

Max-Min Fairness in Input-Queued Switches

Madhusudan Hosaagrahara and Harish Sethu
ECE Department, Drexel University
3141 Chestnut Street
Philadelphia, PA 19104-2875
{madhu,sethu}@ece.drexel.edu

ABSTRACT

This paper describes an algorithm that computes the max-min fair allocation of rates for flows through an input-queued switch. The algorithm is provably max-min fair and can be implemented in a distributed fashion to dynamically determine flow rates.

1. INTRODUCTION

Fairness in traffic management can improve the isolation between traffic streams, offer a more predictable performance, eliminate transient bottlenecks, mitigate the effect of certain kinds of denial-of-service attacks, and can serve as a critical component of a quality-of-service strategy to achieve certain guaranteed services. The *Max-Min* notion of fairness is among the more popular notions, and is based on the following premises: (a) no entity should receive an allocation larger than its demand; and (b) increasing the allocation of any entity should not result in the decrease of the allocation of another entity that received an equal or smaller allocation [1].

Numerous fair scheduling algorithms that compute approximate max-min rates have been developed and successfully deployed in Internet routers [2]. These fair scheduling algorithms have typically assumed an output-queued switch with each set of competing flows sharing a single resource (the output link bandwidth) without any inter-dependencies amongst these flows. However, a new trend toward the design and use of input-queued switches based on the virtual-output-queuing (VOQ) architecture [3] is emerging.

A flow in a VOQ switch, defined as the traffic between a given input port and a given output port, not only shares its output port with a set of flows but also shares its input port with a different set of flows. Further, all the flows through the switch also share the crossbar which also limits the total data transfer rate.

This paper is concerned with extending and applying the notion of max-min fair allocation to a VOQ switch. Fair schedulers proposed for VOQ switches have attempted to calculate the fair rates by using schedulers only at either the input ports, or at the output ports, or by using fair

schedulers at both the input and output ports along with schemes to synchronize these schedulers [4–6]. Our motivation for this paper derives from the fact that simply being fair at each input port, or at each output port, or at both input and output ports does not result in an overall fair allocation. We illustrate this using the traffic pattern, depicted in Fig. 1(a), in a 4-input 4-output VOQ switch, where $\{A, B, C, D\}$ is the set of input ports and $\{W, X, Y, Z\}$ is the set of output ports. We assume that the flows are all best-effort and therefore have demands equal to the line rate, assumed at 1 unit, leading to the demand matrix shown in Fig. 1(b). We further assume that each input port always has packets available to transmit and that the crossbar can transfer data, from or to any given port, at no more than 1 unit of bandwidth.

Fig. 1(c) describes the allocations received by the flows when a max-min fair scheduling algorithm is used only at the output ports and Fig. 1(d) describes the allocations when a max-min fair scheduling algorithm is used only at the input ports. In the former case, input ports C and D are over-subscribed while in the latter case, output ports W and X are over-subscribed since the crossbar can transfer no more than 1 unit of bandwidth to any given port. Applying fair scheduling independently at both input and output ports and without any communication between the schedulers results in an allocation, illustrated in Fig. 1(e), where a flow receives the lesser of its allocation at its input and output ports. Although this allocation does not cause a bottleneck at any input or output port, or at the crossbar, the allocations are not max-min fair since the flows $B-X$, $C-X$ and $C-Y$ may increase their allocations without reducing the allocation of any other flow. Fig. 1(f) depicts the ideal max-min fair allocation, wherein it is impossible to increase the allocation of any flow whose allocation is smaller than its demand without decreasing the allocation of some other flow with a smaller allocation, while also satisfying the constraints of the crossbar.

2. CONTRIBUTIONS

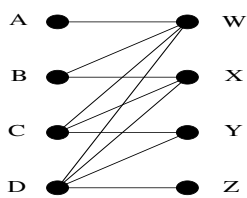
This paper introduces a new, provably fair algorithm, called *Fair Resource Allocation (FRA)* algorithm, that iteratively computes the *max-min* fair rates of allocation for each flow through the switch.

We assume that the demands of the flows are specified in a matrix form, denoted by \mathcal{D} , where the element at the u -th row and v -th column, $\mathcal{D}[u, v]$, denotes the demand of a flow arriving at input port u and headed to output port v . The max-min fair allocation for this flow is determined by the corresponding element of the allocation matrix (denoted by \mathcal{A}) returned by FRA. The bandwidth available at each input and output port are maintained as two row vectors,

Presented at the ACM SIGCOMM 2005 poster session.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Copyright 2005 ACM.



(a) The flow pattern.

	W	X	Y	Z
A	1	0	0	0
B	1	1	0	0
C	1	1	1	0
D	1	1	1	1

(b) The corresponding demand matrix.

	W	X	Y	Z
A	1/4	0	0	0
B	1/4	1/3	0	0
C	1/4	1/3	1/2	0
D	1/4	1/3	1/2	1

(c) Allocation with fair scheduling only at the output ports.

	W	X	Y	Z
A	1	0	0	0
B	1/2	1/2	0	0
C	1/3	1/3	1/3	0
D	1/4	1/4	1/4	1/4

(d) Allocation with fair scheduling only at the input ports.

	W	X	Y	Z
A	1/4	0	0	0
B	1/4	1/3	0	0
C	1/4	1/3	1/3	0
D	1/4	1/4	1/4	1/4

(e) Allocation with independent fair scheduling at both input and output ports.

	W	X	Y	Z
A	1/4	0	0	0
B	1/4	3/8	0	0
C	1/4	3/8	3/8	0
D	1/4	1/4	1/4	1/4

(f) The Max-Min fair allocation.

Figure 1: An illustration of why it is non-trivial to achieve fairness in input-queued switches.

\mathcal{I} and \mathcal{O} , respectively. $\mathcal{I}[u]$, the u -th element of \mathcal{I} , denotes the bandwidth available at input port u . Similarly, $\mathcal{O}[v]$ denotes the bandwidth available at the v -th output port. FRA assumes the presence of a *Max-Min Fair* (MMF) operator that, given a scalar constraint and a vector of demands, calculates the vector of max-min fair allocations. We define two new operators based on the MMF operator: the *row-wise* max-min fair operator MMF_R , and the *column-wise* max-min fair operator MMF_C . Given a matrix of demands and a vector of constraints, MMF_R generates a matrix of allocations where each row of the allocation matrix is max-min fair with the demand vector being given by the corresponding row of the demand matrix and the constraint being given by the corresponding element of the vector of constraints. Similarly, MMF_C operates on a demand matrix and a vector of constraints to calculate an allocation matrix where each column of the allocation matrix is max-min fair, with the demand vector being given by the corresponding column of the demand matrix and the constraint being given by the corresponding element of the vector of constraints. FRA maintains two matrices internally; the request matrix, denoted by \mathcal{R} , and the grant matrix, denoted by \mathcal{G} .

Fig. 2 describes the pseudo-code of the FRA. The operation of the algorithm can be divided into two main phases. In the first phase (lines 02–12), all the output ports which are bottleneck links for at least one flow are allocated, thus finalizing the allocations for such flows. The demand matrix is updated to reflect these allocations and the process is repeated until there is no output port which is the bottleneck

```

01: Function FRA ( $\mathcal{D}, \mathcal{I}, \mathcal{O}$ ) returns  $\mathcal{A}$ 
02: Do
03:   done  $\leftarrow$  true
04:    $\mathcal{R} \leftarrow \text{MMF}_R(\mathcal{I}, \mathcal{D})$ 
05:    $\mathcal{G} \leftarrow \text{MMF}_C(\mathcal{O}, \mathcal{R})$ 
06:   for all  $[i, j]$ 
07:     if  $\mathcal{G}[i, j] < \mathcal{R}[i, j]$ 
08:        $\mathcal{D}[i, j] \leftarrow \mathcal{G}[i, j]$ 
09:       done  $\leftarrow$  false
10:   end if
11: end for
12: While done is false
13:    $\mathcal{R} \leftarrow \text{MMF}_R(\mathcal{I}, \mathcal{D})$ 
14:   for all  $[i, j]$ 
15:     if  $\mathcal{R}[i, j] < \mathcal{D}[i, j]$ 
16:        $\mathcal{D}[i, j] \leftarrow \mathcal{R}[i, j]$ 
17:     end if
18:   end for
19:    $\mathcal{A} \leftarrow \mathcal{D}$ 
20: return  $\mathcal{A}$ 

```

Figure 2: Pseudo-code for the FRA algorithm.

link for any flow. Thus, at the end of the first phase, the only possible bottlenecks for the flows are the input ports. In the second phase (lines 13–20), the algorithm allocates bandwidth at the input ports in a max-min fair fashion, thus resulting in an allocation that is overall max-min fair.

THEOREM 1. The FRA algorithm results in a *max-min* fair allocation.

PROOF. Omitted for brevity.

The rates computed by the FRA algorithm can serve as a reference in the evaluation of the fairness achieved by various schedulers for input-queued switches. Although not presented here, we have derived practical schedulers based on FRA that exhibit superior fairness in comparison to previously proposed schedulers while also achieving comparable performance in simulations using real traffic traces.

3. REFERENCES

- [1] D. Bertsekas and R. Gallager, *Data Networks*, 2nd ed. Prentice Hall, 1992.
- [2] H. Zhang, "Service disciplines for guaranteed performance service in packet-switching networks," *Proc. of the IEEE*, vol. 83, no. 10, Oct. 1995.
- [3] Y. Tamir and G. L. Frazier, "High-performance multi-queue buffers for VLSI communications switches," in *Proc. ISCA*, 1988, pp. 343–354.
- [4] D. C. Stephens and H. Zhang, "Implementing distributed packet fair queueing in a scalable switch architecture," in *Proc. IEEE INFOCOM*, Apr. 1998, pp. 282–290.
- [5] N. Kumar, R. Pan, and D. Shah, "Fair scheduling in input-queued switches under inadmissible traffic," in *Proc. IEEE GLOBECOM*, vol. 3, Nov. 2003, pp. 1713–1717.
- [6] D. Cavendish, M. Lajolo, and H. Liu, "On the evaluation of fairness for input queued switches," in *Proc. IEEE ICC*, vol. 2, May 2002, pp. 996–1000.