Generalized Dynamic Frame Sizing Algorithm for Finite-Internal-Buffered Networks

Ching-Min Lien and Cheng-Shang Chang

***Abstract*—In this paper, we generalize the Dynamic Frame Sizing (DFS) algorithm proposed in [2] for CICQ switches to a network of queues subject to multicast traffic flows. Under the assumption of Bernoulli arrival processes for the multicast flows, we show that the DFS algorithm guarantees 100% throughput and there are at most two packets in each internal queue.**

***Index Terms*—Flow-based networks, frame-based scheduling, packet switching, 100% throughput.**

I. Introduction

In [7], Tassiulas and Ephremides studied the maximal throughput of a network of interacting queues. Since then, there are numerous research results in the literature about the interaction of scheduling policies, network throughput and performance guarantees of switches or networks. In particular, Giaccone, Leonardi and Shah [3] pointed out that most papers in the literature implicitly assumed unlimited buffer size for queues (which might be difficult to implement in practice). As such, they proposed scheduling policies for flow-based networks that can maximize the throughput while keeping the internal queues bounded.

On the other hand, Chang, Hsu, Cheng and Lee proposed the Dynamic Frame Sizing (DFS) algorithm for CICQ switches in [2]. The DFS algorithm for CICQ switches guarantees 100% throughput and finite crosspoint buffer, where the buffer size is independent of the size of switch.

In this paper, we apply the DFS algorithm to the generalized network model as considered in [3]. Under the assumption of Bernoulli arrival processes, we show that the DFS algorithm also provides 100% throughput and it only needs at most two buffers in each internal queue. Similar to the scheduling policies proposed in [3], the DFS algorithm is based on the state (i.e., the lengths of all the queues) of the network, and needs no prior knowledge about arrival processes. Moreover, the DFS algorithm is a frame-based scheme and it only needs to gather the state information at the beginning of each frame. This incurs much less communication overheads than that in [3], where the state information is needed for every time slot.

II. Modeling of Networks of Queues

Consider a network of queues as shown in Fig.1. Suppose that there are J multicast traffic flows and Q queues, indexed from 1 to J and from 1 to Q, respectively. Each flow enters the network at some queue, traverses through a set of queues arranged in a fan-out tree, and then leaves the network. Define $\Phi_I = \{q^{(j)}\}_{j=1}^{J}$ as the collection of all the ingress queues, where $q^{(j)}$ is the ingress queue for the j^{th} flow. The rest of the queues are called internal queues and are collected as the set Φ_M. To clarify the relationship between queues (especially in the same flow), the queues traversed right before and after queue q are named its upstream and downstream queues, respectively. For the clarity of our presentation, we assume at this moment that all the queues are of infinite sizes (including both ingress queues and internal queues) so that no packets are lost. Later, we will show that each internal queue needs to hold at most two packets.

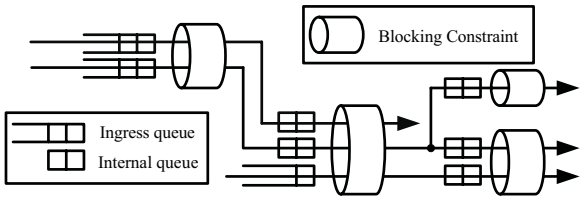

Fig. 1. Example of flow-based network of queues with eight queues, three multicast traffic flows and four blocking constraints.

Throughout this paper, we consider the usual discrete-time setting by assuming that packets are of the same size and that time is slotted so that one packet can be transmitted within a time slot. Also, we assume that each queue is started from an empty system at time 0. Let $x_q(t)$ be the number of packets in queue q at the end of time t. The governing equation for the ingress queue of the j^{th} flow, i.e., $q = q^{(j)}$, can be represented as

$$x_q(t+1) = x_q(t) + a_j(t+1) - d_q(t+1), \tag{1}$$

where $a_j(t)$ and $d_q(t)$ represents the number of external arrival packets of the j^{th} flow and the packets departing from queue q at time t, respectively. On the other hand, the governing equation for an internal queue q can be written as

$$x_q(t+1) = x_q(t) + d_{q_1}(t+1) - d_q(t+1), \tag{2}$$

where q_1 is its upstream queue. In this paper, $x_q(t)$, $a_j(t)$ and $d_q(t)$ are nonnegative integers for all queues and flows. Let R_j be the set of queues traversed by the j^{th} flow. Note from the governing equations in (1) and (2) that each queue is traversed by exactly one flow. As such, R_j's are all disjoint. According to (2), all the downstream queues of queue q would receive one packet before the end of the time slot that their upstream queue q sends a packet. The multicast policy is named no fanout splitting in the literature [5].

To model the interaction among queues, we assume there are blocking constraints for departure vectors, $\{d_q(t)\}_{q=1}^{Q}$.

Manuscript received January 12, 2009. The associate editor coordinating the review of this letter and approving it for publication was F. Theoleyre.

The authors are with the Institute of Communications Engineering, National Tsing Hua University, Hsinchu 300, Taiwan, R.O.C. (e-mail: keiichi@gibbs.ee.nthu.edu.tw, cschang@ee.nthu.edu.tw).

Digital Object Identifier 10.1109/LCOMM.2009.090067

Let S_ℓ be the collection of the queues restricted by the ℓ^{th} constraint, $\ell = 1, 2, \cdots, L$. In each time slot, there is at most one of the queues in S_ℓ that can send a packet, i.e.,

$$\sum_{q \in S_\ell} d_q(t) \leq 1. \tag{3}$$

Here, we further assume that all the queues in S_ℓ are traversed by different flows. Finally, we define F_ℓ as the collection of the flows restricted by the ℓ^{th} constraint. Specifically, the flow index j is in F_ℓ if there is a queue q traversed by flow j and constrained by the ℓ^{th} blocking constraint, i.e., $q \in R_j \cap S_\ell$. Notice that $R_j \cap S_\ell$ contains at most one queue because all the queues in S_ℓ are traversed by different flows.

Now we make the following assumptions on the external arrival processes.

(A1) The J end-to-end flows are independent Bernoulli processes when they arrive at the network. Specifically, $a_j(t)$'s are independent Bernoulli random variables for all j and t.

(A2) Assume that the arrival rate of the j^{th} end-to-end flow is λ_j, $j = 1, 2, \cdots, J$, and this rate information is unknown to the network.

(A3) The arrival rates $\{\lambda_j\}_{j=1}^{J}$ satisfy the following inequality

$$\rho = \max_{1 \leq \ell \leq L} \sum_{j \in F_\ell} \lambda_j < 1.$$

In the literature, the condition in (A3) is commonly known as the no overbooking condition for the blocking constraints.

III. Dynamic Sizing Algorithm

The dynamic frame sizing (DFS) algorithm in [2] is a framed based scheduling. The packets arriving externally would be scheduled at the beginning of the next frame if they were not sent until then. The size of the frame is determined by the minimum clearance time, which is the minimum amount of time to clear all the packets stored in the ingress queues at the beginning of a frame if there were no future arrivals. The implementation of the algorithm can be described as follows.

Dynamic Frame Sizing Algorithm:

(S1) Determine the size of a new frame: Denote by τ_n the last time slot of the $(n-1)^{th}$ frame (and by $\tau_n + 1$ the beginning time slot of the n^{th} frame). Suppose that there are $x_{q^{(j)}}(\tau_n)$ packets stored in ingress queue $q^{(j)}$ at the beginning of the n^{th} frame. For every flow j, let $y_q^{(n)} = x_{q^{(j)}}(\tau_n)$ for every $q \in R_j$. Then the size of the n^{th} frame is set to be

$$T_n = \max_{1 \leq \ell \leq L} \sum_{q \in S_\ell} y_q^{(n)} = \max_{1 \leq \ell \leq L} \sum_{j \in F_\ell} x_{q^{(j)}}(\tau_n). \tag{4}$$

Note that T_n is the minimum time T such that $\sum_{q \in S_\ell} y_q^{(n)} \leq T$ for all ℓ. If we view the backlogs at the ingress queues as fluids, then T_n is the minimum time to clear all the backlogs under the blocking constraints in (3). If $T_n = 0$ in (4), then T_n is set to be 1 and we skip the rest of the steps.

(S2) Schedule packets in the per-flow queues with the rates proportional to their sizes at the beginning of a frame. For each queue q, we generate $y_q^{(n)}$ tokens in the n^{th} frame (for sending packets to its downstream queues or to leave the network). For each $q \in R_j$, the k^{th} token of j^{th} flow in the n^{th} frame is assigned with the eligible time $\tau_n + 1 + \lfloor (k-1)T_n / y_q^{(n)} \rfloor$ and the deadline $\tau_n + \lceil kT_n / y_q^{(n)} \rceil$. To schedule under the blocking constraints in (3), there is a token arbitrator for each blocking constraint. In each time slot of the n^{th} frame, i.e., the time interval $[\tau_n + 1, \tau_n + T_n]$, the ℓth token arbitrator selects one eligible token with the earliest deadline (the EDF policy in the literature) among all the remaining tokens for all the queues in S_ℓ, and removes that token. A queue with a selected token is then allowed to send a packet in that time slot.

In order for showing that the DFS algorithm achieves 100% throughput under (A1-3), we need the following lemma to make sure that there are exactly $y_q^{(n)}$ tokens selected for queue q in the n^{th} frame.

Lemma 3.1: With the DFS algorithm and the EDF policy, the network has the following properties.

(a) Each token is served not later than its deadline. As such, there are exactly $y_q^{(n)}$ tokens selected for queue q in the n^{th} frame.

(b) There are at most two packets in each internal queue.

Proof: Consider the n^{th} frame with length T_n. We mark the beginning of the frame as time 1 for simplicity. According to (S2), it can be verified that the cumulative number of tokens for the j^{th} flow by time t is $\lceil y_q^{(n)} t / T_n \rceil$ and the number of tokens for the j^{th} flow that have deadlines not later than t is $\lfloor y_q^{(n)} t / T_n \rfloor$. According to Theorem 2.3.17 in [1], it is sufficient to show that for every subset S of S_ℓ,

$$\sum_{q \in S} \left\lfloor \frac{y_q^{(n)} t}{T_n} \right\rfloor \leq \min_{0 \leq s \leq t} \left(\sum_{q \in S} \left\lceil \frac{y_q^{(n)} s}{T_n} \right\rceil + (t - s) \right). \tag{5}$$

Notice that

$$\begin{aligned} \sum_{q \in S} \left\lfloor \frac{y_q^{(n)} t}{T_n} \right\rfloor - \sum_{q \in S} \left\lceil \frac{y_q^{(n)} s}{T_n} \right\rceil &\leq \sum_{q \in S} \frac{y_q^{(n)} (t-s)}{T_n} \\ &\leq t - s \end{aligned}$$

for all $0 \leq s \leq t$, where we use the fact that

$$\sum_{q \in S} y_q^{(n)} \leq \sum_{q \in S_\ell} y_q^{(n)} \leq T_n$$

for each token arbitrator ℓ by the definition of T_n in (4). Thus, we have that for all $0 \leq s \leq t$

$$\sum_{q \in S} \left\lfloor \frac{y_q^{(n)} t}{T_n} \right\rfloor \leq \sum_{q \in S} \left\lceil \frac{y_q^{(n)} s}{T_n} \right\rceil + (t - s)$$

and this leads to (5).

The proof of (b) follows exactly the same argument used for (P2) in [2]. Such a result was originally proved in [4].

IV. Logarithm Frame Size

Our main result is that the expectation of frame size $E[T_n]$ is bounded for each n under the DFS algorithm, as shown in Theorem 4.1.

Theorem 4.1: Assume that the input traffic satisfies (A1-3). Then under the DFS algorithms (S1) and (S2), we have for $n > 1$

$$\log E[e^{\theta^* T_n}] \leq \max\left(\theta^*, \frac{2\log L}{1-\rho}\right) \tag{6}$$

where θ^* is the unique positive solution of

$$\frac{e^\theta - 1}{\theta} = \frac{1+\rho}{2\rho}. \tag{7}$$

As a result, the expectation of the frame size $E[T_n]$ is bounded by $\max\left(1, \frac{2\log L}{\theta^*(1-\rho)}\right)$, and the DFS algorithm achieves 100% throughput.

Proof: From (S1),(S2) and Lemma 3.1(a), all the packets stored in ingress queue $q^{(j)}$ at the beginning of the n^{th} frame must arrive during the $n-1^{th}$ frame. Thus, the queue length of ingress queue $q^{(j)}$ at the beginning of a frame is bounded by the number of packets that arrive during the previous frame, i.e.,

$$x_{q^{(j)}}(\tau_{n+1}) \leq \sum_{t=\tau_n+1}^{\tau_{n+1}} a_j(t). \tag{8}$$

With (4) and (8), we have for $\theta > 0$ that

$$\begin{aligned} e^{\theta T_{n+1}} &\leq \max_{1\leq \ell \leq L} \exp\left(\theta \sum_{t=\tau_n+1}^{\tau_{n+1}} \sum_{j\in F_\ell} a_j(t)\right) \\ &\leq \sum_{\ell=1}^{L} \exp\left(\theta \sum_{t=\tau_n+1}^{\tau_{n+1}} \sum_{j\in F_\ell} a_j(t)\right), \end{aligned}$$

where we use the inequality that $\max(x_1, x_2) \leq x_1 + x_2$ for $x_1, x_2 \geq 0$. Since we assume that the arrival processes are independent Bernoulli processes in (A1) and $\{a_j(t)\}_{t=1}^{\infty}$ are i.i.d Bernoulli random variables with parameter λ_j in (A2), it then follows that

$$E[e^{\theta T_{n+1}} | T_n] \leq \sum_{\ell=1}^{L} \left\{ E\left[\exp\left(\theta \sum_{j\in F_\ell} a_j(1)\right)\right]\right\}^{T_n}.$$

Note that

$$\begin{aligned} &\log E\left[\exp\left(\theta \sum_{j\in F_\ell} a_j(1)\right)\right] = \sum_{j\in F_\ell} \log E\left[\exp\left(\theta a_j(1)\right)\right] \\ &= \sum_{j\in F_\ell} \log\left(\lambda_j e^\theta + 1 - \lambda_j\right) \leq \sum_{j\in F_\ell} \lambda_j \left(e^\theta - 1\right), \end{aligned} \tag{9}$$

where we use $\log(1+x) \leq x$ for nonnegative x. According to (A3), we have that $\rho = \max_{1\leq \ell \leq L} \sum_{j\in F_\ell} \lambda_j < 1$, and thus

$$E[e^{\theta T_{n+1}} | T_n] \leq L \exp(\rho T_n (e^\theta - 1)). \tag{10}$$

Taking expectation on both sides of (10) yields

$$E[e^{\theta T_{n+1}}] \leq L E\left[\exp\left(\rho T_n (e^\theta - 1)\right)\right]. \tag{11}$$

As θ^* is the unique positive solution of (7), we can rewrite (11) as

$$E[e^{\theta^* T_{n+1}}] \leq L E[e^{\theta^* T_n (1+\rho)/2}]. \tag{12}$$

Since $\phi(\theta) = \log E\{e^{\theta T_n}\}$ is convex in θ (see e.g., [1, Proposition 7.1.8]) and $\rho < 1$, we have that

$$\log E[e^{\theta^* T_n (1+\rho)/2}] \leq \frac{1+\rho}{2} \log E[e^{\theta^* T_n}]. \tag{13}$$

Using (13) and (12) yields

$$\log E[e^{\theta^* T_{n+1}}] \leq \log L + \frac{1+\rho}{2} \log E[e^{\theta^* T_n}]. \tag{14}$$

Since $T_1 = 1$, it is easy to verify (6) from induction by using (14). Now we use (6) to show the bound of the frame size in Theorem 4.1. Since $e^{\theta x}$ is convex in x, it follows from Jensen's Inequality that

$$E[T_n] \leq \frac{1}{\theta^*} \log E[e^{\theta^* T_n}] \leq \max\left(1, \frac{2\log L}{\theta^*(1-\rho)}\right). \tag{15}$$

Using the standard theory for regenerative processes [6], it can be shown that the DFS algorithm achieves 100% throughput according to (15).

V. Conclusion

We applied the dynamic frame sizing algorithm proposed in [2] to a general network model as considered in [3]. As the policies P1 and P2 in [3], the DFS algorithm is also sufficient to provide the following properties: (i) It guarantees 100% throughput. (ii) Each internal queue is of finite size (two packets at most). On the other hand, there is no need for the DFS algorithm to solve the optimal problem globally in every time slot. Instead, it gathers information from all the ingress queues, determines the frame size and broadcasts to all the queues in the network once a frame. At each time slot, all the token arbitrators schedule locally as shown in section III, and the computing complexity is thus reduced. The problem how to apply the DFS algorithm to networks with cyclic flows is under study.

References

[1] C.-S. Chang, *Performance Guarantees in Communication Networks.* London: Springer-Verlag, 2000.

[2] C.-S. Chang, Y. H. Hsu, J. Chang, and D. S. Lee, "Dynamic frame sizing algorithms for CICQ switches with logarithm delay," in *Proc. IEEE INFOCOM*, pp. 747-755, Apr. 2009.

[3] P. Giaccone, E. Leonardi, and D. Shah, "Throughput region of finite-buffered networks," *IEEE Trans. Parallel and Distributed Systems*, vol. 18, no. 2, pp. 251-263, Feb. 2007.

[4] S.-M. He, S.-T. Sun, H.-T. Guan, Q. Zheng, Y.-J. Zhao, and W.Gao, "On guaranteed smooth switching for buffered crossbar switches," *IEEE Trans. Networking*, vol. 16, no. 3, pp. 718-731, June 2008.

[5] M. A. Marsan, A. Bianco, P. Giaccone, E. Leonardi, and F. Neri, "Multicast traffic in input-queued switches: optimal scheduling and maximum throughput," *IEEE Trans. Networking*, vol. 11, no. 3, June 2003.

[6] S. M. Ross, *Stochastic Processes.* John Wiley & Sons, Inc., 1996.

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