A High-Throughput Scheduling Algorithm for a Buffered Crossbar Switch Fabric

Tara Javidi
EECS Department
The University of Michigan
Ann Arbor, MI 48109-2122

Robert Magill and Terry Hrabik
Tellabs Research Center
3740 Edison Lakes Parkway
Mishawaka, IN 46545

Abstract- We examine high-throughput scheduling algorithms for buffered crossbar switch fabrics containing one buffer per crosspoint. We propose a scheduling system that uses longest queue first (LQF) scheduling for virtual output queues (VOQs) at the inputs and round-robin (RR) scheduling for the crosspoints. It is shown, through fluid model techniques, that this system achieves 100% throughput for input traffic that satisfies the strong law of large numbers and that produces a load $\leq 1/N$ for any input/output pair of an $N \times N$ switching fabric. Simulations indicate that 100% throughput may be attained for a much larger class of admissible loads.

I. INTRODUCTION

Input-queued (IQ) switches with crossbar switch fabrics provide a low cost packet switch architecture [1]. For an $N \times N$ switch fabric with no speedup, costs are controlled because, independent of N, the memory at the inputs need only operate at twice the line speed [1]. Packet data queued at the inputs is switched to the outputs on fixed time intervals, or timeslots. The amount of data switched to an output in one timeslot is called a cell. To prevent head-of-the-line (HOL) blocking, the inputs use virtual output queues (VOQs) to logically divide the input memory into separate queues for each output. In each timeslot, the scheduling operation chooses a matching of input and output pairs such that each input connects to at most one output and vice versa. The performance and complexity of the conflict-free matching algorithm determine the throughput and scalability of the IQ switch.

A maximum weighted matching (MWM) algorithm assigns a weight to each input/output pair and chooses a matching that maximizes the sum of the weights over all possible conflict-free matches. Through a fluid model analysis, Dai and Prabhakar have shown that MWM, where each input/output pair is weighted by its VOQ length, provides 100% throughput for admissible input traffic that obeys the strong law of large numbers [2]. This important result indicates that speedup is not necessary to achieve 100% throughput in an IQ switch for a wide variety of traffic sources.

Unfortunately, the complexity of a MWM algorithm is typically $O(N^3)$ [3] which prohibits scaling the fabric to large N. To increase scalability, an $O(N^2)$ maximal size matching algorithm such as SLIP is often used [1]. SLIP associates a separate Round-Robin (RR) scheduler with each input and output. The output RR schedulers independently choose an input for which the input/output VOQ is backlogged. The input RR schedulers then independently choose from among the output selections to complete the match. The SLIP algorithm yields 100% throughput for i.i.d. uniformly distributed Bernoulli arrivals; however, it achieves less than 100% throughput for other admissible arrival patterns even when the input/output pair loading is $\leq 1/N$ [1].

To achieve higher throughput with the SLIP scheduling structure, an $N \times N$ buffered crossbar switch fabric may be combined with VOQs at the inputs [4] as shown in Fig. 1. (Note that the crosspoint buffers allow multiple inputs to "match" to the same output simultaneously.) This combined input- and crosspoint-queued (CICQ) switch is feasible because, like the input buffers, the crosspoint buffers need only operate at twice the internal line rate. For small buffer sizes at each crosspoint, say 1 cell, the input ports must not overload the N^2 crosspoint buffers. By using a form of credit-based flow control [5], the input and output schedulers operate independently based on the status of the crosspoint buffer credits. Nabeshima uses Oldest Cell First (OCF) scheduling for each input and output scheduler [4]. This system is shown through simulations to produce very high throughput for i.i.d. uniformly distributed Bernoulli arrivals.

In this paper we extend the work of Nabeshima for CICQ switches that have one buffer per crosspoint. First, we simplify the scheduling process by using RR schedulers at the outputs. RR scheduling has been shown to be feasible for high-speed fabrics [6]. Second, we apply the fluid model techniques used for IQ switch fabrics to CICQ fabrics. We prove that a system with Longest Queue First (LQF) input schedulers and RR output schedulers is stable for any loading where the input/output pair loading is $\leq 1/N$ and the arrivals satisfy the strong law of large numbers. The fluid model for our switch and the stability proof are contained in section 2 and in the appendix. In section 3 we compare the throughput, delay and stability performance of various CICQ algorithms through simulations. Simulation results suggest that our scheduling algorithm for the CICQ fabric with one buffer per crosspoint may achieve 100% throughput for a much larger class of admissible loads.

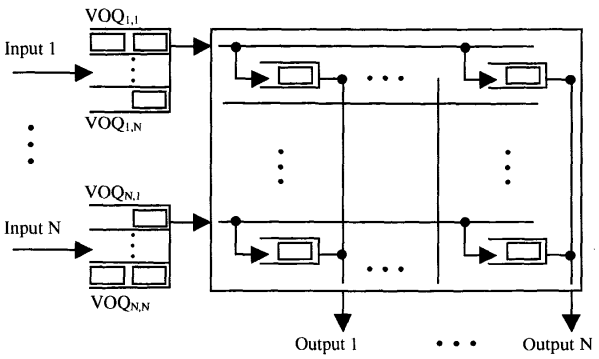

Fig. 1. A CICQ switch fabric

0-7803-7097-1/01/$10.00 ©2001 IEEE

II. THROUGHPUT RESULTS FOR A CICQ SWITCH

This section presents a throughput proof, through fluid model techniques, for the CICQ switch operated by input and output schedulers and regulated by credit based flow control.

A. *CICQ Switch Scheduling*

Consider an $N \times N$ buffered crossbar fabric with a one-cell buffer at each crosspoint. Assume that time is slotted and that packets arrive at the switch at the beginning of a time slot. For concreteness time slot n corresponds to the time interval $[n-1, n)$, $n = 1, 2, \ldots$. Each input buffer is partitioned into N virtual output queues (VOQs), each of infinite capacity. The virtual output queue VOQ_{ij} holds packets arriving at input i destined for output j. Note that a particular crosspoint (i, j) is associated with one VOQ_{ij}. The cells in VOQ_{ij} will be sent to crosspoint (i, j) in the fabric.

In each time slot the scheduling operation consists of two separate "phases": (1) output scheduling, and (2) input scheduling. A form of credit-based flow control [5] is used between the input and output schedulers. Each crosspoint buffer and corresponding VOQ has an associated credit which is used as a flag for the state of the crosspoint buffer (*1 = full, 0 = empty*). During the output scheduling phase, the output schedulers at each output j, select a non-empty crosspoint (i, j) whose credit state is 1 and set the credit state to 0. During the input scheduling phase, the input schedulers, at each input i, select a non-empty VOQ_{ij} whose credit state is 0 and set the credit state to 1. For the sake of analysis, we assume packet transmission from the crosspoint buffers and from the input VOQs occurs at the end of a time slot.

The rest of this paper follows the same notations used in Dai's paper. In summary, $Z_{ij}(n)$ represents the number of packets in VOQ_{ij} at the beginning of time slot n. $A_{ij}(n)$ denotes the cumulative number of packets that have arrived at VOQ_{ij} by time n, $D_{ij}(n)$ denotes the cumulative number of packets that have departed from VOQ_{ij} by time n. Note that $D_{ij}(n)$ represents departures from VOQ_{ij}. The departures from the fabric are at most one less than $D_{ij}(n)$.

Assume that the arrivals follow the strong law of large number; i.e., with probability one,

$$\lim_{n\to\infty} \frac{A_{ij}(n)}{n} = \lambda_{ij} \qquad i, j = 1, \ldots N, \tag{1}$$

where λ_{ij} is the arrival rate at VOQ_{ij}.

By definition, a pair of input and output service algorithms is called *rate stable* if with probability one,

$$\lim_{n\to\infty} \frac{D_{ij}(n)}{n} = \lambda_{ij} \qquad i, j = 1, \ldots N. \tag{2}$$

Note that a switch operating under rate stable service algorithm achieves 100% throughput [2].

Various input and output scheduling disciplines may be used within the outlined CICQ scheduling structure. Consider a LQF_RR scheduling algorithm where each input scheduling algorithm selects the longest VOQ whose credit state is zero, and where each output scheduling algorithm is a simple RR to service one of the crosspoints whose credit state is one. We wish to prove the following theorem.

Theorem 1: If $\lambda_{ij} \le 1/N \quad i, j = 1, \ldots N$, a CICQ switch using the LQF_RR algorithm is rate stable.

Before establishing the proof of theorem 1, we note the following corollary.

Corollary 1: A CICQ switch using the LQF_RR algorithm is rate stable under admissible uniform traffic.

Proof: Uniform traffic implies that the arrivals at any arbitrary input i is uniformly distributed over all outputs, i.e. $\lambda_{ij} = \lambda_{ik}$ $j, k = 1, \ldots, N$. For admissibility, $\sum_{j=1}^{N} \lambda_{ij} \le 1$, which implies that, for uniform traffic, $\lambda_{ij} \le 1/N \quad i, j = 1, \ldots N$. Using theorem 1, the proof of the corollary follows.

B. *Fluid Model*

In order to prove theorem 1, we need to establish the fluid model for the switch. The fluid model results from the general dynamics of the switch. The following equation of evolution should hold for any CICQ switch: for $n \ge 0$

$$Z_{ij}(n) = Z_{ij}(0) + A_{ij}(n) - D_{ij}(n) \qquad i, j = 1, \ldots, N. \tag{3}$$

$D_{ij}(n)$ is determined by the structure of the switch fabric, by the credit-based flow control and by the scheduling algorithm under which the switch is operated. Capturing these dynamics in a fluid model is difficult. Interestingly enough, our proof method does not require complete knowledge of $D_{ij}(n)$, it instead uses qualitative properties of the scheduling algorithm to establish the proof of theorem 1.

Now consider the fluid model equation of the switch:

$$\bar{Z}_{ij}(t) = \bar{Z}_{ij}(0) + \lambda_{ij} t - \bar{D}_{ij}(t) \ge 0, \tag{4}$$

where $\bar{D}_{ij}(t)$ is a function of the input scheduling algorithm and of the availability of the VOQ_{ij} credit at time t (which is dependent on the output scheduling algorithm).

The fluid model equation of a switch can be constructed through a limiting procedure from the difference equation describing the dynamics of the switch. We briefly address this construction in the appendix (See appendix A of [2] for more detail). The important relationship between rate stability of a scheduling algorithm for a switch and the properties of the fluid model is given by Fact 1.

Definition 1: The fluid model of a switch operating under a scheduling algorithm is said to be weakly stable if for every fluid model solution $(\bar{\mathbf{D}}, \bar{\mathbf{Z}})$ with $\bar{\mathbf{Z}}(0) = 0$, $\bar{\mathbf{Z}}(t) = 0$ for almost every $t \ge 0$.

Fact 1: A switch operated by a scheduling algorithm is rate stable if the corresponding fluid model is weakly stable.

Proof: See proof of Theorem 3 of [2].

So our goal now is to prove that for every fluid model solution $(\bar{\mathbf{D}}, \bar{\mathbf{Z}})$ of a CICQ switch using LQF_RR scheduling, $\bar{Z}_{ij}(t) = 0$ for almost every t if $\lambda_{ij} \le 1/N$.

C. Proof of Theorem 1

First, we establish the following fact.

Fact 2: Let f be a non-negative, absolutely continuous function defined on $\mathbb{R}^+ \cup \{0\}$ with $f(0) = 0$. Assume that for almost every t such that $f(t) > 0$, $\dot{f}(t) \le 0$. Then $f(t) = 0$ for almost every $t \ge 0$.

Note that $\mathbb{R}^+$ is the set of positive real numbers, and $\dot{f}(t)$ denotes the derivative of function f at time t. We adopt the convention that symbol $\dot{f}(t)$ implies that f is differentiable at t.

Proof: See proof of Lemma 1 of [2].

Now, for every $t \ge 0$ and every i, define F_i as

$$F_i(t) \triangleq \max_j \bar{Z}_{ij}(t). \tag{5}$$

To prove Theorem 1, we prove that under the LQF_RR algorithm, for almost every $t \ge 0$ and every i,

$$F_i(t) = 0. \tag{6}$$

If (6) is true, since $0 \le Z_{ij}(t) \le F_i(t)$ by definition, we can conclude that for almost every $t \ge 0$ and $\forall i, j$, $Z_{ij}(t) = 0$. In other words, equation (6) implies Theorem 1.

Lemma 1. For a CICQ switch operated by LQF_RR

$$\sum_{j \in \mathrm{K}} \dot{\bar{Z}}_{ij}(t) \le 0 \qquad \text{if } F_i(t) \ne 0 \text{ and } \lambda_{ij} \le \frac{1}{N}, \tag{7}$$

where $\mathrm{K} \triangleq \{ j : Z_{ij}(t) = F_i(t) \}$.

Proof: See appendix A.

Lemma 1 states that under the LQF_RR algorithm the total rate of change among the longest VOQs is negative.

Lemma 2. $\forall j \in \mathrm{K}$, we have $\dot{\bar{Z}}_{ij}(t) = \dot{F}_i(t)$.

Proof: See appendix B.

Lemma 2 implies that the rate of change for all virtual output queues of the longest queue depths is the same.

To prove (6) we first prove that $\forall i = 1, 2, \ldots, N$, F_i is a non-negative, absolutely continuous function. Notice that $\forall i, j \in \{1, 2, \ldots, N\}$, $\bar{Z}_{ij}$ is a non-negative Lipschitz continuous function; i.e., $\exists M_{ij} > 0$ such that $|Z_{ij}(t_1) - Z_{ij}(t_2)| \le M_{ij} |t_1 - t_2|$ for every $t_1, t_2 \ge 0$. (See Appendix A of [2].) We use this to show that F_i is Lipschitz continuous; i.e., $|F_i(t_1) - F_i(t_2)| \le M_i |t_1 - t_2|$, where $M_i \triangleq \max_{1 \le j \le N} M_{ij}$. Without loss of generality assume that $F_i(t_1) \ge F_i(t_2)$. We have

$$\begin{aligned} |F_i(t_1) - F_i(t_2)| &= F_i(t_1) - F_i(t_2) \\ &= Z_{ik_1}(t_1) - \max_{1 \le j \le N} Z_{ij}(t_2) \\ &\le Z_{ik_1}(t_1) - Z_{ik_1}(t_2) \\ &\le M_{ik_1} |t_1 - t_2| \\ &\le M_i |t_1 - t_2| \end{aligned} \tag{8}$$

in which $k_1 = \operatorname{argmax}_{1 \le j \le N} Z_{ij}(t_1)$. Thus, for any i the function F_i is Lipschitz, hence absolutely continuous. In addition, it is easy to check that F_i is also non-negative.

Now we can apply Fact 2 to function F_i, which implies that it is sufficient to show that for every $t \ge 0$ such that $F_i(t) \ne 0$, we have $\dot{F}_i(t) \le 0$. On the other hand, using Lemma 1, we know that if $F_i(t) \ne 0$ and a switch is operated by a LQF_RR algorithm,

$$\sum_{j \in \mathrm{K}} \dot{\bar{Z}}_{ij}(t) \le 0 \quad \text{for } \lambda_{ij} \le \frac{1}{N}. \tag{9}$$

By Lemma 2, we have

$$\sum_{j \in \mathrm{K}} \dot{\bar{Z}}_{ij}(t) = \sum_{j \in \mathrm{K}} \dot{F}_i(t) = m \dot{F}_i(t), \tag{10}$$

where $m = \|\mathrm{K}\| > 0$. Combining (9) and (10), we have $\dot{F}_i(t) \le 0$. Since $F_i(0) = \max_j \bar{Z}_{ij}(0) = 0$, from Fact 2 we conclude that for almost every $t \ge 0$, $F_i(t) = 0$ $\forall i$. Hence the proof of Theorem 1 is complete.

□

III. Simulation Results

This section presents simulation results illustrating the stability, delay and queue length performance of the LQF_RR algorithm. A two-state, Markov-Modulated Bernoulli Process (MMBP) produces the traffic sources for each input/output pair. The state transition probabilities in the two-state MMBP are chosen to produce an average rate and a "burstiness factor" (γ) for the source. When $\gamma = 1$, the two-state MMBP becomes a Bernoulli process. For $\gamma > 1$, the average burst size of the source is equal to γ times the average burst size of a Bernoulli process with the given average rate. For the simulations, arrivals occur at the beginning of the timeslot. Packet transmission follows packet scheduling by one time slot, and a minimum of one transmission slot delay is incurred in a buffered crossbar fabric.

Fig. 2 illustrates a 2x2 switch with unbalanced loading. We vary the average rates, $\lambda_{i,j}$ of the connections and measure the maximum queue size of each VOQ for 10 consecutive intervals of 10,000 cell slots. If the maximum value for a VOQ increases *every* interval, the switch is considered unstable. Fig. 3 illustrates the instability regions for four scheduling algorithms under Bernoulli arrivals. The LQF_RR, RR_RR and OCF_OCF algorithms use a CICQ fabric with one cell buffer per crosspoint regulated by credit-based flow control. The OCF_OCF [2] and RR_RR algorithms use

OCF and RR schedulers, respectively, at both the inputs and outputs. The SLIP algorithm is for an IQ switch. Fig. 3 illustrates that the SLIP algorithm produces an instability region that dips into the "box" indicating the region where $\lambda_{i,j} \leq 1/2$. The RR_RR algorithm produces instability for admissible loads, but it does not intersect the $\lambda_{i,j} \leq 1/2$ region. Neither the LQF_RR nor the OCF_OCF algorithms produced instability for any admissible loads.

Fig. 4 illustrates the average VOQ depths in a 16×16 switch for both uniform Bernoulli and uniform bursty traffic. The OQ algorithm is an output queued switch fabric with OCF (FIFO) scheduling at the output. (The VOQs exist at the output.) The figures show that the LQF_RR and OCF_OCF algorithms produce slightly larger average queue depths than the OQ switch. These results include cells buffered at the crosspoints. The LQF_RR algorithm produces slightly less average queue depths than the OCF_OCF algorithm for high loads where the advantage is more pronounced for the bursty traffic scenario. For loads greater than 50%, both the LQF_RR and the OCF_OCF algorithms produce significantly smaller average queue lengths than the SLIP algorithm.

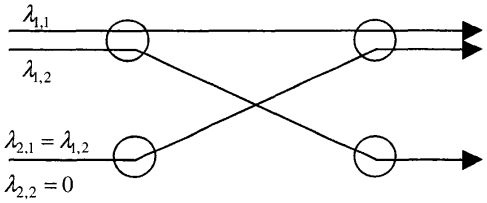

Fig. 2. Unbalanced load for a 2 x 2 switch

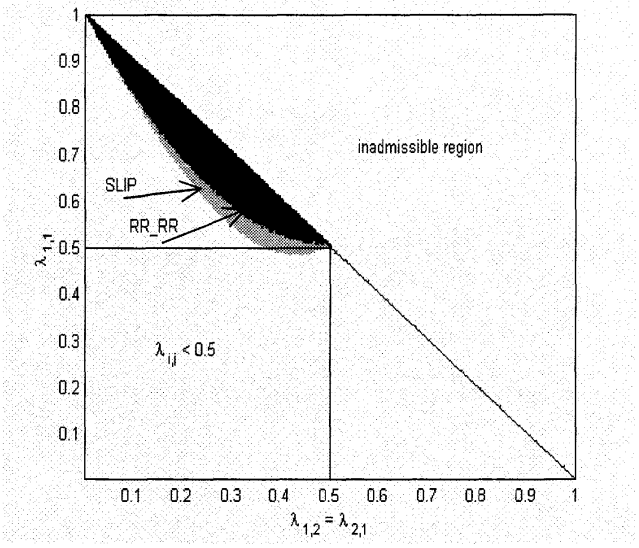

Fig. 3. Instability regions for unbalanced traffic on a 2x2 switch

Fig. 5 shows the average delay results for both the uniform Bernoulli and the uniform bursty traffic. The LQF_RR and OCF_OCF algorithms produce slightly larger average delays than the OQ switch for uniform Bernoulli traffic. For the uniform bursty traffic, the average delay of the LQF_RR algorithm is less than the OCF_OCF switch due to the smaller average queue sizes shown in Fig. 4. It should be noted, however, that the maximum delay (not shown here) for the LQF_RR algorithm is larger than for the OCF_OCF.

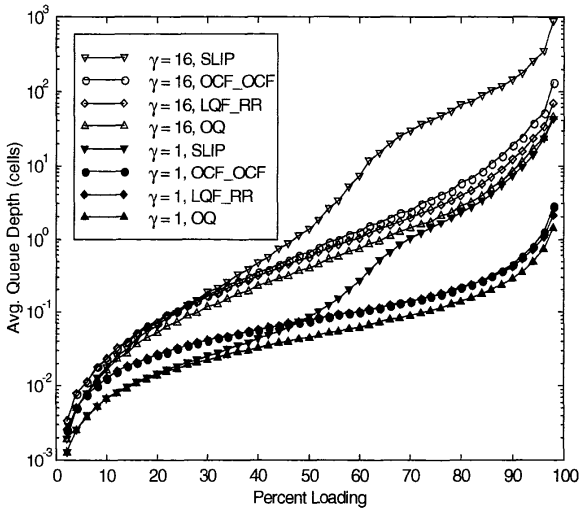

Fig. 4. Average VOQ depths for uniform Bernoulli and uniform bursty traffic

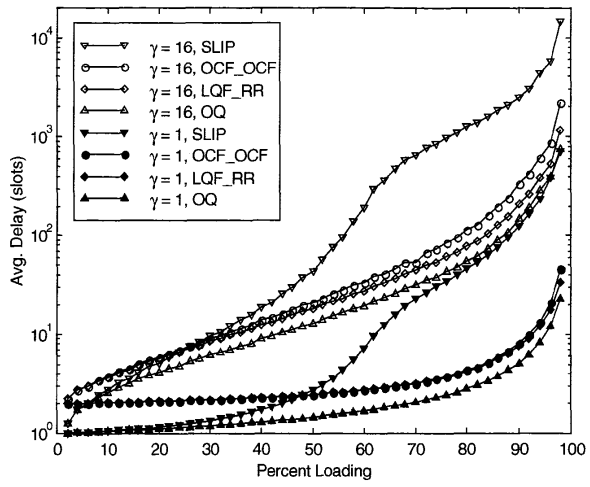

Fig. 5. Average cell delay for uniform Bernoulli and uniform bursty traffic

IV. CONCLUSIONS

An $N \times N$ CICQ switch fabric with one buffer per crosspoint is cost-effective because the required memory speed is only twice the line rate for both the inputs and for the N^2 crosspoints. The proposed LQF_RR scheduling algorithm, using VOQs at the input ports and credit-based flow control, is shown through analysis and simulation to be a high throughput system. It is shown, through fluid model techniques, that 100% throughput is achieved for input traffic that satisfies the strong law of large numbers and that produces a load $\leq 1/N$ for any input/output pair. Simulations indicate that 100% throughput may be achieved for a much larger class of admissible loads.

REFERENCES

[1] N. McKeown, "Scheduling algorithms for input-queued cell switches," *PhD Thesis, University of California at Berkeley*, May 1995.

[2] J. G. Dai, B. Prabhakar, "The throughput of data switches with and without speedup," *INFOCOM 2000*.

[3] A. Kam, K-Y. Siu, "Linear-Complexity algorithms for QoS support in input-buffered switches with no speedup," *IEEE JSAC*, vol. 17, no. 6, pp. 1040-1056, June 1999.

[4] M. Nabeshima, "Performance evaluation of a combined input- and crosspoint-queued switch," *IEICE Trans. Commun.*, Vol. E83-B, No. 3, March 2000.

[5] H.T. Kung, and R. Morris, "Credit-Based flow control for ATM networks," *IEEE Network Magazine*, vol. 9, pp. 40-48, March/April 1995.

[6] P. Gupta and N. McKeown, "Designing and implementation of a fast crossbar scheduler," *IEEE Micro*, vol. 19, pp. 20-28, Jan. – Feb. 1999.

APPENDIX

A. *Proof of Lemma 1*

To prove Lemma 1 we need to show that every *fluid limit* of the switch operating under the LQF_RR algorithm satisfies (7) when $\lambda_{ij} \leq 1/N$. We begin with the definition of fluid limits and the procedure to construct them.

Recall that $A_{ij}(n,\omega)$, $Z_{ij}(n,\omega)$, and $D_{ij}(n,\omega)$ are, respectively, the cumulative number of arrivals to, the number of packets in, and the cumulative number of departures from VOQ_{ij} at the beginning of time slot n on the sample path ω. We can extend these discrete functions for arbitrary time $t \in [n, n+1)$ as follows:

$$\begin{aligned} A_{ij}(t,\omega) &= A_{ij}(n,\omega); \\ Z_{ij}(t,\omega) &= Z_{ij}(n,\omega); \\ D_{ij}(t,\omega) &= D_{ij}(n,\omega) + (t-n)\big(D_{ij}(n+1,\omega) - D_{ij}(n,\omega)\big). \end{aligned} \tag{11}$$

Now for each $r > 0$ define

$$\begin{aligned} \bar{A}_{ij}^{r}(t,\omega) &= r^{-1} A_{ij}(rt,\omega); \\ \bar{Z}_{ij}^{r}(t,\omega) &= r^{-1} Z_{ij}(rt,\omega); \\ \bar{D}_{ij}^{r}(t,\omega) &= r^{-1} D_{ij}(rt,\omega). \end{aligned} \tag{12}$$

It can be shown (See Appendix A of [2]) that for each sample path ω satisfying (1) and any sequence $\{r_n\}$ with $r_n \to \infty$ as $n \to \infty$, there exists a subsequence $\{r_{n_k}\}$ and the continuous functions $\big(\bar{D}_{ij}(.), \bar{Z}_{ij}(.)\big)$ such that for any $t \geq 0$, as $k \to \infty$

$$\begin{aligned} \bar{A}_{ij}^{r_{n_k}}(t,\omega) &\to \lambda_{ij} t; \\ \bar{Z}_{ij}^{r_{n_k}}(t,\omega) &\to \bar{Z}_{ij}(t); \\ \bar{D}_{ij}^{r_{n_k}}(t,\omega) &\to \bar{D}_{ij}(t). \end{aligned} \tag{13}$$

Definition 2: Any function obtained through the limiting procedure in (13) is said to be a fluid limit of the switch.

Proof of Lemma 1: We need to prove that any fluid limit (obtained through the limiting procedure in (13)) satisfies (7). In other words we need to prove that if $F_i(t) > 0$, then for $\forall \varepsilon > 0$ $\exists \delta > 0$ such that

$$\sum_{j \in \mathrm{K}} \frac{\bar{Z}_{ij}(t') - \bar{Z}_{ij}(t)}{t' - t} \leq \varepsilon \qquad \forall t' \in [t, t+\delta]. \tag{14}$$

Suppose that $F_i(t) > 0$. This implies that for $\forall j_1 \in \mathrm{K}$, $\forall j_2 \notin \mathrm{K}$ $\bar{Z}_{ij_1}(t) - \bar{Z}_{ij_2}(t) > 0$. By continuity of these functions, $\exists \delta$ such that $\min_{t' \in [t, t+\delta]} \bar{Z}_{ij_1}(t') - \bar{Z}_{ij_2}(t') > 0$ for $\forall j_1 \in \mathrm{K}$, $\forall j_2 \notin \mathrm{K}$. Set $a = \min_{\substack{j_1 \in \mathrm{K} \\ j_2 \notin \mathrm{K}}} \min_{t' \in [t, t+\delta]} \big\{\bar{Z}_{ij_1}(t') - \bar{Z}_{ij_2}(t')\big\}$. Thus, for large enough k, and $\forall j_1 \in \mathrm{K}$, $\forall j_2 \notin \mathrm{K}$,

$$\bar{Z}_{ij_1}^{r_{n_k}}(t') - \bar{Z}_{ij_2}^{r_{n_k}}(t') \geq a/2 \quad \text{for } t' \in [t, t+\delta]. \tag{15}$$

Also, for large enough k we have $r_{n_k} a/2 \geq 1$. Thus for large enough k, and $\forall j_1 \in \mathrm{K}$, $\forall j_2 \notin \mathrm{K}$,

$$Z_{ij_1}(t') - Z_{ij_2}(t') \geq 1 \quad \text{for } t' \in \big[r_{n_k} t, r_{n_k}(t+\delta)\big]. \tag{16}$$

This means that for a large interval $\big[r_{n_k} t, r_{n_k}(t+\delta)\big]$, the longest VOQ at input i belongs to the set $\mathrm{B} \triangleq \{VOQ_{ij} : j \in \mathrm{K}\}$ and that during this large interval, any VOQ that belongs to set B is nonempty.

Claim: For every $t' \in [t, t+\delta]$ and for any r_{n_k} we have

$$\sum_{j \in \mathrm{K}} \big(Z_{ij}(r_{n_k} t') - Z_{ij}(r_{n_k} t)\big) \leq \sum_{j \in \mathrm{K}} \big(A_{ij}(r_{n_k} t') - A_{ij}(r_{n_k} t)\big) - Lm, \tag{17}$$

where $L \in \mathbb{Z}$, $NL \leq r_{n_k} t' - r_{n_k} t < N(L+1)$ and $m = \|\mathrm{K}\| > 0$.

Proof of the Claim: For any input i, at any time slot n, and for $1 \le p \le m$ define $t_i^p(n) = \inf\{t \ge 0 : \text{the credit states of at least } p \text{ of } VOQ_{ij}, j \in \mathrm{K} \text{ equal 0 before time } t+n\}$. First notice that $0 \le t_i^1(n) \le \cdots \le t_i^p(n) \le \cdots \le t_i^m(n) < N$, where the last inequality holds because the output scheduler operates as a round robin. On the other hand since the input scheduler selects one VOQ at a time, we have $t_i^p(n) < N - m + p$. In addition, we know at input i, during the interval $\left[r_{n_k} t, r_{n_k}(t+\delta)\right]$, the longest virtual output queue(s) belongs to the set B. Hence for any time interval of length N which occurs between time $r_{n_k} t$ and $r_{n_k} t'$ there are at least m departures from set B. Therefore we have (17).

□

Now for $\forall t' \in [t, t+\delta]$, $\forall \varepsilon > 0$, choose $r_{n_k} > m/\varepsilon(t'-t)$. Then, as expressed in the Claim, define L such that $NL \le r_{n_k} t' - r_{n_k} t < N(L+1)$. This implies that

$$\frac{L}{r_{n_k}} > \frac{L(t'-t)}{N(L+1)} \tag{18}$$

and

$$L > \left(\frac{m}{N\varepsilon} - 1\right). \tag{19}$$

Since $x/x+1$ is an increasing function of x, (19) implies

$$\frac{L}{L+1} > \left(1 - \frac{N\varepsilon}{m}\right). \tag{20}$$

Combining (20) and (18), we have

$$\frac{L}{r_{n_k}} > \left(1 - \frac{\varepsilon N}{m}\right)\frac{(t'-t)}{N}. \tag{21}$$

Using the claim we can write

$$\begin{aligned} \frac{1}{r_{n_k}} \sum_{j \in \mathrm{K}} \left(Z_{ij}(r_{n_k} t') - Z_{ij}(r_{n_k} t)\right) &\le \frac{1}{r_{n_k}} \sum_{j \in \mathrm{K}} \left(A_{ij}(r_{n_k} t') - A_{ij}(r_{n_k} t)\right) - \frac{Lm}{r_{n_k}} \\ &\le \frac{1}{r_{n_k}} \sum_{j \in \mathrm{K}} \left(A_{ij}(r_{n_k} t') - A_{ij}(r_{n_k} t)\right) - \frac{m}{N}(t'-t) + \varepsilon(t'-t). \end{aligned} \tag{22}$$

Letting $k \to \infty$, yields

$$\begin{aligned} \sum_{j \in \mathrm{K}} \left(\bar{Z}_{ij}(t') - \bar{Z}_{ij}(t)\right) &\le \sum_{j \in \mathrm{K}} \lambda_{ij}(t'-t) - \frac{m}{N}(t'-t) + \varepsilon(t'-t) \\ &\le \sum_{j \in \mathrm{K}} \left(\lambda_{ij} - \frac{1}{N}\right)(t'-t) + \varepsilon(t'-t) \\ &\le \varepsilon(t'-t). \end{aligned} \tag{23}$$

in which the last inequality holds because for every $i, j = 1, 2, \ldots, N$, $\lambda_{ij} \le 1/N$.

Equivalently, (23) can be written as

$$\sum_{j \in \mathrm{K}} \frac{\bar{Z}_{ij}(t') - \bar{Z}_{ij}(t)}{t'-t} \le \varepsilon. \tag{24}$$

So the proof of equation (14), hence the proof of Lemma 1, is complete.

□

B. *Proof of Lemma 2*

In order to prove Lemma 2, we prove the following fact which is a generalization of Lemma 2.

Fact 3: Consider $G(t) = \max_{j \in \mathrm{A}} G_j(t)$, where A is a finite set (Note that in Lemma 2, $\mathrm{A} = \{1, 2, \ldots, N\}$), and for every $j \in \mathrm{A}$, G_j is absolutely continuous. If G is differentiable at point t, and $i = \arg\max_{j \in \mathrm{A}} G_j(t)$ (in Lemma 2, this condition implies that $i \in \mathrm{K}$) we have $\dot{G}_i(t) = \dot{G}(t)$.

Proof: Choose two sequences $\{t_n\}$ and $\{s_n\}$ with $s_n \nearrow t$ and $t_n \searrow t$. Since G (hence, G_i) is differentiable at t,

$$\dot{G}(t) = \lim_{n \to \infty} \frac{G(s_n) - G(t)}{s_n - t} = \lim_{n \to \infty} \frac{G(t_n) - G(t)}{t_n - t} \tag{25}$$

and

$$\dot{G}_i(t) = \lim_{n \to \infty} \frac{G_i(s_n) - G_i(t)}{s_n - t} = \lim_{n \to \infty} \frac{G_i(t_n) - G_i(t)}{t_n - t}. \tag{26}$$

On the other hand for every s_n we have

$$\frac{G(s_n) - G(t)}{s_n - t} \le \frac{G_i(s_n) - G_i(t)}{s_n - t}, \tag{27}$$

while for every t_n

$$\frac{G(t_n) - G(t)}{t_n - t} \ge \frac{G_i(t_n) - G_i(t)}{t_n - t}. \tag{28}$$

Combining (25), (26), (27), and (28) yields

$$\dot{G}_i(t) = \dot{G}(t). \tag{29}$$

And the proof of Lemma 2 is complete.

□