Delay and Throughput Analysis of the High Speed Variable Length Self-Routing Packet Switch

Jingshown Wu, Hsien-Po Shiang, Kun-Tso Chen and Hen-Wai Tsao
Department of Electrical Engineering and Institute of Communication Engineering
National Taiwan University
Taipei, Taiwan 107

Abstract

In this paper, we analyze the performance of a high-speed variable length self-routing packet switch. Conventional crossbar switches need a powerful central control unit, complex matching algorithms, and speed up to have high throughput and low delay. Contrary, in this switch the routing function is performed by each switching element with an address correlator. In addition this switch employs multiplane structure and input port expansion scheme to alleviate head of line (HOL) blocking. We study delay, and throughput of this switch for various numbers of planes and expansion ratio under uniform traffic assumption. Results show that with reasonable number of planes and expansion ratio, the self-routing switch performs almost the same as output queue (OQ) switches, which have low input delay and 100% throughput. The simulation results agree with the analytical calculation very well.

I. Introduction

The Internet becomes very popular. The demand of network bandwidth increases rapidly. The optical transmission with dense wavelength division multiplexing technology may provide several tens terabits per second by a single fiber. It is doubtless the optical transmission can meet the demand increase. On the other hand, a single stage switch may have only a few hundred gigabits per second capacity currently. Recently, high speed high throughput switches had received a lot of attention. Many switching architectures such as Banyan, Knock-out, speed-up crossbar, etc. were proposed. OQ switches have 100% throughput from input ports to output ports. However, because of the limitation of memory bandwidth, it is difficult to implement a high speed high capacity OQ switch by today's very large scale integrated circuit (VLSI). For input queue (IQ) switches, if there are two or more packets in input ports with first-in-first-out (FIFO) queueing discipline, HOL blocking occurs. The maximum throughput of these switches is limited to about 58.6% under fixed length packet and uniform traffic condition. Many schemes such as window policy, channel grouping, virtual output queue (VOQ), speeding up switching fabric, etc. are proposed to improve the maximum throughput [1-8].

In VOQ switches, each input port maintains a separate queue for an each output port. The queues of each input-output pair can be constructed by either individual FIFO queues or shared buffer with certain queueing disciplines [1]. However, VOQ switches need very powerful and sophisticated scheduling algorithms to achieve high performance. The scheduling problem is equivalent to the matching problem of bipartite graph. It is proved that a VOQ with maximum matching algorithm can achieve 100% throughput. However, the complexity of maximum matching is very high and difficult to implement by today's VLSI technology to meet the high speed requirement. More practical arbitration mechanisms based on a maximal matching computation procedure, such as parallel iterative matching (PIM), iterative SLIP matching (iSLIP), dual round-robin matching (DRRM), etc. are proposed [3-5]. It is proved that a VOQ switch with the four-iteration PIM algorithm can have throughput more than 99% under uniform traffic condition [3].

Combined input/output queue (CIOQ) switches with speed up have been studied intensively. The maximum throughputs of these switches can reach to 88.5% and 97.6% for the speedup factor s = 2 and s = 3 under uniform traffic and infinite buffer assumption [8]. Combining VOQ and CIOQ schemes can reach 100% throughput [5-7].

Multiplane switches with arbiters are also proposed [9]. The results show that these switches with highly parallel structure can emulate high-speed OQ switches (e.g. 1.28 Tb/s) [6]. However, these switches that require VOQ and arbiters with DRR Matching algorithm are still very complex.

In this paper, we present a novel multiplane variable length self-routing switch, called NTU switch, which consists of preprocessors, sequencers, multiplexers, and multiplane switching fabric. The packets enter the switching fabric sequentially controlled by sequencers, and a busy port indicator line in each column of switching elements is used to solve the output port contention. Because packets enter the switching fabric sequentially, we do not need to perform centralized arbitration or matching. In addition, this switch can handle variable length packets directly without segmentation and assembly. In order to alleviate HOL blocking, which occurs in a single plane crossbar switch, we employ multiplane switching fabric and input

[1]Part of this work is supported by the National Science Council and the Ministry of Education Taiwan, R.O.C. under the Grants NSC89-2215-E-002-061 and 89-E-FA-06-2-4.

[2]The third author is supported by Silicon Integrated Systems Corp. fellowship.

expansion scheme. The routing function is performed by each switching element in the plane. The proposed switch is highly scalable. The performance including throughput, delay, and packet loss is studied. Results show that a reasonable-size NTU switch with 4 planes and 1.5 input expansion ratio will have 100% throughput and little input port delay like an OQ switch.

The remainder of this paper is organized as follows. Section II describes the architecture of the NTU switch. Section III analyzes the performance of the NTU switch with exponential packet lengths, Poisson packet arrivals, and under uniform traffic assumption. Section IV gives numerical results based on analytical calculation and simulation. The throughput, average packet delay time, average queue length, and loss probability of an $N \times N$ NTU switch with 4 planes and 1.5 expansion ratio are discussed in detail. Finally, Section V concludes our work.

II. Architecture of the multi-plane switch

An $N \times N$ NTU switch architecture is shown in Fig. 1, which consists of preprocessors, switching fabric planes, sequencers, FIFO queues on both input and output sides, and multiplexers.

Preprocessors, which have k inputs and n outputs ($n > k$), distribute packets from linecards (input queues) to appropriate sequencers. The expansion ratio r can then be defined as n/k. For an $N \times N$ switch, we have N/k preprocessors, which convert each byte into a parallel nine-bit word, in which the most significant bit is regarded as control signal. The most significant bit of the word for the header and the tailer is marked by "1", otherwise is "0". In order to have better performance, the n output lines of a preprocessor are connected to the sequencers of different switching planes. Besides spreading the input traffic, a preprocessor should send the contiguous packets that have identical output destination to the same sequencer to solve the reordering problem.

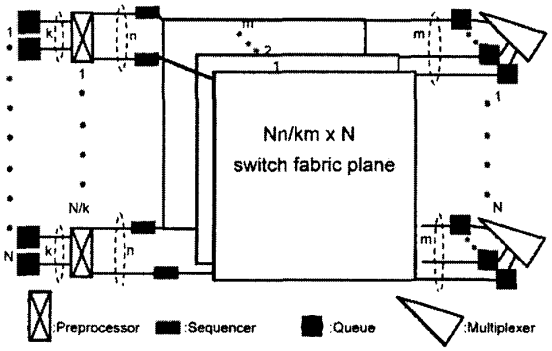

Fig. 1 The architecture of the NTU switch

Sequencers are located at each input of switching fabric planes. We have Nr sequencers in an $N \times N$ NTU switch. Assume that the number of switching planes is m; there are Nr/m sequencers connected to a single switching plane and these m planes identically consist of rN^2/m switching elements in a form of $rN/m \times N$ matrix, as shown in Fig. 2. In each switching fabric plane, sequencers control the packet entering time and ensure that only one packet is allowed to enter a plane at one time by maintaining different time points for packet entrance in a round robin fashion. If the entering packet is successfully transmitted to the output destination via a certain switching element, a feedback signal will be sent to inform the sequencer continuing the transmission; otherwise the sequencer will retransmit the packet at the next allowable entering time. The interval between two allowable time points, τ, must be at least the propagation delay of the busy indicator signal through the switching element column to avoid output contention. In general, τ is in the order of hundred picoseconds and is quite small compared with a packet transfer time, because practically the packet length distribution is from 40 to 1500 bytes as shown in Fig. 3 [10].

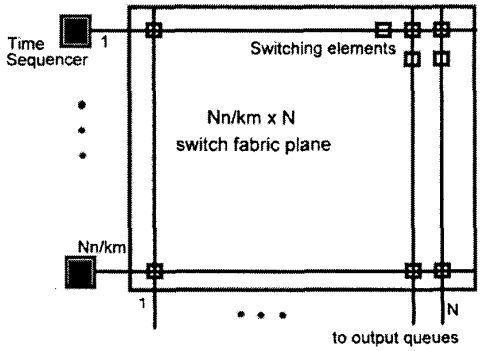

Fig. 2 the architecture of a single switching plane

The block diagram of a switching element is illustrated in Fig. 4. The output port address of a packet is checked while the packet being transmitted along the horizontal bus. If the two-word packet address matches the pre-stored address pattern and the output port busy indicator signal is idle, the correlator will send a match signal to change the state of the connector and route the packet to its destined output port until a tailer is detected. In the meantime, the output port busy indicator signal is set to busy to prevent the disturbance of other packets from different inputs.

The number of inputs for each switching plane is Nr/m, and the total switching elements of the NTU switch is rN^2. There are N output queues behind each switching fabric plane, and packets destined to the same output port may queue in the dual port output queues of different planes. A multiplexer selects these queues in an FIFO order and delivers the packets out. Obviously, there are limitations on expansion ratio r and plane number m. In this paper, we show that the throughput

and delay for various m and r.

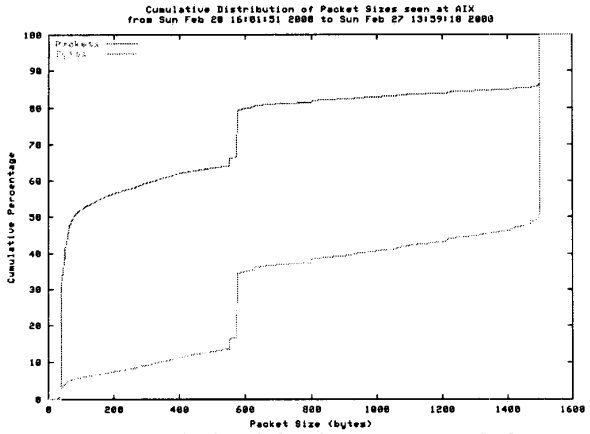

Fig. 3 The Packet length distribution on Internet.[10]

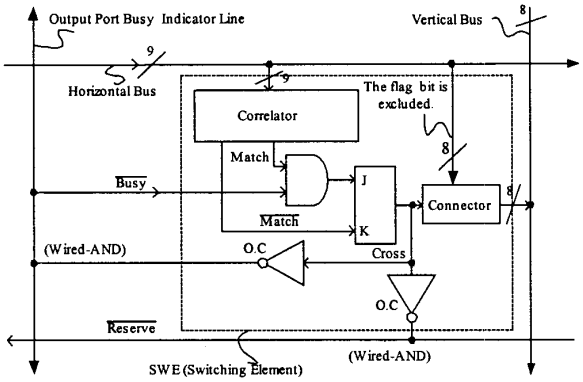

Fig. 4 The switching element.

III. The analysis

We analyze the NTU switch in following conditions: (1) Exponentially distributed packet length, (2) Poisson packet arrival, and (3) Uniform traffic. Consider an $N \times N$ architecture as shown in Fig. 1. For simplicity, we assume that (1): The number of input/output ports N is large enough. (2): The interval between two allowable time points, τ, for packet entering the plane is negligible.

Let the arrival rate of each input port be λ (packet per unit time) and the packet transmission time is exponential with mean S. The normalized input load, ρ, should be λS for each input port. The normalized throughput of each output port, ρ_o can be given by:

$$\rho_o = \begin{cases} \rho, & if\ \rho \le \rho_{o\max} \\ \rho_{o\max}, & if\ \rho > \rho_{o\max} \end{cases}, \qquad (1).$$

where $\rho_{o\max}$ is the normalized maximum throughput.

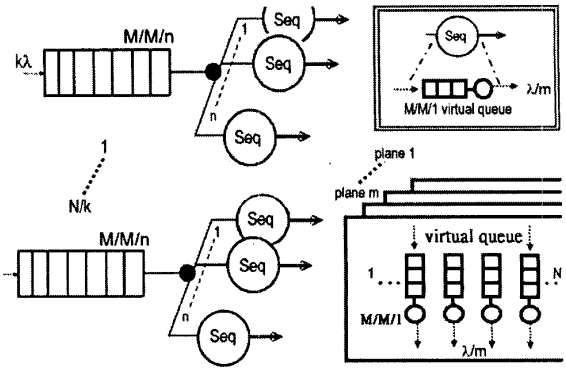

Fig. 5 The approximate model.

A. Queueing Model

Fig. 5 shows the approximate queueing model for input queues and switching planes. For a preprocessor, we group k input queues into a single queue with input rate $\lambda_{in} = k\lambda$. The n servers could represent different sequencers connected to certain planes. The output conflict will happen if more than one packet in the sequencers associated with the same crossbar switch plane have the same output destination. These contending packets will form a virtual queue corresponding to the desired output port, and the average service time of the front queue will be the average total sojourn time of a virtual queue. The output rate of a virtual queue is given by $\lambda_o = \lambda/m$, because totally there are m planes connected to an output port that should have output rate λ and the traffic to each plane is fairly distributed.

Since N is assumed to be large enough, the arrival process of a virtual queue can be modeled as a Poisson process [11]. In addition, the packet length distribution is exponential, thus, a virtual queue is equivalent to an M/M/1 queue. Since the next packet to be routed in a virtual queue is determined by the sequencers' time point, the queuing discipline is approximately random, but for simplicity, we approximate the virtual queue by a FIFO M/M/1 queue.

Note that the pdf of the sojourn time of an FIFO M/M/1 virtual queue is $(1/S - \lambda_o)e^{-(1/S-\lambda o)t}$ [12]. Thus, we can model the front queue as an M/M/n queue.

B. Maximum throughput

Assume X is the random variable representing the service time of the M/M/n queue. The average total sojourn time of M/M/1 is well known as [12]:

$$\overline{X} = \frac{S}{1-\lambda_o S} \qquad (2).$$

We can obtain the maximum throughput $\rho_{o\max}$ when the M/M/n queue saturates, that is: $\frac{\lambda_{in}}{n} = \frac{1}{\overline{X}}$. As

mentioned before, substituting $\lambda_{in} = k\lambda$ and $\lambda_o = \frac{\lambda}{m}$ in (1), we have:

$$mk\rho_{o\max} = nm - n\rho_{o\max} \Rightarrow \rho_{o\max} = \frac{nm}{mk+n} = \frac{mr}{m+r} \quad (3).$$

C. Average delay

Define the waiting time of the M/M/n queue as W_{in}, which is also the time that packets spent in the actual input queue, is given by [12]:

$$\overline{W_{in}} = \frac{P_Q \overline{X}}{(r-\lambda\overline{X})k} = \frac{P_Q mS}{(rm-r\rho-m\rho)k}, \quad (4).$$

where $P_Q = \frac{P_0(k\lambda\overline{X})^n}{n!(1-\lambda\overline{X}/r)}$, and

$$P_0 = \left[\sum_{i=1}^{n-1}\frac{(k\lambda\overline{X})^i}{i!} + \frac{(k\lambda\overline{X})^n}{n!(1-\lambda\overline{X}/r)}\right]^{-1}.$$

Note that $\rho = \lambda S = \rho_{o\max}$, $\overline{W_{in}} \to \infty$, and the average waiting time of a packet in an input queue is deeply related to the group number of input ports for a preprocessor, k. Under the same m and r, the larger group number we have, the shorter waiting time will be. Since we approximate the Random Selection Service by FIFO queueing discipline, the second moment of the service time of the M/M/n queue should be modified. Because in average there should be r/m sequencers contributing to a virtual queue, $\overline{W_{in}}$ is evaluated from the simulation results by multiplying a factor r/m. Thus,

$$\overline{W_{in}} = \frac{r}{m} \times \frac{P_Q \overline{X}}{(r-\lambda\overline{X})k} = \frac{P_Q rS}{(rm-r\rho-m\rho)k}.$$

Let Ws denote the packet delay time from its arrival to completion of transmission to the output queue. Ws consists of two components; the actual waiting time in the input queue, W_{in}, and the waiting time in the sequencer plus packet transmission time, X. We have:

$$\overline{W_s} = \overline{W_{in}} + \overline{X} \quad (6).$$

As for the output queues, we roughly regard the m output queues connected to the same output port as an equivalent M/M/1 queue. Let W_{out} be the delay of an output queue, thus, we have:

$$\overline{W_{out}} = \frac{S}{1-\rho_o}, \quad (7).$$

where ρ_o is the throughput of the NTU switch. Note that the average delay of output queues is bounded if throughput saturates.

Because a dual port memory can read and write a packet at the same time, the total delay, D, for an individual packet arriving at an input queue to its departure time from the destined output port is given by:

$$\overline{D} = \overline{W_{in}} + \overline{W_{out}} \quad (8).$$

D. Average queue length

Now that we have $\overline{W_{in}}$, from Little's formula, $\overline{L_{in}} = \lambda_{in} \times \overline{W_{in}}$, the average queue length of each M/M/n queue is given by:

$$\overline{L_{in}} = \frac{P_Q r\rho}{(rm-r\rho-m\rho)} = k\overline{L_{iq}} \quad (9)$$

,where $\overline{L_{iq}}$ is the average input queue length. Since output queues are M/M/1, we can compute the average queue length of each output queues as:

$$\overline{L_{oq}} = \frac{\rho_o}{m(1-\rho_o)} \quad (10)$$

There are m output queues for a single output port. Average output queue length is also bounded when the throughput saturates due to output line saturation

IV. Numerical results

In this section, some numerical results of the NTU switch architecture are presented. The simulation results are based on computer program with 95% confidence interval no more than 5%. The distribution of the packet transfer time is exponential with 10 unit time in average (S=10), while the interval between two adjacent allowable time points is 0.01 unit time.

Maximum throughputs for various expansion ratio and plane number are shown in Fig. 6. When r=1, the maximum throughput is well-known 50%, 67%, and 80% for one, two, and four switching planes, respectively. It shows that for plane number m ≥ 3, expansion ratio no more than 1.5 is sufficient to achieve 100% throughput, and the complexity of the switch planes only increase half of their original size (rN^2) for N less than 256.

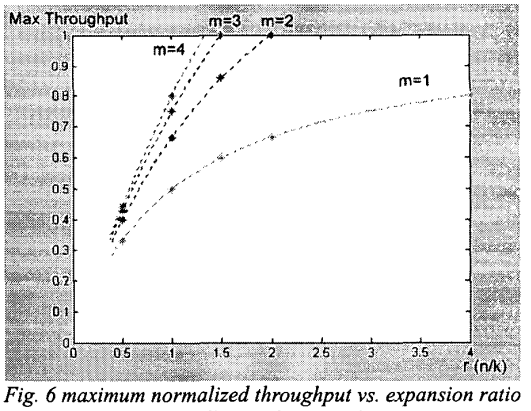

Fig. 6 maximum normalized throughput vs. expansion ratio with different plane number

The average input and output queue lengths are shown in Fig. 7 in the case of m=4, r = 1 and 1.5. When r=1.5, the input packets are quickly transferred from the input queues to the output queues. Fig. 8 shows average delay times, Ws and D. Note that for r = 1.5, most delay occurs in the output queue, even though all the input ports are nearly full loaded.

Although we assume that τ is small compared to the packet length, the performances degrade when the switch size grows. The selective input scheme that

bypasses the empty or busy sequencers improves the performance significantly for large N. The simulations using exponential packet distribution give fine results. The measured packet length distribution is illustrated in Fig. 3. Fig. 9 shows the simulation results of throughputs with nonnegligable τ using an approximate packet length distribution based on Fig. 3: the probabilities for 40 bytes, 60 bytes, 550 bytes and 1500 bytes are 0.35, 0.15, 0.2 and 0.15 respectively, and others are uniformly distributed.

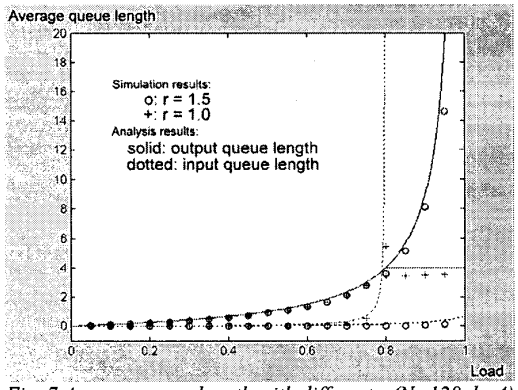

Fig. 7 Average queue length with different r (N=128, k=4)

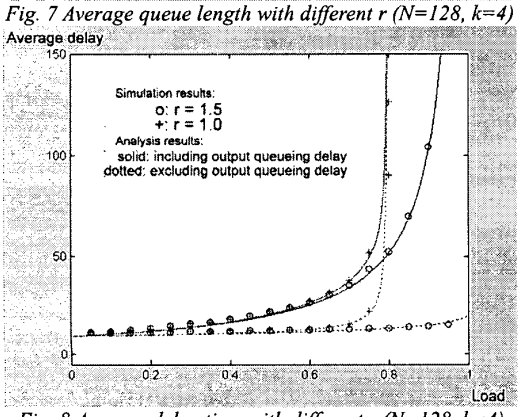

Fig. 8 Average delay time with different r (N=128, k=4)

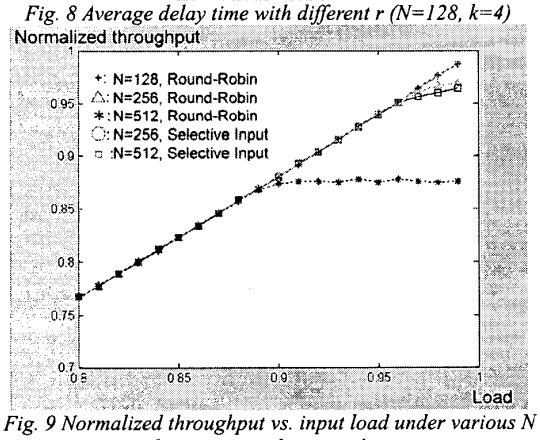

Fig. 9 Normalized throughput vs. input load under various N and sequencer selecting schemes (simulation results, m=4, r=1.5)

V. Conclusion

Most of current switching architectures can only handle fixed length packets. In this paper, we present the NTU switch, which employs multiplane architecture and input port expansion scheme. This switch is highly scalable and can take care variable length packets directly. We study the throughput and delay of this switch analytically. The simulation result is also provided. Results show that this switch is able to emulate the ideal OQ switch in terms of throughput and average queueing delay. Currently we are implementing a field programmable gate array 8 x 8 NTU prototype switch and designing VLSI circuits.

Because the proposed NTU switch has neither complex scheduling nor arbiter, we believe that this switch with OC-192 input line rate can be realized by today's VLSI technology.

References

[1].M.G. Hluchyj, and M. J. Karol, "Queueing in High-Performance Packet Switching," *IEEE J. Selected Areas Commun., Vol. 6, No. 9, pp. 1587-1597, Dec. 1988.*

[2].N. K. Sharma, "Comparison of windowing policies for input buffered packet switch," *IEEE, IPCCC 1997, pp. 245-251.*

[3].G. Nong, J. K. Muppala, M. Hamdi, "Analysis of nonblocking ATM switches with multiple input queues," *IEEE/ACM Trans. Networking, vol. 7, no. 1, Feb. 1999, pp. 60-74.*

[4].N. McKeown, "The iSLIP scheduling algorithm for input-queued switches," *IEEE/ACM Trans. Networking, vol. 7, no. 2, Apr. 1999, pp. 188-201.*

[5].H. J. Chao and J. S. Oark, "Centralized contention resolution schemes for a large-capacity optical ATM switch," *Proc. IEEE Wksp., Fairfax, VA, May. 1998.*

[6].H. J. Chao, "Saturn: A Terabit Packet Switch Using Dual Round-Robin," *IEEE Comm. Meg. Vol. 38, No. 12, pp 78-84, Dec, 2000.*

[7].S-T. Chuang, A. Goel, N. McKeown, and B. Prabhakar, "Matching output queueing with a combined input output queued switch," *IEEE JSAC, vol. 17, no. 6, June 1999, pp. 1030-1039.*

[8].A. K. Gupta and N. D. Georganas, "Analysis of a packet switch with input and output buffers and speed constraints" *in Proc. INFOCOM'91, Bal Harbour, FL, Apr. 1991, pp. 694-700.*

[9].C. Kolias and L. Kleinrock, "Performance Analysis of Multiplane, nonblocing ATM switches," *Tech. Rep., CS, UCLA, July 1998. http://mullennium .cs.ucla.edu/~ck/tr1.ps*

[10].Packet length distrivutions – CAIDA: ANALYSIS:AIX:plen_hist, Page URL: http://www.caida.org/analysis/AIX/plen_hist/index.xml.

[11].M. J. Karol, M. G. Hluchyj, and A. P. Morgan, "Input Versus Output Queueing on a Space-Division Packet Switch," *IEEE Trans. Commun., Vol. Com-35, No. 12, pp. 1347-1356, Dec. 1987.*

[12].Dimitri Bertsekas, Robert Gallager, "Data Networks," *2nd Edition.*