Convergence to multi-resource fairness under end-to-end window control

Thomas Bonald, James Roberts, Christian Vitale

Convergence to multi-resource fairness under end-to-end window control

Thomas Bonald
Télécom ParisTech,
Université Paris-Saclay, France
thomas.bonald@telecom-paristech.fr

James Roberts
SystemX, France
(now retired)
james.walter.roberts@gmail.com

Christian Vitale
IMDEA Networks Institute and
Universidad Carlos III de Madrid, Spain
christian.vitale@imdea.org

***Abstract*—The paper relates to multi-resource sharing between flows with heterogeneous requirements as arises in networks with wireless links or software routers implementing network function virtualization. Bottleneck max fairness (BMF) is a sharing objective in this context with good performance. The paper shows that BMF results when local fairness is imposed at each resource while flow rates are controlled by an end-to-end window. We analytically prove convergence to BMF under a fluid model when flows share a network limited to 2 resources while numerical results confirm BMF convergence for larger networks. Simulation results illustrate the impact of packetized transmission.**

I. Introduction

Elastic flows in the Internet share multiple resources under the joint impact of end-to-end congestion control protocols and router queue management mechanisms. In the wired Internet the resources in question are links and all flows have the same requirement for each bit/s of rate. Requirements are not homogeneous for a wireless link, however, where the amount of spectrum consumed for each bit/s depends on the flow's radio conditions. Heterogeneous requirements occur also when flows share compute resources in a software router. Some flows require simple forwarding while others require complex processing, for encryption, say, or other virtualized functions, and consume more CPU.

Resource sharing by flows with homogeneous requirements has been widely studied over many years. Of particular interest for the present work is the observation made some 30 years ago that network-wide max-min fairness is realized by implementing fair queuing in router queues and performing window-based flow control [8], [10]. This claim was proved by Hahne for a synchronous time-slotted network model with flow rates controlled by *hop-by-hop* windows [7]. To our knowledge there is no published proof that max-min fair sharing occurs with *end-to-end* window control, as used in TCP/IP.

Our main objective here is to derive the equivalent result in the case of heterogeneous resource requirements. This is needed for today's network, where most flows use at least one wireless link, and tomorrow's, where compute resources are potential bottlenecks in dynamically provisioned software routers. Explicitly, we show that imposing local fairness at each resource, coupled with end-to-end flow control results in a desirable generalization of max-min fairness called bottleneck max fairness (BMF) [2].

Proving this result is hard due to the complex dynamics of per-flow resource queue backlogs. For homogeneous requirements, the proof that backlogs eventually converge and rates stabilize at their max-min fair shares was the culmination of several years doctoral thesis work by Hahne. Chrysos and Katevenis have since derived a somewhat simpler proof, thanks to their use of a fluid model, but this is still highly non-trivial and again confined to hop-by-hop window control [3].

We apply the same fluid model as Chrysos and Katevenis to prove convergence to BMF for flows with heterogeneous requirements in a network limited to two bottleneck resources. It is considerably harder to account for heterogeneous requirements because the water filling characterization of max-min fairness used in [7] and [3] does not generalize to BMF. In addition, we have used water filling with insights gained from the analysis of BMF to derive an original proof of convergence for a general network with homogeneous requirements under end-to-end window control.

The fluid model enables the mathematical analysis but, in practice, the theoretical objectives of multi-resource sharing can only be achieved approximately. It is therefore important to examine the behavior of a more realistic packet-based model to understand how this deviates from the ideal. We simulate a network where resources implement start time fair queuing [6] and investigate the impact on convergence times of window size, the number of competing flows and their particular requirements.

In the next section, we argue the need to perform fair resource scheduling for heterogeneous requirements and discuss properties of the resulting BMF allocation. Section III introduces the dynamical system describing the evolution of backlogs in the fluid model limit. The main convergence results for this dynamical system are given in Section IV. Section V presents simulation results that illustrate deviations from the fluid model ideal when accounting for finite sized packets.

This work was performed at LINCS, www.lincs.fr. C. Vitale was supported by the Spanish Ministry of Economy and Competitiveness under HyperAdapt grant (ref: TEC2014-55713-R) and by the European Commission in the framework of the H2020-ICT-2014-2 project Flex5Gware (grant agreement no. 671563). His research was also partially supported by the Madrid Regional Government through the TIGRE5-CM program (S2013/ICE-2919).

II. Multi-resource sharing

We discuss why scheduling is required for fair multi-resource sharing before recalling the desirable properties of BMF. The main symbols used are listed in Table I.

R	number of resources
n	number of flows
C_j	capacity of resource j
A_{ij}	requirement of one unit of flow i at resource j
$a_{ij} = A_{ij}/C_j$	normalized requirement
$p(i,j)$	resource visited by flow i prior to visiting resource j
ϕ_{ij}	rate at which flow i leaves resource j
φ_i	rate allocated to flow i
f_j	fair share at resource j
Q_{ij}	backlog of flow i at resource j
b_{ij}	backlog indicator, 1 if $Q_{ij} > 0$, 0 otherwise
W_i	end-to-end window of flow i in bits
$W^{(p)}$	end-to-end window in packets

TABLE I
Summary of notation.

A. Need for scheduling

Consider a network with R resources shared by n flows. Resource j has capacity of C_j units per second where units are resource dependent. Flow i requires A_{ij} units of resource j to process each bit. The rate in bit/s, φ_i, allocated to flow i must satisfy the capacity constraints

$$\sum_{i=1}^{n} A_{ij}\varphi_i \leq C_j \tag{1}$$

for $1 \leq j \leq R$. $A_{ij}\varphi_i$ is the amount of resource j used per second by flow i.

The following are examples of envisaged resource types,

- wired link: C_j is measured in bit/s; $A_{ij} = 1$ if flow i uses link j and $A_{ij} = 0$ otherwise;
- LTE wireless link: C_j is measured in time slot/s; A_{ij} is the fractional number of slots needed to transmit each bit of flow i accounting for the flow's radio conditions; the requirement can be more than 20 times smaller for a user close to the antenna than for a user at the cell edge;
- software router CPU: C_j is measured in cycle/s and A_{ij} is the number of cycles needed to process one bit of flow i; this number varies widely depending on the type and number of functions to be processed (e.g., simple forwarding, encryption, transcoding).

Scheduling is generally absent in the wired Internet and bandwidth sharing is realized by means of congestion control protocols like TCP that react to drop signals received from FIFO buffers. Sharing is generally fair enough if users implement the same protocol [15] though it has often been noted that fair queuing implemented in router queues would provide more robust control, e.g., [11], [1].

For a wireless link, where requirements are highly variable, it is generally considered preferable to aim for equal resource shares, $A_{ij}\varphi_i$, rather than equal bit rates. This is broadly what the proportional fair scheduler achieves [16], as implemented in 3G and 4G cellular networks. That the IEEE 802.11b scheduler tends to realize max-min fair *bit rates* was recognized as a performance anomaly [9].

In NFV, the dynamic provision of compute capacity implies CPU may become a temporary bottleneck and the way it is shared is therefore an issue. Simple FIFO queuing coupled with end-to-end congestion control would lead to approximately equal flow bit rates, as in a wired Internet. However, if requirements differ significantly, max-min fair rates would produce the same "performance anomaly" as in 802.11b. In an early paper considering dual, CPU and bandwidth, resource sharing [13], Shin and co-authors proposed an ECN marking scheme intended to realize proportional fairness through TCP congestion control. We would argue that the use of a scheduler to equalize CPU usage among flows constitutes a more satisfactory resource sharing solution.

In this paper we adopt the position that flows using a single resource considered in isolation should receive max-min fair resource shares. Let ϕ_{i0} be the incoming bit rate of flow i at resource j. The scheduler allocates a fair share $A_{ij}\varphi_i = f_j C_j$ to any flow i such that $\varphi_i < \phi_{i0}$, and allocates $A_{ij}\varphi_i = A_{ij}\phi_{i0}$ to the others, where fair share f_j is determined by the capacity constraint (1). Rates φ_i defined thus are weighted max-min fair with weights $1/A_{ij}$. It is well-known that this allocation can be realized approximately using packet schedulers such as DRR [14] or SFQ [6].

B. Bottleneck max fairness

Ghodsi and co-authors introduced the problem of multi-resource sharing in compute clusters [5] and extended their analysis to networks [4]. They advocate so-called dominant resource fairness (DRF). In networking applications, DRF requires schedulers at each resource to implement weighted max-min fairness with the same flow weight w_i applied at each resource determined from the dominant relative resource requirement, $w_i = 1/\max_j\{A_{ij}/C_j\}$. This choice is motivated by a requirement that the allocation be *strategyproof*: flows should not be able to gain a greater bit rate by falsely stating their requirements.

The plausibility of designing and implementing such a gaming strategy in a context of dynamic demand in a network setting is highly debatable, however. We have previously advocated an alternative allocation that sacrifices strict strategyproofness in order to achieve a better efficiency–performance tradeoff [2]. This allocation is called bottleneck max fairness (BMF).

Like max-min fairness, BMF is defined for a fluid model where packet size is infinitesimally small and resource capacity is perfectly divisible among flows. The allocation is such that resource sharing is Pareto efficient (i.e., all capacity is used if possible) and every flow receives the maximum allocation at some resource that is fully used. This may be recognized as one of the definitions of max-min fair resource sharing, e.g., [12]. The significant difference here derives from the heterogeneous requirements A_{ij} in capacity constraints (1).

It was shown in [2] that the BMF allocation always exists and has all the desirable sharing properties identified by Ghodsi *et al.* [4] except strategyproofness. On the other hand, it has an alternative property called *single-bottleneck fairness*:

if the network has a unique bottleneck j, the allocation is such that $A_{ij}\varphi_i = 1/n$ for all flows i. That this property is not shared by DRF largely explains its inferior throughput performance.

A significant advantage of BMF in networking applications is that the allocation can be realized simply by implementing weighted fair queuing independently at each resource with weights for flow i at resource j equal to $1/A_{ij}$. The main objective of this paper is to justify this statement.

C. Scheduling and window-based flow control

We suppose flow i maintains a fixed volume W_i of unacknowledged data and every resource j realizes weighted max-min fair sharing with weights $1/A_{ij}$. Assume the network attains a steady state with constant flow rates φ_i, constant queue backlogs and constant round trip times. We have the following proposition [2].

Proposition 1: Suppose flows implement a large enough fixed window and resources realize weighted fair queuing. If the network attains a steady state, the realized flow rates are bottleneck max fair.

Proof. The proof is immediate as the window can be made large enough that every flow has at least one bottleneck resource (i.e., it has a backlog and the resource is therefore fully used) while the scheduler ensures its share of that resource is maximal. These, with Pareto efficiency, are the conditions that define BMF. □

This proposition also applies to max-min fairness as a special case of BMF and its equivalent was stated by Hahne [7]. All the difficulty in proving the controls yield BMF is in proving the system does in fact converge to a steady state.

III. A DYNAMICAL SYSTEM

We present the dynamical system governing the evolution of the resource queue backlogs and flow rates under a fluid model with zero propagation times. We assume resources are consumed successively in the order defined by a flow-specific route: $p(i,j)$ designates the resource visited by flow i prior to its visit to resource j[1]. In this and the next section, to simplify the formulas, we use normalized requirements $a_{ij} = A_{ij}/C_j$.

A. Persistent binary system states

Let $\phi_{ij}(t)$ denote the rate at which flow i is served by resource j at time t. For brevity, we generally omit the explicit dependence on time in this and other variables. Flow rates ϕ_{ij} depend on the backlogs at each queue Q_{ij} or, more succinctly, on the backlog status indicators b_{ij},

$$b_{ij} = \begin{cases} 1, & \text{if } Q_{ij} > 0, \\ 0, & \text{if } Q_{ij} = 0. \end{cases}$$

[1]The formulation could be extended to allow simultaneous consumption of sets of resources. Assuming packets are processed in parallel at these resources one at a time, the corresponding fluid model would instantly realize the same flow rate at each resource in question.

In periods where the $b_{ij}(t)$ are constant, rates ϕ_{ij} are also constant and satisfy the following equations

$$a_{ij}\phi_{ij} = \begin{cases} f_j, & \text{if } b_{ij} = 1, \\ a_{ij}\phi_{ip(i,j)}, & \text{if } b_{ij} = 0, \end{cases}$$

where

$$f_j = \frac{1 - \sum_{k=1}^n a_{kj}\phi_{kj}\mathbb{1}\{b_{kj} = 0\}}{\sum_{k=1}^n \mathbb{1}\{b_{kj} = 1\}}.$$

These equations can be rewritten:

$$a_{ij}\phi_{ij} = a_{ij}\phi_{ip(i,j)}(1-b_{ij}) + \frac{b_{ij}(1 - \sum_{k=1}^n a_{kj}\phi_{kj}(1-b_{kj}))}{\sum_{k=1}^n b_{kj}}. \tag{2}$$

They express the result of per-resource weighted max-min fair schedulers, as described at the end of Section II-A.

To avoid unhelpful complications, we suppose the a_{ij} are such that linear equations (2) are independent and therefore yield a unique set of ϕ's for each binary state vector b. For many such vectors, the computed ϕ's will not in fact be feasible (e.g., they might be negative). Vectors that do yield a set of feasible rates constitute the space of valid *persistent binary states*.

B. Evolution between persistent states

The system evolves between persistent states as follows. All backlogs for which $\phi_{ij} > \phi_{ip(i,j)}$ are decreasing in time. Let the queue of flow i^* at resource j^* be the one to empty first. At this instant the system enters a new state b' where $b'_{i^*j^*} = 0$ and $b'_{ij} = b_{ij}$ for the other queues.

If b' is a persistent state (i.e., there is a feasible solution to the new instance of equations (2)), the system will enter a new phase with a new set of rates ϕ' which persist until a new queue empties. If b' is not persistent, some of the non-backlogged queues will immediately become backlogged because the flow's incoming rate exceeds its weighted fair share. In case b' is not persistent, the rates ϕ' satisfy the following equations:

$$a_{ij}\phi'_{ij} = \begin{cases} f_j, & \text{if } a_{ij}\phi'_{ip(i,j)} \geq f_j \text{ or } b'_{ij} = 1, \\ a_{ij}\phi'_{ip(i,j)}, & \text{otherwise,} \end{cases} \tag{3}$$

where the fair share satisfies

$$f_j = \frac{1 - \sum_k a_{kj}\phi'_{kj}\mathbb{1}\left\{b'_{kj} = 0 \text{ and } a_{kj}\phi'_{kp(k,j)} < f_j\right\}}{\sum_k \mathbb{1}\left\{b'_{kj} = 1 \text{ or } a_{kj}\phi'_{kp(k,j)} \geq f_j\right\}}.$$

The new persistent state is b'' where $b''_{ij} = 1$ if $b'_{ij} = 0$ and $\phi'_{ip(i,j)} > \phi'_{ij}$, and $b''_{ij} = b'_{ij}$ otherwise. Notice that b'' is indeed a persistent state since the ϕ' satisfy (2) with b replaced by b''.

C. Convergence

The network evolves between persistent states until it enters one in which the service rates at all resources are the same for each flow. If this occurs, the rates in question are BMF by Proposition 1.

The equations defining the dynamical system can be rapidly solved numerically allowing us to explore convergence over a wide range of parameter values (e.g., 1 million random choices). Convergence indeed always occurs in all our experiments. The graph defined by valid transitions between persistent states is acyclic. There are cases where the BMF allocation is not unique [2]. In such cases, numerical experiments show the system converges to one or another of the possible allocations, depending on the assumed initial backlogs $Q_{ij}(0)$.

Unfortunately, it proves very difficult to analytically prove convergence in general. The next section proves convergence for some significant special cases.

IV. Proof of convergence

We first prove convergence to BMF for networks of 2 resources consumed successively before discussing the difficulty of extending this result to more resources[2]. We then prove convergence for $R \geq 2$ resources in the special case where BMF reduces to max-min fairness.

A. BMF for 2 resources

We consider a system with 2 resources and n flows. The capacity of all resources is normalized to 1. To avoid some tedious qualifications, we suppose here that $a_{ij} > 0$ and the ratios a_{i1}/a_{i2} are distinct.

Inspecting the results of numerous simulations, we observed that the evolution of backlogs is such that one queue in particular is never empty in any persistent state. The resource in question is $j^* = \arg\max\{\sum_i a_{ij}/a_{ip(i,j)}\}$ and the flow is $i^* = \arg\max\{a_{ij^*}/a_{ip(i,j^*)}\}$. For convenience and without loss of generality we therefore renumber the resources such that

$$\sum_{i=1}^{n} \frac{a_{i1}}{a_{i2}} > \sum_{i=1}^{n} \frac{a_{i2}}{a_{i1}}, \tag{4}$$

and the flows such that

$$\frac{a_{11}}{a_{12}} > \frac{a_{21}}{a_{22}} > \cdots > \frac{a_{n1}}{a_{n2}}. \tag{5}$$

The queue that never empties is then that of flow 1 at resource 1. The following three lemmas allow us to affirm convergence in Theorem 1. Lemmas 1 and 3 are proved in the appendix while the proof of Lemma 2 is symmetrical to that of Lemma 1 and is therefore omitted.

Lemma 1: Flow 1 always stabilizes to a backlog only at resource 1. Given flows 1 to $r-1$ are backlogged only at resource 1, for $1 < r \leq n$, a sufficient condition that flow r also stabilizes to a backlog only at 1 is

$$\frac{a_{r1}}{a_{r2}} > \frac{n - r + 1 - \sum_{i=r}^{n} \frac{a_{i1}}{a_{i2}}}{r - 1 - \sum_{i=1}^{r-1} \frac{a_{i2}}{a_{i1}}}. \tag{6}$$

Lemma 2: A sufficient condition for flow n to stabilize to a backlog only at resource 2 is $\sum_{i=1}^{n} a_{i2}/a_{i1} > n$. Given flows $n-s+1$ to n, for $s \geq 1$, are backlogged only at resource 2, a sufficient condition that flow $n-s$ also stabilizes to a backlog only at 2 is

$$\frac{a_{(n-s)2}}{a_{(n-s)1}} > \frac{n - s - \sum_{i=1}^{n-s} \frac{a_{i2}}{a_{i1}}}{s - \sum_{i=n-s+1}^{n} \frac{a_{i1}}{a_{i2}}}. \tag{7}$$

Lemma 3: Given the first $r-1$ flows for $r \geq 1$ have converged to a backlog only at resource 1 and the last s flows for $s \geq 0$ have converged to a backlog only at resource 2, either flow r will converge to a resource 1 backlog, or flow $n-s$ will converge to a resource 2 backlog.

Lemma 3 shows that the system will eventually converge to a state where the first k flows stabilize with a backlog only at resource 1 while the remainder stabilize with a backlog only at resource 2 for some k, $1 \leq k \leq n$. The order of convergence depends on initial backlogs and the various flow window sizes. The lemma implies that convergence will occur at least as fast as if first flow 1 stabilizes, then either flow 2 or flow n, and so on, proceeding from either the lowest or the highest numbered remaining flow. In this order, the last flow to stabilize is either flow k or flow $k+1$. The value of k characterizes the BMF allocation and is specified in the following theorem.

Theorem 1: When each resource locally realizes weighted max-min fair sharing with respective weights $1/a_{ij}$, flow rates converge to the BMF allocation. When resources and flows are labelled such that (4) and (5) hold, flows 1 to k are backlogged only at resource 1 and the remainder only at resource 2, where $k \geq 1$ is the index such that

$$\frac{\sum_{i=k}^{n}(1 - \frac{a_{i1}}{a_{i2}})}{\sum_{i=1}^{k-1}(1 - \frac{a_{i2}}{a_{i1}})} < \frac{a_{k1}}{a_{k2}} \quad \text{and} \quad \frac{\sum_{i=k+1}^{n}(1 - \frac{a_{i1}}{a_{i2}})}{\sum_{i=1}^{k}(1 - \frac{a_{i2}}{a_{i1}})} > \frac{a_{(k+1)1}}{a_{(k+1)2}}. \tag{8}$$

The BMF allocations are $\varphi_i = f_1/a_{i1}$, for $1 \leq i \leq k$, and $\varphi_i = f_2/a_{i2}$, for $k < i \leq n$, where f_1 and f_2 are the fair shares

$$f_1 = \frac{n - k - \sum_{i=k+1}^{n} \frac{a_{i1}}{a_{i2}}}{k(n-k) - \sum_{i=k+1}^{n} \frac{a_{i1}}{a_{i2}} \sum_{i=1}^{n} \frac{a_{i2}}{a_{i1}}}, \tag{9}$$

$$f_2 = \frac{k - \sum_{i=1}^{k} \frac{a_{i2}}{a_{i1}}}{k(n-k) - \sum_{i=k+1}^{n} \frac{a_{i1}}{a_{i2}} \sum_{i=1k}^{n} \frac{a_{i2}}{a_{i1}}}. \tag{10}$$

Proof. We know from Lemmas 1, 2 and 3 that the backlogs of all flows eventually stabilize and Proposition 1 shows the resulting allocation is BMF.

Condition (6) may be written $a_{r1}/a_{r2} > N(r)/D(r)$ where $N(r)$ and $D(r)$ are the numerator and denominator of the right hand side, respectively. It may readily be verified that $a_{k1}/a_{k2} > N(k)/D(k) \Rightarrow a_{r1}/a_{r2} > N(r)/D(r)$ for $r < k$. This condition therefore ensures by Lemma 1 that the first k flows stabilize with a backlog only at resource 1.

If now $a_{(k+1)1}/a_{(k+1)2} < N(k+1)/D(k+1)$, which is the second inequality in (8), we can similarly show that $a_{r1}/a_{r2} < N(r+1)/D(r+1)$ for $k+1 \leq r < n$. Condition (7) is thus satisfied for flows $k+1$ to n completing the proof that (8) characterizes the stabilized backlogs.

[2]If the resources are consumed simultaneously, the fluid model is instantaneously stable since there is only one common queue which holds the entire window.

Finally, fair shares (9) and (10) follow from the equations

$$f_1 = \frac{1 - \sum_{i=k+1}^{n} \frac{a_{i1}}{a_{i2}} f_2}{k},$$

$$f_2 = \frac{1 - \sum_{i=1}^{k} \frac{a_{i2}}{a_{i1}} f_1}{n-k}.$$

□

B. More than 2 resources

While all our simulations of the dynamical fluid system confirm convergence, at the time of writing, we have not been able to prove this analytically. We observe in the numerical results that, in every case, the backlog of one flow at one resource is always decreasing or empty. Unfortunately, this queue is not identified by a simple generalization of the characterization discovered for 2-resource networks. It depends on the routing but the same BMF allocation results from all possible routings. The next section provides an original proof of convergence for a network of any size in the special case of max-min fairness.

C. Max-min fairness

The resources considered here are wired network links. Consider a network of R links of capacities C_j, for $1 \leq j \leq R$, shared with max-min fairness by n flows. Flow i has requirement $A_{ij} = 1$ when it uses link j and $A_{ij} = 0$ otherwise. Flow i maintains a window of W_i unacknowledged data. The proof of convergence of the corresponding dynamical system derives from the well-known water filling definition of max-min fairness, as used previously by Hahne [7] and Chrysos and Katevenis [3] for hop-by-hop window control.

Let $\mathcal{J}_1$ be the set of order 1 bottlenecks defined by

$$\mathcal{J}_1 = \arg\min_{1 \leq j \leq R} \frac{C_j}{\sum_i A_{ij}}.$$

$\mathcal{J}_1$ contains the links that are simultaneously saturated first in the water filling procedure. Let $\mathcal{I}_1$ be the set of flows i such that $A_{ij} = 1$ for $j \in \mathcal{J}_1$, i.e., the flows that use at least one of the order 1 bottlenecks. Now define recursively bottleneck sets of order x and corresponding flow sets $\mathcal{I}_x$ by

$$\mathcal{J}_x = \arg\min_{j \notin \{\cup_{y<x} \mathcal{J}_y\}} \frac{C_j - \sum_{i \in \mathcal{I}_{x-1}} \varphi_i A_{ij}}{\sum_{i \notin \mathcal{I}_{x-1}} A_{ij}},$$

where φ_i is the weighted max-min fair rate for flow i and

$$i \in \mathcal{I}_x \iff A_{ij} = 1 \text{ for } j \in \mathcal{J}_x \text{ or } i \in \mathcal{I}_{x-1}.$$

Links $\mathcal{J}_x$ are saturated in step x of the water filling procedure.

The weighted max-min rates are given recursively by

$$\varphi_i = \min_{1 \leq j \leq R} \frac{C_j}{\sum_k A_{kj}}, \tag{11}$$

for $i \in \mathcal{I}_1$, and

$$\varphi_i = \min_{j \notin \{\cup_{y<x} \mathcal{J}_y\}} \frac{(C_j - \sum_{k \in \mathcal{I}_{x-1}} \varphi_k A_{kj})}{\sum_{k \notin \mathcal{I}_{x-1}} A_{kj}}, \tag{12}$$

for $i \in \mathcal{I}_x \setminus \mathcal{I}_{x-1}$ and $x > 1$.

Theorem 2: When flows have homogeneous requirements and are controlled by an end-to-end window, rates converge from any initial state to the max-min fair rates.

Proof. The proof is by induction. We first prove that buffers at links $j \in \mathcal{J}_1$ for flows $i \in \mathcal{I}_1$ are drained at rate φ_i and their backlog is either increasing or stable.

If the backlog of flow $i \in \mathcal{I}_1$ at some link $j \in \mathcal{J}_1$ is less than W_i, flow i must be backlogged at some other link. Denote by l the first backlogged link in the path of i preceding j. The rate ϕ_{il} of flow i leaving link l satisfies

$$\phi_{il} \geq \frac{C_l}{\sum_k A_{kl}}.$$

Rate ϕ_{il} is the rate into $j \in \mathcal{J}_1$ and, by the definition of $\mathcal{J}_1$, $\phi_{il} \geq \varphi_i$. This is true for all flows using j whose backlog at that link is less than W_i (including those that have no backlog because link j follows some other link in $\mathcal{J}_1$ in the flow i path).

Let ϕ_{ij} be the service rate of flow i at link $j \in \mathcal{J}_1$. Note that φ_i is the max-min fair rate realized locally by flows using j when all flows are backlogged at j. For any flow i backlogged at j, ϕ_{ij} cannot be less than φ_i. On the other hand, any non-backlogged flow must be served at its input rate and we have just shown that this is at least equal to φ_i. Clearly, the only allocations that satisfy these conditions and the capacity constraint, $\sum_i \phi_{ij} A_{ij} \leq C_j$, are the φ_i given by (11).

Increasing queues (i.e., where $\phi_{il} > \varphi_i$) must stabilize before some time t_1 after which no link outside $\mathcal{J}_1$ has a backlog for flows $\mathcal{I}_1$. When $\mathcal{J}_1$ contains more than 1 link, the stable backlogs of flow i can be any partition of the window W_i.

Now suppose flows $\mathcal{I}_{x-1}$ have converged to their fair rates and that after some epoch t_{x-1} their windows are entirely contained in the buffer of some link in $\{\cup_{y<x} \mathcal{J}_y\}$. We need to prove flows $\mathcal{I}_x \setminus \mathcal{I}_{x-1}$ will then similarly converge after some epoch $t_x \geq t_{x-1}$.

Consider $i \in \mathcal{I}_x \setminus \mathcal{I}_{x-1}$. If the queue at some link $j \in \mathcal{J}_x$ is less than W_i, flow i must be backlogged at some other link. Denote by l the first link in the path of i preceding j to have a flow i backlog. The rate ϕ_{il} of flow i leaving link l satisfies

$$\begin{aligned} \phi_{il} &\geq \frac{C_l - \sum_{i \in \mathcal{I}_{x-1}} \varphi_i A_{il}}{\sum_{i \notin \mathcal{I}_{x-1}} A_{il}} \\ &\geq \frac{C_j - \sum_{i \in \mathcal{I}_{x-1}} \varphi_i A_{ij}}{\sum_{i \notin \mathcal{I}_{x-1}} A_{ij}}. \end{aligned}$$

As for $x = 1$, this implies the output rate is at least that given by (12) for these flows. As any other flow in $\mathcal{J}_x$ with a full backlog cannot receive a lower rate, we conclude they must all receive the same rate given by (12).

The backlog is either increasing or stable. If all queues are already stable, the induction hypothesis is satisfied and $t_x = t_{x-1}$. If not, all increasing queues will have stabilized at some later epoch t_x after which no link other than $\mathcal{J}_x$ has a backlog for flows $\mathcal{I}_x \setminus \mathcal{I}_{x-1}$. □

Note that the special case of BMF where requirements A_{ij} are either equal to some flow dependent value A_i or 0 is

equivalent to max-min fairness for the resource shares $\varphi_i A_i$. Theorem 2 therefore has the corollary that this special case indeed converges to the BMF allocations. The theorem can be easily extended for weighted max-min fair allocations, as considered in [3], though these allocation are not BMF.

V. Packet system behavior

Simulation is used to evaluate the convergence behavior of packetized flows. Results show that flows indeed converge to the fluid BMF rates as long as the window in packets is large enough.

A. Packet model

We simulate a network of 2 unit capacity resources where all flows use both resource. Packets are of constant size L bits and are distinguished by flow requirements $a_{ij} > 0$. Propagation times are zero. The source of each flow maintains $W^{(p)}$ unacknowledged packets in the backlog of either resource ($W_i = W^{(p)}L$ in the fluid model notation). As packet size and window are fixed, it is as if packets circulate from one resource to the other and, in results below, we measure time in round trip times (RTTs), the variable time between successive service completions of the "same" packet.

Each resource implements start time fair queuing (SFQ) [6]. The start time tag S_{ij}^k of the k^{th} packet of flow i to arrive at resource j is computed recursively on its arrival epoch u_{ij}^k by,

$$S_{ij}^k = \max(V_j(u_{ij}^k), S_{ij}^{k-1} + a_{ij}L), \tag{13}$$

where $V_j(t)$ is the start time of the last packet to have begun service at j. Packets are served in increasing order of start time tags. In terms of the algorithm described in [6], it is as if packets have length $a_{ij}L$.

SFQ only approximates weighted max-min fairness and resource sharing realized by the packetized flows differs from the fluid ideal. In particular, the BMF rates may not be attained if the window is too small. The following proposition gives a lower bound on the required window size.

Proposition 2: A sufficient condition for the flows to attain the BMF allocation in the considered 2 resource network is that the end-to-end window in packets satisfies,

$$W^{(p)} \geq \frac{a_{kj}}{a_{ij}} + 1, \tag{14}$$

for all flows i, k and all resources j.

Proof. Th. 1 of [6] shows that the difference in the amount of service received by two continuously backlogged flows in a given interval is bounded. Reinterpreting the parameters of the bound in terms of the present network, we have, for two flows i and k, backlogged throughout interval (t_1, t_2),

$$\mid a_{ij}N_i(t_1,t_2) - a_{kj}N_k(t_1,t_2) \mid \leq a_{ij}L + a_{kj}L,$$

where $N_i(t_1,t_2)$ and $N_k(t_1,t_2)$ are the number of flow i and k packets served in the interval. In order that the flows be backlogged, it is necessary that $W^{(p)}$ be large enough to absorb the fluctuations. The requirement for flow i is for at least $a_{ij}L + a_{kj}L$ bits in the backlog to effectively satisfy the momentary excess service rate. This translates to a window $W^{(p)} \geq \frac{a_{kj}}{a_{ij}} + 1$ since each packet has an effective "length" of $a_{ij}L$ bits. Repeating this reasoning for each possible couple of backlogged flows and each resource yields (14). □

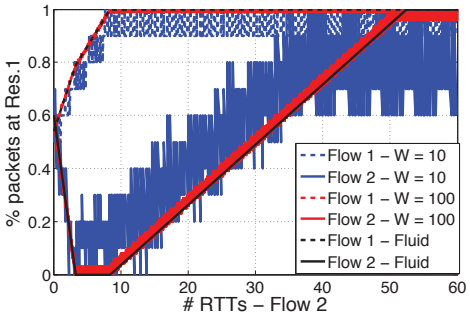

Fig. 1. Example of Backlog evolution.

B. Tending to the fluid limit

The fluid model occurs in the limit where the packet size tends to zero while the window in volume of data remains fixed. The backlog and throughput results derived for the fluid model in fact predict the performance of the packetized system for quite large packets.

Fig.1 shows the evolution of the share of $W^{(p)}$ in the resource 1 backlog as a function of the number of RTTs experienced by flow 2. The figure relates to a particular instance where 2 flows share the 2 resources with respective requirements $a_{11} = .4$, $a_{12} = .3$, $a_{21} = .09$, $a_{22} = .11$. The window of each flow is initially evenly split between both resources.

The red lines, for $W^{(p)} = 100$, almost coincide with the results of the fluid model (black lines). For larger packets, with $W^{(p)} = 10$, the backlogs follow the fluid trends but naturally exhibit greater variability.

In Fig.2 we show the impact of $W^{(p)}$ on the evolution of flow rates at resource 2. The rate received by flow i at resource j in the packet model is defined as

$$\hat{\phi}_{ij}(t) = \frac{1}{W^{(p)}} \sum_{k=P_{ij}(t)-W^{(p)}+1}^{P_{ij}(t)} \frac{1}{T_{kj}}, \tag{15}$$

where $P_{ij}(t)$ is the number of packets of flow i served before time t at resource j and T_{kj} is the RTT of packet k at resource j. Variable $\hat{\phi}_{ij}(t)$ is an average of the flow i rate leaving resource j over the last $W^{(p)}$ packets.

Convergence to BMF rates occurs if $W^{(p)}$ is larger than the bound of Proposition 2. In the present example, the rate of flow 2 for $W^{(p)} = 3$ converges to a value less than the BMF allocation. On the other hand, for $W^{(p)} = 10$ and greater, convergence to the fluid limit rates occurs within the first few RTTs for resource 1 (not shown). Convergence takes 50 RTTs however for rates at resource 2.

C. Speed of convergence

The time to converge to the BMF rates depends on the requirement parameters. We have explored this dependence

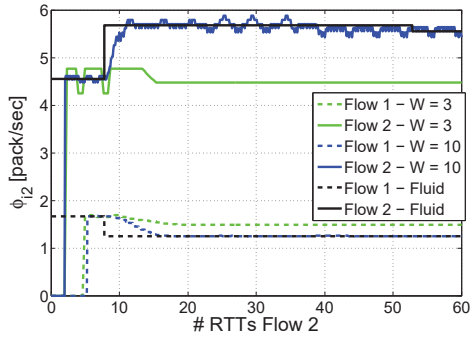

Fig. 2. Example of unfair convergence.

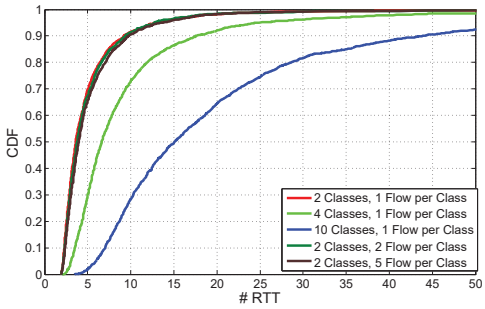

Fig. 3. CDF of convergence time to BMF rates.

by simulating the 2 resource network with many different random requirements. Flows of the same "class" have the same requirements drawn uniformly at random between 0 and 1. There is a variable number of flows per class. We set $W^{(p)} = 20$ for this experiment. The initial assignment of the $W^{(p)}$ packets of each flow at each resource is also set at random.

Figure 3 plots the cumulative distribution function of the time to convergence for 5000 realizations. The rates are deemed to converge when all measured rates, $\hat{\phi}_{ij}$, are simultaneously within 5% of the calculated BMF rates. Time to convergence for each flow is measured in its own RTTs and the overall convergence time is the maximum of these counts.

The figure shows that convergence time can be quite long and increases statistically with the number of flows. Slow convergence occurs when a flow backlog drains from one resource to the other at a very slow rate since the fair share at each is momentarily very similar. The occurrence of such an event is more likely as the number of flows with different requirements is greater. This explains why the convergence time in this experiment is statistically longer for the cases with more classes. Note, however, that even when convergence to precise BMF rates may be relatively long, it is very rare ($< 1\%$ of cases) that the rate attained by any flow after 20 RTTs differs by more than 20% from its BMF rate.

We have also investigated the impact on convergence time of the packet window size $W^{(p)}$. The convergence time in RTTs has roughly the same CDF if $W^{(p)}$ is somewhat greater than the bound in Proposition 2. This CDF coincides with that of the fluid model evaluated using the same random choice of requirements and initial backlogs.

VI. Conclusion

This paper considers multi-resource sharing between flows with heterogeneous resource requirements arising notably in networks with wireless links or software routers implementing virtualized network functions. Bottleneck max fairness is a sharing objective in this context that yields a satisfactory efficiency fairness tradeoff. Our aim in this paper has been to show BMF can be realized by locally imposing fair sharing at each resource and performing end-to-end window-based flow control.

Results of a large number of simulations of both fluid and packet-based models demonstrate that flow rates indeed converge in finite time to the BMF allocation from whatever initial backlog state. This empirical evidence is probably sufficient justification for engineering purposes but remains unsatisfactory from an analytical perspective.

Our main contribution has been to build a rather intricate proof of convergence for a fluid model of a 2 resource network. The system is significantly more complex than a network with homogeneous requirements and extension of the proof to more than 2 resources remains a challenging open problem. Insights gained in the present analysis did, however, lead to an original proof of convergence to max-min fairness for a general network with homogeneous requirements.

References

[1] T. Bonald, M. Feuillet, and A. Proutiere. Is the "law of the jungle" sustainable for the internet? In *INFOCOM 2009, IEEE*, pages 28–36, April 2009.

[2] T. Bonald and J. Roberts. Multi-resource fairness: Objectives, algorithms and performance. In *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '15, pages 31–42, New York, NY, USA, 2015. ACM.

[3] N. Chrysos and M. Katevenis. Distributed WFQ scheduling converging to weighted maxmin fairness. *Computer Networks*, 55(3):792806, 2011.

[4] A. Ghodsi, V. Sekar, M. Zaharia, and I. Stoica. Multi-resource fair queueing for packet processing. In *Proceedings of ACM SIGCOMM 2012*, pages 1–12, New York, NY, USA, 2012. ACM.

[5] A. Ghodsi, M. Zaharia, B. Hindman, A. Konwinski, S. Shenker, and I. Stoica. Dominant resource fairness: Fair allocation of multiple resource types. In *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation*, NSDI'11, pages 24–24, Berkeley, CA, USA, 2011. USENIX Association.

[6] P. Goyal, H. Vin, and H. Cheng. Start-time fair queueing: a scheduling algorithm for integrated services packet switching networks. *Networking, IEEE/ACM Transactions on*, 5(5):690–704, Oct 1997.

[7] E. Hahne. Round-robin scheduling for max-min fairness in data networks. *Selected Areas in Communications, IEEE Journal on*, 9(7):1024–1039, Sep 1991.

[8] E. L. Hahne and R. G. Gallager. Round robin scheduling for fair flow control in data communication networks. In *ICC*, pages 103–107, 1986.

[9] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda. Performance anomaly of 802.11b. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 2, pages 836–843 vol.2, March 2003.

[10] M. Katevenis. Fast switching and fair control of congested flow in broadband networks. *IEEE Journal on Selected Areas in Communications*, 5(8):1315–1326, October 1987.

[11] J. Nagle. On packet switches with infinite storage. RFC 970, 1985.

[12] B. Radunovic and J. Y. L. Boudec. A unified framework for max-min and min-max fairness with applications. *IEEE/ACM Transactions on Networking*, 15(5):1073–1083, Oct 2007.

[13] M. Shin, S. Chong, and I. Rhee. Dual-resource tcp/aqm for processing-constrained networks. *IEEE/ACM Trans. Netw.*, 16(2):435–449, Apr. 2008.

[14] M. Shreedhar and G. Varghese. Efficient fair queueing using deficit round robin. *SIGCOMM Comput. Commun. Rev.*, 25(4):231–242, Oct. 1995.
[15] R. Srikant. *The Mathematics of Internet Congestion Control*. Birkhauser Basel, 2004.
[16] P. Viswanath, D. Tse, and R. Laroia. Opportunistic beamforming using dumb antennas. *Information Theory, IEEE Transactions on*, 48(6):1277–1294, Jun 2002.

APPENDIX

The following is used in the proofs of Lemmas 1 – 3.

Lemma 4: Inequality (4) implies $\sum_{i=1}^{n} \frac{a_{i1}}{a_{i2}} > n$.

Proof. Since the harmonic mean of a set of numbers is always smaller than the arithmetic mean,

$$\frac{1}{n}\sum_{i=1}^{n}\frac{a_{i1}}{a_{i2}} > n\left(\sum_{i=1}^{n}\frac{a_{i2}}{a_{i1}}\right)^{-1} > n\left(\sum_{i=1}^{n}\frac{a_{i1}}{a_{i2}}\right)^{-1}.$$

Thus,

$$\left(\sum_{i=1}^{n}\frac{a_{i1}}{a_{i2}}\right)^{2} > n^2$$

and the lemma follows since the a_{ij} are all positive. □

Lemma 1 – a sufficient condition

Proof. When $r = 1$, the denominator in (6) is zero and the condition reads $\sum_{i=1}^{n} a_{i1}/a_{i2} > n$ since the right hand side must be negative. With this interpretation we prove that (6) is a sufficient condition for flow r to be backlogged for $r \geq 1$. Since $\sum_{i=1}^{n} a_{i1}/a_{i2} > n$ by Lemma 4, sufficiency is enough to prove the first statement.

We only need to consider evolutions from valid (i.e., persistent) system states with each flow backlogged at at least one resource. If all flows are initially backlogged only at resource 1 and this is a persistent state, convergence has already occurred. If all flows were initially backlogged only at resource 2 we would have $\phi_{i2}a_{i2} = 1/n$ and $\phi_{i1} = \phi_{i2}$. This yields a combined consumption of resource 1, $\sum a_{i1}\phi_{i1}$, that is greater than 1 (by Lemma 4) proving that this state is in fact impossible, i.e., not persistent.

When both resources have at least one backlogged flow, the fair shares f_1 and f_2 are well-defined and characterize the system state. If flow i is backlogged at 1 but not at 2, we must have $a_{i1}\phi_{i1} = f_1$, $a_{i2}\phi_{i2} < f_2$ and $\phi_{i1} = \phi_{i2}$ yielding $a_{i1}/a_{i2} > f_1/f_2$. If flow i is backlogged at both resources with the queue at 1 increasing, we have $a_{i1}\phi_{i1} = f_1$, $a_{i2}\phi_{i2} = f_2$ and $\phi_{i1} < \phi_{i2}$ again yielding $a_{i1}/a_{i2} > f_1/f_2$. Similarly, for flows that are not backlogged at 1 or have a decreasing queue, we must have $a_{i1}/a_{i2} < f_1/f_2$.

The value of the ratio f_1/f_2 partitions flows into two categories. The first denoted $\mathcal{C}_1$ consists of flows such that $a_{i1}/a_{i2} > f_1/f_2$. These flows are backlogged only at 1 or have an increasing resource 1 queue. The second category $\mathcal{C}_2$ comprises the remainder and these are backlogged only at 2 or have a decreasing queue at 1. The statement of the lemma is true if flow r belongs to category $\mathcal{C}_1$ in every valid state.

To establish the sufficient condition, first suppose flow r is not in category $\mathcal{C}_1$ for some valid state since $a_{r1}/a_{r2} < f_1/f_2$. No flow other than flows 1 to $r-1$ can then be in $\mathcal{C}_1$ since we would require $a_{i1}/a_{i2} > f_1/f_2$ and $a_{i1}/a_{i2} < a_{r1}/a_{r2}$. Let $\mathcal{B}_2 \subseteq \mathcal{C}_2$ be the set of flows backlogged only at 2 and b_2 its cardinality. The fair shares f_1 and f_2 then satisfy

$$f_1 = \frac{1 - \sum_{i\in\mathcal{B}_2}\frac{a_{i1}}{a_{i2}}f_2}{n - b_2},$$

$$f_2 = \frac{1 - \sum_{i=1}^{r-1}\frac{a_{i2}}{a_{i1}}f_1}{n - r + 1}.$$

Solving we find,

$$\frac{f_1}{f_2} = \frac{n - r + 1 - \sum_{i\in\mathcal{B}_2}\frac{a_{i1}}{a_{i2}}}{n - b_2 - \sum_{i=1}^{r-1}\frac{a_{i2}}{a_{i1}}}.$$

The right hand side is maximized when $\mathcal{B}_2$ consists of flows x_1 to n for a particular value of x_1. These are the flows with the smallest values of a_{i1}/a_{i2}. To see this suppose the maximizing $\mathcal{B}_2$ instead includes flow k but not j with $a_{j1}/a_{j2} < a_{k1}/a_{k2}$; replacing k by j would increase f_1/f_2 contradicting the initial assumption. Thus,

$$\frac{f_1}{f_2} \leq \frac{n - r + 1 - \sum_{i=x_1}^{n}\frac{a_{i1}}{a_{i2}}}{x_1 - 1 - \sum_{i=1}^{r-1}\frac{a_{i2}}{a_{i1}}}.$$

Note that $x_1 \geq r$ and, for the positivity of f_1 and f_2, we also require,

$$n - r + 1 > \sum_{i=x_1}^{n}\frac{a_{i1}}{a_{i2}}, \tag{16}$$

$$x_1 - 1 > \sum_{i=1}^{r-1} a_{i2}/a_{i1}. \tag{17}$$

It is true that f_1 and f_2 would also be positive if the inequality in both (16) and (17) were inverted. We show that this is impossible for $x_1 \geq r$. Write

$$n - r + 1 - \sum_{i=x_1}^{n}\frac{a_{i1}}{a_{i2}} = x_1 - r + \sum_{i=x_1}^{n}(1 - \frac{a_{i1}}{a_{i2}}).$$

For this expression to be negative it is clearly necessary that the smallest term in the sum be negative, i.e., $a_{x_1 1}/a_{x_1 2} > 1$.

Similarly, writing

$$x_1 - 1 - \sum_{i=1}^{r-1}\frac{a_{i2}}{a_{i1}} = x_1 - r + \sum_{i=1}^{r-1}(1 - \frac{a_{i2}}{a_{i1}})$$

we deduce that the expression can only be negative if the smallest term in the sum is negative, i.e., if $a_{(r-1)2}/a_{(r-1)1} > 1$. Given the ordering of the flows following (5), this condition is incompatible with the previous one completing the proof that (16) and (17) are indeed necessary conditions on the value of x_1.

Let

$$F(x) = \frac{n - r + 1 - \sum_{i=x}^{n}\frac{a_{i1}}{a_{i2}}}{x - 1 - \sum_{i=1}^{r-1}\frac{a_{i2}}{a_{i1}}},$$

so that x_1 maximizes $F(x)$ over states satisfying the above inequalities. We now show that $x_1 = r$ is actually the only

possibility by successively considering all possible values of x_1.

i) $1 < x_1 - 1 - \sum_{i=1}^{r-1} a_{i2}/a_{i1}$ and $x_1 > r$.
If x_1 maximizes $F(x)$ we have $F(x_1) > F(x_1 - 1)$ or

$$\frac{n - r + 1 - \sum_{i=x_1}^{n} \frac{a_{i1}}{a_{i2}}}{x_1 - 1 - \sum_{i=1}^{r-1} \frac{a_{i2}}{a_{i1}}} > \frac{n - r + 1 - \sum_{i=x_1-1}^{n} \frac{a_{i1}}{a_{i2}}}{x_1 - 2 - \sum_{i=1}^{r-1} \frac{a_{i2}}{a_{i1}}} \quad (18)$$

where the denominators are both positive in the assumed range. Cross-multiplying and simplifying, we find,

$$F(x_1) < \frac{a_{(x_1-1)1}}{a_{(x_1-1)2}}.$$

If, as assumed, $a_{r1}/a_{r2} < f_1/f_2$ for some valid state, then certainly $a_{r1}/a_{r2} < F(x_1) < a_{(x_1-1)1}/a_{(x_1-1)2}$ implying $r > x_1 - 1$. As r and x_1 are integers, this inequality is incompatible with the assumption $x_1 > r$. We conclude from this contradiction that x_1 cannot in fact be in the considered range.

ii) $0 < x_1 - 1 - \sum_{i=1}^{r-1} \frac{a_{i2}}{a_{i1}} < 1$ and $x_1 > r$.
We again have (18) but the denominator on the left is positive while the one on the right is negative. Cross-multiplying therefore inverts the inequality and we deduce,

$$F(x_1) > \frac{a_{(x_1-1)1}}{a_{(x_1-1)2}}. \quad (19)$$

Let $N_f(x) = n - r + 1 - \sum_{i=x}^{n} \frac{a_{i1}}{a_{i2}}$ and $D_f(x) = x - 1 - \sum_{i=1}^{r-1} \frac{a_{i2}}{a_{i1}}$ so that $F(x) = N_f(x)/D_f(x)$. We can write

$$F(x_1) = \frac{N_f(x_1 - 1) + \frac{a_{(x_1-1)1}}{a_{(x_1-1)2}}}{D_f(x_1 - 1) + 1} > \frac{a_{(x_1-1)1}}{a_{(x_1-1)2}}.$$

As $D_f(x_1 - 1) + 1$ is positive in the considered range, the inequality yields

$$N_f(x_1 - 1) > D_f(x_1 - 1)\frac{a_{(x_1-1)1}}{a_{(x_1-1)2}}.$$

On the other hand, $D_f(x_1-1)$ is negative so that the inequality is inverted on division giving,

$$F(x_1 - 1) = \frac{N_f(x_1 - 1)}{D_f(x_1 - 1)} < \frac{a_{(x_1-1)1}}{a_{(x_1-1)2}}. \quad (20)$$

We can also write,

$$F(x_1 - 1) = \frac{N_f(x_1) - \frac{a_{(x_1-1)1}}{a_{(x_1-1)2}}}{D_f(x_1) - 1}.$$

From (19) and $D_f(x_1) > 0$, we have $N_f(x_1) > D_f(x_1)a_{(x_1-1)1}/a_{(x_1-1)2}$ and, therefore,

$$F(x_1 - 1) > \frac{D_f(x_1)\frac{a_{(x_1-1)1}}{a_{(x_1-1)2}} - \frac{a_{(x_1-1)1}}{a_{(x_1-1)2}}}{D_f(x_1) - 1} = \frac{a_{(x_1-1)1}}{a_{(x_1-1)2}}.$$

This contradicts (20) proving, therefore, that x_1 cannot be in the considered range.

iii) $x_1 = r$.
First consider $r = 1$. In the state where all flows are backlogged only at resource 2, we would have $f_2 = 1/n$ and a combined flow into resource 1 of $\sum_{1=1}^{n} \frac{a_{i1}}{a_{i2}}/n$. However, the latter expression is greater than 1 by Lemma 4 proving that this state is impossible. Flow 1 must therefore belong to $\mathcal{C}_1$.

For $r > 1$, in order for the initial assumption to be satisfied we require $a_{r1}/a_{r2} < f_1/f_2$ for some state. This is clearly not satisfied if $a_{r1}/a_{r2} > F(x_1) = F(r)$. In other words, a sufficient condition for flow $r \in \mathcal{C}_1$ is indeed (6). □

Lemma 3 - convergence to BMF

Proof. We know from Lemma 1 that the backlog of flow 1 always converges and from Lemma 2 that the backlog of flow n converges independently of r if $\sum_1^n a_{i2}/a_{i1} > n$. It remains to consider states with $r > 1$ and $s > 0$ where the sufficient conditions for convergence are (6) and (7), respectively.

Let $N(r)$ denote the numerator and $D(r)$ the denominator of the right hand side of (6):

$$N(r) = n - r + 1 - \sum_{i=r}^{n} \frac{a_{i1}}{a_{i2}} = \sum_{i=r}^{n} \left(1 - \frac{a_{i1}}{a_{i2}}\right),$$

$$D(r) = r - 1 - \sum_{i=1}^{r-1} \frac{a_{i2}}{a_{i1}} = \sum_{i=1}^{r-1} \left(1 - \frac{a_{i2}}{a_{i1}}\right).$$

It is easily verified that $N(r)$ and $D(r)$ cannot both be negative for the same value of r. Moreover, the values of r where $N(r)$ may be negative are the smallest possible (it is easy to see that $N(r+1) < 0 \Rightarrow N(r) < 0$ for $1 \le r < n$) and the values of r where $D(r)$ may be negative are the largest possible (since $D(r-1) < 0 \Rightarrow D(r) < 0$ for $1 < r \le n$).

The sufficient condition (7) that flow $n - s$ stabilizes to a resource 2 backlog may be expressed

$$\frac{a_{(n-s)2}}{a_{(n-s)1}} > \frac{D(n - s + 1)}{N(n - s + 1)}.$$

If the right hand side is negative, the condition is trivially satisfied. If positive both numerator and denominator are positive and we can invert the inequality to read

$$\frac{a_{(n-s)1}}{a_{(n-s)2}} < \frac{N(n - s + 1)}{D(n - s + 1)}.$$

If $a_{r1}/a_{r2} < N(r)/D(r)$ condition (6) is not satisfied. The proof is complete if we prove that $(a_{r1}/a_{r2} < N(r)/D(r)) \Rightarrow (a_{(n-s)1}/a_{(n-s)2} < N(n - s + 1)/D(n - s + 1))$, i.e., (7) is satisfied.

If $a_{r1}/a_{r2} < N(r)/D(r)$ then $N(r)$ and $D(r)$ must both be positive and $N(r) > D(r)a_{r1}/a_{r2}$. We deduce,

$$\begin{aligned} \frac{N(r+1)}{D(r+1)} &= \frac{N(r) - 1 + a_{r1}/a_{r2}}{D(r) + 1 - a_{r2}/a_{r1}} \\ &> \frac{D(r)a_{r1}/a_{r2} - 1 + a_{r1}/a_{r2}}{D(r) + 1 - a_{r2}/a_{r1}} \\ &= \frac{a_{r1}}{a_{r2}}. \end{aligned}$$

Now, $a_{r1}/a_{r2} > a_{(r+1)1}/a_{(r+1)2}$ so $N(r+1)/D(r+1) > a_{(r+1)1}/a_{(r+1)2}$. Applying the same argument repeatedly we finally deduce $a_{(n-s)1}/a_{(n-s)2} < N(n-s+1)/D(n-s+1)$. □