer, "Multicasting in a Growable Packet (ATM) Switch," *INFOCOM'91*, pp.850-858, 1991.
[18].B. Prabhakar, N. McKeown, and R. Ahuja, “Multicast scheduling for input-queued switches,” *IEEE J. Selected Areas Commun.*, vol. 15, no. 5, pp. 855-866, 1997.
[19].P. U. Tagle and N. K. Sharma, “Multicast packet switch based on dilated network,” *IEEE GLOBECOM*, pp.849-853, 1996.
[20].J. S. Turner, "Design of a Broadcast Packet Switching Network," *IEEE Trans. Commun.*, vol. 36, pp734-743, Jun., 1988.
[21].R. J. F. Vries, "ATM Multicast Connections Using the GAUSS Switch," *GLOBECOM'90*, pp.211-217, 1990.
[22].Y. Yang and G. M. Masson, "Broadcast Ring Sandwich Networks," *IEEE Trans. Computers*, vol. 44, pp.1169-1180, Oct., 1995.
[23].Y. S. Yeh, M. G. Hluchyj, and A. S. Acampora, "The Knockout switch: A simple, modular architecture for high-performance packet switching," *IEEE, J. Select. Areas Commun.*, vol. SAC-5, pp.1274-1283, Oct., 1987.
[24].W. D. Zhong and K. Yukimatsu, "Design requirements and architectures for multicast ATM switching," *IEICE Trans. Commun.*, vol. E77-B,Nov., 1994.