The throughput of data switches with and without speed up

By
J.G. Jim Dai, Balaji Prabhakar

Presented by
Mohammad Hosein Yarmand

ECE 731
December 2006
IQ vs. OQ crossbar switches

- In input-queued (IQ), buffers which queue packets at the inputs need only run twice as fast the line rates.

- 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 up to two transactions per time slot: (1) write in an incoming packet, and (2) copy a buffered packet onto the crossbar fabric.

- The buffers of an N x N output-queued (OQ) switch are required to run at least N + 1 times the line rate.
IQ vs. OQ crossbar switches (cont)

- 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.

- 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 [12].

- 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.
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 up to 100% by using “virtual output queuing” and suitable packet scheduling algorithms.

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. ([13], [14], [15])

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. ([13], [14], [15])
First result of paper

- These results 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.

- 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 up to 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.
Matching \[^{[16]}\]

- Given a graph $G = (V,E)$, a **matching $M$** in $G$ is a set of pairwise non-adjacent edges; that is, no two edges share a common vertex.

- A **maximum matching** is a matching that contains the largest possible number of edges. There may be many maximum matchings.

- A **maximal matching** is a matching $M$ of a graph $G$ with the property that if any edge not in $M$ is added to $M$, it is no longer a matching. In other words, a matching $M$ of a graph $G$ is maximal if every edge in $G$ has non-empty intersection with at least one edge in $M$. Note that every maximum matching must be maximal, but not every maximal matching must be maximum.
The strong law of large numbers

The strong law of large numbers states that if $X_1, X_2, X_3, \ldots$ is an infinite sequence of random variables that are pairwise independent and identically distributed with $E(|X_i|) < \infty$ (and where the common expected value is $\mu$), then

$$\lim_{n \to \infty} \bar{X}_n = \mu$$

i.e., the sample average converges almost surely to $\mu$. 

[17]
Switches with Speed up

- With a speedup of 4 or 5 one can achieve up to 100% throughput when arrivals are i.i.d. at each input, and the distribution of packet destinations is uniform across the outputs. ([2], [3], [4], [5], [6])

- It is possible to remove the i.i.d. restrictions on the input traffic patterns.

- There are 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. ([7], [8], [9])

- All input traffic patterns and switch sizes are allowed

- CIOQ switch delivers a throughput of 100%
Second result of paper

- how well a CIOQ switch that employs an arbitrary, but well-chosen, scheduling algorithm performs as its fabric speedup is increased.

- When the speedup of a CIOQ switch is at least 4 and the input traffic is leaky bucket constrained, any maximal matching algorithm delivers 100% throughput. ([10], [11])

- 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.
Leaky Bucket [16]

- Typically, the algorithm is used to control the rate at which data is injected into a network, smoothing out "burstiness" in the data rate.
Third result of paper

- 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 network under very mild assumptions on the input traffic.

- 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.
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.
Consider an N x N crossbar switch such as the following Figure. 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 $[n-1,n)$, $n = 1,2,\ldots$
The buffer at an input is partitioned into $N$ “virtual output queues” (VOQs), each of infinite capacity. The virtual output queue VOQij holds packets arriving at input i destined for output j.

A “scheduling cycle” (or phase) 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 VOQij is non-empty.
A matching may be represented by a permutation matrix \( \pi \). Let \( \Pi \) 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.

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.
Definition 1: A matching algorithm $m$ is a specification of a sequence of permutations $\{\pi_{ij}^m(n + \frac{k}{\delta})\}_{n,k}$, where $\pi_{ij}^m(n + \frac{k}{\delta})$ 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 $n$. $A_{ij}(n)$ is the cumulative number of packets that have arrived at VOQij by time $t$. 
Model and notation (cont)

- We assume that the arrival processes satisfy a strong law of large numbers (SLLN): with probability one,

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

We call \( \lambda_{ij} \) the arrival rate at VOQij.

- This assumption very mild. It is satisfied, for example, when the arrival processes are jointly stationary and ergodic with arrival rates \( \lambda_{ij} \).
Model and notation (cont)

Let $D_{ij}(n)$ be the number of departures from $VOQ_{ij}(n)$ 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,

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

(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

\[
\sum_i \lambda_{ij} \leq 1, \quad \sum_j \lambda_{ij} \leq 1.
\]  

(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

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 \(n - 1\).
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.

There is no need for output buffers and 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 \( \pi \) 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, \( w \),

\[
\pi^w(n) = \arg \max_{\pi} \{ f_{\pi}(n) \}.
\]  \hspace{1cm} (4)

Let \( f(n) = f_{\pi^w(n)}(n) \) be the weight of the maximum

Theorem 1: A maximum weight matching algorithm is efficient.
Definition 5: A matching algorithm \( \pi \) 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. \( Z_{ij'}(n + \frac{k}{s}) \pi_{ij'}(n + \frac{k}{s}) > 0 \)
2. \( Z_{i'j}(n + \frac{k}{s}) \pi_{i'j}(n + \frac{k}{s}) > 0 \),

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 $s \geq 2$. 
Suppose the switch employs some (yet to be specified) matching algorithm $m$. For a $\pi \in \Pi$, let $T^m_\pi(n)$ be the cumulative amount of time that permutation $\pi$ has been used by time slot $n$. Again, we assume $T^m_\pi(0) = 0$. The following equations of evolution hold for the switch: for $n \geq 0$ and $i, j = 1, \ldots, N$,

$$Z_{ij}(n) = Z_{ij}(0) + A_{ij}(n) - D_{ij}(n),$$
Fluid models – switch dynamics (cont)

\[ D_{ij}(n) = \sum_{\pi \in \Pi} \sum_{\ell=1}^{n} \pi_{ij} 1\{Z_{ij}(\ell)>0\} (T^{m}_\pi(\ell) - T^{m}_\pi(\ell - 1)), \]

\( T^{m}_\pi(\cdot) \) is non-decreasing, and \( \sum_{\pi \in \Pi} T^{m}_\pi(n) = n. \)

The first equation tracks the evolution of \( Z_{ij} \) in terms of the total number of arrivals at and departures from \( \text{VOQ}_{ij} \).

The second equation keeps a count of the cumulative number of departures from \( \text{VOQ}_{ij} \). And the third equation expresses the fact that input \( i \) is matched to some output or the other at each time.
Fluid models – Fluid equations

\[ Z_{ij}(t) = Z_{ij}(0) + \lambda_{ij} t - D_{ij}(t) \geq 0, \quad (5) \]

\[ \dot{D}_{ij}(t) = \sum_{\pi \in \Pi} \pi_{ij} T^m_{\pi}(t), \text{ if } Z_{ij}(t) > 0, \quad (6) \]

\[ T^m_{\pi}(\cdot) \text{ is nondecreasing, and } \sum_{\pi \in \Pi} T^m_{\pi}(t) = t, \quad (7) \]

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 be differentiable at \( t \).
Equation (6) has an equivalent characterization: Whenever $Z_{ij}(t) > 0$, there exists a $\delta > 0$ such that

$$D_{ij}(t') - D_{ij}(t) = \sum_{\pi \in \Pi} \pi_{ij}(T^m_\pi(t') - T^m_\pi(t)), \quad t' \in [t, t+\delta].$$

Equation (8) 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 $[t, t']$. 
to matching algorithm $m$. For example, if $m$ equals $w$, the maximum weight matching algorithm, the additional fluid equation takes the form: for each $\pi \in \Pi$,

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

(9)

The above equation says that under the maximum weight matching algorithm, a matching $\pi$ which has weight less than another matching $\pi'$ at some time $t$ will not be employed at that time. Thus, equation (9) characterizes the
**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.
Proof of Theorem 1

Lemma 1: Let $f : [0, \infty) \rightarrow [0, \infty)$ be an absolutely continuous function with $f(0) = 0$. Assume that $f(t) \leq 0$ 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 of Theorem 1 (cont)

Let \((D, T, Z)\) be a fluid model solution satisfying (5)-(7) and (9) with \(Z(0) = 0\). For a permutation matrix \(\pi\), define \(f_\pi(t) = \langle \pi, Z(t) \rangle\). Let \(f(t) = \max_\pi f_\pi(t)\). Let \(\lambda\) be the \(N \times N\) matrix with entries \(\lambda_{ij}\). It is well-known that under condition (3)

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

Briefly, this is because under condition (3) \(\lambda\) is doubly substochastic 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.
Proof of Theorem 1 (cont)

Let $t$ be a fixed value such that $f$ and $Z$ are differentiable at $t$. Let $\Pi'$ be the set of matchings $\pi$ 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 (9),

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

It follows that

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

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

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

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

$$= f(t).$$
\[ \langle Z(t), \dot{Z}(t) \rangle = \langle Z(t), \lambda \rangle - \langle Z(t), \dot{D}(t) \rangle \]
\[ = \langle Z(t), \lambda \rangle - f(t) \]
\[ \leq 0. \]

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.
Proof of Theorem 2

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

$$\dot{C}_{ij}(t) \leq \sum_{j'} \lambda_{ij'} + \sum_{i''} \lambda_{i''j} - s \quad \text{whenever } Z_{ij}(t) > 0.$$
Proof of Theorem 2 (cont)

itive 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.
Proof of Theorem 2 (cont)

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

$$C(t) = QZ(t) + Z(t)Q \quad t \geq 0. \quad (11)$$

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
Proof of Theorem 2 (cont)

\[ \dot{f}(t) = 2(Z(t), \dot{C}(t)). \quad (12) \]

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 (10). Therefore \( \dot{f}(t) \leq 0 \) whenever \( f(t) > 0 \), proving Theorem 2.
References

References (cont)


References (cont)

[17] www.mathworld.com