Performance analysis of large multicast switches with multicast virtual output queues

Min Song\textsuperscript{a,}*\textsuperscript{-}, Weiying Zhu\textsuperscript{a}, Andrea Francini\textsuperscript{b}, Mansoor Alam\textsuperscript{c}

\textsuperscript{a}Department of Electrical and Computer Engineering, Old Dominion University, 231 Kaufman Hall, Norfolk, VA 23529, USA
\textsuperscript{b}Data Networking and Wireless Systems Department, Bell Laboratories—Lucent Technologies, Murray Hill, NJ, USA
\textsuperscript{c}EECS Department, The University of Toledo, 2801 Bancroft Street, Toledo, OH 43606, USA

Received 13 January 2004; revised 23 August 2004; accepted 14 September 2004

Abstract

In multicast switches, the accommodation of multicast traffic in multiple queues per input buffer reduces the throughput degradation caused by head-of-line (HOL) blocking. Such an arrangement, called multicast virtual output queuing (MC-VOQ), is very promising in theory but can only be implemented in practice with heavy approximation. Complete avoidance of the HOL blocking problem would in fact require a distinct queue for each fanout set possible, leading to an exponential growth of the number of queues needed with the switch size. If only a limited number of queues can be used per input buffer, criteria must be identified for setting the number of queues, for associating the fanout sets with the individual queues, and for scheduling the transmission of packets out of the queues.

This paper presents an analytical model for the investigation of saturation throughput and packet delay in MC-VOQ multicast switches. The model relies on the assumption of Poisson-distributed uniform input traffic and random queuing and scheduling policies. Extensive simulation experiments validate the results of the analysis for large switch sizes.

Keywords: Performance analysis; Multicast; Packet switch; Delay analysis; Input queuing; Queuing theory

1. Introduction

As the Internet has evolved from a convenience to a mission-critical platform for conducting research, education, and business, the spread of multicast applications such as distributed interactive simulations, distance learning, and video-on-demand is rapidly growing. As a result, there is an increasing demand for high-performance packet switches that efficiently support multicast traffic.

The distinguishing feature of a multicast switch is its ability to deliver a multicast packet to multiple output ports. A multicast packet switch must be capable of replicating packets, so that all destinations of a multicast packet can receive their respective copies. We focus on crossbar-based switch fabrics, because of their intrinsic multicast capabilities [3,7]. We only address crosbars with speedup one (in a crossbar, the speedup is the capacity ratio between the fabric ports and the switch ports).

While virtual output queuing is a widely accepted queuing scheme for unicast traffic, it is not practical for multicast switches in its exhaustive form, which offers one queue per possible fanout set [5,10]. One alternative queuing scheme for multicast switches consists in the allocation of a single FIFO (first-in first-out) queue at each input port to handle all multicast traffic that transits through the port [9]. The performance of such multicast switches was studied by Hayes et al. [4] and Hui [5]. While the former focuses on small-sized switches and the latter focuses on large-sized switches, both studies assume one FIFO queue at each input port and the random packet selection policy with fanout splitting. The performance of multicast switches with one input queue per input buffer with no fanout splitting was studied in [1,8]. Although the single-queue scheme can be easily implemented due to its
simplicity, head-of-line (HOL) blocking heavily impairs its performance.

To overcome HOL blocking, window-based queuing was introduced [2], allowing packets behind the HOL one to be transmitted prior to the HOL packet. The increased scheduling space results in an approximate 10% throughput increase on average. A major price paid for this enhancement is the increased complexity of the input queues, which must have random access capabilities. Another approach to alleviate HOL blocking consists in the allocation of a plurality of multicast queues at each input port, so that the packet scheduler can choose among multiple HOL packets. Simulation studies for such multicast switches with non-exhaustive multicast virtual output queuing (MC-VOQ) were performed by Gupta and Aziz [3], Song et al. [10], and Bianco et al. [11]. Particularly interesting are the results presented in [11], which indicate that the number of input queues, the queuing policy that couples the destination masks with the available input queues, and the scheduling policy that sets the service order over the available HOL packets are all critical to the throughput performance of the switch. All such indications are obtained under gathered traffic conditions, i.e. under traffic scenarios where only a small number of input ports supply multicast traffic to the switch. Uniform distribution of the traffic feeds over all input ports makes the performance impact of different design choices almost irrelevant.

The contribution of this paper consists of a first step towards the analytical validation and possible generalization of the simulation results presented in [11]. We provide analytical characterization of the behavior of a large multicast switch when Poisson traffic is uniformly distributed over all input ports. Our results confirm that under non-gathered traffic conditions a small number of multicast queues is sufficient to maximize the saturation throughput, even if low-end queuing and scheduling policies are deployed. We show simulation results that match the analytical ones for reasonably large switch sizes. Our next step, to be presented in a follow-up paper, will be the extension of the model to the case of gathered traffic.

The rest of the paper is organized as follows. Section 2 illustrates the multicast switch architecture used in this paper. Section 3 introduces the initial model and a modified equivalent model. The saturation throughput analysis is given in Section 4. In Section 5, we derive the packet delay and service time. The theoretical and experimental results are jointly presented in Section 6. Finally, conclusions are given in Section 7.

2. Multicast switch architecture

Fig. 1 shows the \(N \times N\) multicast packet switch studied in this paper. The switch size \(N\) is assumed to be a large number. At each input port, there are \(M\) FIFO queues dedicated to multicast traffic. A multicast packet arriving at the input interface is first queued into one of the \(M\) multicast queues and then switched from the input port to its target output ports. To preserve the order of packet delivery on the output links, packets of the same flow get queued at the same FIFO. The switch fabric is a bufferless crossbar with speedup one, i.e., at each time slot, each input can transmit no more than one packet and each output can receive no more than one packet. However, it is obviously possible that multiple copies of the same HOL multicast packet are forwarded to different output ports during the same time slot. In essence, the service discipline is based on fanout splitting [9]. The random scheduling policy [11] resolves contention at the outputs. When more than one HOL packet contend for the same output, one of them is selected randomly with equal probability.

We focus on saturation throughput and delay performance. We define throughput as the average number of packets delivered to an output per time slot. Delay is the number of time slots between the arrival and departure times of a packet at its assigned input queue.

3. Modeling

3.1. Initial model

Some assumptions need to be made before any further discussion. The incoming multicast traffic at each input link consists of a mix of multicast Poisson-distributed flows. The traffic arrival rates at the input links are the same for all inputs. We logically organize the input multicast queues in groups: all the \(k\)th queues across all input ports belong to group \(Q^k\). Each multicast flow is randomly associated with one of the \(M\) queues with equal probability: packets of the same multicast flow always get queued to the same input queue and serviced in the same order they came in.
The aggregate traffic at each input port still follows the Poisson process.

All packets are assumed to have equal size: the time slot coincides with the time needed to transfer one packet across the crossbar fabric. Each packet is destined for a random number of output ports. The average number of destinations per new arriving packet (this number is termed fanout) is assumed to be much smaller than \( N \). A packet has fanout \( f \) with probability \( r_f \). The \( f \) destinations are assumed distinct and uniformly distributed among the \( N \) output ports. Thus, the Poisson characterization and the uniform distribution properties still apply to each queue.

More than one HOL packets are likely to be available at a single input port. Due to the assumption of speedup one, only one of them can be selected for transmission in one time slot. Input contention occurs in this case. It is solved by matching input and output ports in \( M \) subsequent rounds during each time slot. In the \( m \)th \((1 \leq m \leq M)\) round, one of the groups that are still un-served is randomly chosen with equal probability. The HOL packets in queues of the selected group are considered for service during this round.

The HOL packets of the selected group at the available inputs send requests to all its residual destinations. Once an available output receives the requests, it randomly grants one of them. As a result, the input receiving the grant and the output initiating the grant get matched. An input (output) port is considered available if it was not matched as of the previous round within the same time slot. A HOL packet is removed from the input queue when its copies are transmitted to all its target output ports. The number of time slots spent by a packet at the head of its queue is called the service time of the HOL packet.

In the initial model above, we assume that an observer samples the state of the \( N \times M \) queues at the beginning of each time slot. The observer sees \( N \times M M/G/1 \) queues with identical statistical properties. Because all the input queues served in the \( m \)th round are subject to statistically identical arrival and service processes, the number of packets counted by the observer in any one of the queues has identical statistical properties as in all other queues. We define this number as the queue length \( N_m \) of the \( m \)th round. Similarly, all the queues in \( Q^k \) are also statistically identical. The number of packets counted by the observer in any of the \( Q^k \) queues is defined as the queue length \( L_k \) of the \( k \)th group. In the initial model, there is no fixed relationship between the queues served in the \( m \)th round and the queues in group \( Q^m \).

Because of the adoption of the random queuing and random serving policies, all groups have identical statistical properties. Therefore, for any \( i \in [1,2,\ldots,N] \), we have

\[
E[L_i] = E[L] = L,
\]

and

\[
\Pr(\text{group } Q^k \text{ is served in the } m \text{th round}) = \frac{1}{M},
\]

where \( L \) is the average queue length, identical for all queues.

Consequently, we have

\[
E[N_m] = \sum_{k=1}^{M} (\Pr(\text{group } Q^k \text{ is served in the } m \text{th round}) \times E[L_k]) = L.
\]

Thus, the average queue lengths during any two rounds are always identical.

### 3.2. Modified model

We now define a model that is logically equivalent to the initial one but much easier to analyze. In this modified model, we set a fixed relationship between the queues that are served in the \( m \)th round and the queues in group \( Q^m \); the queues in \( Q^m \) are always served in the \( m \)th round. As a result, the frequency at which queues in group \( Q^m \) are served is obviously higher than the frequency of service of queues in group \( Q^j \) if \( i < j \). In order for the queue statistics to remain identical over all groups, the arrival rates of multicast packets to queues of different groups must be redefined accordingly. However, the explicit derivation of the rates of the new Poisson arrival processes is not required, because such rates are irrelevant to the completion of the analysis. The main element of relevance remains the statistical identity and independence of the \( N \times M M/G/1 \) queues that compose the model.

Our analysis of the saturation throughput and delay performance of the multicast switch will be based on the modified model. The theoretical results will be validated through simulation of the initial model.

### 4. Saturation throughput analysis

The following notation holds for the \( m \)th round.

- \( \lambda_{m\text{-input}} \): packet arrival rate for a queue in \( Q^m \).
- \( N_m^j \): number of HOL packets from the available inputs in \( Q^m \) whose residue destinations contain output \( j \).
- \( N_m^f \): value of \( N_m^j \) in the next time slot.
- \( q_m \): probability that one destination of the HOL packet in \( Q^m \) is serviced in each time slot.
- \( q_m^c \): conditional probability that one destination of the HOL packet in \( Q^m \) is serviced in each time slot given that this input port is available for the \( m \)th round.
- \( X_m \): service time of the HOL packet in \( Q^m \).
- \( T_m \): delay of a packet that transits in a queue of \( Q^m \).
- \( A_m^m \): number of HOL packets arriving in \( Q^m \) at the current time slot from the available inputs whose set of destinations contains output \( j \). For a large \( N \), the distribution of \( A_m^m \) converges to a Poisson distribution.
4.1. The first round

Each destination of the HOL packets in \( Q^1 \) is served independently with identical probability \( q_1 \), across the inputs as well as from slot to slot. The throughput in the first round \( (\lambda_1) \) is defined as the average number of packets delivered to an output per time slot in the first round. Therefore, \( \lambda_1 = E(N_j^1) \), where the indicator function \( \epsilon(x) = 1 \) if \( x > 0 \) and \( \epsilon(x) = 0 \) if \( x = 0 \). It is obvious that in the first round, the system behavior is exactly the same as the behavior of a multicast switch with single input queue. Therefore, the following equations from [5] can be invoked to express the maximal value of \( \lambda_1 \) (the contribution of the first round to the saturation throughput):

\[
q_1 = \frac{2(1 - \lambda_1)}{2 - \lambda_1}, \quad (4)
\]

\[
E[X_1] = \sum \left( r_f \left( \sum_{k=0}^{j} \binom{j}{k} \frac{(-1)^{k+1}}{1 - (1 - q_1)^k} \right) \right), \quad (5)
\]

\[
\sum j f r_f = \lambda_1 E[X_1], \quad (6)
\]

4.2. The second round

The destinations of the new arriving HOL packets at the current time slot from the available inputs are randomly distributed over all the output ports. The distributions are uniform and independent of one another. For a large \( N \), the distribution of \( A_j^2 \) converges to a Poisson distribution. The throughput in the second round \( (\lambda_2) \) coincides with the average number of packets delivered to an output per time slot in the second round. Note that output \( j \) is matched in the second round if and only if it is not matched in the first round and there is at least one HOL packet in available queues of \( Q^2 \) that are destined for it. Therefore, \( \lambda_2 = E[[1 - \epsilon(N_j^2)]\epsilon(N_j^2)] \). The dynamic equation for output \( j \) is

\[
N_j^2 \lambda_2 = [1 - \epsilon(N_j^2)]\epsilon(N_j^2) + A_j^2. \quad (7)
\]

Our first objective is to find the equation for \( q_2 \). To this end, we define \( q_2^2 \) as the conditional probability that one destination of the HOL packet in the second queue of an input port gets serviced in each time slot given that this input port is available. Based on the relationship between un-conditional probability and conditional probability, \( q_2 = P[\text{the input is available for the second round}] \times q_2^2 \). Since

\[
P[\text{the input is available for the second round}]
\]

\[
= \frac{\text{Average number of available inputs for the 2nd round}}{\text{Number of inputs}}
\]

\[
= \frac{E[N - \sum_{j=1}^{N} \epsilon(N_j^1)]}{N} = \frac{N - NE[\epsilon(N_j^1)]}{N} = 1 - \lambda_1,
\]

and \( q_2^2 \rightarrow \frac{E[[1 - \epsilon(N_j^1)]\epsilon(N_j^2)]}{E[N_j^2]} \) as \( N \rightarrow \infty \), we have

\[
q_2 = (1 - \lambda_1)q_2^2 = (1 - \lambda_1)\frac{E[[1 - \epsilon(N_j^1)]\epsilon(N_j^2)]}{E[N_j^2]]. \quad (8)
\]

For the steady-state system, through normalization of both sides of (7):

\[
E[A_j^2] = E[[1 - \epsilon(N_j^1)]\epsilon(N_j^2)] = \lambda_2. \quad (9)
\]

Recalling the definition of \( \epsilon(x) \), we have \( (\epsilon(x))^2 = \epsilon(x) \) and \( x \times \epsilon(x) = x \). Therefore

\[
((1 - \epsilon(N_j^1))\epsilon(N_j^2))^2 = [1 - \epsilon(N_j^1)]\epsilon(N_j^2), \quad (10)
\]

\[
N_j^2[1 - \epsilon(N_j^2)]^2 \epsilon(N_j^2) = [1 - \epsilon(N_j^1)]N_j^2. \quad (11)
\]

Because of the assumption of Poisson distribution of \( A_j^2 \), we also obtain

\[
E[(A_j^2)^2] = (E[A_j^2])^2 + E[A_j^2]. \quad (12)
\]

Then, by squaring both sides of (7), substituting (9)–(12), and normalizing both sides, we have

\[
E[N_j^2] = \frac{(2 - \lambda_2)\lambda_2}{2(1 - \lambda_1 - \lambda_2)}. \quad (13)
\]

Substituting (9) and (13) into (8), \( q_2 \) can be expressed as follows

\[
q_2 = \frac{2(1 - \lambda_1)(1 - \lambda_1 - \lambda_2)}{(2 - \lambda_2)}. \quad (14)
\]

Based on the analysis result in [5],

\[
E[X_2] = \sum \left( r_f \left( \sum_{k=0}^{j} \binom{j}{k} \frac{(-1)^{k+1}}{1 - (1 - q_2)^k} \right) \right), \quad (15)
\]

Based on the P-K formula [5,6], the average delay for a packet transiting in a queue of \( Q^2 \) can be expressed as

\[
E[T_2] = E[X_2] + \frac{\lambda_2 \text{input} E[X_2^2]}{2(1 - \lambda_2 \text{input} E[X_2])}. \quad (16)
\]

We can see that the average delay becomes infinite when \( 1 = \lambda_2 \text{input} E[X_2] \). Considering that \( \lambda_2 = \lambda_2 \text{input} \sum f r_f \), we can express the second round contribution to the saturation throughput as follows:

\[
\sum \left( r_f \left( \sum_{k=0}^{j} \binom{j}{k} \frac{(-1)^{k+1}}{1 - (1 - q_2)^k} \right) \right). \quad (17)
\]
Table 1
Saturation throughput with two input queues

<table>
<thead>
<tr>
<th>$f$</th>
<th>1</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda_1$</td>
<td>0.586</td>
<td>0.695</td>
<td>0.779</td>
<td>0.849</td>
<td>0.868</td>
</tr>
<tr>
<td>$\lambda_2$</td>
<td>0.127</td>
<td>0.092</td>
<td>0.068</td>
<td>0.047</td>
<td>0.041</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>0.713</td>
<td>0.787</td>
<td>0.847</td>
<td>0.896</td>
<td>0.910</td>
</tr>
</tbody>
</table>

4.3. The $m$th round

In this section, we generalize the throughput analysis to a multiclass switch with $M$ queues per input buffer. We focus on the throughput and service time at the $m$th round ($2 \leq m \leq M$). The throughput in the $m$th round ($\lambda_m$) is defined as the average number of packets delivered to an output per time slot in the $m$th round. Note that output $j$ is matched in the $m$th round if and only if it is not matched in the previous rounds and there is at least one HOL packet in available queues of $Q^m$ that are destined for it. Therefore

$$\lambda_m = E \left[ \prod_{k=1}^{m-1} (1 - \epsilon(N_j^k)) \epsilon(N_j^m) \right].$$

The total throughput is the sum of the throughputs achieved in $M$ rounds, $\lambda = \sum_{m=1}^{M} \lambda_m$. During each round, the switch could be modeled as $N$ identical and independent $M/G/1$ queues, where each queue is associated with a distinct switch output. The dynamic equation for output $j$ is

$$N_j^m = N_j^{m-1} - \prod_{k=1}^{m-1} (1 - \epsilon(N_j^k)) \epsilon(N_j^m) + A_j^m. \quad (18)$$

Taking the average of both sides of (18) and considering that for the steady-state system $E[N_j^m] = E[N_j^m]$, we have

$$E[A_j^m] = E \left[ \prod_{k=1}^{m-1} (1 - \epsilon(N_j^k)) \epsilon(N_j^m) \right] = \lambda_m. \quad (19)$$

Based on the assumption of randomly granting to one of the requests at each output, we have

$$\Pr\{\text{the input is available for the } m\text{th round}\} = \frac{\text{Average number of available inputs for the } m\text{th round}}{\text{Number of inputs}}$$

$$= \frac{E[N - \sum_{k=1}^{m-1} \left( \sum_{j=1}^{N} \prod_{k=1}^{m-1} (1 - \epsilon(N_j^k)) \epsilon(N_j^k) \right)]}{N}$$

$$= \frac{N - N \sum_{k=1}^{m-1} E \left[ \prod_{k=1}^{m-1} (1 - \epsilon(N_j^k)) \epsilon(N_j^k) \right]}{N}$$

$$= \frac{1 - \sum_{k=1}^{m-1} \lambda_k}{N}. \quad (20)$$

Thus

$$q_m = \left( 1 - \sum_{k=1}^{m-1} \lambda_k \right) E \left[ \prod_{k=1}^{m-1} (1 - \epsilon(N_j^k)) \epsilon(N_j^m) \right] \quad (21)$$

Similar to (12), we have

$$E[(A_j^m)^2] = E[A_j^m] + E[A_j^m]. \quad (22)$$

Taking into account the definition of $\epsilon$, we have

$$N_j^m \prod_{k=1}^{m-1} (1 - \epsilon(N_j^k)) \epsilon(N_j^m) = \prod_{k=1}^{m-1} (1 - \epsilon(N_j^k)) N_j^m. \quad (23)$$

Since $m \ll N$, the $N_j^k$ values ($1 \leq k \leq m$) are independent of each other. We can then derive the following equation

$$E \left[ \prod_{k=1}^{m-1} (1 - \epsilon(N_j^k)) \right] = 1 - \sum_{k=1}^{m-1} \lambda_k, \quad (24)$$

proved in Appendix. In addition, we have

$$\left( \prod_{k=1}^{m-1} (1 - \epsilon(N_j^k)) \right)^2 = \prod_{k=1}^{m-1} (1 - \epsilon(N_j^k)) \epsilon(N_j^k). \quad (25)$$

Combining this result with (19), (21)–(23), and squaring and normalizing both sides of (18), we obtain:

$$E[N_j^m] = \frac{(2 - \lambda_m) \lambda_m}{2(1 - \sum_{k=1}^{m-1} \lambda_k)}. \quad (26)$$

Substituting (19) and (24) into (20), we reach the following result:

$$q_m = \frac{2(1 - \sum_{k=1}^{m-1} \lambda_k) (1 - \sum_{k=1}^{m} \lambda_k)}{(2 - \lambda_m)} \quad (27)$$

Similar to (15), the average service time in the $m$th round is given by

$$E[T_m] = E[X_m] + \frac{\lambda_m \text{-input } E[X_m]^2}{2(1 - \lambda_m \text{-input } E[X_m])}. \quad (28)$$

Based on the $P$–$K$ formula [5,6], the average delay for a packet transiting in a queue of $Q^m$ can be expressed as

$$E[T_m] = E[X_m] + \frac{\lambda_m \text{-input } E[X_m]^2}{2(1 - \lambda_m \text{-input } E[X_m])}. \quad (29)$$

We can see that the average delay for a queue in the $m$th round becomes infinite when $1 = \lambda_m \text{-input } E[X_m]$, where $\lambda_m = \lambda_m \text{-input } E[X_m]$. Thus, we have the following equation to express the saturation throughput in the $m$th round

$$\sum_{f} (r_f f) = \lambda_m \sum_{f} \left( \frac{r_f}{k+1} \right) \frac{1}{1 - (1 - \lambda_m f)^2} \quad (30)$$
where $q_m$ is given by (25) and all the $\lambda_k$ terms ($1 \leq k \leq m - 1$) are calculated during the previous rounds. The theoretical results of the saturation throughput for a multicast switch with three queues and a constant fanout of $f$ are listed in Table 2, where $\lambda = \lambda_1 + \lambda_2 + \lambda_3$.

### Table 2
Saturation throughput with three input queues

<table>
<thead>
<tr>
<th>$f$</th>
<th>1</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda_1$</td>
<td>0.586</td>
<td>0.695</td>
<td>0.779</td>
<td>0.849</td>
<td>0.868</td>
</tr>
<tr>
<td>$\lambda_2$</td>
<td>0.127</td>
<td>0.092</td>
<td>0.068</td>
<td>0.047</td>
<td>0.041</td>
</tr>
<tr>
<td>$\lambda_3$</td>
<td>0.066</td>
<td>0.048</td>
<td>0.035</td>
<td>0.024</td>
<td>0.021</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>0.779</td>
<td>0.835</td>
<td>0.882</td>
<td>0.921</td>
<td>0.931</td>
</tr>
</tbody>
</table>

5. Delay analysis

We use the following notation in the analysis of the delay performance:

- $\lambda_{\text{input}}$: offered load at an input link.
- $T_m$: delay of a packet in any queue.
- $N_m$: the queue length of the queue in $Q^m$.

When the overall packet arrival rate is less than the saturation packet arrival rate, i.e. before the system delay becomes infinite, the queuing system is stable and the following condition holds:

$$\lambda_{\text{input}} = \sum_{m=1}^{M} \lambda_{m-\text{input}}.$$  \hfill (29)

According to [5], we have

$$E[X_m^2] = \sum_{f} \left( r_f \left( \sum_{k=1}^{f} \binom{f}{k} \left( \frac{-1}{2} \right)^{f-1} \frac{2(1-q_m^k)}{(1-q_m^k)^2} \right) \right) + E[X_m],$$  \hfill (30)

where $q_m$ and $E[X_m]$ are given by (25) and (26), respectively.

According to Little’s theorem, we obtain

$$E[N_m] = \lambda_{m-\text{input}} E[T_m].$$  \hfill (31)

As stated early, the average queue lengths of any two queues in the modified model are all identical:

$$E[N_1] = E[N_2] = \ldots = E[N_M].$$  \hfill (32)

We also have $\lambda_m = \lambda_{m-\text{input}} \sum_f (r_f)$. Therefore, combining (25)–(27) and (29)–(32), $\lambda_{m-\text{input}}$ and $E[T_m]$ can be calculated for a given offered load. The resulting average system delay is as follows:

$$E[T] = \frac{\sum_{m=1}^{M} \lambda_{m-\text{input}} E[T_m]}{\sum_{m=1}^{M} \lambda_{m-\text{input}}}. \hfill (33)$$

As an example, let us consider a multicast switch with three queues per input buffer ($M = 3$). Assuming that all multicast packets have a constant fanout of 4 ($f = 4$), we substitute different values of $\lambda_{\text{input}}$ ($0 < \lambda_{\text{input}} < 0.25$) into the derived equations. $E[T_m]$ ($m = 1, 2, 3$) and the average system delay $E[T]$ are calculated and plotted in Fig. 2. As we expected, $E[T_1] < E[T_2] < E[T_3]$ follows for all admissible offered load. The validity of the delay analysis will be further verified by simulations in Section 6.

6. Numerical results

Extensive simulations were performed for different switch sizes and fanouts to verify the analysis results and
to infer further conclusions. The duration of all simulation runs is one million time slots. Data are gathered for statistical elaboration during the last half million time slots. Infinite queue size is assumed to avoid packet loss. The simulation results are obtained on a system that reflects our initial model of a multicast switch. The model is implemented in the switch simulator SIM [12]. Fig. 3 shows the theoretical and simulation saturation throughput vs. fanout for multicast switches with \( M \) queues \((M=2, 3, 4)\). The simulation saturation throughput is determined as follows. The input load \( (\lambda_{\text{input}}) \) is kept increasing until achieving a maximal match size, which remain constant regardless how big the input load is. The maximal match size divided by \( N \) is then the saturation throughput for the given \( N, M \), and \( f \). The discrepancy between analysis and simulation results is always far below 2%, which confirms the accuracy of the analysis.

**Fig. 4.** Saturation throughput vs. switch size \((N)\) for switches with two input queues. Fanouts of 1, 2, 4, 8, and 10.

**Fig. 5.** Saturation throughput vs. switch size \((N)\) for switches with three input queues. Fanouts of 1, 2, 4, 8, and 10.

**Fig. 6.** Average delay vs. offered load for switches with \( M \) input queues \((M=2, 3, 4)\) and constant fanout 2.

**Fig. 7.** Average delay vs. offered load for switches with \( M \) input queues \((M=2, 3, 4)\) and constant fanout 4.

Figs. 4 and 5 show the simulation results for the saturation throughput vs. switch size in the presence of two and three input queues. The dash lines show the corresponding theoretical saturation throughputs. The purpose of these two plots is to highlight the convergence of the saturation throughput to its asymptotic value as the switch size increases, for different fanout values. One can see that the convergence is faster for smaller fanouts. The saturation throughput remains fairly constant for \( N \geq 80 \) in all scenarios. For \( N > 80 \) and bigger fanouts, the difference between simulation results and respective theoretical results under the assumption of a very big \( N \) becomes indistinguishable, hinting that the analytical results are indeed valid for practical switches that lie in the upper end \((N > 80)\) of the size scale. It should be noticed that we can only claim that the difference between the simulation saturation throughputs and theoretical saturation throughputs is very
close to each other. It is hard to justify which one is bigger than the other.

Simulated and analytical average delays are plotted in Figs. 6–8, vs. the load offered to a $256 \times 256$ switch with $M$ queues ($M = 2, 3, 4$). Each plot corresponds to a different fanout value ($f = 2, 4, 8$). Simulation results and theoretical results always agree well.

The match between the simulation data and the theoretical data shown in Figs. 3–8 testifies to the correctness of our analysis under the assumptions of our model. In addition, based on the results shown in Figs. 6–8, a number of interesting conclusions can be drawn. When the offered load is much less than the saturation arrival rate, which indicates the switch capacity, there is no obvious difference among the average delay of multicast switches with two input queues, three input queues and four input queues. However, as the offered load increases and approaches the saturation arrival rate, the average delay difference between multicast switches with different numbers of input queues becomes more and more obvious.

Furthermore, using the equations derived in Section 5, it is possible to calculate the number of input queues needed in order to meet specific delay requirements for a multicast switch where no prior knowledge of the input and output distribution of multicast traffic is available. Under the same conditions, when the number of input queues exceeds a certain threshold the gain in delay performance guaranteed by additional queues is not obvious.

Fig. 9 shows the increment of saturation throughput $\lambda_m$ ($m = 2, \ldots, 10$) by adding the $m$th queue. The insert of Fig. 9 illustrates the saturation throughput $\lambda_1$ of a multicast switch with one queue per input buffer. One can see that the bigger the $f$, the bigger the $\lambda_1$. This is because a big $f$ alleviates the HOL blocking and thus increases the throughput. Meanwhile an increasing $\lambda_1$ results in a decreasing space in the subsequent iterations. Thus, the bigger the $\lambda_1$, the smaller $\lambda_m$ ($m = 2, \ldots, 10$).

We also observe that while the saturation throughput of a multicast switch increases with the number of input queues, the benefit of additional queues becomes less and less sensible as new queues get added. In the particular case of Fig. 9, the throughput contribution of the 10th queue with two input queues, three input queues and four input queues.

<table>
<thead>
<tr>
<th>$f$</th>
<th>$\lambda$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.930</td>
</tr>
<tr>
<td>4</td>
<td>0.950</td>
</tr>
<tr>
<td>6</td>
<td>0.961</td>
</tr>
<tr>
<td>8</td>
<td>0.967</td>
</tr>
<tr>
<td>9</td>
<td>0.969</td>
</tr>
<tr>
<td>10</td>
<td>0.972</td>
</tr>
</tbody>
</table>

Table 3: Theoretical saturation throughput of a switch with nine queues and increment induced by the tenth queue.
approaches zero when the overall saturation throughput with nine queues is already above the 90% mark. For clarity, the analytical values of $\lambda = \sum_{m=1}^{9} \lambda_m$ and $\lambda_{10}$ under different fanouts are listed in Table 3.

We conclude that high throughput can be achieved in a large multicast switch with few input queues (less than 10) if multicast traffic is uniformly distributed over all inputs and outputs. To corroborate this conclusion, we plot in Fig. 10 the values of saturation throughput supplied by our modified model as the number of queues $M$ ranges from 1 to 10. The benefit of using multiple queues is initially obvious, but fades as the number of queues grows larger.

We ran the last set of simulation experiments on a 256×256 multicast switch with number $M$ of input queues equal to 1, 2, 3, 9, and 10. Figs. 11–13 show the delay as a function of the offered load for fanouts 2, 4, and 8, respectively.

Since we adopted the log-scale for the y-axis in order to show the result clearly, these curves are slightly s-shaped. The delay difference is obvious between the cases with one, two, and three queues. Conversely, the difference of delay performance becomes indistinguishable between switches with 9 and 10 multicast queues per input buffer.

7. Conclusion

Using the $M/G/1$ model and the queuing theory, we analyzed the performance of multicast switches with multiple input queues under the assumption of uniform traffic distribution over all inputs and outputs. We provided closed-form expressions for the saturation throughput, the average service time, and the average delay. We performed extensive simulations to verify the analysis. Our results show that the throughput and delay performance of the multicast switch improves as the number of queues per input buffer increases. In a large multicast switch, the performance indices converge to their asymptotic values when the number of input queues is relatively small. Thus, a small number of input queues per input buffer is sufficient to achieve sub-optimal performance in large switches with uniform distribution of multicast traffic. The extension of our analytical model to more general traffic conditions will be the topic of a follow-up paper currently in preparation.

Appendix. Proof for (23)

When $m=2$, Eq. (23) is $E[1 - e^{\lambda_m}] = 1 - \lambda_1$, which is obviously correct according to Section 3. Assuming that
Eq. (23) is correct when \( m = l \), i.e.

\[
E \left[ \prod_{k=1}^{l-1} (1 - \epsilon(N^k_l)) \right] = 1 - \sum_{k=1}^{l-1} \lambda_k.
\]

We prove that Eq. (23) is also correct when \( m = l + 1 \) as follows.

Consider we have:

\[
\lambda_1 = E \left[ \prod_{k=1}^{l-1} (1 - \epsilon(N^k_l)) \epsilon(N^l_l) \right] = E \left[ \prod_{k=1}^{l-1} (1 - \epsilon(N^k_l)) \right] E[\epsilon(N^l_l)].
\]

Then

\[
E[1 - \epsilon(N^l_l)] = 1 - \frac{\lambda_1}{E \left[ \prod_{k=1}^{l-1} (1 - \epsilon(N^k_l)) \right]}
\]

\[
= 1 - \frac{\lambda_1}{1 - \sum_{k=1}^{l-1} \lambda_k} = 1 - \frac{\sum_{k=1}^{l} \lambda_k}{1 - \sum_{k=1}^{l-1} \lambda_k}.
\]

Thus

\[
E \left[ \prod_{k=1}^{l+1-1} (1 - \epsilon(N^k_l)) \right] = E \left[ \prod_{k=1}^{l-1} (1 - \epsilon(N^k_l)) \right] E[1 - \epsilon(N^l_l)]
\]

\[
= 1 - \sum_{k=1}^{l+1} \lambda_k.
\]

This proves Eq. (23) for \( 2 \leq m \leq M \).

References


