Technical Report

UCAM-CL-TR-142
ISSN 1476-2986

Number 142

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

A fast packet switch for the integrated services backbone network

Peter Newman

July 1988

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

https://www.cl.cam.ac.uk/

© 1988 Peter Newman

Technical reports published by the University of Cambridge
Computer Laboratory are freely available via the Internet:

https://www.cl.cam.ac.uk/techreports/

ISSN 1476-2986

version 2.2
July 1988

A Fast Packet Switch for the Integrated Services Backbone Network[1]

Peter Newman

The Computer Laboratory, University of Cambridge

ABSTRACT

With the projected growth in demand for bandwidth and telecommunications services, will come the requirement for a multi-service backbone network of far greater efficiency, capacity, and flexibility than the ISDN is able to satisfy. This class of network has been termed the Broadband ISDN, and the design of the switching nodes of such a network is the subject of much current research. This paper investigates one possible solution. The design and performance, for multi-service traffic, is presented of a fast packet switch based upon a non-buffered, multi-stage interconnection network. It is shown that for an implementation in current CMOS technology, operating at 50 MHz, switches with a total traffic capacity of up to 150 Gbit/sec may be constructed. Furthermore, if the reserved service traffic load is limited on each input port to a maximum of 80% of switch port saturation, then a maximum delay across the switch of the order of 100 μsecs may be guaranteed, for 99% of the reserved service traffic, regardless of the unreserved service traffic load.

1. Introduction

The growing acceptance of the Integrated Services Digital Network (ISDN) promises increased bandwidth and new telecommunications services at reasonable cost [1]. User demand for bandwidth and services is forecast to rise rapidly following the widespread adoption of ISDN access standards. In addition, when research into multi-service metropolitan area networks (MANs) [2], and private networks, reaches the market we may expect an acceleration in demand, possibly fuelled by an increasing availability of video services. To meet this demand for increased bandwidth and for an expanding diversity of services in the backbone network, evolution towards the Broadband ISDN, with the consequent requirement for new switching techniques, will become increasingly desirable [3]. The enhancements offered by a suitable switching mechanism should include: increased flexibility, high traffic capacity, enhanced bandwidth efficiency for 'bursty' services, inherent rate adaption, and the service independent support of multi-service traffic.

A recent study of switching techniques appropriate to a multi-service backbone network [4] concludes in favour of fast packet switching, a statistical switching mechanism also known as asynchronous time division [5,6]. Two problems require attention before a statistical switching mechanism may be employed in a multi-service backbone network, whether public or private. First a fast packet switch of high maximum traffic capacity must be designed and implemented, in current technology, at an acceptable cost. Secondly, for multi-service

[1] To appear in the IEEE Journal on Selected Areas in Communications, December 1988.

operation, a mechanism is required to support real-time traffic across a fast packet switch at a level of service at least commensurate with that offered by the ISDN. Thus, for example, voice traffic requires a blocked calls lost service with a guaranteed maximum delay performance on each voice connection throughout the entire duration of a call.

This paper presents a simulation study of the multi-service traffic performance of a fast packet switch based upon a non-buffered, multi-stage interconnection network. The design of the switch is first discussed followed by a simulation of its throughput at saturation. Then follows an investigation of the performance of the switch for multi-service traffic, in which a simple hardware mechanism is employed to offer a guaranteed maximum delay performance for real-time services such as voice. The results suggest, for example, that fast packet switches may be realised with a total switch capacity of up to 150 Gbit/sec, constructed from identical switching elements, in current CMOS gate array technology operating at 50MHz. Furthermore, if the reserved service traffic load is limited on each input port to a maximum of 80% of switch port saturation, then a maximum switch delay may be guaranteed of the order of 100 μsecs, for 99% of all reserved service packets, while the switch is continuously loaded to saturation with multi-service traffic.

2. Design of a Fast Packet Switch

Fast packet switching is a connection oriented packet switching mechanism which achieves high throughput and low delay by reducing the processing required per packet to an absolute minimum [7,8,9,10] and then implementing it in hardware. Routing is performed at call set-up and a virtual circuit is allocated which is fixed for the duration of the call. All flow control and error recovery protocol functions are performed on an end-to-end basis. The packet length across any virtual circuit is constant and small, and the packet format is very simple: a packet header containing a priority field and a label, (to identify the virtual circuit,) of fixed length, eg. 16 bits, followed by the information component typically in the region 4 to 64 octets.

Two fundamental components are required to construct a fast packet switch: switching and buffering. This results in three possible classes of fast packet switch design: **Input Buffered,** in which the buffering precedes the switching using a non-buffered switch fabric [11,13]; **Output Buffered,** in which the buffering follows the switching also using a non-buffered switch fabric [14]; and the **Buffered Switch Fabric,** where buffering occurs internally within the switch fabric [6,15–18,33,34]. The decision to investigate the design of a fast packet switch based upon a non-buffered switch fabric was taken on the basis that a non-buffered switching element is much simpler to implement than is its buffered counterpart. This implies the possibility of implementation in gate array technology offering greater flexibility in the cost, performance, and other design parameters than that available from a dedicated VLSI solution. Furthermore, a simple design permits switching elements of greater degree to be fabricated leading to a reduction in the number of interconnections required to form a given size switch fabric compared to that of a buffered design. The long term goal of an all optical implementation of the switch fabric, or at least the switch fabric data paths, also motivated the selection of a non-buffered switch fabric.

Pure input buffering has a performance which is approximately 58% that of pure output buffering [19] all other factors being equal. However, pure output buffering requires an order of magnitude more hardware and switch fabric interconnections than does an input buffered solution [14]. The fast packet switch to be described is thus based upon a pure input buffered switch fabric, but to improve performance, to facilitate maintenance, and to accommodate real-time traffic, a two-plane structure has been adopted which permits a limited amount of output buffering to be implemented if desired. The design may be extended to more than two switch planes in parallel but results suggest that this is unlikely to be necessary unless extremes of performance or reliability are required.

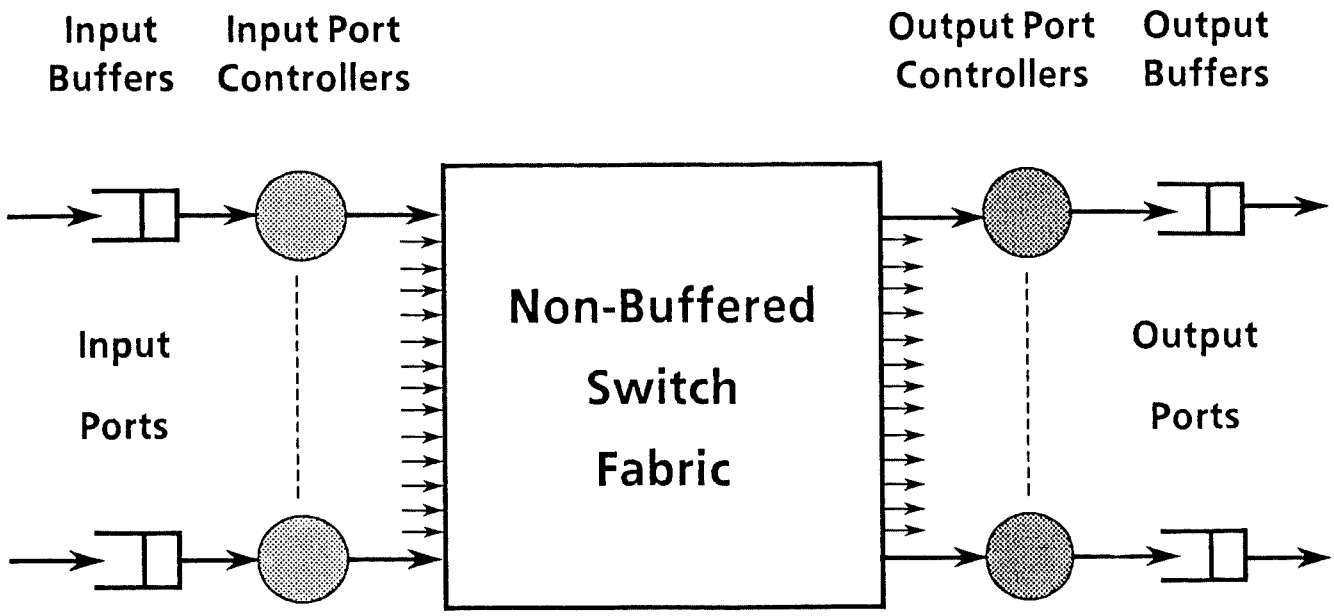

Fig. 1. Basic structure of the fast packet switch

2.1 The Switch

The basic structure of the fast packet switch is given in fig. 1. An incoming packet arrives in a first in first out (FIFO) queue. When free, the respective input port controller extracts the label from the packet at the head of the queue and uses it to reference a connection table. Each input port controller operates asynchronously, at the packet level, and independently of all other controllers. From the table it receives two components, an outgoing label and a tag. The outgoing label is used to replace the incoming label within the packet. The tag specifies the required destination output port of the switch and is attached to the front of the packet. The input port controller then initiates a set-up attempt by launching the packet into the switch fabric, tag first and in bit serial form. There are two possible outcomes, either the packet will be successful and reach the desired output buffer, or it will fail. A set-up attempt may fail either because it is blocked by other traffic within the switch fabric or because the requested output port is busy serving another packet. If the set-up attempt fails, the switch fabric will assert a collision signal which is returned to the input port controller, along a reverse path, typically within a few bit times of emission of the packet tag. On receiving the collision signal the input port controller removes the set-up attempt from the switch fabric and waits for a delay typically equivalent to 10% of the length of a packet. This is the retry delay and at the end of this period the input port controller begins a fresh attempt to transmit the packet. It continues to do so until it is successful or until it exceeds a limit designed to detect fault conditions.

A slightly more complex algorithm that offers an improvement in performance at high loads does not repeatedly attempt to transmit the same packet but on the failure of a set-up attempt searches through the input queue and attempts to transmit the second packet. If that attempt fails the third packet on the queue is attempted and so on cyclically through the queue until a successful transmission is achieved. This overcomes the so called 'head of line' blocking problem [13,19] but care has to be taken not to get packets on the same virtual circuit out of sequence. This algorithm will be referred to as input queue by-pass [15].

A simple model of the operation of the fast packet switch may be drawn by analogy with the operation of a well known local area network: Ethernet. Ethernet may be considered as a

fast packet switch which distributes the switching function across the local area using a single shared medium switch fabric. The fast packet switch described above merely confines the switching function within a box so that a multi-path medium of much higher bandwidth may be implemented. The input port controller of the fast packet switch corresponds to the media access controller of Ethernet and in both cases the controller throws a packet at the switch fabric and if it is unsuccessful the switch fabric informs it immediately. The difference between the two lies in the fact that in Ethernet a collision destroys both colliding packets therefore an exponential random back-off algorithm is required. In the fast packet switch, however, collisions are non-destructive in the sense that one of the colliding packets always survives, so a simple retransmission algorithm is sufficient.

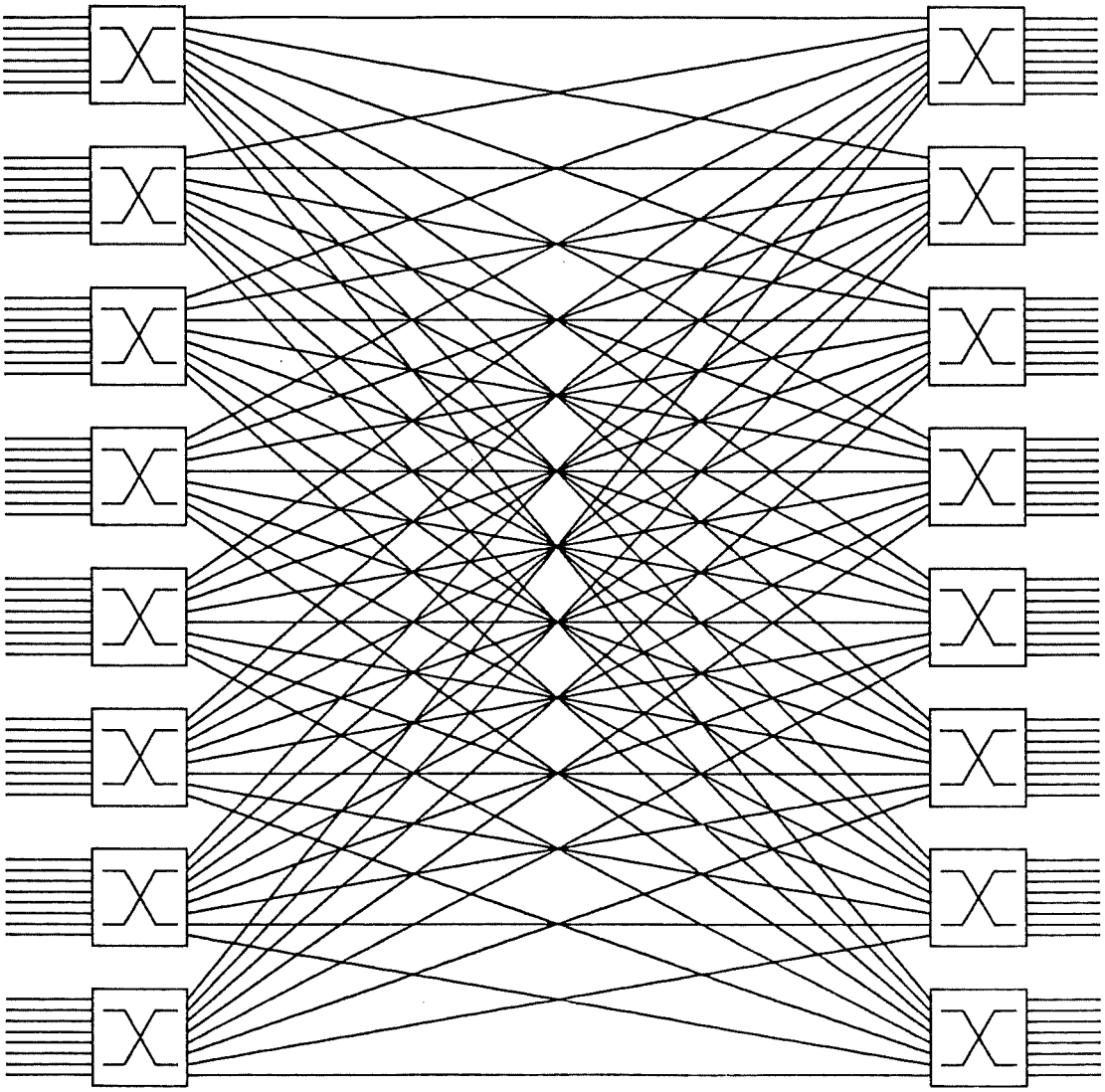

Fig. 2. A 64×64 delta network of 8×8 switching elements

2.2 The Switch Fabric

2.2.1 The Routing Fabric

A fast packet switch requires a highly parallel structure for the switch fabric both in the number of switching elements and in the number of interconnection paths between switching

elements. Also, control of the switch fabric must be distributed, with each active switching element operating independently of all others upon control information at the head of each packet. This suggests the use of a self-routing, multi-stage interconnection network. Such a network consists of many identical and independent switching elements, organised in stages, with the interconnection pattern of links between stages so arranged that each switching element may be controlled by the relevant digit from within a tag prefixed to the head of each packet. The tag simply contains the required destination port number of the switch. For switching elements of degree d each digit within the tag contains $\log_2(d)$ bits and the first digit controls the first stage of switches, the second digit controls the second stage and so on. Multi-stage interconnection networks that display this self-routing property belong to the class of banyan networks [20] and have been called delta networks [21], and although many examples of such networks are discussed within the literature [22], they have been proven topologically equivalent [23]. An example of a single plane 64×64 delta network constructed from switching elements of degree 8 is given in fig. 2. In general a delta network of size N requires $\log_d(N)$ stages with N/d switching elements per stage. Each interconnection link in the delta network consists of two paths, a forward path to carry the data and a reverse path, set up in parallel with the forward path, to carry the collision signal.

While the majority of research interest has been expended upon delta networks constructed from 2×2 switching elements, our previous investigations [24] suggested that it might be possible to implement non-blocking switching elements of up to degree 16, in gate array technology. The use of switching elements of degree greater than 2 raises the problem that delta networks are only defined in sizes that are an integer power of the degree of the switching element. This would result in large increments between valid sizes of network. The proposed solution is to replicate the interconnection links between stages which permits networks to be built to any size that is an integer power of 2, from switching elements of any degree that is also an integer power of 2, [25,26]. Thus a modified delta network of size N requires s stages, where $s = \lceil \log_d(N) \rceil$, of N/d switching elements per stage and each link of the pure delta network is replicated d^s/N times.[2] Strictly speaking the modified delta network is no longer a member of the class of banyan networks and fig. 3 illustrates a 16×16 modified delta network of switching elements of degree 8.

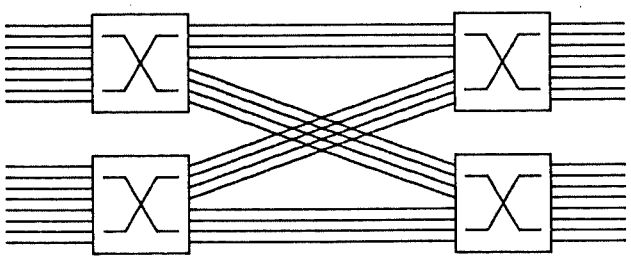

Fig. 3. A 16×16 modified delta network of 8×8 switching elements

We now have the possibility of multiple paths existing between the same pair of input and output ports. This increases the performance and fault tolerance of the switch but requires an algorithm to select between equivalent paths. Fortunately, as there is no buffering within the switch fabric, each incident packet may be routed independently without the risk of out-of-sequence errors between packets travelling on the same virtual circuit. Two algorithms have been investigated: searching and flooding. In the searching mechanism the input port controller attempts to transmit across each of the equivalent paths in sequence until it meets with success. In the flooding method the incoming packet is broadcast simultaneously over all paths that lead to the destination such that the destination selects one of the incident copies and all

[2] $\lceil x \rceil$ signifies the smallest integer equal to or greater than x.

others collide and are removed immediately.

2.2.2 The Distribution Fabric

The above switch fabric performs well for traffic which has a random destination distribution but its performance can be markedly impaired for incident traffic with a worst case distribution of destinations. For some applications this may not be significant, however, for high performance switches, and in order to handle traffic sources which have an average bandwidth in excess of about 10% of the switch port bandwidth, extra stages of switching must be introduced to distribute the incident traffic across the routing fabric. This has been termed the distribution fabric and to distribute the incoming traffic across an entire s stage delta network requires $s-1$ distribution stages and results in a Beneš topology [28]. Fig. 4 illustrates a 64×64 Beneš network of switching elements of degree 8.

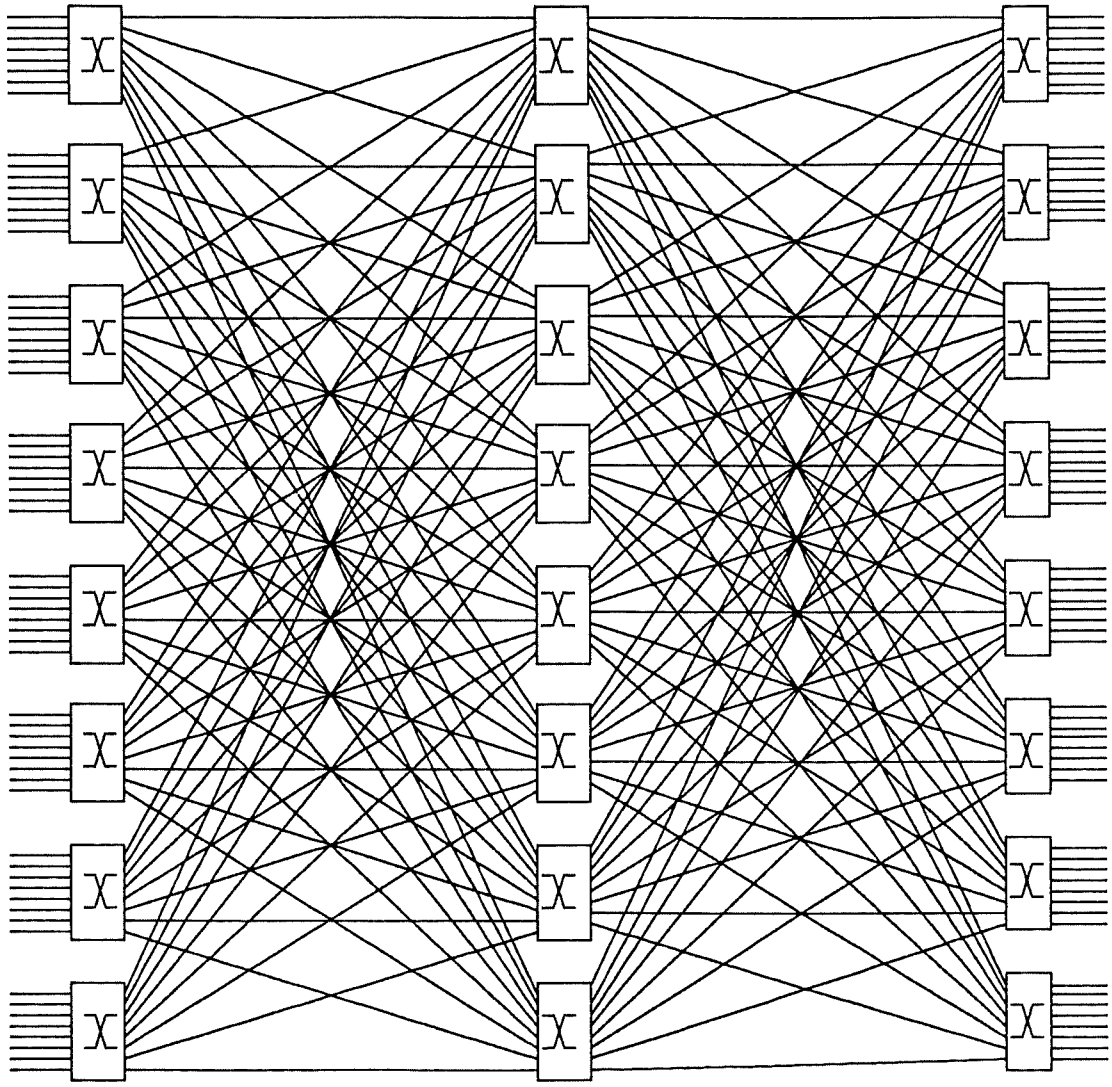

Fig. 4. A 64×64 Beneš network of 8×8 switching elements

Clearly we have now introduced a large number of equivalent paths into the switch fabric and again for each incident packet we are free to select any free path independently. The simplest method of achieving this is to implement the distribution stages of the switch fabric with

switching elements that select any free output at random.

2.3 The Two-Plane Switch Structure

It is common practice in the design of a telecommunications switch to duplicate or even replicate the switch fabric and control hardware for reliability and ease of maintenance. If this is achieved in a load sharing manner the performance of the switch is also enhanced. The general structure of a two-plane switch is shown in fig. 5 and may be extended to form a multi-plane switch of any arbitrary number of planes.

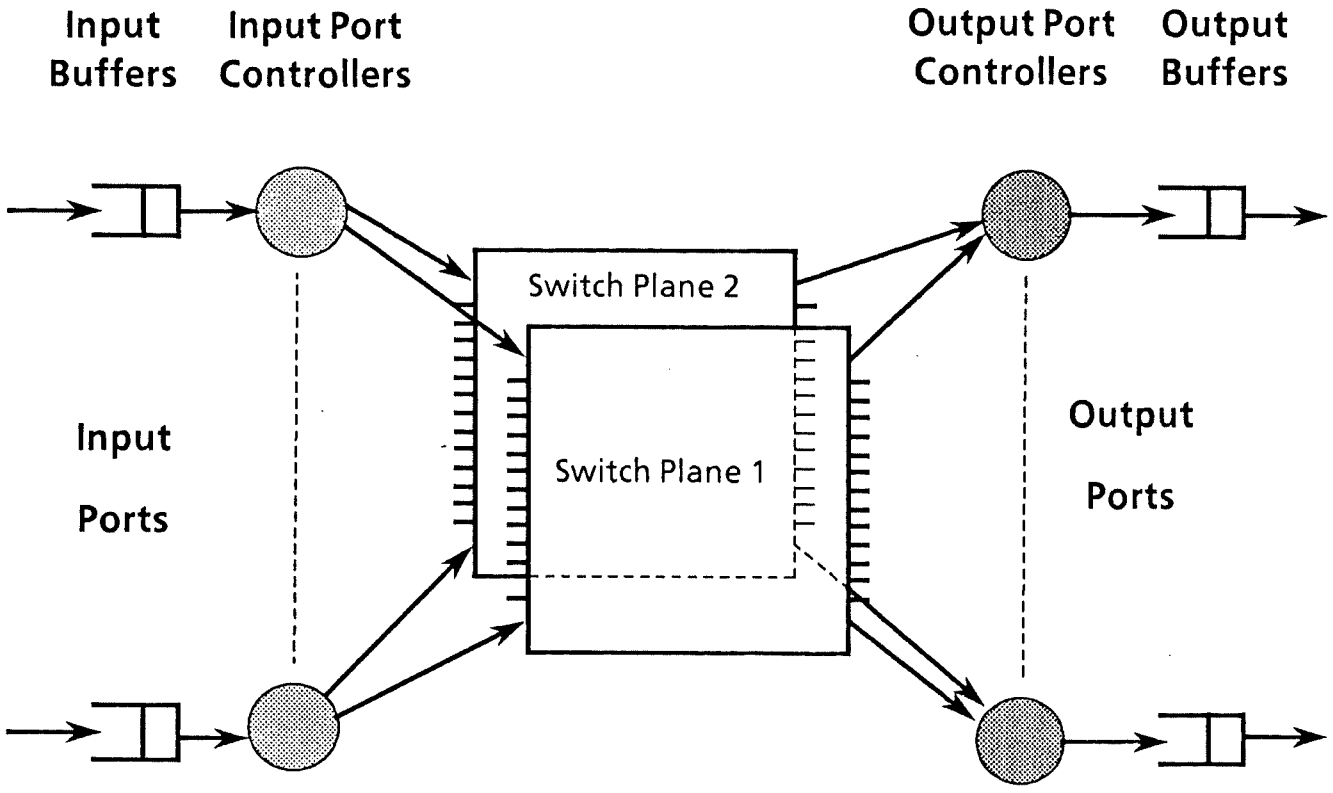

Fig. 5. A two-plane switching structure

It consists of two identical switch planes, each switch plane being a complete delta network with or without a distribution fabric. The two switch planes are connected in parallel to form a load sharing arrangement [26,27]. Once again we are introducing multiple paths and at the input port controller we may use either the searching or the flooding algorithm to select a path. Considering the ouput port controller: a simple implementation is only able to handle a single packet at a time and thus rejects set-up attempts arriving across the free plane while it is busy serving a packet. A more complex output port controller is capable of handling two packets arriving at the same time and buffering them in a first in first out manner in the output buffer. Thus a measure of output buffering may be provided at the cost of a more complex output port controller.

3. A Simulation Study of Switch Performance at Saturation

The above design of fast packet switch features a number of design parameters the effect of which, on switch performance, needs to be investigated. The simplest way to quantify the performance of a particular switch implementation is to specify the normalised average

throughput of the switch when saturated with traffic with a uniform random destination distribution. A simulation model has thus been developed to investigate the throughput at saturation of the switch with respect to the design parameters summarised in table 1.

Table 1: Switch fabric design parameters

Parameter	Range
Switch Fabric Size	2x2 to 4096x4096
Interconnection Networks	Crossbar Delta Benes
Degree of Switching Element	2x2 to 16x16
Multiple Path Algorithms	Searching Flooding Random
Multiple Switch Planes	1 to 4
Port Controllers	Regular Input Queue By-Pass Double Buffered Output De-Luxe

3.1 The Simulation Model

In order to reduce the amount of computer time required by the simulation model to reasonable proportions, the set-up of a packet has been modelled as an instantaneous event. In reality a packet will set-up on a stage by stage basis, thus a packet which fails set-up could itself cause blocking during its set-up attempt. The effect of this simplification is to over-estimate the throughput at saturation and the results of a more detailed simulation model show that the error introduced by this assumption is in general no more than about 2%.

In the model used to determine the throughput at saturation of the switch fabric each packet source supplies a new packet immediately upon completion of transmission of the previous packet and all output ports act as a perfect sink. Packet destinations follow a uniform random distribution and all packets are of the same length. No limit is placed upon the number of set-up attempts allowed. The simulation was initialised with random time relationships between all packets and run to attain stability before measurements commenced. Simulations were run for a total of 200,000 packets minimum which yielded results with a standard deviation of about 0.8% of the mean for the smaller network sizes to about 0.2% for the larger networks. The results are normalised and presented as the throughput per port at saturation which represents the average utilisation of an output port at saturation. The total traffic capacity of a fast packet switch is thus the product of the normalised throughput per port at saturation, the size of the switch, and the system clock.

3.2 The Crossbar Switch Fabric

First we consider the operation of the crossbar switch fabric as it gives the ideal performance for a non-buffered switch against which other interconnection networks may be compared. In the crossbar switch blocking proceeds solely from the probability of multiple sources attempting to transmit to the same destination at the same time. The upper two curves of fig. 6 show the difference between the simulator output and the analysis [21] under the assumptions of synchronous operation and blocked packets discarded.

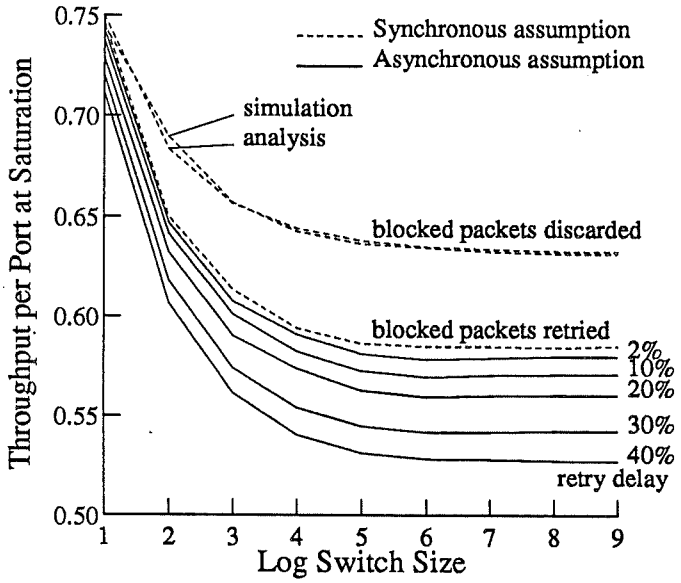

Fig. 6. Throughput at saturation for the crossbar switch

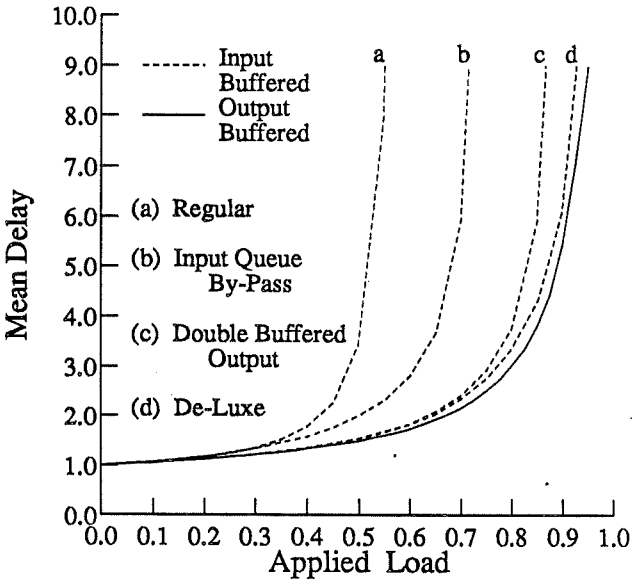

Fig. 7. Mean delay performance of crossbar switch fabrics for slotted traffic

(The switch size (N×N) is expressed as $\log_2(N)$ and the curves are discrete, points being connected purely for visual convenience.) The next curve shows the effect of re-submitting blocked packets under the assumption of synchronous switch operation and its asymptote agrees with the analytical results of [19]. This is followed by a set of curves assuming asynchronous arrival of packets, with asynchronous switch operation and blocked packets retried, at different values of retry delay, expressed as a percentage of the packet length, (ie. the emission delay of a packet).

Whilst discussing the performance of the crossbar switch fabric it is interesting to introduce a simulation study of the delay performance for slotted traffic which has been analysed in [19]. Fig. 7 shows how input queue by-pass and the use of a two-plane output buffered crossbar switch fabric improves the average delay performance of the pure input buffered switch. For the case of a two-plane crossbar switch fabric with output buffering and input queue by-pass, (the de-luxe model,) a performance very close to that of the pure output buffered switch may be achieved but at a much reduced cost in terms of hardware and interconnections within the switch fabric. The detailed results of the simulation model for the throughput at saturation of crossbar switch fabrics under the various design parameters are given in appendix I.

3.3 The Delta Network

Fig. 8 gives the maximum throughput performance of a single plane pure input buffered delta network constructed from switching elements of degree 2, 4, 8, and 16 using a flooding algorithm and a retry delay of 10% of the packet length. The corresponding curve for the

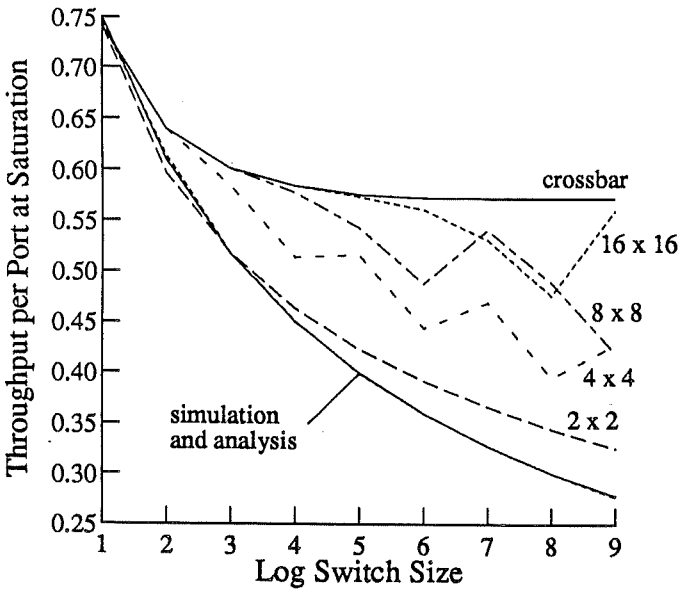

Fig. 8. Throughput at saturation for single plane, pure input buffered, flooding delta networks

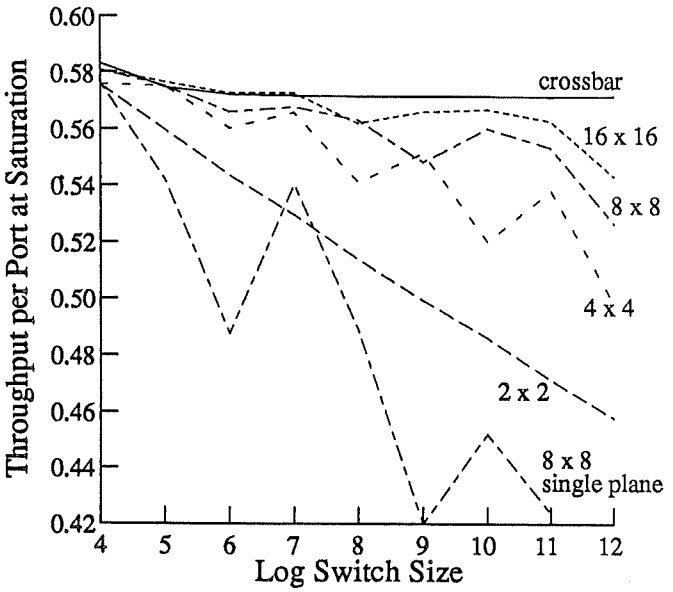

Fig. 9. Throughput at saturation for two-plane, pure input buffered delta networks

crossbar switch is included for comparison. The perturbations in the curves are due to the number of equivalent paths through the network with the minima indicating the pure delta network. Curves are also presented of the analysis [21] and simulation results for the 2×2 delta network, under the assumptions of synchronous operation and blocked packets discarded, demonstrating an agreement which renders the curves virtually co-incident. Comparison with the simulation results of [32] also reveals a close agreement. In fig. 9 the improvement in throughput obtained with a two-plane, pure input buffered delta network is shown using a routing algorithm which floods between planes but searches within a plane, commencing with a random selection from all equivalent paths within a plane. (The hardware for this hybrid mechanism is easier to implement, is more flexible, and its performance differs only marginally from that of the pure flooding case.) An investigation of multi-plane, pure input buffered delta networks with more than two planes shows that, in the case of switching elements of degree 8 and 16, little is gained in increased throughput as the asymptote of crossbar network performance is approached rapidly. Further, for a two-plane, pure input buffered network, the pure searching algorithm yields a performance that is only slightly inferior to that of a flooding mechanism, (no more than 2% with 8×8 switching elements). The detailed results of the simulation model for the throughput at saturation of delta networks with respect to the various design parameters are tabulated in appendix II.

3.4 The Distribution Fabric

The performance of the Beneš network as a switch fabric for a fast packet switch has been reported in [29] and for the purposes of this discussion we state the obvious that the introduction of a distribution stage into the switch fabric does not degrade its throughput performance, but rather, enhances it to approach the performance of the equivalent crossbar switch fabric. The results reported for the delta network routing fabric may thus be taken as a lower bound on performance when considering a switch fabric with distribution stages and the results for the crossbar switch fabric taken as an upper bound. Appendix III gives the throughput at saturation of the single plane pure Beneš network for comparison.

4. Multi-Service Integration over a Fast Packet Switch

From the results presented of switch performance at saturation it may be seen that switches of very high total traffic capacity may be constructed from LSI switching elements operating at conventional speeds. We now consider how to integrate multiple services, (voice, video, image, text, data, etc.) onto the structure.

4.1 Multi-Service Traffic Requirements

We argue that all communications services may be classified into two fundamental categories according to the delay requirement they present to the network, and for lack of better terminology we will refer to them as reserved and unreserved services. A reserved service exacts an inflexible, low delay and low variance of delay requirement, whereas unreserved services are much more flexible in the range of delay that can be tolerated. The majority of reserved services derive from information based upon a physical property that changes rapidly with time, eg. voice and video, and often contain a high degree of redundancy, thus permitting an appreciable packet loss before any noticeable deterioration in quality is perceived. There are some reserved services, however, that are highly sensitive to error, eg. process control, in which the delay constraint proceeds from the requirement for a high priority service, yet such services are generally of low bandwidth. Unreserved services include the bulk of data transfer, interactive and transaction services at various priorities.

The delay constraint is not the only difference between these two basic service classifications. A reserved service requires a guaranteed bandwidth and delay performance throughout the entire duration of the connection, else the connection request must be refused. An unreserved service expects the bandwidth and delay associated with a connection to vary according to the traffic load on the network. Hence, if a minimum of these two fundamental service priorities are implemented within the hardware of the switch, a diverse range of communications services may be supported [7,35].

4.2 Extensions to the Switch

In order to support the two basic services, reserved service traffic must be given priority at all input and output ports. At the input ports, the single input queue at every port of fig. 1 is replaced by two queues, one for reserved service packets and one for unreserved service packets. A priority field is also added to the tag to distinguish the two classes of packet. The input port controller is modified so as to transmit unreserved service packets only when the reserved service packet queue is empty, and to postpone repeated set-up attempts of an unsuccessful unreserved service packet on the arrival of a reserved service packet. The transmission of a successful unreserved service packet is not interrupted by the arrival of a reserved service packet. Reserved service priority at the output port is ensured by a simple mechanism implemented in hardware in each of the output port controllers [30]. If there is competition between packets from different input ports for access to an output port, this mechanism ensures that reserved service packets are given priority.

4.3 Simulation Traffic Models

Two models of unreserved service traffic have been used, saturation and Poisson. In the saturation model, unreserved service traffic is generated to keep each input port continuously busy while in the Poisson model, unreserved service packets are generated according to a Poisson arrival process. Both models generate traffic with a uniform random destination distribution. Three models of reserved service traffic were investigated: Poisson, talkspurt voice and TDM voice. In the Poisson model, reserved service packets are generated according to a Poisson arrival process with a uniform random distribution of packet destinations. In the talkspurt voice case, a superposition of individual voice sources has been modelled, on every input port of the switch, in which the on-off characteristics of speech have been used for bandwidth compression, (ie. packet voice with silence detection.) Each voice source is assumed to exhibit two states, active and silent, representing the talkspurts and pauses present

in conversational speech [36]. In the active state each voice source generates packets at a regular rate representing 32 Kbit/sec voice coding, 256 bit packets with a further 32 bits overhead, and a 20 MHz system clock. No packets are generated in the silent state. The two states are modelled by an exponential distribution with means of 1.2 and 1.8 seconds respectively [37], and each voice source transmits packets to a single destination which is selected at random during initialisation. The TDM voice model is simply a talkspurt model with silent periods of zero duration to represent packet voice without silence detection. A random phase relationship is assumed between all voice sources.

4.4 Multi-Service Switch Performance

The simulation results for a 64×64 fast packet switch constructed from 8×8 switching elements using a two-plane, pure input buffered delta network are now presented for various combinations of the multi-service traffic models. Investigations suggest that the major characteristics of the results are general to all sizes of fast packet switch constructed from switching elements of any degree according to any permutation of the design parameters discussed above. A good approximation to the throughput and delay performance for other sizes and designs of fast packet switch may be obtained by scaling the measurements presented for this example in proportion to the throughput at saturation of the desired switch fabric.

The measurement of delay selected for the performance of the reserved service is that of the 99^{th} percentile of the delay distribution [38]. It is assumed that packet voice traffic may withstand a 1% random packet loss, for small packet sizes [39,12], without perceptible loss of quality. Hence, our measure of guaranteed maximum delay is the delay within which 99% of all reserved service packets arrive at their destination. The consequence is that the accuracy of the maximum delay measurements is much lower than that of throughput as we are examining the tail of the delay distribution.

Delay is normalised to the packet length and all measurements are taken with a retry delay of 10% of the packet length. Applied load and throughput per port are also normalised and reflect the average utilisation of input and output ports respectively.

4.4.1 Poisson Reserved Service Traffic

Fig. 10 gives the basic result for a switch with a Poisson reserved service traffic source and a saturated unreserved service traffic source on each of the switch input ports. As the reserved service traffic load is increased, the maximum unreserved service traffic load that the switch is able to sustain falls, so as to maintain the total load on the switch reasonably constant at saturation. The reserved service throughput response, in the absence of any unreserved service traffic, is identical to that in the presence of unreserved service sources. Fig. 11 gives the corresponding maximum delay curves for reserved service traffic with and without the presence of saturated unreserved service traffic. The maximum delay for reserved service traffic in the presence of saturated unreserved service traffic is approximately 50% greater than in the absence of unreserved service traffic. This difference is due to the probability of an incident reserved service packet finding the input node already busy serving an unreserved service packet that has achieved set-up. Further, the throughput and maximum delay performance of reserved service traffic is not adversely affected by a non-uniform distribution of packet destinations for unreserved service traffic. Investigations also suggest that it is possible to operate a fast packet switch with input and output ports running at widely different mean traffic loads, as might be the case, for example, between ports connected to inter-switch trunks and those connected to local area networks.

In figs. 12 and 13 a Poisson reserved service traffic source is multiplexed with a Poisson unreserved service source at every input port of the switch. Fig. 12 shows the throughput performance of unreserved service traffic for several reserved service traffic loads. Fig. 13 shows the corresponding average delay for unreserved service traffic. Both curves saturate at a level that reflects the remaining switch bandwidth available after serving the requirements of

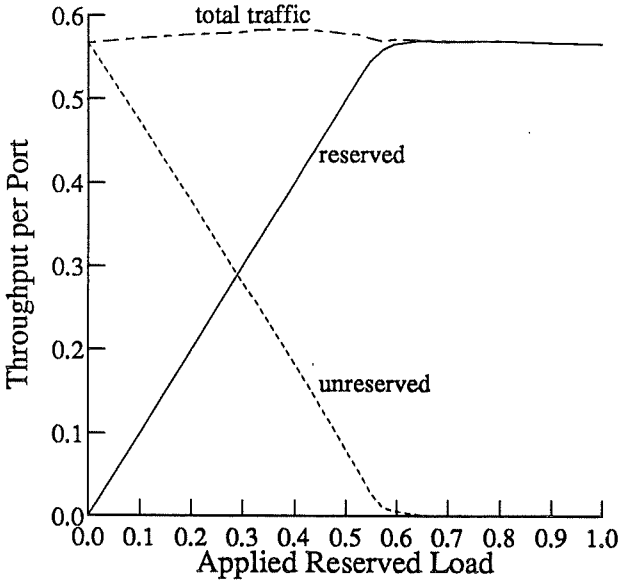

Fig. 10. Throughput performance for the Poisson reserved service + saturated unreserved service traffic model

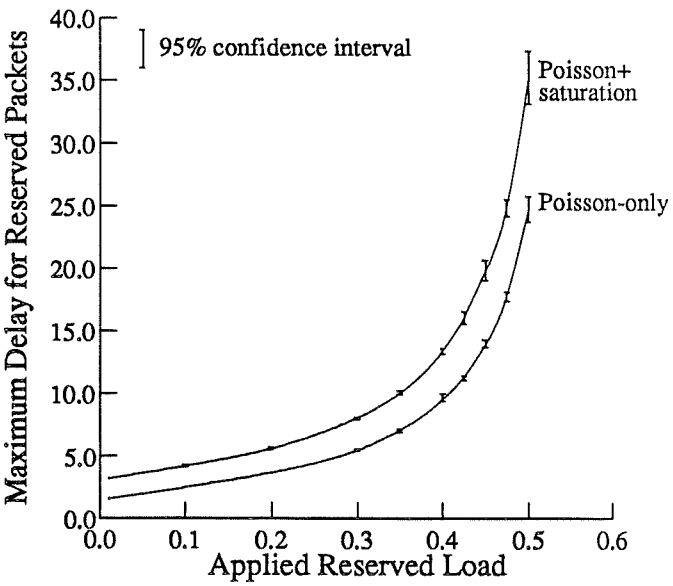

Fig. 11. Maximum reserved service packet delay for the Poisson reserved service traffic model with and without saturated unreserved service traffic

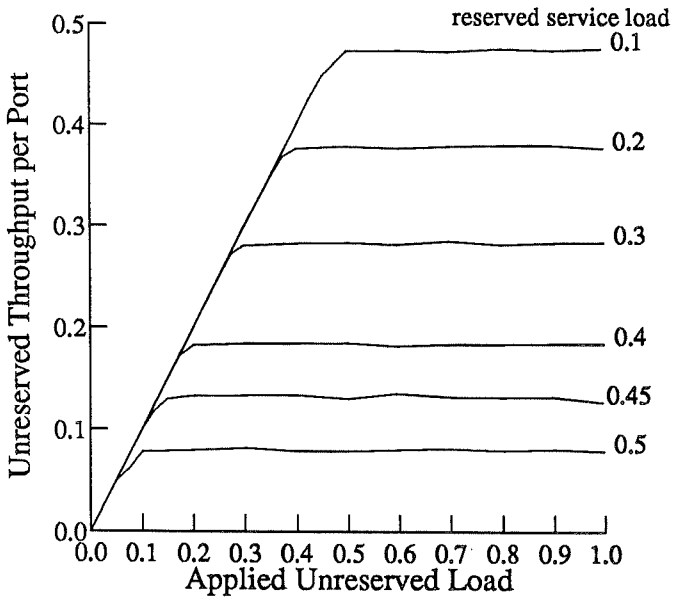

Fig. 12. Unreserved service throughput performance for the Poisson reserved service + Poisson unreserved service traffic model

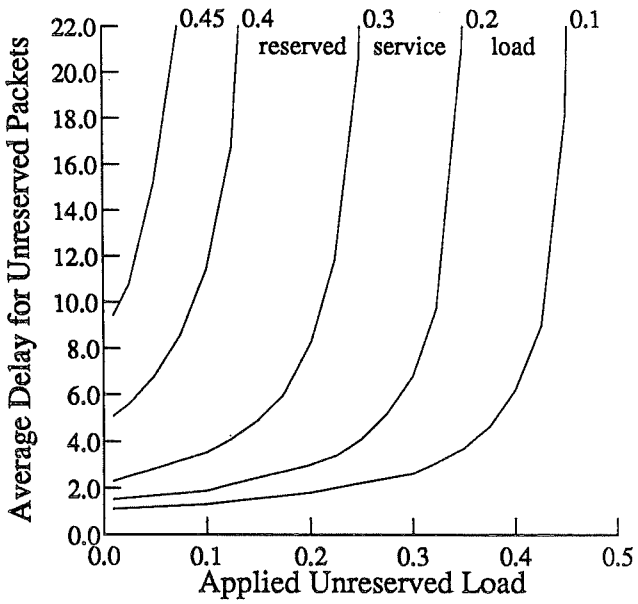

Fig. 13. Average unreserved service packet delay for the Poisson reserved service + Poisson unreserved service traffic model

reserved service traffic. The reserved service throughput characteristic in this case is identical to that observed with a saturated unreserved service traffic source while the maximum reserved service delay is reduced in proportion to the amount that the total load on the switch falls below saturation.

To give a comparative impression of switch performance fig. 14 shows the maximum delay performance of various designs of fast packet switch for Poisson traffic. Once again it may be seen that the performance of the pure output buffered switch [14] is only slightly greater than that of the highest performance two-plane delta design. This in turn is of slightly greater performance than a two-plane Batcher-Banyan, (ie. crossbar switch,) [11,13] as the latter is synchronous at the packet level and therefore cannot take advantage of input queue by-pass.

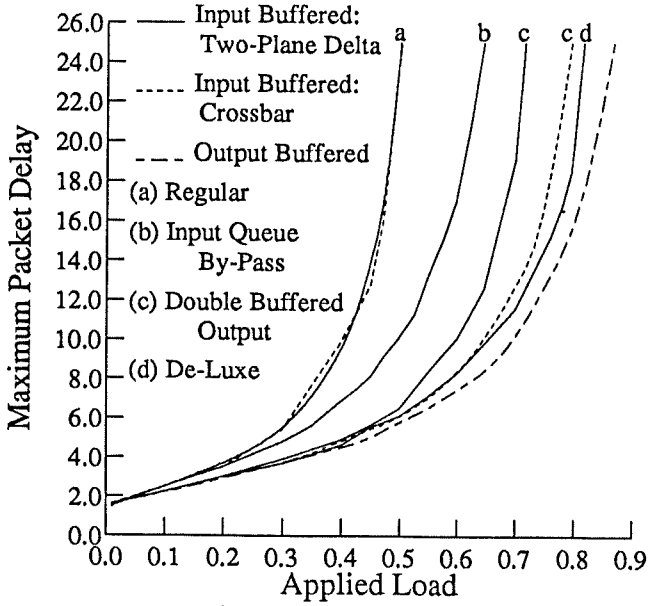

Fig. 14. Comparison of maximum delay performance of various switch designs for Poisson traffic

4.4.2 Talkspurt Voice

For the above 64×64 switch with Poisson traffic sources the queue lengths at the input ports were observed to be short and to stabilise rapidly for traffic loads below about 0.45. This figure represents a load of 80% of saturation and is a valid conservative estimate for the upper bound of the applied reserved service traffic load for stable operation of all sizes and designs of fast packet switch. The maximum mean reserved service traffic load for any switch port may therefore be fixed at 80% of the saturation load for that switch. The maximum delay performance of the talkspurt and TDM voice source models, to the above value of maximum mean reserved service traffic load, is now compared to the result for the Poisson reserved service traffic model.

The maximum delay performance, in the absence of unreserved service traffic, is given by fig. 15 and in the presence of saturated unreserved service traffic by fig. 16. (For the talkspurt voice model an applied load of 0.45 corresponds to 625 voice sources per switch port, and to 250 voice sources per switch port for the TDM voice model.) It is evident that within the region of stable operation there is no significant difference in the guaranteed maximum delay across the switch for Poisson, talkspurt and TDM voice sources, either in the presence or absence of saturated unreserved service traffic. Furthermore an observation of the inter-arrival times of packets generated by the talkspurt model on a single input port reveals a very close approximation to the exponential distribution [40]. Thus the superposition of a large number of talkspurt voice sources may be modelled by a Poisson arrival process, with reasonable accuracy, for applied loads below about 80% of saturation [41,42].

4.4.3 Packet Length

Finally, we consider the effect of variable unreserved service packet length upon performance. In the results presented so far we have assumed constant packet length and normalised all results to become independent of the absolute value. Now we assume that all packets consist of a header and an information component and we normalise results to the value of the information component. First we consider the case in which reserved service packets and unreserved service packets are of different but constant length. The length of the

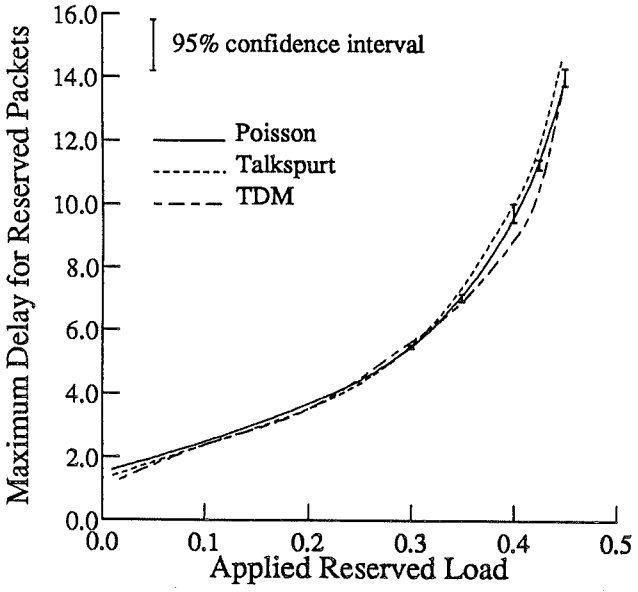

Fig. 15. A comparison of maximum reserved service packet delay for Poisson, talkspurt and TDM voice models in the absence of unreserved service traffic

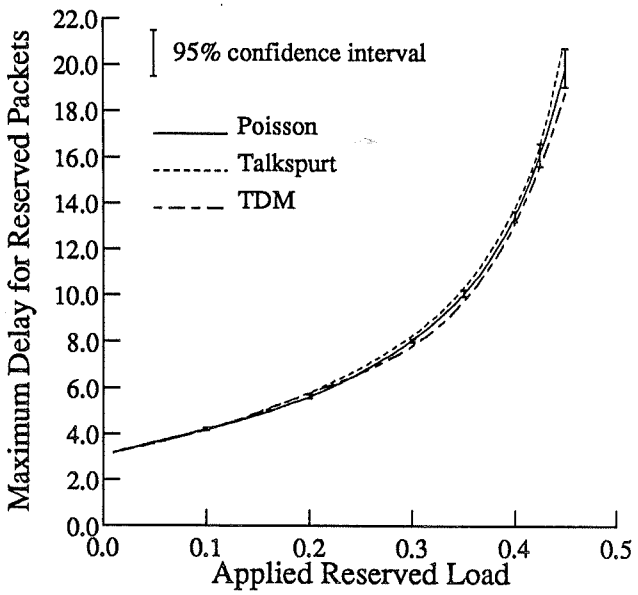

Fig. 16. A comparison of maximum reserved service packet delay for Poisson, talkspurt and TDM voice models in the presence of saturated unreserved service traffic

unreserved service packet is expressed in terms of the reserved service packet information field, and all packets have a header of one eighth of the length of the reserved service packet information field.

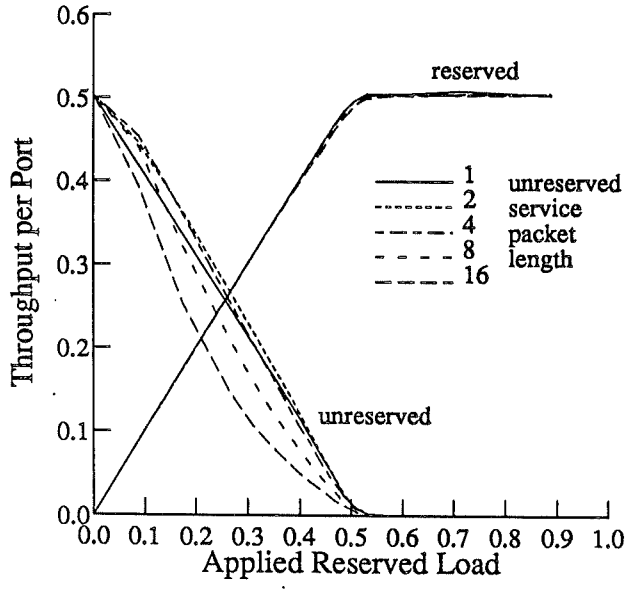

Fig. 17. Effect of unreserved service packet length on throughput performance for Poisson reserved service + saturated unreserved service traffic model

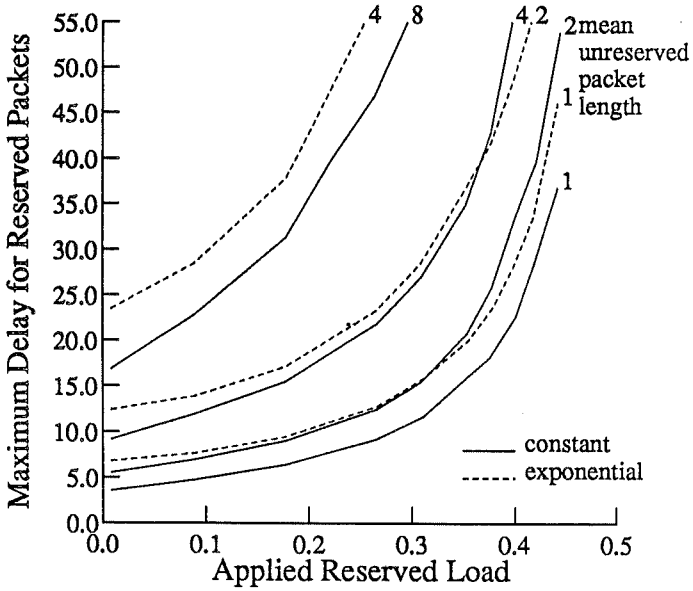

Fig. 18. Effect of unreserved service packet length, constant and exponentially distributed, on maximum reserved service packet delay.

The throughput results are presented in fig. 17 where it may be seen that the reserved service throughput performance is not unduly affected by the unreserved service packet length. However, the unreserved service throughput at saturation, for large unreserved service packet lengths, is lower than that for small packets showing that the advantage of low packet

overhead is rapidly outweighed by the superior multiplexing capability of small packet sizes. An examination of the case in which the length of the information component of all unreserved service packets is given by an exponential distribution reveals similar results with a reduction in unreserved service throughput performance of between 10% to 20%, due to the variability in packet length. An investigation of the case in which all packet lengths follow a uniform random distribution of ±10% about a mean value reveals no drop in performance when compared to that of constant length. Thus the switch is insensitive to the variation in packet length that might be introduced by a line code employing 'bit-stuffing'. The effect of the unreserved service packet length upon reserved service packet delay performance is given in fig. 18. As expected, a variable length unreserved service packet exerts a greater detrimental influence than one of constant length, and the shorter the mean packet length the less the reserved service packet delay performance is affected. Hence, conventional sizes of data packet must clearly be broken down into short packets for multiplexing with real-time traffic but this may not be necessary for a 'data-only' environment.

5. Implementation

An experimental implementation of the fast packet switch has been completed in low cost 3μ HCMOS gate arrays [30,31]. A 4×4 crossbar switching element and an experimental input port controller, with a standard 8-bit microprocessor bus interface, have been fabricated and demonstrated to operate as expected at a clock rate of 8 MHz. The throughput at saturation and delay performance of the switching element have been measured and agree with the simulation results to within 1%. The switching element required a total of 378 gates and the input port controller 292 gates which allows an estimate of the gate complexity of fully implemented parts to be made for crossbar switching elements of various sizes, table 2.

Table 2: Estimated complexity of crossbar switching elements

Size	Gate Count
2x2	250
4x4	600
8x8	1900
16x16	6000
32x32	21000

It is reasonable to expect an implementation in 2μ CMOS to achieve speeds of around 50 MHz without great difficulty and beyond this we observe that only the data path within the switching element is required to operate at high speed. The majority of the logic in the switching element handles packet set-up and if a small increase in overhead is permitted in the packet set-up time then this logic can operate at a slower speed than that within the data path. (In the current design the data path passes through no more than two gates and a flip-flop.) We may thus consider implementation in BiCMOS, ECL and even GaAs at speeds approaching 500 MHz and beyond without exceeding the power budget, table 3.

For even higher speed operation the switching and data paths within the switching element may be implemented optically with the control logic in ECL or GaAs to form an electro-optic switching element [43]. Switching times down to a few nanoseconds might thus become feasible on switch ports handling several gigabits/sec to yield a total switch capacity measured in terabits/sec.

Table 3: Estimated maximum bandwidth per switch port for various implementation technologies

Technology	Bandwidth Per Port
3 µ CMOS	10 Mbit/sec
2 µ CMOS	50 Mbit/sec
BiCMOS	250 Mbit/sec
ECL	500 Mbit/sec
GaAs	1 Gbit/sec
Photonic	>1 Gbit/sec

6. Conclusions

The design of a fast packet switch based on a non-buffered interconnection network has been reported and simulation results of its throughput performance at saturation discussed. The design is modular and will operate at any speed, with any device technology, including integrated optics. Maximum switch size is limited only by implementation considerations for the technology and operating speed selected. This design of fast packet switch uses fewer active elements than the equivalent crossbar switch, whilst offering a similar performance at saturation, for all sizes of switch greater than 16×16.

An extension to the design of the switch has been proposed in order to support multi-service traffic. Simulation results indicate that with a reserved service traffic loading of up to 80% of switch port saturation, the upper bound on delay for 99% of all incident reserved service packets is in the region of 20 packet lengths. Further, unreserved service traffic may be multiplexed with reserved service traffic, at every input port of the switch, so as to operate the switch continuously at saturation, without affecting the bounded delay performance of the reserved service. These results hold for voice traffic modelled as Poisson sources, talkspurt voice sources and TDM voice sources which yield a very similar maximum delay performance. The reserved service throughput and delay performance also appears insensitive to the arrival distribution and to the destination distribution of unreserved service traffic.

For delay sensitive, reserved service performance, the packet length for unreserved service traffic should be short and constant. No performance impairment is introduced by a ±10% variation in packet length. For a single service implementation, moderately insensitive to delay, variable length packets of any reasonable maximum length may be supported.

An experimental implementation of the fast packet switch in 3µ HCMOS gate arrays has demonstrated that the switch can be implemented at low cost in conventional gate array technology and that the performance of a 4×4 switching element agrees closely with that predicted by the simulation model.

Work is currently in progress on the problem of supporting multicast operation across the switch, for both reserved and unreserved traffic, with a similar throughput and delay performance to that of unicast traffic. Initial results suggest that this may be achieved with the same philosophy of simple implementation in gate array technology. The much more interesting problem of how to organise, manage, control and interface to a network of such fast packet switches is also under consideration.

Finally, by way of summary, we observe that the Cambridge Fast Packet Switch is but: "One small chip for MANs . . . . ."

REFERENCES

(1) S.N. Pandhi, "The universal data connection," IEEE Spectrum, July 1987, pp 31–37.

(2) R.W. Klessig, "Overview of metropolitan area networks," IEEE Communications Magazine, Vol 24, No 1, Jan 1986, pp 9–15.

(3) J.S. Turner, "Design of an Integrated Services *Packet* Network," IEEE JSAC, Vol SAC-4, No 8, Nov 1986, pp 1373–1380.

(4) J.J. Kulzer, W.A. Montgomery, "Statistical switching architectures for future services," ISS '84, Florence, May 1984, 43A1 pp1–6.

(5) M. Littlewood, I.D. Gallagher, J.L. Adams, "Evolution toward an ATD multi-service network," British Telecom Tech. Journal, Vol 5, No 2, April 1987, pp 52–62.

(6) A. Thomas, J.P. Coudreuse, M. Servel, "Asynchronous time-division techniques: an experimental packet network integrating video communication," ISS '84, Florence, May 1984, 32C2 pp 1–7.

(7) J.W. Forgie, A.G. Nemeth, "An efficient packetized voice/data network using statistical flow control," Proc. ICC '77, 38.2, pp 44–48.

(8) J.S. Turner, L.F. Wyatt, "A packet network architecture for integrated services," Globecom '83, Dec 1983, pp 45–50.

(9) A.G. Fraser, "DATAKIT – A modular network for synchronous and asynchronous traffic," ICC '79, pp 20.1.1–3, June 1979.

(10) P. Kirton, J. Ellershaw, M. Littlewood, "Fast packet switching for integrated network evolution," ISS '87, March 1987, pp B6.2.1–7.

(11) A. Huang, S. Knauer, "Starlite: A wideband digital switch," Globecom'84, 5.3.1-5, pp 121–125.

(12) J. G. Gruber, N. Le, "Performance requirements for integrated voice/data networks," IEEE J-SAC, SAC-1, No. 6, Dec 1983, pp 981–1005.

(13) J. Y. Hui, E. Arthurs, "A broadband packet switch for integrated transport," IEEE J-SAC, SAC-5, No. 8, Oct 1987, pp 1264–1273.

(14) Y. S. Yeh, M. G. Hluchyj, A. S. Acompora, "The Knockout Switch: A simple, modular architecture for high-performance packet switching," IEEE J-SAC, SAC-5, No. 8, Oct 1987, pp 1274–1283.

(15) R. G. Bubenik, J. S. Turner, "Performance of a broadcast packet switch," Proc. Int. Conf. Communications, ICC'87, June 1987, 31.6, pp 1118–1122.

(16) S. Nojima et. al, "Integrated services packet network using bus matrix switch," IEEE J-SAC, SAC-5, No. 8, Oct 1987, pp 1284–1292.

(17) M. De Prycker, M. De Somer, "Performance of an independent switching network with distributed control," IEEE J-SAC, SAC-5, No. 8, Oct 1987, pp 1293–1301.

(18) G. Perucca, "Research on advanced switching techniques for the evolution to ISDN and broadband ISDN," IEEE J-SAC, SAC-5, No. 8, Oct 1987, pp 1356–1364.

(19) M. J. Karol, M. G. Hluchyj, S. P. Morgan, "Input versus output queueing on a space-division packet switch," IEEE Trans. Communications, COM-35, No. 12, Dec 1987, pp 1347–1356.

(20) R. J. McMillen, "A survey of interconnection networks," Globecom '84, 5.1.1-9, pp 105–113.

(21) J.H. Patel, "Performance of processor to memory interconnections for multiprocessors," IEEE Trans. Computers, C-30, No 10, Oct 1981, pp 771–780.

(22) T. Feng, "A survey of interconnection networks," IEEE Computer, Vol 14, No 12, Dec 1981, pp 12–27.

(23) C. Wu, T. Feng, "On a class of multistage interconnection networks," IEEE Trans. Computers, C-29, No 8, Aug 1980, pp 694–702.

(24) P. Newman, "Message switching: an experimental model," The GEC Hirst Research Centre, April 1983, unpublished manuscript.

(25) G.B. Adams, M.J. Siegal, "The extra stage cube: a fault tolerant interconnection network for supersystems," IEEE Trans. Computers, C-31, No 5, May 1982, pp 443–454.

(26) M. Kumar, J.R. Jump, "Performance of unbuffered shuffle-exchange networks," IEEE Trans. Computers, C-35, No 6, June 1986, pp 573–578.

(27) C.P. Kruskal, M. Snir, "The performance of multi-stage interconnection networks for multiprocessors," IEEE Trans. Computers, C-32, No 12, Dec 1983, pp 1091–1098.

(28) V.E. Beneš, "On rearrangeable three-stage connecting networks," BSTJ, Vol 41, No 5, Sept 1962, pp 1481–1492.

(29) P. Newman, "A broad-band packet switch for multi-service communications," Proc IEEE Infocom '88, New Orleans, March 1988, 1A3 pp19–28.

(30) P. Newman, "Self-routing switching element for an asynchronous time switch," Priority Patent Application No. 8724208, 15 Oct 1987.

(31) P. Newman, "Data Signal Switching Systems," UK Patent GB 2 151 880 B, 16 Dec 1983.

(32) D.R. Milway, "Binary routing networks," The University of Cambridge Computer Laboratory, Technical Report No 101, Dec 1986.

(33) Y. Jenq, "Performance analysis of a packet switch based on single-buffered banyan network," IEEE J-SAC, SAC-1, No 6, Dec 1983, pp 1014–1021.

(34) D.M. Dias, J.R. Jump, "Analysis and simulation of buffered delta networks," IEEE Trans. Computers, C-30, No 4, April 1981, pp 273–282.

(35) R.M. Falconer, J.L. Adams, "Orwell: a protocol for an integrated services local network," British Telecom Tech. Journal, Vol 3, No 4, Oct 1985, pp 27–35.

(36) P.T. Brady, "A statistical analysis of on-off patterns in 16 conversations," BSTJ Vol 47, Jan 1968, pp 73–91.

(37) J.N. Daigle, J.D. Langford, "Models for analysis of packet voice communications systems," IEEE J-SAC, SAC-4, No 6, Sept 1986, pp 847–855.

(38) J.M. Appleton, M.M.Peterson, "Traffic analysis of a token ring PBX," IEEE Trans. Communications, COM-34, No 5, May 1986, pp 417–422.

(39) J. Gruber, L. Strawczynski, "Judging speech in dynamically managed voice systems," Telesis 1983 two, pp 30–34.

(40) B.G. Kim, "Characterisation of arrival statistics of multiplexed voice packets," IEEE J-SAC, SAC-1, No 6, Dec 1983, pp 1133–1139.

(41) K. Sriram, W. Whitt, "Characterising superposition arrival processes in packet multiplexers for voice and data," IEEE J-SAC, SAC-4, No. 6, Sept 1986, pp 833–846.

(42) H. Heffes, D. M. Lucantoni, "A marcov modulated characterisation of packetized voice and data traffic and related statistical multiplexer performance," IEEE J-SAC, SAC-4, No. 6, Sept 1986, pp 856–867.

(43) R.W. Blackmore, W.J. Stewart, I. Bennion, "An opto-electronic exchange for the future," ISS '84, Florence, May 1984, 41A4 pp 1–7.

Appendix I Throughput at saturation for crossbar switch fabrics

Size	Synchronous		Asynchronous (10% retry delay)			
	Regular	Double Buffer	Regular	Queue By-Pass	Double Buffer	De-Luxe
2	.747	1.0	.742	.878	1.0	1.0
4	.650	.941	.640	.808	.931	.981
8	.613	.916	.601	.771	.897	.971
16	.594	.905	.583	.752	.883	.965
32	.586	.900	.575	.746	.876	.961
64	.585	.898	.572	.740	.872	.958
128	.585	.896	.572	.740	.872	.957
256	.584	.897	.572	.734	.872	.954
512 +	.585	.897	.572	.739	.872	.951

Appendix II Throughput at saturation of delta networks with switching elements of degree 2 to 16

Degree 2

Size	Single Plane		Two-Plane			
	Regular	Queue By-Pass	Regular	Queue By-Pass	Double Buffer	De-Luxe
2	.742	.878	.742	.878	1.0	1.0
4	.597	.791	.648	.813	.930	.979
8	.517	.711	.599	.775	.850	.950
16	.462	.642	.576	.743	.786	.916
32	.423	.583	.560	.723	.733	.879
64	.392	.536	.543	.702	.688	.842
128	.366	.496	.529	.682	.650	.807
256	.344	.463	.514	.662	.618	.773
512	.325	.435	.499	.642	.589	.741

Degree 4

Size	Single Plane		Two-Plane			
	Regular	Queue By-Pass	Regular	Queue By-Pass	Double Buffer	De-Luxe
4	.640	.808	.640	.808	.931	.981
8	.583	.693	.605	.767	.887	.947
16	.513	.684	.576	.752	.836	.942
32	.516	.599	.575	.725	.818	.895
64	.444	.596	.560	.724	.755	.892
128	.468	.530	.566	.697	.762	.842
256	.395	.530	.541	.696	.690	.840
512	.428	.478	.551	.668	.714	.791

Degree 8

Size	Single Plane		Two-Plane			
	Regular	Queue By-Pass	Regular	Queue By-Pass	Double Buffer	De-Luxe
8	.601	.771	.601	.771	.897	.971
16	.576	.652	.581	.741	.861	.927
32	.542	.649	.575	.738	.848	.924
64	.488	.647	.566	.731	.799	.922
128	.540	.560	.568	.707	.806	.865
256	.489	.559	.563	.705	.783	.864
512	.420	.558	.548	.706	.718	.863
1024	.452	.493	.560	.674	.759	.804
2048	.424	.493	.553	.674	.726	.804
4096	.371	-	.526	.674	.652	.801

Degree 16

Size	Single Plane		Two-Plane			
	Regular	Queue By-Pass	Regular	Queue By-Pass	Double Buffer	De-Luxe
16	.583	.752	.583	.752	.883	.965
32	.573	.635	.577	.728	.849	.916
64	.560	.632	.573	.728	.842	.913
128	.531	.632	.573	.725	.829	.913
256	.476	.630	.562	.725	.786	.912
512	.559	.544	.566	.698	.797	.851
1024	.485	.543	.567	.695	.794	.847
2048	.459	.542	.563	.695	.769	.847
4096	.409	.541	.543	-	.703	.850

Appendix III Throughput at saturation of the single plane pure Beneš network

Degree	Size	Regular	Queue By-Pass
8	8	.601	.771
	64	.538	.648
	512	.496	.559
16	16	.583	.752
	256	.535	.632