Nash Equilibria as a Fundamental Issue Concerning Network-Switches Design

George F. Georgakopoulos
Dept. of Computer Science, University of Crete, and
Inst. of Computer Science, Foundation for Research & Technology - Hellas (FORTH)
Heraklion, Crete, Greece, GR-71409
ggeo@csd.uoc.gr

***Abstract*—We view the 'packet-switching problem' (from N inputs towards N outputs) from the perspective of game theory and we prove that, if the rates of flows are weighed then 'weighed max-min fair service rates' are the unique Nash equilibrium point of a natural strategic game in which throughput is granted on a 'least-demanding first-served' principle. We prove that a crossbar switching device with suitably randomized schedulers converges to this equilibrium point without pre-computing it.**

Crossbar network switches, non-cooperative strategic games.

I. INTRODUCTION

In the 'network-switching problem' data packets from N sources are forwarded through a single node towards N destinations creating $N \times N$ flows of data. A switching device in this node has to be assigned the task of handling the relevant traffic control. Suppose that each flow of data f from a source to a destination is assigned a weight w_f by which we weigh its rate of service r_f. The question is: *on what principle(s)* should our device operate? If the *utility* given to f is defined as r_f/w_f, what service rates should be given to the flows in order to achieve 'fairness', i.e., to equalize utilities as far as possible? And how can we be convinced that a specific device performs as desired?

On this question various active themes of research converge, and among a quite extensive literature various interesting starting points, more closely related to this work, are [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]. In an attempt to face the afore-mentioned issue it has been proposed that the rates of service for the $N \times N$ flows should be granted according to the 'weighted max-min fairness' (WMMF) principle [1, 2, 3, 10]: the rate granted to each flow is 'maximized' in the sense that it cannot be further increased unless the rate of some other flow receiving equal or less utility is decreased. However reasonable one may consider the WMMF-rates to be, they do raise a *reverse engineering* issue: a *solution* is suggested (a switching fabric or a relevant algorithm), yet we are seemingly lacking a rigorous definition of the *problem* it solves—a situation quite suggestive of a fiercely advancing technological era. (This state of affairs is met again: see [12] about the TCP/IP protocol.)

In this work we offer such a problem: Following a *game theory* approach, we suggest *to let the flows decide the utility they will get*. More specifically we define a natural *auction* game which grants utilities to flows according to a 'least-demanding first-served' principle and prove that (a) this game has a *unique Nash equilibrium* which moreover corresponds to the 'WMMF' principle; (b) this equilibrium can be attained by at least one *implementable* device (here a crossbar switch with randomized schedulers). Thus we are able to answer on firm foundational ground at least a pair of crucial issues: (a) Which is the problem and which is its solution; and (b) that some implementable device does indeed offer this solution.

Notice (a) that the *'least-demanding first-served'* principle applied is a quite reasonable principle for allocating a common resource (here: throughput); and (b) that this principle is compatible with the, seemingly forthcoming, *'pay-more get-more'* principle: a flow f can simply 'pay' for a higher weight w_f. The reader might want to compare our result with [13] where it is shown that if users are allowed to *select* their 'priorities' by paying a higher cost for a higher priority, then in a network supporting a continuum of priorities, a weighted max-min fair allocation (with suitably defined weights) is, again, achieved as a Nash equilibrium. Our result complements [13] (since we deal with how *given* weights should be interpreted), yet we should indicate that the game we consider here is defined *differently* from that of [13].

Since WMMF is an already widely accepted principle, our results should not be interpreted simply as arguments in favor of WMMF, but mainly as a further analysis of its properties.

In Section II we review the 'weighted max-min fairness' idea. In Section III we present a switching game and prove that its unique Nash equilibrium coincides with 'weighted max-min fairness'. In Section IV we prove that a crossbar switching device with suitably randomized schedulers indeed attains the afore-mentioned Nash equilibrium. Section V is an epilogue.

II. A 'FAIR' SOLUTION FOR THE SWITCHING PROBLEM

A. An abstract switching device

Our switching device D is abstractly defined by the following: (a) A *size* $N \geq 2$; (b) N *inputs* $i = 1, ..., N$, and N *outputs* $j = 1, ..., N$; (c) An $N \times N$ matrix of positive *weight parameters* $w_{i,j}$. (We can avoid zero weights by replacing them with a sufficiently small value $\varepsilon > 0$.)

The intended meaning of the above is the following: (1) Each pair $(i, j) \in [1, N] \times [1, N]$ is a *flow* (of *data packets*) entering through input i and destined to leave through output j;.

(2) Device D operates at discrete *time-steps*, $t = 1, 2, 3, ...$. At

each time-step a subset $M \subseteq [1, N] \times [1, N]$ of the flows is *served*, i.e., for each $(i, j) \in M$ a packet belonging to flow (i, j) is either received from input i, or forwarded to output j, or both.

(3) During T time-steps, for each (i, j) let $s_{i,j}$ be the number of data packets of flow (i, j) transferred from input i to output j. If D has devoted $p_{i,j}T$ units of time to receive input from (i, j) then we say that (i, j) has been served with *input rate* $p_{i,j}$. Symmetrically for *output rates* $q_{i,j}$. If a steady state is reached as $T \to \infty$, then flow (i, j) is *served with rate* $r_{i,j} = \lim_{T\to\infty} s_{i,j}/T$.

The ratio $u_{i,j} = r_{i,j} / w_{i,j}$ is the *utility* granted to flow (i, j).

(4) For any $n \times n$ matrix $\mathbf{a}$ we define $R_i(\mathbf{a})$ as the sum $\sum_{j=1}^{n} a_{i,j}$ of the elements of the i^{th} row. Similarly we define $C_j(\mathbf{a})$ as the sum $\sum_{i=1}^{n} a_{i,j}$. For every flow (i, j) its weight $w_{i,j}$ is a measure of its *priority* depending in whatever way on this flow. In principle we would like the service rate $r_{i,j}$ of each flow (i, j) to be proportional to its weight $w_{i,j}$, $i, j = 1, ..., N$. This is achievable along a row of matrix $\mathbf{w}$ by setting $r_{i,j} = w_{i,j}/R_i(\mathbf{w})$ (or along a column of $\mathbf{w}$) yet this is not always achievable for all i, j without sacrificing some throughput of D. (In this work we do not deal with the issue of handling flows (i, j) composed of many sub-flows with individual weights.)

B. The WMMF rates for servicing N × N *flows*

Given a matrix of weights $\mathbf{w}$ we recall an algorithm, proceeding in $2N$ rounds, that computes the WMMF-rates $\mathbf{r} = [r_{i,j}] = \text{WMMF}(\mathbf{w})$: We characterize each row or column of $\mathbf{w}$ as *'fixed'* or not—initially none of the rows or columns is fixed—and at each round we consider only non-fixed rows or columns. An element (i, j) is said to be *fixed* if either row i or column j is fixed. For each row i let $F_{i,0}$ denote the sum of the rates $r_{i,j}$ assigned to fixed elements along i, and let $W_{i,0}$ denote the sum of the weights of the non-fixed elements along i. Analogously, for each column j. The WMMF-algorithm is:

```
Set W_{i,0}←R_i(w), F_{i,0}←0 & W_{0,j}←C_j(w), F_{0,j}←0
Repeat
 For non-fixed rows ρ_i ← (1−F_{i,0})/W_{i,0}
 For non-fixed columns κ_j ← (1−F_{0,j})/W_{0,j}
 Find i with min ρ_i among non-fixed rows
 Find j with min κ_j among non-fixed columns
 If ρ_i < κ_j Then
 { For (i,j)=non-fixed elements of row i;
   Set u_{i,j} ← ρ_i; 'Fix' row i }
 Else
 { For (i,j)=non-fixed elements of column j;
   Set u_{i,j} ← κ_j; 'Fix' column j }
 Update W_{i,0}, F_{i,0} & W_{0,j}, F_{0,j}
Until all elements have been fixed
For all (i,j) { set r_{i,j} to u_{i,j} · w_{i,j} }.
```

Another view of the WMMF algorithm is the following: Given $\mathbf{w}$, a pair of matrices can be defined by $p_{i,j} = w_{i,j}/R_i(\mathbf{w})$ and $q_{i,j} = w_{i,j}/C_j(\mathbf{w})$, $i, j = 1, ..., N$. In such a pair $(\mathbf{p}, \mathbf{q})$ a *majorized column* j is one such that $p_{i,j} \geq q_{i,j}$ for all rows i. Similarly a *majorized row* i is one such that $q_{i,j} \geq p_{i,j}$ for all columns j. In any such pair of matrices, $(\mathbf{p}, \mathbf{q})$, at least one majorized row or column will *always* exist: Setting $V = \{R_k(\mathbf{w}): k = 1, ..., N\} \cup \{C_l(\mathbf{w}): l = 1, ..., N\}$ then either the row i for which $R_i(\mathbf{w}) = \max V$, or the column j for which $C_j(\mathbf{w}) = \max V$, is easily seen to be majorized. If column j is majorized then we fix for each row $i = 1, ..., N$ the rate $r_{i,j}$ of flow (i, j) by setting $r_{i,j} = q_{i,j} \leq p_{i,j}$, and distribute the excess-rate $(p_{i,j} - q_{i,j})$ to the other flows (i, l) $l = 1, ..., N$, $l \neq j$, in the same row, proportionally to their weights $w_{i,l}$. Column j is further ignored. We act analogously if we have a majorized row. Repeating the above procedure we obtain the WMMF-rates.

A matrix $\mathbf{a} = [a_{i,j}]$ where $a_{i,j} \geq 0$, $i, j = 1, ..., N$, is said to be *doubly stochastic* iff $R_i(\mathbf{a}) = 1$ for all $i = 1, ..., N$, and $C_j(\mathbf{a}) = 1$ for all $j = 1, ..., N$. Matrix $\mathbf{a}$ is said to be in *max-min form* iff every element $a_{i,j}$ is the maximum element either in row i or in column j. Matrix $\mathbf{a}$ is said to *majorize* matrix $\mathbf{b}$ iff for all i, j we have $a_{i,j} \geq b_{i,j}$. The following are 'folklore' facts about WMMF:

Fact 1: The WMMF-algorithm returns a matrix $\mathbf{r}$ of rates of service for which the following three hold: (a) $\mathbf{r}$ is a doubly stochastic matrix; (b) the utility matrix $\mathbf{u}$ is in max-min form; (c) $\mathbf{r}$ majorizes the matrix $\mathbf{f} = [\min\{p_{i,j}, q_{i,j}\}]$, where $p_{i,j} = w_{i,j}/R_i(\mathbf{w})$ and $q_{i,j} = w_{i,j}/C_j(\mathbf{w})$, $i, j = 1, ..., N$. The converse is also true. ■

III. NETWORK SWITCHING AS A STRATEGIC GAME

A. A 'throughput game' and its Nash equilibria

We shall view the switching problem as a *strategic game*. Our game can be supposed to be *non-cooperative* simply because the breathtaking speed at which switching fabrics are able and expected to be operated, renders any 'cooperation' a prohibitively time-consuming luxury.

Let us recall the notion of a (*strategic*) *game* [14]: Let $k = 1, ..., m$ be the *m players* of our game. Each player k has a set of available *strategies* S_k to follow. A vector of strategies $(s_1, ..., s_m)$ where $s_k \in S_k$, $k = 1, ..., m$ is a (*strategy*) *profile*. For any strategy profile $S \in S_1 \times S_2 \times ... \times S_m$, each player k enjoys a *gain* (or *payoff*) defined by a function $\text{gain}_k(S) \in \mathbf{R}$. For any strategy profile S, any player $k = 1, ..., m$, and any strategy $s \in S_k$, we denote by $S \leftarrow [k, s]$ the strategy profile obtained by replacing in S the strategy s_k of k^{th} player by s. A *Nash equilibrium point* is a profile S^* such that *no player can increase its payoff* by modifying S^* *unilaterally* (here is where 'non-cooperativeness' enters the scene) i.e.: for all players $k = 1, ..., m$ and all strategies of k, $s \in S_k$: $\text{gain}_k(S^* \leftarrow [k, s]) \leq \text{gain}_k(S^*)$. Our switching game is defined as follows:

Definition 2 (throughput-auction game): (a) We have $N \times N$ players, each corresponding to a flow (i, j), $i, j = 1, ..., N$. Each player (flow) is characterized by a weight $w_{i,j}$, $i, j = 1, ..., N$ (the highest the weight, the highest its 'priority'); (b) The strategy of each player (i, j) is a positive real number, $U_{i,j} \in \mathbf{R}$, (to be interpreted as the *required utility*, expressing that flow (i, j) 'requires' a rate of service equal to $U_{i,j} \cdot w_{i,j}$). Thus a strategy profile is an $N \times N$ matrix of required utilities $\mathbf{U}$; (c) the game is played as follows: for each input i, the required utilities $U_{i,j}$, $j = 1, ..., N$, are examined in increasing order and are granted a *tentative* input rate $P_{i,j} = U_{i,j} \cdot w_{i,j}$ as long as the

total rate granted for input i remains ≤ 1. Flows for which the remaining input rate is not sufficient to cover what they require receive zero input rate $P_{i,j} = 0$. Symmetrically for each output j a tentative output rate $Q_{i,j} = U_{i,j} \cdot w_{i,j}$ is granted. The *gain* for each (i, j), is defined by: $\text{gain}_{(i,j)}(\mathbf{U}) = g_{i,j} = \min(P_{i,j}, Q_{i,j})$. ■

We characterize the Nash-equilibria of the game of Def. 2:

Theorem 3: Referring to the game of Def. 2, let $g_{i,j}$ be the granted rates. Profile $\mathbf{U}$ is a *Nash equilibrium* for this game if and only if the following three conditions hold: (a) The rates $g_{i,j}$ form a *doubly stochastic matrix*; (b) The finally *granted* utilities matrix $u_{i,j} = g_{i,j} / w_{i,j}$ is in *max-min form*; (c) The rate-matrix $\mathbf{g}$ majorizes $\mathbf{f} = [\min\{p_{i,j}, q_{i,j}\}]$, where $p_{i,j} = w_{i,j}/R_i(\mathbf{w})$ and $q_{i,j} = w_{i,j}/C_j(\mathbf{w})$, $i, j = 1, ..., N$.

Proof: *Necessity* is proved as follows: (a) The proof has two steps: (1) If in matrix $\mathbf{g}$ for some i, j we have $R_i(\mathbf{g}) < 1$ and $C_j(\mathbf{g}) < 1$ then we examine two cases: Case 1.1: If $g_{i,j} = 0$ then flow (i, j) can set (possibly *reducing*) its strategy $U_{i,j}$ to a sufficiently small value $\delta > 0$ so as to be served before other flows and thus be granted an amount δ of both input and output rate. So (i, j) can unilaterally increase its payoff from 0 to δ, therefore strategy profile $\mathbf{U}$ is not a Nash equilibrium point; Case 1.2: If $g_{i,j} > 0$ then flow (i, j) has received the utility (thus rate also) it has required, and since by our hypothesis the rate for input i and the rate for output j have not been exhausted, flow (i, j) can alter its strategy $U_{i,j}$ *increasing* it by a sufficiently small value $\delta > 0$ so as to be still served (perhaps as the last one, either in row i or column j). Both input and output tentative rates $(P_{i,j}, Q_{i,j})$ will be increased thus securing greater payoff. So again $\mathbf{U}$ is not Nash. (2) So if for some row i we have $R_i(\mathbf{g}) < 1$ then by (1) we must have $C_j(\mathbf{g}) = 1$ for all $j = 1, ..., N$, which gives: $\sum_{i=1}^{N} R_i(\mathbf{w}) = \sum_{j=1}^{N} C_j(\mathbf{w}) = N$. But since for all $i = 1, ..., N$, $R_i(\mathbf{g}) \leq 1$, all $R_i(\mathbf{g})$'s must be also equal to 1—a contradiction. A symmetric argument holds if for some j we have $C_j(\mathbf{g}) < 1$. Thus $\mathbf{g}$ is a doubly stochastic matrix.

(b) Let $u_{i,j} = g_{i,j} / w_{i,j}$ be the granted utilities and suppose that for some i, j and k, l the following two hold: $u_{i,j} < u_{i,l}$ and $u_{i,j} < u_{k,j}$, i.e., $u_{i,j}$ is not the maximum either in the i-row or in the j-column. But in this case a sufficiently small increase in (i, j)'s strategy $U_{i,j}$ can be granted (both for the input and output rates) because (i, j) cannot be the last served flow either in row i (since flow (i, l) has been granted a strictly greater rate) or in column j (since flow (k, j) has been granted a strictly greater rate). So (i, j) can unilaterally increase its payoff, therefore strategy profile $\mathbf{U}$ is not Nash.

(c) Let $g_{i,j} < \min\{w_{i,j}/R_i(\mathbf{w}), w_{i,j}/C_j(\mathbf{w})\}$ for some i, j. Since by (a) above for all $i = 1, ..., N$ we have $R_i(\mathbf{g}) = 1$ and for all $j = 1, ..., N$, we have $C_j(\mathbf{g}) = 1$, there must exist k, l so that $g_{i,l} > w_{i,l}/R_i(\mathbf{w})$ and $g_{k,j} > w_{k,j}/C_j(\mathbf{w})$. So granted rates $g_{i,l}$ and $g_{k,j}$ are both greater than $g_{i,j}$; yet by (b) this cannot happen.

Sufficiency also holds: Let $g_{i,j}$ satisfy all three conditions of Theorem 3. Then the strategy profile $\mathbf{U} = [g_{i,j} / w_{i,j}]$, $i, j = 1, ..., N$, is a Nash equilibrium point: By (c) all flows are granted a non-zero rate; by (b) they are the last served either in their row or column; finally, by (a) they exhaust either all the remaining input or all the remaining output rate. Thus no flow can unilaterally increase its payoff since: (1) decreasing its requirement ('strategy') does not help because it has already been granted what it requires; (2) increasing its requirement also does not help because it will not be served earlier, while it already exhausts either all input / output rate available to it. ■

By Fact 1 and Theorem 3, our game has a unique Nash equilibrium point: the granted rates equal the WMMF-rates.

B. *The game approach: a short informal discussion*

The following two issues might be of interest to the reader: The first issue is: In an *actually* played game shouldn't everyone involved be aware of it? We consider this issue as a delicate one: On the one hand the answer is 'no': As evolutionary game theory has revealed [14, 15] game theory can explain phenomena involving very simple organisms in which no rationality or even awareness is observed. On the other hand awareness is not something we 'begin-with', but something we *'become-of'*. After all, history has reserved a very distinctive role for scientific research in this latter process... The second issue is the difference between the *'fairness'* and the *'game'* approach: In the latter case the designer *observes* the strategic game users are indeed—knowingly of unknowingly—playing, computes its equilibrium points, and (possibly) claims: «This is the game you are involved in, this is provably the best each one of you can obtain from it, however selfishly each of you may act, and here is a device that obtains the same instead of you». Notice that instead of discussions about what is fair, or what could be a reasonable approximation to it, an exact optimum is offered. We see no guarantee that the two approaches will always coincide—as it happens in our case.

IV. A RANDOMIZED SWITCH CONVERGING TO NASH

Notice that, with a device cycle of just a few nano-seconds, *even computing the WMMF-rates by running the algorithm of Section* II.B *is prohibitively time and hardware consuming.* Instead we shall show (inspired by [2, 3]) that a suitably randomized device can converge to the WMMF-rates *without pre-computing them*. An attractive architecture for such a device is the 'crossbar switch' (implementable easily and cheaply *on-chip* [2, 16]). More specifically D is defined by what is mentioned in Section II.A, plus the following: (a) An $N \times N$ matrix of *buffers* of size B, i.e., variables $b_{i,j}$ which take values in $[0, B]$. If $b_{i,j} = 0$ then the (i, j) buffer is called *empty*, else if $b_{i,j} = B$ it is called *full*. (b) N *schedulers*: one S_i^{in} for each row i, and one S_j^{out} for each column j. Schedulers run a scheduling algorithm returning a number in $[0, N]$. At each time-step for each $i = 1, ..., N$ input-scheduler S_i^{in} selects a column $l \in [0, N]$. If $l \neq 0$ and (i, l) is not full then a packet is transferred from input i to buffer (i, l) which is set to value $b_{i,l} + 1$. Output-schedulers S_j^{out} operate symmetrically. (Here we suppose that input comes from a set of input queues and backpressure is applied.). The scheduling algorithm of our device D will be an *'oblivious repetitive sampling'*:

S_i^{in} : if all buffers along row i are full return 0, else pick

$j \in [1, N]$ with probability $p_{i,j} = w_{i,j} / R_i(\mathbf{w})$ until a non-full buffer (i, j) is encountered.

S_j^{out} : (symmetrically).

Theorem 4: Let D be a crossbar switching device with buffers of size B, weight parameters $\mathbf{w}$, and schedulers performing oblivious repetitive sampling, and let Rate(T, B) be the matrix of service rates $r_{i,j}$ achieved in T time-steps. The following holds: $\lim_{T\to\infty} \lim_{B\to\infty} \text{Rate}(T, B) = \text{WMMF}(\mathbf{w})$.

Proof: Let C be the following finite Markov chain: A single buffer of size B, can be in one of $B+1$ states $b \in [0, B]$: at state b the buffer holds b packets. At discrete time-steps our buffer is (independently of its state and of other buffers) *probed* for input with probability p and if it is not full it receives a packet (otherwise the 'chance is lost'). At the same step our buffer is similarly *probed* for output with probability q. We define the service-rate function $s(\cdot,\cdot)$ as follows:

$s(p, q)$ = the rate of service achieved by our buffer if it is probed for input (resp. output) with probability p (resp. q). (1)

Let $e(p, q)$ be the probability that the buffer will be empty, and let $f(p, q)$ be the probability that it will be full. A packet will be received from input with probability $p(1–f(p, q))$ and will be delivered to output with probability $(1–e(p, q))\, q$. These expressions must be equal to the service rate $s(p, q)$ achieved by this buffer, so the following holds:

$$f(p,q) = [p - s(p,q)]/p\,, \quad e(p,q) = [q - s(p,q)]/q \qquad (2)$$

If at row i we are *probing* buffer (i, j) for input with probability $p_{i,j}$ then at each time-step we are '*visiting (at least once)*' this buffer with an actual probability $\overline{p}_{i,j}$ no less than $p_{i,j}$, since it may happen to probe (i, j) on our first probe or after some other buffer. Similarly we are output-probing (i, j) with probability $\overline{q}_{i,j} \geq q_{i,j}$. Thus the service rate of (i, j) will actually be $s(\overline{p}_{i,j}, \overline{q}_{i,j})$. The following hold for every (i, j):

$$\overline{p}_{i,j} = \frac{\overline{p}_{i,1} f(\overline{p}_{i,1}, \overline{q}_{i,1}) p_{i,j}}{1 - p_{i,1}} + \ldots + p_{i,j} + \ldots + \frac{\overline{p}_{i,N} f(\overline{p}_{i,N}, \overline{q}_{i,N}) p_{i,j}}{1 - p_{i,N}} \qquad (3)$$

$$\overline{q}_{i,j} = \frac{\overline{q}_{1,j} e(\overline{p}_{1,j}, \overline{q}_{1,j}) q_{i,j}}{1 - q_{1,j}} + \ldots + q_{i,j} + \ldots + \frac{\overline{q}_{N,j} e(\overline{p}_{N,j}, \overline{q}_{N,j}) q_{i,j}}{1 - q_{N,j}} \qquad (4)$$

Equation (3) arises from the following considerations: We may probe buffer (i, j) on our first probe (with probability $p_{i,j}$), or we may probe buffer (i, j) after some other buffer (i, k) ($k = 1, \ldots, N$, $k \neq j$) as follows: (1) Probe buffer (i, k) at least once (with probability $\overline{p}_{i,k}$ by the definition of $\overline{p}_{i,k}$); (2) Find that buffer (i, k) is full (with probability $f(\overline{p}_{i,k}, \overline{q}_{i,k})$); (3) 'Switch' to buffer (i, j) (with the relative probability $p_{i,j}/(1–p_{i,k})$): notice that probing subsequently the same buffer (i, k) obliviously and repetitively $m \geq 0$ times, and switching afterwards to (i, j), has total probability $p_{i,j} \sum_{m=0}^{\infty} p_{i,k}^m = p_{i,j}/(1 - p_{i,k})$. Equation (4) arises analogously.

Let $\mathbf{P}$ (or $\mathbf{Q}$) be the space of all $N \times N$ matrices with elements in the range [0,1]. Probabilities $(\overline{\mathbf{p}}, \overline{\mathbf{q}})$ appear on both sides of (3) and (4), so we have to view the above $2N^2$ equations as an operator $\mathbf{F}(\cdot,\cdot)$: $\mathbf{P} \times \mathbf{Q} \to \mathbf{P} \times \mathbf{Q}$ of which we seek a fixed point $(\overline{\mathbf{p}}, \overline{\mathbf{q}})$. Our next lemma gives a fixed point of $\mathbf{F}$ under the ideal circumstances:

Lemma 5: Let the pair $(\mathbf{p}, \mathbf{q}) \in \mathbf{P} \times \mathbf{Q}$ arise from $\mathbf{w}$ by $p_{i,j} = w_{i,j}/R_i(\mathbf{w})$ and $q_{i,j} = w_{i,j}/C_j(\mathbf{w})$, and suppose that for all buffers the service-rate function $s(x, y) = \min\{x, y\}$. Let $(\overline{\mathbf{p}}, \overline{\mathbf{q}})$ be the fixed point of the operator $\mathbf{F}$. Then $\min(\overline{\mathbf{p}}, \overline{\mathbf{q}})$ = WMMF($\mathbf{w}$), i.e., the service rates achieved are the WMMF-rates.

Proof: Using (2) we rewrite (3) and (4) as follows:

$$\overline{p}_{i,j} = \frac{\left(\overline{p}_{i,1} - s(\overline{p}_{i,1}, \overline{q}_{i,1})\right) p_{i,j}}{1 - p_{i,1}} + \ldots + p_{i,j} + \ldots + \frac{\left(\overline{p}_{i,N} - s(\overline{p}_{i,N}, \overline{q}_{i,N})\right) p_{i,j}}{1 - p_{i,N}} \qquad (5)$$

$$\overline{q}_{i,j} = \frac{\left(\overline{q}_{1,j} - s(\overline{p}_{1,j}, \overline{q}_{1,j})\right) q_{i,j}}{1 - q_{1,j}} + \ldots + q_{i,j} + \ldots + \frac{\left(\overline{q}_{N,j} - s(\overline{p}_{N,j}, \overline{q}_{N,j})\right) q_{i,j}}{1 - q_{N,j}} \qquad (6)$$

Suppose that $s(\cdot,\cdot) = \min\{\cdot,\cdot\}$ and let the WMMF-algorithm fix at his first step the j^{th} column. Then the j^{th} column is majorized: for $i = 1, \ldots, N$ we have $\min\{p_{i,j}, q_{i,j}\} = q_{i,j}$ = the WMMF-rate for (i, j).

For the fixed point $(\overline{\mathbf{p}}, \overline{\mathbf{q}})$ (to be defined stepwise) we have $\overline{p}_{i,j} = p_{i,j} + (\textit{other terms}) \geq p_{i,j}$, and we can 'fix' $\overline{q}_{i,j}$ to be equal to $q_{i,j}$. Since finally we shall have $s(\overline{p}_{i,j}, \overline{q}_{i,j}) = \min\{\overline{p}_{i,j}, \overline{q}_{i,j}\} = q_{i,j}$, we shall achieve along the j^{th} column the same rates as those given to the flows by the WMMF-algorithm. We act symmetrically if WMMF-algorithm fixes at its first step the i^{th} row. Passing these values to the rest of equations of $\mathbf{F}$ and ignoring in further rounds all fixed rows or columns suffices to prove, inductively, our Lemma. ■

Lemma 6: For the afore-mentioned finite Markov chain C, $\lim_{B\to\infty} s(p, q) = \min(p, q)$ for all $p, q \in [0, 1]$.

Proof: Define parameters λ and the vector $\mathbf{V_B}$ by :

$$\mathbf{V_B} = \frac{\left\langle (1-q)\lambda^B, \lambda^{B-1}, \ldots, \lambda^2, \lambda^1, (1-p) \right\rangle}{\left((1-q)\lambda^B + \lambda^{B-1} + \ldots + \lambda^2 + \lambda^1 + (1-p) \right)}$$

where $\lambda = q(1–p)/[p(1–q)]$. Vector $\mathbf{V_B}$ is an eigenvector of C with eigenvalue 1: this can be easily verified since the transition matrix of C has only 5 types of columns. Thus $\mathbf{V_B}$ gives the steady-state probabilities for our Markov chain C. For $q < p$ we get that the probability $e(p, q)$ of the buffer to be empty (i.e., the 1st component of $\mathbf{V_B}$) tends to zero as $B \to \infty$, so by (2) we get $s(p, q) \to q = \min\{p, q\}$. Symmetrically for $p < q$ we get $s(p, q) \to p = \min\{p, q\}$. ■

Proof of Theorem 4 (continued): By Lemma (6) $s(\cdot,\cdot)$ tends to $\min\{\cdot,\cdot\}$ as $T, B \to \infty$. Thus by Lemma (5) the actual probing probabilities for the buffers—given by the fixed point $(\overline{\mathbf{p}}, \overline{\mathbf{q}})$ of $\mathbf{F}$—will satisfy in-the-limit $\min(\overline{\mathbf{p}}, \overline{\mathbf{q}})$ = WMMF($\mathbf{w}$). Since $s(\cdot,\cdot)$

tends to $\min\{\cdot,\cdot\}$ the achieved rates will be asymptotically equal to the WMMF-rates, thus establishing Theorem 4. ■

Finally, since 'oblivious repetitive sampling' is too time consuming, we may turn to *direct-sampling* schedulers:

Lemma 6: Let **p** be a probability distribution over elements $[1, n]$ and let $G \subseteq [1, n]$ be the only 'eligible' ones. We select repetitively an element in $[1, n]$ according to **p** until we select one in G. Let p_k^* be the probability that element k is selected. Then $p_k^* = p_k / \sum_{i \in G} p_i$ for $k \in G$ and $p_k^* = 0$ for $k \notin G$.

Proof: (Straightforward.) ■

So, finally, the schedulers for our crossbar switch are:

S_i^{in} : Let the non-full buffers in row i be $\{(i, l): l \in G\}$, where $G \subseteq [1, N]$. If $G = \varnothing$ return 0 else return column $j \in G$ with probability $p_{i,j}^* = w_{i,j} / \sum_{l \in G} w_{i,l}$.

S_j^{out} : (Symmetrically).

V. Epilogue and Further Work

Concerning Section III all directions are open: a maximal target would be to apply it to all sorts of similar problems in network design. Concerning Section IV one important issue is left open: Random bits are not so cheap and several $^{G}/_{sec}$ of them may be needed in modern large network switching fabrics. Can we apply instead some form of fast deterministic sampling [16, 17, 18, 19] for the same purpose?

Acknowledgment

The author thanks prof. *Manolis Katevenis* as well as *Nikos Chrysos* (Computer Science Department, University of Crete, and ICS, FORTH, Greece) for introducing him to the problem, as well as for various helpful discussions.

References

[1] D. P. Bertsekas and R. Gallager, "Data Networks", Englewood-Cliffs, New Jersey, 1992.

[2] N. Chrysos: "Weighted Max-Min Fairness in a Buffered Crossbar Switch with Distributed WFQ Schedulers: a First Report", FORTH-ICS/TR-309, Inst. of Computer Science, FORTH, Crete, Greece; M.Sc. Thesis, Univ. of Crete, 2002.

[3] N. Chrysos and M. Katevenis, "Weighted Fairness in Buffered Crossbar Scheduling", Proc. of IEEE Workshop On High Performance Switching and Routing, June 2003, Torino, Italy, pp. 17–22.

[4] T. Javidi, R. Magill and T. Hrabik, "A High-Throughput Scheduling Algorithm for a Buffered Crossbar Switch Fabric", Proceedings of IEEE Int. Conf. on Communications 5, 2001, pp. 1586–1591.

[5] C. Lund, S. Phillips and N. Reingold (1996), "Fair Prioritized Scheduling in an Input-Buffered Switch" , Proc. IFIP-IEEE Conf. on Broadband Communications, Montreal, 1996, pp. 358–369.

[6] R. O. LaMaire and D. N. Serpanos, "Two-dimensional Round-Robin Schedulers for Packet Switches with Multiple Input Queues", IEEE/ACM Transactions on Networking 2(5), 1994, pp. 471–482.

[7] N. McKeown, "The iSLIP Scheduling Algorithm for Input-Queued Switches", IEEE/ACM Transactions on Networking 7(2), 1999, pp. 188–201.

[8] A. Charny, P. Krishna, N. Patel and R. Simcoe, "Algorithms for Providing Bandwidth and Delay Guarantees in Input-Buffered Crossbars with Speedup", Proc. of IEEE 6th Int. Workshop on Quality of Service, Napa, California, 1998, pp. 235–244.

[9] H. Ahmadi and W. Denzel, "A Survey of Modern High-Performance Switching Techniques", IEEE J. on Selected Areas in Communication 7(7), 1989, pp. 1091–1103.

[10] E. L. Hahne, "Round-Robin Scheduling for Max-Min Fairness in Data Networks", IEEE J. on Selected Areas in Communication 9(7), 1991, pp. 1024–1039.

[11] L. Zhang, "Virtual Clock: A New Traffic Control Algorithm for Packet Switching Networks", ACM Trans. on Computer Systems 9(2), 1990, pp. 101–124.

[12] C. H. Papadimitriou, "Algorithms, Games and the Internet", 33rd ACM Symp. on Theory of Computing, Hersonissos, Crete, Greece, 2001, pp. 749–753.

[13] P. Marbach, "Priority Service and Max-Min Fairness", Proc. of INFOCOM'02, New York, USA, 2002.

[14] R. Myerson, "Game Theory: Analysis of Conflict", Harvard University Press, Cambridge, Massachusetts, 1997.

[15] H. Gintis, "Game Theory Evolving", Princeton University Press, New Jersey, 2000.

[16] K. G. I. Harteros, "Fast Parallel Comparison Circuits for Scheduling", M.Sc. Thesis, Univ. of Crete, Greece, (http://archvlsi.ics.forth.gr/muqpro/cmpTree), TR FORTH-ICS/TR-304, 2002.

[17] D. C. Stephens and H. Zhang, "Implementing Distributed Packet Fair Queueing in a Scalable Switch Architecture", Proc. of INFOCOM'98, San Francisco, CA, 1998, pp. 282–290.

[18] H. Zhang, "Service Disciplines For Guaranteed Performance Service in Packet-Switching Networks", Proc. of IEEE 83(10), 1995, pp. 1374–1396.

[19] A. Demers, S. Keshav and S. Shenker (1990), "Analysis and Simulation of a Fair Queueing Algorithm", J. of Internetworking: Research and Experience 1(1), 1990, pp. 3–26.