Submitted 24-06-2003. *Current copy is an 'on-going' version.*

Nash Equilibria as a Fundamental Issue Concerning Network-Switches Design

GEORGE F. GEORGAKOPOULOS

Dept. of Computer Science, University of Crete, Knossou Av., Heraklion, Greece, GR-71409

ggeo@csd.uoc.gr

Abstract. This work deals with an issue of foundational value concerning the design principles of network switches: We view the 'packet-switching problem' (from N inputs towards N outputs) from the perspective of *game theory* and we prove that the already proposed 'fair solution' of *weighted max-min fair service rates* is the unique Nash equilibrium point of a natural strategic game: namely that of an auction where throughput is granted by the principle 'least-demanding first-served'. Subsequently we prove that a crossbar switching device with suitably randomized schedulers converges to this equilibrium point (and, moreover, without the need to pre-compute it).

1 Introduction

In large networks, since connecting directly each node with each other is unaffordable, switching devices are necessarily used in order to route information from many sources to many destinations. Research in this area may investigate materials, fabrication processes, designs or architectures in order to decrease production cost and achieve reliability, efficiency, etc. But since the *raison d' être* of a manufactured digital device is to be incorporated into a system and offer a *function* within this system, the engineering community faces also a more theoretically oriented issue: given some application context, which are the 'proper' functions which we should attempt to implement by our devices? And how can we be convinced that we have succeeded so?

In the 'switching problem', specifically, data packets from N sources are directed through a single network node towards N destinations. A switching device in this node has to be assigned the task to handle the relevant traffic control. The question is: *on what principle(s)* should our device operate?

On this question various active themes of research converge: designs, architectures, analyses, simulations, experiments, etc. (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].) In the literature one frequently meets 'fairness' or 'quality of service' as guiding principles regarding the functional design of a switching device. (The former principle is usually interpreted as offering to the 'users' equal shares of service and the latter as offering at least a minimum amount of service.) But terms like 'fairness' or 'quality'—however useful they may sound for everyday reference—are not, to the author's opinion, appropriate for *foundational* purposes: We should make a 'general theory of fairness', or 'of quality' available to us for this purpose...

Instead, in this work we follow and support another approach: namely that *game theory*—a deep and rich discipline—should be preferred as a tool for analyzing situations, like network design and operation, in which agents meet and compete for receiving a service.

From this perspective we offer in this work two main results regarding our subject:

(1) That under the conditions switching devices are used, they 'should' operate according to the so called 'weighted max-min fairness' principle since, as it turns out, this principle corresponds to the *unique Nash equilibrium* of a fairly natural non-cooperative strategic *auction game*.

(2) That this function (i.e., offering 'weighted max-min fairness') can indeed be attained by at least one

implementable device—in our case this can be a crossbar switch with randomized schedulers.

If something is of interest in the afore-mentioned results is that, in the practically rather important case we here deal with (network switches), we are able to answer—on firm foundational ground—both crucial issues: (a) Which is the proper function to be offered; and (b) that a specific implementable device does indeed offer this prescribed function.

This work is organized as follows: In Section 2 we review the 'weighted max-min fairness' idea referring to an abstract switching device. In Section 3 we review the basic definitions for non-cooperative strategic games, in Section 3.1 we present our switching game and prove that its unique Nash equilibrium coincides with 'weighted max-min fairness', and in Section 3.2 we digress shortly into an informal discussion about the results of Section 3.1. In Section 4 we prove that a crossbar switching device with suitably randomized schedulers indeed attains the afore-mentioned Nash equilibrium. Section 5 is an epilogue with some comments for further work.

2 The $N \times N$ switching problem and a 'fair' solution for it

2.1 An abstract switching device

Our switching device D is abstractly defined by the following:

- A *size* $N \geq 2$.
- N *inputs* $i = 1, ..., N$, and N *outputs* $j = 1, ..., N$.
- 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:

- 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.
- 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 directed to output j, or both.
- 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 our device D has devoted $p_{i,j} T$ units of time to receive input from flow (i, j) then we say that (i, j) has been served with *input rate* $p_{i,j}$. Analogously if our D has devoted $q_{i,j} T$ units of time to direct flow (i, j) to output j then we say that (i, j) has been served with *output rate* $q_{i,j}$. Assuming that a steady state is reached as $T \to \infty$, flow (i, j) is *served with rate* $r_{i,j} = \lim_{T \to \infty} s_{i,j}$.
- The ratio $u_{i,j} = r_{i,j} / w_{i,j}$ is said to be the *utility* granted to flow (i, j).

To continue, the following two definitions will be useful: 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 $\sum_{i=1}^{n} a_{i,j}$ of the elements of the j^{th} column.

- The role of the weights $\mathbf{w} = [w_{i,j}]$, $i, j = 1, ..., N$, is slightly more complex: For every flow (i, j) its weight $w_{i,j}$ is a measure of its *priority* depending in whatever way on the type of 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 **w** by setting $r_{i,j} = w_{i,j}/C_j(\mathbf{w})$— yet this is not always achievable for all i, j without sacrificing some throughput of D.

2.2 The WMMF rates for servicing $N \times N$ flows

In an attempt to find a compromise for the issue mentioned in the previous paragraph, it has been proposed that the rates of service for the $N \times N$ flows should be granted according to what is called the 'weighted max-min fairness' (WMMF) principle [1, 8]: the rate granted to any flow (i, j) 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.

We present below a simple algorithm which given a matrix of weights **w** returns the matrix $\mathbf{r} = [r_{i,j}] = \text{WMMF}(\mathbf{w})$ of the WMMF-rates w.r.t. **w**. The algorithm proceeds in rounds: We characterize each row or column of **w** as *'fixed'* or not—initially none of the rows or columns is fixed—and at each round we pay attention only to the 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 let $F_{0,j}$ denote the sum of the rates $r_{i,j}$ assigned to fixed elements along j, and let $W_{0,j}$ denote the sum of the weights of the non-fixed elements along j. We use N variables ρ_i (one for each row i), and N variables κ_j (one for each column j). The following invariant is maintained:

'for every row i: $\rho_i = (1-F_{i,0})/\ W_{i,0}$ and for every column j: $\kappa_j = (1-F_{0,j})/\ W_{0,j}$'

The WMMF-algorithm is:

```
Initialize W_{i,0}←R_i(w), F_{i,0}←0, ρ_i←1/R_i(w) and W_{0,j}←C_j(w), F_{0,j}←0, κ_j←1/C_j(w);
Repeat
  Find row i with minimum ρ_i among non-fixed rows
  Find column j with minimum 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}, ρ_i and W_{0,j}, F_{0,j}, κ_j;
Until all elements have been fixed;
For all (i,j) {set the rate r_{i,j} for flow (i,j) to u_{i,j} · w_{i,j}}
```

Notice that in the above algorithm the 'remaining throughput' $1-F_{i,0}$ for row i, or $1-F_{0,j}$ for column j, is divided among the remaining non-fixed flows along the row i or column j to be fixed, according to their relative weights: $r_{i,j} = (1-F_{i,0})\ w_{i,j}/W_{i,0}$, or $r_{i,j} = (1-F_{0,j})\ w_{i,j}/W_{0,j}$.

We can describe what the WMMF-algorithm achieves with the help of the following: 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 **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 **a** is said to *majorize* matrix **b** iff for all i, j we have $a_{i,j} \geq b_{i,j}$.

Fact 1: The WMMF-algorithm returns a matrix **r** of rates of service for which the following three hold:

(a) throughput is exhausted, i.e., **r** is a doubly stochastic matrix.

(b) the rate granted to every flow (i, j) is maximized in a 'fair sense' in the sense that it cannot be

increased unless the rate of some other flow with equal or less utility is decreased, or equivalently (as it is easily seen) the utility matrix **u** is in max-min form.

(c) every flow is granted a rate at least as large as its possible 'fair' share along its row and column, i.e., the matrix **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: if a matrix **r** satisfies conditions (a) and (b) (from which (c) follows), then **r** consists of the WMMF-rates.

Proof: The first part consists of 'folklore' facts about the WMMF-algorithm, which can be derived from it in a straightforward manner. Notice that condition (c) follows from (a) and (b): If (c) does not hold then for some i, j we must have $r_{i,j} < w_{i,j}/R_i(\mathbf{w})$ and $r_{i,j} < w_{i,j}/C_j(\mathbf{w})$. Since by (a) we have for all $i = 1, ..., N$ $R_i(\mathbf{r}) = 1$, there must exist a column l such that $r_{i,l} > w_{i,l}/R_i(\mathbf{w})$. Similarly there must exist a row k such $r_{k,j} > w_{k,j}/C_j(\mathbf{w})$. Thus $u_{i,l}$ and $u_{k,j}$ are both greater than $u_{i,j}$; yet by (b) this cannot happen.

To prove the converse suppose that matrix **r** satisfies conditions (a) and (b). Let $u_{i,j}$ be the minimum utility appearing in **u**. By condition (b) $u_{i,j}$ is the maximum element either in row i or in column j, so the utility must be constant either along row i or along column j. Let us suppose that this holds along row i (columns are treated symmetrically). By (a) we get that for $j = 1, ..., N$, $u_{i,j} = 1/R_i(\mathbf{w})$. Along row k ($k = 1, ..., N$) we cannot have $r_{k,l} > w_{k,l}/R_k(\mathbf{w})$ for all $l = 1, ..., N$, because then we would have $R_k(\mathbf{w}) > 1$ contrary to condition (a). So for some l we must have $1/R_i(\mathbf{w}) = u_{i,j} \le u_{k,l} \le 1/R_k(\mathbf{w})$, for $k = 1, ..., N$. Similarly we get $1/R_i(\mathbf{w}) \le 1/C_l(\mathbf{w})$, for $l = 1, ..., N$. Thus the minimum utility $u_{i,j}$ equals the minimum of $\rho_k = 1/R_k(\mathbf{w})$, $k = 1, ..., N$ and of $\kappa_l = 1/C_l(\mathbf{w})$, $l = 1, ..., N$. Therefore along the i^{th} row the utility $u_{i,j} = 1/R_i(\mathbf{w})$ corresponding to $r_{i,j}$, $j = 1, ..., N$, is the same as that computed by the WMMF-algorithm: in this algorithm we have initially $\rho_k = 1/R_k(\mathbf{w})$ and $\kappa_l = 1/C_l(\mathbf{w})$ and the minimum (here: ρ_i) of these is selected to become the utility along the corresponding row or column (here: i^{th} row). It is straightforward to proceed inductively by 'fixing' i^{th} row and mimicking the steps of the WMMF-algorithm. ■

However reasonable one may consider these WMMF-rates to be, adopting such a device means that a *specific* way of sharing the common resource of switching time is *enforced* to the flows by the 'authority' of this device (or its designer). Thus an issue can be raised whether (or how, when, why, etc.) such an enforcement is *justified*. On a high level we could say that WMMF-algorithm solves the problem of assigning service rates to the $N \times N$ flows, but this is uninformative since any algorithm which returns any matrix **r** such that $R_i(\mathbf{r}) \le 1$, $C_j(\mathbf{r}) \le 1$, $i, j = 1, ..., N$ does the same. On a low level we could say that the WMMF-algorithm computes an WMMF-matrix, but this is a void tautology. We do of course feel intuitively that the WMMF-algorithm 'tries' and, in some sense, succeeds to be 'fair'; yet this is a simple every-day language statement, and certainly not a mathematical characterization.

Notice that we face here a sort of *reverse engineering* issue: a *solution* is suggested (a switching fabric or, at least, a relevant algorithm, namely WMMF)—yet what we are seemingly lacking is the *problem* it solves: a situation quite suggestive of a fiercely advancing technological era. (This state of affairs is met again in other cases: see [10] about the TCP/IP protocol.) What we need here is a *well posed problem*, defined independently and irrespectively of the WMMF-algorithm, yet one of which this algorithm is the solution. In the next section we describe a natural strategic 'switching' game and we show that the WMMF-rates correspond to a Nash equilibrium point for it.

3 Network switching as a strategic game

Let us try to figure out what a network switch should achieve in order to 'satisfy' the flows passing through it. We shall view the situation as a *strategic game* with the $N \times N$ flows as its players. The game will be an 'auction' where a limited common resource—the operating time of our device—is to be granted to the players. Our game can be supposed to be *a non-cooperative* one simply because the breathtaking speed at which switching fabrics are able and expected to be operated, renders any 'cooperation' a prohibitively time-consuming luxury.

For non-cooperative games *Nash equilibrium* is a time-honoured concept, so let us recall the basic notions [3]: 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*. Every strategy profile is supposed to determine completely the result of the game. Given any strategy profile $S \in S_1 \times S_2 \times ... \times S_m$, the game is played, a final outcome is reached, and each player k enjoys a *gain* (or *payoff*) defined by a function $\text{gain}_k(S)$: $S_1 \times S_2 \times ... \times S_m \to \mathbf{R}$. The following notation will be useful: 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 strategy profile S^* such that *no player can increase its payoff* by modifying S^* *unilaterally* (here is where 'non-cooperativeness' enters the scene) i.e.:

$$\text{for all players } k = 1, ..., m \text{ and all strategies of } k,\ s \in S_k\text{: } \text{gain}_k(S^* \leftarrow [k, s]) \le \text{gain}_k(S^*)$$

Thus our question takes the following form: is there a switching game defined independently of any specific 'device' and played by the $N \times N$ flows in a network node, such that the WMMF-rates form a Nash equilibrium point of it? If yes, then achieving this equilibrium offers true service, welcomed by the players involved, since it is nothing else but the best they can obtain by themselves 'under-the-circumstances'. In the next section we answer this question in the affirmative.

3.1 Defining the 'throughput game' and characterizing its Nash equilibria

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 $P_{i,j} = 0$. Similarly for each output j, utilities $U_{i,j}$, $i = 1, ..., N$, are examined in increasing order and are granted a tentative output rate $Q_{i,j} = U_{i,j} \cdot w_{i,j}$ as long as the total rate granted remains ≤ 1. Remaining flows receive zero output rate $Q_{i,j} = 0$. The *gain* for each flow (i, j), $i, j = 1, ..., N$, is given by: $\text{gain}_{(i,j)}(\mathbf{U}) = g_{i,j} = \min(P_{i,j}, Q_{i,j})$. (Clearly, service rates cannot be greater than corresponding input or output rates.)

The following theorem characterizes the Nash equilibria for the game of Def. 2:

Theorem 3: Referring to the game of Def. 2, let $g_{i,j}$ be the granted rates, i.e., the payoff for each player (i, j) due to a strategy profile $\mathbf{U} = [U_{i,j}]$, $i, j = 1, ..., N$. Profile **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 **g** majorizes the matrix $\mathbf{f} = [\min\{p_{i,j}, q_{i,j}\}]$, where $p_{i,j} = w_{i,j}/R_i(\mathbf{w})$ and $q_{i,j} = w_{i,j}/C_j(\mathbf{w})$, $i, j = 1, ..., N$.

Proof: Necessity is proved as follows:

(a) The proof has two steps:

1. If in matrix **g** for some i, j we have $R_i(\mathbf{g}) < 1$ and $C_j(\mathbf{g}) < 1$ then we examine two cases:
   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 **U** is not a Nash equilibrium point.
   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 **U** is not Nash.
2. So if for some row i we have $R_i(\mathbf{g}) < 1$ then by (1.) above 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 $i = 1, ..., N$, $R_i(\mathbf{g}) \le 1$, 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, finally, **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 **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, and asking for less makes no sense; (2) increasing its requirement also does not help because it will not be served earlier, while it already exhausts either all input, or all output, rate available to it. ■

Combining in the obvious way Fact 1 and Theorem 3, we get that our switching game has a unique Nash equilibrium point, the granted rates of which are those computed by the WMMF-algorithm.

Unused throughput is a waste of time, and 'time is money'—if not for all, at least for communication companies, either private or public. In this latter context our result is to be interpreted as a *sound implication*: *if* throughput is to be exhausted and you have to offer your best, *then* by Nash you cannot do otherwise but strive for the WMMF-rates. Except if someone can change the (rules of the) game...

3.2 An informal discussion about the game-theoretical approach

We have shown that a natural strategic game leads to the WMMF-rates of service in switching devices. The auction-principle underlying the suggested game is a (weighted) *'least-demanding first-served'* principle, and indeed it seems quite difficult to the author to imagine a straighter rule for allocating a limited common and/or public resource. We devote below few paragraphs to discussing two issues that may be of interest to the reader:

The first issue is a reasonable question: Do we have here an *actually* played game? In real life each flow (i, j) represents two persons (or systems, *made by* and *for* persons) communicating by transferring information from i to j. If persons are in fact the players, shouldn't everyone involved be already aware of the game? We consider this issue as a delicate one: On the one hand the answer is, by now, 'no': As evolutionary game theory has revealed [11, 12] game theory can explain phenomena involving very simple organisms in which no rationality or even awareness is observed. On the other hand awareness is usually 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 clarification of the difference—of foundational value—between the *fairness* and the *game-equilibrium* approach: In the former the designer invents a notion of fairness (in our case 'equal utilities') and designs a system or device that deviates from it the least possible (in our case: the 'WMMF-rates'). Notice that in our case the WMMF-rates do not correspond to equal utilities: utilities are finally granted unequally; yet this is excused by an appeal to the principle of 'exhausting throughput'. But in what way is this latter principle connected to 'fairness'? To dramatize this state of affairs let us imagine the designer appearing in front of the users and claiming: «This is what I have designed. You have to accept it because it is fair and because you must not act selfishly». But in this scenario nobody never asks the users nothing. For example, what could be the answer to an objecting user: «So you offer to my market-opponent higher utility in the name of throughput—but why must I consider this as 'fair' to me?».

In the latter case (of obtaining a game-equilibrium) the designer *observes* the strategic game users are indeed—knowingly of unknowingly—playing and computes its equilibrium points. (In our case, this game is an auction for throughput, in which the least demanding user is served first.) Subsequently she designs a system or device that offers such an equilibrium, she reappears in front of the users and says: «This is the game you are involved in, and this is *provably* the best each one of you can obtain from it, *however selfishly each of you may act*. The device I have invented *obtains the same instead of you*». No discussions about what is fair, or, moreover, what could be a reasonable approximation to it. What is offered instead is an *exact optimum*—something that no one can improve for itself. The complaining user can still stand up and say: «Well, I don't like this game after all... », but now there is an answer: «We respect your objections, but the game *is your* choice, *not ours*. If you change the game we shall (try to) provide you with the new corresponding solution».

These two described approaches, miraculously (?) coincide in the case of the WMMF-rates for network

switches. However relieving this may be, we see no guarantee that this will be always the case.

4 A randomized switch converging to the Nash equilibrium

We now know, by the discussion in Sections 2 and 3, what rates of service our switching device D 'must' offer (the Nash-rates), and we know how to compute these rates (the WMMF-algorithm). But our device must be *operational*, i.e., achieve actually this rates. An attractive architecture for such a device is the 'crossbar switch': N inputs are connected to N outputs through an $N \times N$ grid of buffers of size B (implementable easily and cheaply *on-chip* [1]). Data directed from input i to output j can be stored temporalily, if needed, to buffer (i, j).

More specifically D is defined by what is mentioned in Section 2.1, plus the following:

- 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*.
- N *schedulers*: one S_i^{in} for each row i, and one S_j^{out} for each column j. Schedulers run a scheduling algorithm (normally the same for all i and j, yet applied on different data). Each scheduler returns a number in $[0, N]$.

The intended meaning of the above is the following:

- Device D operates at discrete *time-steps*, $t = 1, 2, 3, \ldots$. At each time-step for each $i = 1, \ldots, N$ 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 $b_{i,l} + 1$. Similarly for each $j = 1, \ldots, N$ output-scheduler S_j^{out} selects a row $k \in [0, N]$. If $k \neq 0$ and (k, j) is not empty then a packet is transferred from buffer (k, j) to output j and $b_{k,j}$ is set to value $b_{k,j} - 1$.

Thus we have to design practically implementable schedulers S_i^{in} and S_j^{out} capable of obtaining the Nash equilibrium point of Section 3.1. Notice that, with a device cycle of just a few nano-seconds, even computing the WMMF-rates by running the algorithm of Section 2.2 is prohibitively time and hardware consuming. Instead we shall show (inspired by [1]) that a suitably randomized set of schedulers can make our device converge to the WMMF-rates without pre-computing them.

Another view of the WMMF-algorithm of Section 2.2 is the following: Given a weight-matrix $\mathbf{w}$, a pair of matrices can be defined by $p_{i,j} = w_{i,j} / R_i(\mathbf{w})$ and $q_{i,j} = w_{i,j} / C_j(\mathbf{w})$, $i, j = 1, \ldots, N$. In such a matrix-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. In any such pair of matrices, at least one majorized row or column will *always* exist: Setting $V = \{R_k(\mathbf{w}): k = 1, \ldots, N\} \cup \{C_l(\mathbf{w}): l = 1, \ldots, N\}$ 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 $i = 1, \ldots, N$ 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) $l = 1, \ldots, N$, $l \neq j$, 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. If we repeat the above procedure until all rows and all columns have been examined, we shall have obtained the WMMF-rates.

The scheduling algorithm of our device D is the following *oblivious repetitive sampling*:

S_i^{in} : 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} : pick $i \in [1, N]$ with probability $q_{i,j} = w_{i,j} / C_j(\mathbf{w})$ until a non-empty buffer (i, j) is encountered.

We shall prove the following:

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

Proof: In order to prove Theorem 4 we have to do some preliminary work. Let C be the following finite Markov chain: We consider one buffer of size B, which 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 *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 also *probed* for output with probability q and if it is not empty it delivers a packet. Input an output probing is performed independently of each other and independently of the state of our buffer. We give below the transition matrix for this Markov chain for $B = 4$, (expression in row $i = 0, ..., 4$ and column $j = 0, ..., 4$ denotes the probability for the buffer to pass from state i to that state j):

	0	1	2	3	4
0	$(1-p)$	p	0	0	0
1	$(1-p)q$	$pq+(1-p)(1-q)$	$p(1-q)$	0	0
2	0	$(1-p)q$	$pq+(1-p)(1-q)$	$p(1-q)$	0
3	0	0	$(1-p)q$	$pq+(1-p)(1-q)$	$p(1-q)$
4	0	0	0	q	$(1-q)$

Consider the steady state of this Markov chain. We define the service-rate function $s(\cdot,\cdot)$ as follows:

$$s(p, q) = \text{the rate of service achieved by our buffer if it is probed for input with probability } p, \text{ and it is probed for output with probability } q. \tag{1}$$

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. Since receiving and delivering are made independently of each other and independently of the state of our buffer, 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 (since what comes-in eventually gets-out of a finite buffer). So the following holds for $e(p, q)$ and $f(p, q)$:

$$f(p,q) = \frac{p - s(p,q)}{p}, \text{ and } e(p,q) = \frac{q - s(p,q)}{q} \tag{2}$$

With the help of the service rate $s(\cdot,\cdot)$ we can formulate a set of equations describing how the schedulers of our abstract device D decide which row or column to choose. 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 two equations hold for every (i, j):

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

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

Eq. (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) ($k = 1, ..., N,\ k \neq j$) as follows:
  - Probe buffer (i, k) at least once (with probability $\overline{p}_{i,k}$ by the definition of $\overline{p}_{i,k}$),
  - Find that buffer (i, k) is full (with probability $f(\overline{p}_{i,k}, \overline{q}_{i,k})$),
  - '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 $p_{i,j} \sum_{m=0}^{\infty} p_{i,k}^{m} = p_{i,j}/(1 - p_{i,k})$).

To get each term in Eq. (3) we multiply the relevant probabilities because the sampling steps are made independently of each other and independently of the state of our buffer, and we add all terms since they refer to disjoint sets of probing sequences. Notice that from Eq. (3) we get $\sum_{j=1}^{N} \overline{p}_{i,j} \left(1 - f(\overline{p}_{i,j}, \overline{q}_{i,j})\right) = 1$ (as expected). Eq. (4) arises similarly.

Let $\mathbf{P}$ (resp. $\mathbf{Q}$) 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 Eq. (3) and (4), so we have to view the above $2N^2$ equations as an operator $\mathbf{F}(\cdot,\cdot)$: $\mathbf{P} \times \mathbf{Q} \rightarrow \mathbf{P} \times \mathbf{Q}$ 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 $\min(\overline{\mathbf{p}}, \overline{\mathbf{q}}) = \text{WMMF}(\mathbf{w})$, i.e., the service rates achieved are the WMMF-rates.

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

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

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

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 $i = 1, ..., N$ 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.

A straightforward induction—passing these fixed values to the rest of equations of $\mathbf{F}$ and ignoring in further rounds all fixed rows or columns—suffices to prove our lemma. ■

The lemma below states that the service-rate function $s(\cdot,\cdot)$ of a single buffer (recall the finite Markov chain C that we have already defined) behaves 'in-the-limit' as desired:

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

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

$$\lambda = \frac{q(1-p)}{p(1-q)}, \quad \mathbf{V_B} = \sigma\left((1-q)\frac{\lambda^B}{1-p}, \frac{\lambda^{B-1}}{1-p}, ..., \frac{\lambda^2}{1-p}, \frac{\lambda^1}{1-p}, 1\right) \tag{7}$$

Vector $\mathbf{V_B}$ is an eigenvector of C with eigenvalue 1, where σ is just a scale factor selected to make the

sum of the components equal to 1: this can be easily verified since the transition matrix of C has only 5 types of columns. For example for the 1st component of $\mathbf{V_B}$ we have to verify that:

$$\left((1-q)\frac{\lambda^B}{1-p},\ \frac{\lambda^{B-1}}{1-p},\ ....,\ 1\right)\cdot\begin{pmatrix}(1-p) & (1-p)q & \dots & 0\end{pmatrix}^{\mathrm{T}} =_{?} \ (1-q)\frac{\lambda^B}{1-p}, \tag{8}$$

which is indeed the case, etc. Thus $\mathbf{V_B}$ gives indeed 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 Eq. (2) we get $s(p,q) \to q = \min\{p,q\}$. If $p < q$ we get similarly that the probability $f(p, q)$ of the buffer to be full (i.e., the last component V_B) tends to zero as $B \to \infty$, so again by Eq. (2) 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 probabilies for the buffers—given by the fixed point $(\overline{\mathbf{p}},\overline{\mathbf{q}})$ of $\mathbf{F}$—will satisfy in-the-limit $\min(\overline{\mathbf{p}},\overline{\mathbf{q}}) = \text{WMMF}(\mathbf{w})$. Since $s(\cdot,\cdot)$ tends to $\min\{\cdot,\cdot\}$ the achieved rates will be asymptotically equal to the WMMF-rates. This establishes Theorem 4, by appealing to the continuity of the fixed point of $\mathbf{F}$ w.r.t. to $s(\cdot,\cdot)$ (a technical fact, the proof of which is omitted). ■

Yet our probes, for the sake of being independent, are too oblivious (we may again and again probe for input an already probed full buffer) and too repetitive (sampling repetitively, until we find an eligible buffer, may take an unbounded amount of time). Thus our schedulers are not suitable for an actual fast implementation unless we modify them so that they sample directly only eligible buffers. The following simple fact allows us to do so:

Lemma 6: Let $\mathbf{p}$ be a probability distribution over elements $[1, n]$ and let a subset $G \subseteq [1, n]$ of them be the only 'eligible' ones. We select repetitively an element in $[1, n]$ according to $\mathbf{p}$ until we select one in G. Let p_k^* be the probability that element k is selected. Then this probability equals that of selecting directly an eligible element w.r.t. the relative probability distribution, i.e., $p_k^* = p_k / \sum_{i\in G} p_i$ for $k \in G$ and $p_k^* = 0$ for $k \notin G$.
Proof: (Straightforward.) ■

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

5 Epilogue and further work

In this work we followed a game-theoretical approach for investigating what should be the proper function of a network switch 'from N inputs to N outputs', and we concluded that if throughput is granted by an 'auction-game' then the WMMF-rates form its unique Nash equilibrium. Thus if throughput is our main concern anything else than the WMMF-rates would be sub-optimal for the users. Subsequently we proved that this equilibrium can be achieved by a crossbar switching device with randomized schedulers.

Concerning the first part of this work all directions are open: a maximal target would be to apply it to all sorts of similar problems in network design.

Concerning the second part of this work 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 [13, 14, 15, 16] for the same purpose? We do have some positive results along this direction but we cannot yet be fully conclusive.

Acknowledgements

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

References

[1] N. Chrysos and M. Katevenis (2002), *"Transient Behavior of a Buffered Crossbar Converging to Weighted Max-Min Fairness"*, Institute of Computer Science, FORTH, Greece, (http://archvlsi.ics.forth.gr/bufxbar/).

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

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

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

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

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

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

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

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

[10] C. H. Papadimitriou (2001), *"Algorithms, Games and the Internet"*, 33rd ACM Symposium on Theory of Computing, Hersonissos, Crete, Greece, 749–753.

[11] R. Myerson (1997), *"Game Theory: Analysis of Conflict"*, Harvard University Press, Cambridge, Massachusetts.

[12] H. Gintis (2000), *"Game Theory Evolving"*, Princeton University Press, New Jersey.

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

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

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

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