The throughput of data switches with and without speedup

J. G. "Jim" Dai
Schools of Industrial and Systems Engineering, and Mathematics
Georgia Institute of Technology

Balaji Prabhakar
Departments of Electrical Engineering and Computer Science
Stanford University

Abstract

In this paper we use fluid model techniques to establish two results concerning the throughput of data switches. For an input-queued switch (with no speedup) we show that a maximum weight algorithm for connecting inputs and outputs delivers a throughput of 100%, and for combined input- and output-queued switches that run at a speedup of 2 we show that any maximal matching algorithm delivers a throughput of 100%. The only assumptions on the input traffic are that it satisfies the strong law of large numbers and that it does not oversubscribe any input or any output.

1 Introduction

Packet switches based on an input-queued (IQ) crossbar architecture are attractive for use in high speed networks. This is because the buffers which queue packets at the inputs need only run twice as fast the line rates. That is, if time were slotted so that at most one packet arrived at each input of the switch per time slot, then an input buffer potentially needs to make upto two transactions per time slot: (1) write in an incoming packet, and (2) copy a buffered packet onto the crossbar fabric. Hence the bandwidth of the input buffers is no more than twice the line rate. In contrast, the buffers of an $N \times N$ output-queued (OQ) switch are required to run at least $N + 1$ times the line rate. Even for moderately sized switches running at high speeds, memories with such large speedups are either very expensive or simply unavailable (see, for example, [20] for an elaboration of this point).

However, IQ switches which maintain a single first-in-first-out (FIFO) buffer at the inputs are known to suffer from the so-called head-of-line (HoL) blocking problem. The paper of Karol et al [17] shows that this problem can limit the throughput of the switch to about 58% when the input traffic is independent, identically distributed (i.i.d.) Bernoulli and the output destinations are uniform. This is in contrast to OQ switches which always deliver 100% throughput, since no output will idle as long as there is a packet in the switch destined for it.

It has since been shown that the low throughput of IQ switches is merely an artifact of HoL blocking caused due to a FIFO organization of the input buffers, and that IQ switches can achieve a throughput of upto 100% by using a simple scheme known as "virtual output queueing" and by using suitable packet scheduling algorithms [20, 21, 28]. However, all of these results are shown to hold only when the input traffic is i.i.d. although they allow a non-uniform loading of the switch.

It has been believed for some time now that an IQ switch can deliver 100% throughput for *arbitrarily* distributed input patterns so long as no input or output is oversubscribed. That is, the results of [20, 21, 28] ought to be true for a wider class of input distributions and that the i.i.d. assumption is only required by their method of proof. The first result of this paper, Theorem 1, provides a proof of this belief using fluid model techniques. More precisely, Theorem 1 proves that an IQ switch using a maximum weight matching algorithm can achieve a throughput of upto 100% when subjected to arbitrarily distributed input traffic that satisfies the following mild conditions: (i) It obeys the strong law of large numbers, and (ii) it does not oversubscribe any input or output. Theorem 1, therefore, builds upon and extends the work of [21, 28].

After the appearance of [17], a number of researchers (for example, [6, 8, 15, 16, 25]) considered improving the throughput of an IQ switch by using fabrics with a moderate "speedup"[1]. A common conclusion of these studies is that with a speedup of 4 or 5 one can achieve upto 100% throughput when arrivals are i.i.d. at each input, and the distribution of packet destinations

[1]A switch with a fabric speedup of s can remove up to s packets from each input and deliver up to s packets to each output within a time slot. Hence, an OQ switch has a speedup of N while an IQ switch has a speedup of 1. For values of s between 1 and N packets need to be buffered at the inputs before switching as well as at the outputs after switching. We shall refer to this type of a switch as a combined input- and output-queued (CIOQ) switch.

is uniform across the outputs.

One hopes that it is again possible to remove the i.i.d. restrictions on the input traffic patterns. In fact, more is true. Prabhakar and McKeown [23], Chuang et al [9], and Krishna et al [18] have recently devised a number of algorithms that allow a combined input- and output-queued (CIOQ) switch with an internal speedup of between two and four to *exactly emulate* (packet-by-packet) an OQ switch. Furthermore, these algorithms have been shown to work for all input traffic patterns and switch sizes. Since an OQ switch always delivers a throughput of 100%, the previously mentioned exact emulation ensures that the CIOQ switch also delivers a throughput of 100%.

The results of [9, 18, 23] are obtained with specific packet scheduling algorithms. It is interesting to ask just how well a CIOQ switch that employs an arbitrary, but well-chosen, scheduling algorithm performs as its fabric speedup is increased. Charny [4] and Charny et al [5] have recently obtained the following answer to this question: When the speedup of a CIOQ switch is at least 4 and the input traffic is leaky bucket constrained, any maximal matching algorithm (see Definition 5) delivers 100% throughput. Theorem 2 of this paper generalizes this result in two ways: (i) It lowers the minimum required speedup to 2, and (ii) it removes the restriction of leaky bucket constrained inputs.

The results of this paper are derived by considering the fluid model analogs of an IQ or a CIOQ switch. The framework of fluid models has proved to be powerful in obtaining the maximum throughput region (or, the stability region) of a variety of stochastic networks under very mild assumptions on the input traffic (see [26, 10, 14, 27, 7, 13, 22, 11, 3, 24]). For a general exposition of the stability analysis of stochastic networks using fluid models, please refer to the recent set of notes by Dai [12]. In the fluid model framework, in order to prove that a switch delivers a throughput of 100% it is enough to prove that the corresponding fluid model is *weakly stable*. This is the gist of Theorem 3.

We conclude the introduction with a few words about the organization of the paper and about the practical significance of the results obtained. Since fluid model techniques are relatively new in the computer networking context, we have included an appendix in which the procedure for obtaining fluid limits for a discrete stochastic network (in this case, the network consists of a single switch) is given in detail. As mentioned previously, the fluid model method applies to very general traffic processes. Indeed, the only requirement is that they satisfy a strong law of large numbers. Since almost all real traffic processes satisfy this property, the results of this paper have a high practical significance. A second aspect of this paper is that it shows that *any maximal* matching algorithm delivers a 100% throughput under a speedup of 2. The significance of this result derives from the fact that maximal matchings are easier to find than maximum matchings, and hence better suited for implementation. In particular, and to the best of our knowledge, this is the first proof that the popular and well-studied PIM [1] and iSLIP [19] schedulers (which find maximal matchings) deliver a 100% throughput under arbitrary packet arrival patterns.

2 Model and notation

Consider an $N \times N$ crossbar switch such as the one shown in Figure 1. 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 has a buffer of infinite capacity for holding packets prior to switching them to their respective outputs. Likewise each output has an infinite capacity buffer for holding packets that will be placed on the outgoing line. The buffer at an input 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. The queueing discipline at each VOQ and at the output buffer, which typically determines the quality-of-service (QoS) that a flow obtains from the switch, can be entirely arbitrary and is not of concern in this note.

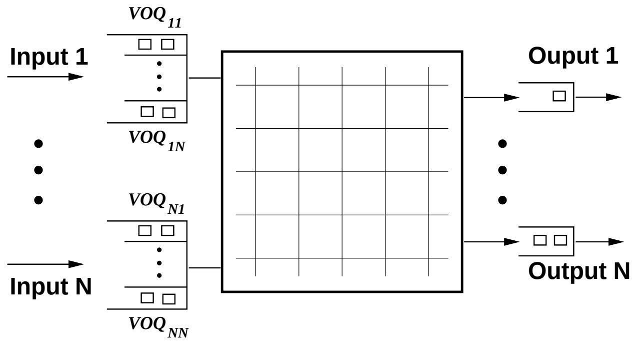

Figure 1: *A CIOQ switch*

A "scheduling cycle" consists of two parts: (a) the matching part, and

(b) the switching part. During the matching part a matching algorithm, m, selects a matching between inputs and outputs in such a way that no input (respectively, output) may be matched to more than one output (respectively, input). During the switching part input i transfers a packet to output j if they are matched to each other and VOQ_{ij} is non-empty.

A matching may be represented by a permutation matrix π. That is, input i is matched to output j if $\pi_{ij} = 1$, otherwise input i is not matched to output j. Let Π be the set of all $N!$ permutation matrices.

The switch is said to have a speedup of s, where $s \in \{1, \ldots, N\}$, if during every time slot there are s scheduling cycles. We will also refer to each of the scheduling cycles in a time slot as a "phase". When the speedup s is bigger than 1, any packets that are transferred from an input i to an output j during a phase will be assumed to be transferred at the end of the phase. As mentioned earlier, when $s > 1$ buffers are required at the ouputs as well. This leads to the combined input- and output-queued (CIOQ) architecture.

In addition to assuming that packets arrive at the switch just at the beginning of a time slot, we shall also assume that packets depart from the switch just prior to the end of a time slot.

Definition 1. A matching algorithm m is a specification of a sequence of permutations $\{\pi_{ij}^m(n + \frac{k}{s})\}_{n,k}$, where $\pi_{ij}^m(n + \frac{k}{s})$ indicates the event that input i is matched to output j during phase k of time slot n.

Let $A_{ij}(n)$ denote the number packets that have arrived at input i destined for output j up to time slot n. Since we assume that packet arrivals occur at the beginning of a time slot, for any time $t \in (n-1, n)$, $A_{ij}(n)$ is the cumulative number of packets that have arrived at VOQ_{ij} by time t. We adopt the convention that $A_{ij}(0) = 0$. We assume that the arrival processes $\{A_{ij}(\cdot), i, j = 1, \ldots, N\}$ satisfy a strong law of large numbers (SLLN): with probability one,

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

We call λ_{ij} the arrival rate at VOQ_{ij}. Assumption (1) on arrival processes is very mild. It is satisfied, for example, when the arrival processes $\{A_{ij}(\cdot), i, j = 1, \ldots, N\}$ are jointly stationary and ergodic with arrival rates λ_{ij}.

Let $D_{ij}(n)$ be the number of departures from $\mathrm{VOQ}_{ij}(n)$ up to time slot n. We also adopt the convention that $D_{ij}(0) = 0$.

Definition 2. A switch operating under a matching algorithm is said to be *rate stable* if, with probability one,

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

for any arrival processes satisfying (1).

Definition 3. A matching algorithm is said to be *efficient* if (2) holds for any arrival processes satisfying (1) and

$$\sum_i \lambda_{ij} \le 1, \quad \sum_j \lambda_{ij} \le 1. \tag{3}$$

Since, each output link can potentially transmit one packet in each time slot,

$$\lim_{n\to\infty} \frac{\sum_i D_{ij}(n)}{n}$$

is the long-run fraction of time that output link j is busy. A switch operating under an efficient matching algorithm can keep each output link 100% busy, equally the switch can achieve upto 100% throughput, if there is enough offered load.

Write $Z_{ij}(n)$ for the number of packets in VOQ_{ij} at the beginning of time slot n, including any packet that might have just arrived at time $n-1$.

3 Stability results

In this section, we state the two major results of this paper. The first states that a maximum weight matching algorithm is efficient for a switch with speedup 1. The second states that any maximal matching algorithm is efficient for a switch with speedup $s \ge 2$.

3.1 Speedup of 1

When the switch speedup is 1 there is only one scheduling cycle and hence no more than one packet may be removed from each input or transferred to each output in one time slot. A packet that reaches its output at the end of a time slot will depart immediately from the switch, and hence there is no need for output buffers. Thus, at a speedup of 1, we are led to the input-queued architecture. For each permutation (or matching of inputs and outputs) $\pi \in \Pi$, let the "weight" under matching π equal

$$f_\pi(n) = \langle \pi, Z(n) \rangle,$$

where for two matrices A and B of the same size, $\langle A, B \rangle = \sum_{ij} A_{ij} B_{ij}$.

Definition 4. Under the *maximum weight matching algorithm*, w,

$$\pi^w(n) = \arg\max_\pi \{f_\pi(n)\}. \tag{4}$$

Let $f(n) = f_{\pi^w(n)}(n)$ be the weight of the maximum weight matching at time n.

When there are multiple matchings that all have equal weight we choose one of these matchings arbitrarily to break the tie.

We shall prove the following theorem in Section 5.

Theorem 1. *A maximum weight matching algorithm is efficient.*

3.2 Speedup of 2

Recall that for a switch with a speedup of s there are s scheduling cycles.

Definition 5. A matching algorithm x is said to be a *maximal matching algorithm* or a *nonidling matching algorithm* if for every phase k of every time n, $Z_{ij}(n + \frac{k}{s}) > 0$ implies that at least one of the following holds:
(1) $Z_{ij'}(n + \frac{k}{s})\, \pi^x_{ij'}(n + \frac{k}{s}) > 0$
(2) $Z_{i'j}(n + \frac{k}{s})\, \pi^x_{i'j}(n + \frac{k}{s}) > 0$,
for some $i', j' \in \{1, \ldots, N\}$.

Thus under a maximal matching algorithm if input i has a packet for output j at the beginning of a scheduling cycle, then either (i) Input i is matched to output j, or (ii) Input i is matched to an output $j' \neq j$ for which it has a packet, or (iii) Output j is matched to an input $i' \neq i$ which has a packet for output j.

Theorem 2. *Any maximal weight matching algorithm is efficient, so long as the speedup $s \geq 2$.*

4 Fluid Models

We now introduce the fluid model of a switch. To do this, we first write down the equations that govern the (discrete) dynamics of a switch. We then write down the corresponding fluid model equations of the switch.

4.1 Switch dynamics

Suppose the switch employs some (yet to be specified) matching algorithm m. For a $\pi \in \Pi$, let $T_\pi^m(n)$ be the cumulative amount of time that permutation π has been used by time slot n. Again, we assume $T_\pi^m(0) = 0$. The following equations of evolution hold for the switch: for $n \geq 0$ and $i, j = 1, \ldots, N$,

$$Z_{ij}(n) = Z_{ij}(0) + A_{ij}(n) - D_{ij}(n), \tag{5}$$

$$D_{ij}(n) = \sum_{\pi \in \Pi} \sum_{\ell=1}^{n} \pi_{ij} \, 1_{\{Z_{ij}(\ell)>0\}} (T_\pi^m(\ell) - T_\pi^m(\ell - 1)), \tag{6}$$

$$T_\pi^m(\cdot) \text{ is non-decreasing, and } \sum_{\pi \in \Pi} T_\pi^m(n) = n. \tag{7}$$

The first equation tracks the evolution of Z_{ij} in terms of the total number of arrivals at and departures from VOQ_{ij}. The second equation keeps a count of the cumulative number of departures from VOQ_{ij}. And the third equation expresses the fact that input i is matched to some output or the other at each time.

4.2 Fluid equations

We now describe a deterministic, continuous fluid model of a switch operating under some matching algorithm m, with offered traffic satisfying (1). Let $T_\pi^m(t)$ be the cumulative amount of time in $[0, t]$ that matching π was employed under the matching algorithm m. The fluid model is governed by the following set of equations:

$$Z_{ij}(t) = Z_{ij}(0) + \lambda_{ij} t - D_{ij}(t) \geq 0, \quad t \geq 0, \quad i, j = 1, \ldots, N, \tag{8}$$

$$\dot{D}_{ij}(t) = \sum_{\pi \in \Pi} \pi_{ij} \dot{T}_\pi^m(t), \quad \text{if } Z_{ij}(t) > 0, \quad t \geq 0, \quad i, j = 1, \ldots, N, \tag{9}$$

$$T_\pi^m(\cdot) \text{ is nondecreasing, and } \sum_{\pi \in \Pi} T_\pi^m(t) = t, \quad t \geq 0, \tag{10}$$

where, for a function f, $\dot{f}(t)$ denotes the derivative of f at t. We adopt the convention that whenever symbol $\dot{f}(t)$ is used, f is assumed to a differentiable at t.

Equations (8)-(10) are fluid model equations. Each solution (D, T, Z) to (8)-(10) is said to be a fluid model solution. One interprets $Z_{ij}(t)$ as the buffer level at time t in VOQ_{ij} and $D_{ij}(t)$ as the total amount of fluid departing from VOQ_{ij} in $[0,t]$. Equation (8) is a basic flow equation. Equation (9) has an equivalent characterization: Whenever $Z_{ij}(t) > 0$, there exists a $\delta > 0$ such that

$$D_{ij}(t') - D_{ij}(t) = \sum_{\pi \in \Pi} \pi_{ij}(T_\pi^m(t') - T_\pi^m(t)), \quad \text{for } t' \in [t, t+\delta]. \tag{11}$$

Equation (11) says that if the amount of fluid in VOQ_{ij} is positive at time t, then, for small enough $\delta > 0$, the amount of fluid drained from VOQ_{ij} in the interval $[t, t'] \subset [t, t+\delta]$ equals the amount of time that input i and output j were matched to each other during $[t, t']$.

Depending on the matching algorithm m used, often there are additional fluid model equations corresponding to matching algorithm m. For example, if m equals w, the maximum weight matching algorithm, the additional fluid equation takes the form: for each $\pi \in \Pi$,

$$\dot{T}_\pi^w(t) = 0 \text{ whenever } \langle \pi, Z(t) \rangle < \langle \pi', Z(t) \rangle \text{ for some } \pi' \in \Pi. \tag{12}$$

The above equation says that under the maximum weight matching algorithm, a matching π which has weight less than another matching π' at some time t will not be employed at that time. Thus, equation (12) characterizes the maximum weight matching algorithm and is added to the basic fluid model equations (8)-(10) whenever we consider the fluid model of a switch employing the maximum weight matching algorithm.

In general, deciding which equation can be added to a fluid model is related to *fluid limits* and is discussed in Section A.1.

Definition 6. The fluid model of a switch operating under a matching algorithm is said to be *weakly stable* if for every fluid model solution (D, T, Z) with $Z(0) = 0$, $Z(t) = 0$ for $t \geq 0$.

Theorem 3. *A switch operating under a matching algorithm is rate stable if the corresponding fluid model is weakly stable.*

We defer the proof to the appendix.

5 Proofs of Theorems 1 and 2

In this section, we prove Theorems 1 and 2. In light of Theorem 3, it suffices to prove that, in each case, the corresponding fluid model is weakly stable. We first state the following simple lemma.

Lemma 1. *Let $f : [0, \infty) \to [0, \infty)$ be an absolutely continuous function with $f(0) = 0$. Assume that $\dot{f}(t) \leq 0$ for almost every t (wrt Lebesgue measure) such that $f(t) > 0$ and f is differentiable at t. Then $f(t) = 0$ for almost every $t \geq 0$.*

Proof. For almost every $t \geq 0$, $f^2(t) - f^2(0) = 2\int_0^t f(s)\dot{f}(s)\, ds \;\leq\; 0$, since $f(s)\dot{f}(s) \leq 0$ a.e. in $[0, t]$. Now $f(0) = 0$ and $f(t) \geq 0$ imply that $f(t) = 0$ for almost every t. □

5.1 Proof of Theorem 1

Let (D, T, Z) be a fluid model solution satisfying (8)-(10) and (12) with $Z(0) = 0$. For a permutation matrix π, define $f_\pi(t) = \langle \pi, Z(t) \rangle$. Let $f(t) = \max_\pi f_\pi(t)$. Let λ be the $N \times N$ matrix with entries λ_{ij}. It is well-known that under condition (3)

$$\langle \lambda, Z(t) \rangle \leq f(t) \text{ for } t \geq 0.$$

Briefly, this is because under condition (3) λ is doubly sub-stochastic and can therefore be written as a convex combination of permutation matrices, from which the above inequality follows. See Lemma 2 of [21] for details.

Let t be a fixed value such that f and Z are differentiable at t. Let Π' be the set of matchings π such that $f_\pi(t) = f(t)$. Then we have $\dot{f}_\pi(t) = \dot{f}(t)$ for $\pi \in \Pi'$ (see, for example, the proof of Lemma 3.2 of [14]), and by (12),

$$\sum_{\pi \in \Pi'} \dot{T}_\pi(t) = 1.$$

It follows that

$$\begin{aligned}
\langle Z(t), \dot{D}(t) \rangle &= \langle Z(t), \sum_{\pi \in \Pi'} \pi \dot{T}_\pi(t) \rangle \\
&= \sum_{\pi \in \Pi'} \langle Z(t), \pi \dot{T}_\pi(t) \rangle
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{\pi \in \Pi'} f_\pi(t) \dot{T}_\pi(t) \\
&= f(t) \sum_{\pi \in \Pi'} \dot{T}_\pi(t) \\
&= f(t).
\end{aligned}
$$

Thus,

$$
\begin{aligned}
\langle Z(t), \dot{Z}(t) \rangle &= \langle Z(t), \lambda \rangle - \langle Z(t), \dot{D}(t) \rangle \\
&= \langle Z(t), \lambda \rangle - f(t) \\
&\leq 0.
\end{aligned}
$$

It follows that $d\langle Z(t), Z(t) \rangle / dt \leq 0$ for any $Z(t) \neq 0$. Since $Z(0) = 0$, from Lemma 1 we have that the fluid model is weakly stable.

5.2 Proof of Theorem 2

Consider the fluid model of a switch having a speedup of s, operating under a maximal matching algorithm. Let (D, T, Z) be a fluid model solution with $Z(0) = 0$. Let $L_i(t) = \sum_{j'} Z_{ij'}(t)$ denote the total amount of fluid queued at input i at time t. Similarly, let $M_j(t) = \sum_{i'} Z_{i'j}(t)$ be the total amount of fluid destined for output j and queued at some input at time t. Define $C_{ij}(t) = L_i(t) + M_j(t)$. In addition to the fluid model equations (8)-(10), under a maximal matching algorithm for a switch having a speedup of s, the fluid model solution satisfies the following additional equation:

$$
\dot{C}_{ij}(t) \leq \sum_{j'} \lambda_{ij'} + \sum_{i'} \lambda_{i'j} - s \quad \text{whenever } Z_{ij}(t) > 0. \tag{13}
$$

Equation (13) can be added to the fluid model because of the following lemma.

Lemma 2. *For a switch with speedup of s operating under a maximal matching algorithm, each fluid limit must satisfy (13).*

We leave the proof to the appendix. We provide an intuitive explanation here. Suppose that at the beginning of a time slot, the number of packets at VOQ_{ij} is at least s. Then during each of the s scheduling cycles within the time slot, there is at least one packet at VOQ_{ij}. Therefore, during a scheduling cycle, either (1) a packet moves from input i to an output j', or (2) a packet moves from an input i' to output j. Hence C_{ij} reduces by at least s during a time slot due to departures. It increases by the number of packets that arrive at input i or for output j. Hence the change in C_{ij} (measured in the fluid model by its derivative) is no more than the difference between the sum of the arrivals and the departures.

We now return to the proof of Theorem 2. Let Q be the $N \times N$ matrix with each entry being 1. One can check that

$$C(t) = QZ(t) + Z(t)Q \quad t \geq 0. \tag{14}$$

Define

$$f(t) = \langle Z(t), C(t) \rangle.$$

It follows that $f(t) \geq 0$ for $t \geq 0$ and $f(0) = 0$. It is also clear that $f(t) = 0$ implies that $Z(t) = 0$. We would like to show that $f(t) > 0$ implies $\dot{f}(t) \leq 0$, from which and Lemma 1 the weak stability of the fluid model follows. We claim (and will shortly prove) that

$$\dot{f}(t) = 2\langle Z(t), \dot{C}(t) \rangle. \tag{15}$$

Equivalently,

$$\dot{f}(t) = 2\sum_{i,j} Z_{ij}(t)\dot{C}_{ij}(t) = 2 \sum_{\{i,j: Z_{ij}(t)>0\}} Z_{ij}(t)\dot{C}_{ij}(t) \leq 0,$$

where the inequality is a consequence of (13). Therefore $\dot{f}(t) \leq 0$ whenever $f(t) > 0$, proving Theorem 2.

To establish (15), one first observes that

$$\begin{aligned} f(t) &= \sum_{i,j} Z_{ij}(t)C_{ij}(t) \\ &= \sum_{i,j} Z_{ij}(t)\Big(\sum_k Z_{ik}(t) + \sum_k Z_{kj}(t)\Big) \\ &= \sum_{i,j,k}\Big(Z_{ij}(t)Z_{ik}(t) + Z_{ij}(t)Z_{kj}(t)\Big). \end{aligned}$$

Therefore,

$$\dot{f}(t) = \sum_{i,j,k} \dot{Z}_{ij}(t)Z_{ik}(t) + \sum_{i,j,k} Z_{ij}(t)\dot{Z}_{ik}(t)$$

$$
\begin{aligned}
& + \sum_{i,j,k} \dot{Z}_{ij}(t) Z_{kj}(t) + \sum_{i,j,k} Z_{ij}(t) \dot{Z}_{kj}(t) \\
= \; & 2 \sum_{i,j,k} Z_{ij}(t) \dot{Z}_{ik}(t) + 2 \sum_{i,j,k} Z_{ij}(t) \dot{Z}_{kj}(t) \\
= \; & 2 \sum_{i,j} Z_{ij}(t) \dot{C}_{ij}(t),
\end{aligned}
$$

which proves (15).

References

[1] T. Anderson, S. Owicki, J Saxe and C. Thacker: "High speed switch scheduling for local area networks", *ACM Transactions on Computer Systems*, pp 319-352, Nov 1993.

[2] P. Billingsley. *Convergence of Probability Measures.* John Wiley & Sons, New York, 1968.

[3] M. Bramson: "Stability of two families of queueing networks and a discussion of fluid limits", *Queueing Systems: Theory and Applications*, 28:7–31, 1998.

[4] A. Charny: "Providing QoS Guarantees in Input Buffered Crossbar Switches with Speedup", PhD Thesis, MIT, 1998.

[5] A. Charny, P. Krishna, N. Patel and R. Simcoe: "Algorithms for Providing Bandwidth and Delay Guarantees in Input-buffered Crossbars with Speedup", *Presented at the 6^{th} IEEE/IFIP IWQ0S '98*, May 1998.

[6] C-Y. Chang, A.J. Paulraj and T. Kailath: "A Broadband Packet Switch Architecture with Input and Output Queueing", *Proc. Globecom '94*, pp.448-452.

[7] H. Chen: "Fluid approximations and stability of multiclass queueing networks I: Work-conserving disciplines", *Annals of Applied Probability*, 5:637–665, 1995.

[8] J.S.-C. Chen and T.E. Stern: "Throughput analysis, optimal buffer allocation, and traffic imbalance study of a generic nonblocking packet switch," *IEEE J. Select. Areas Commun.*, Apr. 1991, vol. 9, no. 3, pp. 439-49.

[9] S-T. Chuang, A. Goel, N. McKeown and B. Prabhakar: " Matching Output Queueing with a Combined Input Output Queued Switch", *IEEE J. Select. Areas Commun.*, **17**, no. 6, pp 1030-1039. A short version appears in *The Proceedings of Infocom '99.*

[10] J. G. Dai: "On positive Harris recurrence of multiclass queueing networks: A unified approach via fluid limit models", *Annals of Applied Probability*, 5:49–77, 1995.

[11] J. G. Dai: "A fluid-limit model criterion for instability of multiclass queueing networks", *Annals of Applied Probability*, 6:751–757, 1996.

[12] J. G. Dai: "Stability of fluid and stochastic processing networks", Miscellanea Publication, No. 9, Centre for Mathematical Physics and Stochastics, Denmark (`http://www.maphysto.dk/`), January 1999.

[13] J. G. Dai and S. P. Meyn: "Stability and convergence of moments for multiclass queueing networks via fluid limit models", *IEEE Transactions on Automatic Control*, 40:1889–1904, 1995.

[14] J. G. Dai and G. Weiss: "Stability and instability of fluid models for reentrant lines", *Mathematics of Operations Research*, 21:115–134, 1996.

[15] A.L. Gupta and N.D. Georganas: "Analysis of a packet switch with input and output buffers and speed constraints", *Proc. Infocom '91*, Bal Harbour, FL, April 1991, pp.694-700.

[16] I. Iliadis and W.E. Denzel: "Performance of packet switches with input and output queueing", *Proc. ICC '90*, Atlanta, GA, April 1990. pp.747-53.

[17] M. Karol, M. Hluchyj and S. Morgan: "Input Versus Output Queueing on a Space Division Switch", *IEEE Trans. Comm*, vol.35, no.12, pp.1347-1356.

[18] P. Krishna, N. Patel, A. Charny and R. Simcoe: "On the Speedup Required for Work-conserving Crossbar Switches" *Presented at the 6th IEEE/IFIP IWQ0S '98*, May 1998. To appear in *IEEE JSAC.*

[19] N. McKeown: "iSLIP: A Scheduling Algorithm for Input-Queued Switches", *IEEE Transactions on Networking*, v.7, pp 188-201, April 1999.

[20] N. McKeown, V. Anantharam, J. Walrand: "Achieving 100% Throughput in an Input-Queued Switch", *INFOCOM '96*, pp.296-302.

[21] N. McKeown, A. Mekkittikul, V. Anantharam and J. Walrand: "Achieving 100% throughput in an input-queued switch", to appear in *IEEE Transactions on Communications.*

[22] S. P. Meyn: "Transience of multiclass queueing networks via fluid limit models", *Annals of Applied Probability*, 5:946–957, 1995.

[23] B. Prabhakar and N. McKeown: "On the speedup required for combined input- and output-queued switching", *Automatica*, no. 35, v.12, pp. 1909-1920, Dec 1999.

[24] A. Puhalskii and A. N. Rybko: "Non-ergodicity of queueing networks under non-stability of their fluid models", Technical Report 141, University of Colorado at Denver, Center for Computational Mathematics, April 1999.

[25] Y. Oie, M. Murata, K. Kubota and H. Miyahara: "Effect of speedup in nonblocking packet switch", *Proc. ICC '89*, Boston, MA, June 1989, pp. 410-14.

[26] A. N. Rybko and A. L. Stolyar: "Ergodicity of stochastic processes describing the operation of open queueing networks", *Problems of Information Transmission*, 28:199–220, 1992.

[27] A. L. Stolyar: "On the stability of multiclass queueing networks: a relaxed sufficient condition via limiting fluid processes", *Markov Processes and Related Fields*, pages 491–512, 1995.

[28] L. Tassiulas and A. Ephremides: "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks", *IEEE Transactions of Automatic Control*, vol. 37, no. 12, pp 1936-1948, 1992.

A Appendix

In this appendix, we first introduce fluid limits, which will be used to prove Theorem 3. We then prove Lemma 2. The proof provides an example showing how one can add additional fluid model equations to a fluid model.

A.1 Fluid limits

In this section, we introduce fluid limits associated with a switch and prove that each fluid limit must be a fluid model solution to (8)-(10).

Recall that $Z_{ij}(n)$ is the number of packets in VOQ_{ij} at the beginning of time slot n. We extend the definition of $Z_{ij}(t)$, for arbitrary time $t \geq 0$, to be $Z_{ij}(\lfloor t \rfloor)$, where $\lfloor t \rfloor$ is the largest integer less than or equal to t. Then $Z_{ij}(\cdot) \in \mathbb{D}[0,\infty)$, where, for an integer d, $\mathbb{D}^d[0,\infty)$ is the space of functions $f : [0,\infty) \to \mathbb{R}^d$ that are right continuous and have left limits in $(0,\infty)$. Similarly, we can extend the definition of $A(t)$ so that it is defined for $t \geq 0$. Note that the functions $A_{ij}(\cdot)$ and $Z_{ij}(\cdot)$ are random elements of $\mathbb{D}[0,\infty)$, in general.

For purely technical reasons (which will soon become apparent), we wish to define $D(t)$ and $T_\pi(t)$ for $t \geq 0$ so that they are *continuous* functions. This merely involves making the following piecewise linear interpolation: For $t \in (n, n+1)$, let $D(t) = D(n) + (t-n)(D(n+1) - D(n))$ and let $T_\pi(t) = T_\pi(n) + (t-n)(T_\pi(n+1) - T_\pi(n))$.

Note that the functions $D_{ij}(t)$ and $T_\pi(t)$ are random elements of $\mathbb{C}[0,\infty)$. We shall sometimes use the notation $A(\cdot,\omega)$, $D(\cdot,\omega)$, $T_\pi(\cdot,\omega)$ and $Z(\cdot,\omega)$ to explicitly denote the dependency on the randomness ω.

For a switch with a speedup of s and for a fixed randomness ω, we have

$$D_{ij}(t+t',\omega) - D_{ij}(t',\omega) \leq t\,s, \quad t,t' \geq 0, \tag{16}$$

$$T_\pi(t+t',\omega) - T_\pi(t',\omega) \leq t, \quad t,t' \geq 0. \tag{17}$$

It should be clear to the reader that $D(t)$ and $T_\pi(t)$ were defined to be continuous in order to obtain the above uniform continuity properties.

Now, for each $r > 0$ define

$$\bar{A}^r(t,\omega) = r^{-1}A(rt,\omega), \quad \bar{D}^r(t,\omega) = r^{-1}D(rt,\omega), \tag{18}$$

$$\bar{T}^r(t,\omega) = r^{-1}T(rt,\omega), \quad \bar{Z}^r(t,\omega) = r^{-1}Z(rt,\omega). \tag{19}$$

It follows from (16) and (17) that

$$\bar{D}^r_{ij}(t) - \bar{D}^r_{ij}(t') \leq s(t-t'), \quad \bar{T}^r_\pi(t) - \bar{T}^r_\pi(t') \leq (t-t'), \tag{20}$$

for any $r > 0$ and $t \geq t' \geq 0$. Recall that a sequence of functions $f_n(\cdot)$ is said to converge to $f(\cdot)$ uniformly on compact (u.o.c.) intervals if, for every $t \geq 0$, $\sup_{0 \leq t' \leq t} |f_n(t') - f(t')| \to 0$ as $n \to \infty$. By the Arzela-Ascoli Theorem (see, e.g., Billingsley [2], pp 221), for a fixed ω, the family

$\{(\bar{D}^r(\cdot,\omega),\bar{T}^r(\cdot,\omega)), r>0\}$ is tight, as $r\to\infty$, in the space of continuous functions endowed with u.o.c. topology. That is, for each sequence $\{r_n\}$, there exists a subsequence $\{r_{n_k}\}$ and a continuous function $(\bar{D}(\cdot),\bar{T}(\cdot))$ such that, for any $t\ge 0$,

$$\lim_{k\to\infty}\sup_{0\le t'\le t}|\bar{D}_{ij}^{r_{n_k}}(t',\omega)-\bar{D}_{ij}(t')|=0, \tag{21}$$

$$\lim_{k\to\infty}\sup_{0\le t'\le t}|\bar{T}_{\pi}^{r_{n_k}}(t',\omega)-\bar{T}_{\pi}(t')|=0. \tag{22}$$

Note that

$$\lim_{r\to\infty}\sup_{0\le t'\le t}|\bar{A}_{ij}^{r}(t',\omega)-\lambda_{ij}t'|=0 \tag{23}$$

for all ω satisfying the SLLN assumption (1). Combining (21) and (23), we have that, for each randomness 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 function $(\bar{D}(\cdot),\bar{T}(\cdot),\bar{Z}(\cdot))$ such that

$$(\bar{D}^{r_{n_k}}(\cdot,\omega),\bar{T}^{r_{n_k}}(\cdot,\omega),\bar{Z}^{r_{n_k}}(\cdot,\omega))\to(\bar{D}(\cdot),\bar{T}(\cdot),\bar{Z}(\cdot)) \tag{24}$$

u.o.c. as $k\to\infty$.

Definition 7. Any function $(\bar{D}(\cdot),\bar{T}(\cdot)\bar{Z}(\cdot))$ obtained through the limiting procedure in (24) is said to be a *fluid limit* of the switch.

One can check that each fluid limit $(\bar{D},\bar{T},\bar{Z})$, obtained from (24), satisfies the fluid model equation (8). Since $\bar{Z}^r(0,\omega)=r^{-1}Z(0,\omega)\to 0$ as $r\to\infty$, one must have $\bar{Z}(0)=0$. We now check that the fluid limit also satisfies the fluid model equation (9). As discussed in Section 4.2, it is enough to check that the fluid limit satisfies (11). Consider a VOQ$_{ij}$ and a time $t\ge 0$. Suppose that $\bar{Z}_{ij}(t)>0$. By the continuity of $\bar{Z}$, there exists a $\delta>0$ such that $\min_{t'\in[t,t+\delta]}\bar{Z}_{ij}(t')>0$. Set $a=\min_{t'\in[t,t+\delta]}\bar{Z}_{ij}(t')$. Thus, for large enough k, we have

$$\bar{Z}_{ij}^{r_{n_k}}(t')\ge a/2 \quad \text{for } t'\in[t,t+\delta] \quad \text{and} \quad r_{n_k}a/2\ge 1.$$

Thus,

$$Z_{ij}(t')\ge 1 \quad \text{for } t'\in[r_{n_k}t, r_{n_k}(t+\delta)]. \tag{25}$$

Equation (25) says that, for a large time interval $[r_{n_k}t, r_{n_k}(t+\delta)]$, the VOQ$_{ij}$ has at least one packet in it. We have, for each $t'\in[t,t+\delta]$,

$$0\le D_{ij}(r_{n_k}t')-D_{ij}(r_{n_k}t)-\sum_{\pi\in\Pi}\pi_{ij}\Big(T_\pi(r_{n_k}t')-T_\pi(r_{n_k}t)\Big)\le 1.$$

Dividing each side by r_{n_k} and letting $k \to \infty$, one has (11).

Finally, because of (20), each fluid limit $(\bar{D}, \bar{T}, \bar{Z})$ is Lipschitz continuous and therefore is absolutely continuous.

A.2 Proof of Theorem 3

Assume that the fluid model is weakly stable. Recall (Definition 6) that this means $Z_{ij}(0) = 0$ and $Z_{ij}(t) = 0$ for $t > 0$. By Section A.1, for each ω satisfying (1), $\{\bar{D}^r(\cdot, \omega), r > 0\}$ is tight as $r \to \infty$, and the fluid limit $(\bar{D}, \bar{Z})$ is uniquely given by $\bar{Z}_{ij}(t) = 0$ for $t \geq 0$. Using this in (8), we get that $\bar{D}_{ij}(t) = \lambda_{ij} t$ for $t \geq 0$.

Thus,

$$\bar{D}^r_{ij}(t, \omega) \to \lambda_{ij} t$$

u.o.c. as $r \to \infty$. In particular, $\bar{D}^r_{ij}(1, \omega) \to \lambda_{ij}$ as $r \to \infty$ or

$$\lim_{r \to \infty} \frac{D_{ij}(r, \omega)}{r} = \lambda_{ij}.$$

Restricting r to the integers on the left hand side yields (2), thus proving the theorem.

A.3 Proof of Lemma 2

We prove the following lemma, which implies Lemma 2 via the fluid limit procedure.

Lemma 3. *A switch employing a maximal matching algorithm at a speedup of s possesses the following property: If $Z_{ij}(n) \geq s$, then*

$$\begin{aligned} C_{ij}(n+1) - C_{ij}(n) \quad &\leq \quad \sum_{j'} A_{ij'}(n+1) - A_{ij'}(n) \\ &\quad + \sum_{i'} A_{i'j}(n+1) - A_{i'j}(n) - s. \end{aligned} \qquad (26)$$

Proof. Let V_{ij} denote the set of all VOQs holding packets at input i or for output j. Then $C_{ij}(n+1) - C_{ij}(n)$ is the difference in the number of arrivals at time $n+1$ to V_{ij} and the number of departures from V_{ij} at time n. The number of arrivals to V_{ij} at time $n+1$ equals $\left(\sum_{j'} A_{ij'}(n+1) - A_{ij'}(n)\right)$ $+ \left(\sum_{i'} A_{i'j}(n+1) - A_{i'j}(n)\right)$.

Since $Z_{ij}(n) \geq s$ and at most one packet may be removed from input i in each phase of the n^{th} time slot, $Z_{ij}(n + \frac{k}{s}) > 0$ for $1 \leq k \leq s$. As the switch employs a maximal matching algorithm,

$$\pi_{ij}^m(n + \frac{k}{s}) + Z_{ij'}(n + \frac{k}{s})\, \pi_{ij'}^m(n + \frac{k}{s}) + Z_{i'j}(n + \frac{k}{s})\, \pi_{i'j}^m(n + \frac{k}{s}) > 0$$

for some $j' \neq j$ and $i' \neq i$.

Therefore, during each phase k of time n, either input i transfers a packet to some output or output j receives a packet from some input. In either case, at least one packet is removed from a VOQ in the set V_{ij} during each phase of time n. Since there are s phases, the number of departures from V_{ij} is at least s and we get the bound on the right-hand-side of (26). $\square$