Fairness in overloaded parallel queues

Carri W. Chan
Division of Decision, Risk and Operations, Columbia Business School cwchan@columbia.edu

Mor Armony
Stern School of Business, New York University marmony@stern.nyu.edu

Nicholas Bambos
Departments of Electrical Engineering and Management Science & Engineering, Stanford University bambos@stanford.edu

This version: February 20, 2011

Maximizing throughput for heterogeneous parallel server queues has received quite a bit of attention from the research community and the stability region for such systems is well understood. However, many real-world systems have periods where they are temporarily overloaded. Under such scenarios, the unstable queues often starve limited resources. This work examines what happens during periods of temporary overload. Specifically, we look at how to fairly distribute stress. We explore the dynamics of the queue workloads under the MaxWeight scheduling policy during long periods of stress and discuss how to tune this policy in order to achieve a target fairness ratio across these workloads.

1. Introduction

Queueing systems are generally designed with sufficient allocation of resources to ensure that the system is stable. Subsequently, a large body of research in queueing has focused on characterizing the stability region and on evaluating and optimizing performance when the queues are operated within this region. However, in reality, it is inevitable that a system will repeatedly enter long periods of overload, either due to an unpredictable increase in load or because some service resources may become unavailable/breakdown. In such periods, which may last for a fairly long duration of time, the system may have more load than it can handle. Hence, it is natural to want to understand what happens during these periods of *temporary instability*.

Some examples of real systems which experience periods of instability can be found in internet communication networks, hospitals, and call centers. Consider systems where traffic is bursty. For instance, in communications networks, with the release of a new online game, the system may be over-stressed as many people attempt to simultaneously enter the virtual world. At a hospital, there may be a surge in demand for the limited hospital resources during a catastrophic event or viral outbreak. In these cases, the system load may increase to potentially unstable levels. A separate scenario of temporary overload may occur when resources are lost. In call centers, a power outage may effectively remove a pool of servers. Consequently, while the system load may remain constant, the available service resources have been reduced, stressing the

system.

In long periods of overload, it is inevitable that the system backlog will grow significantly, so the question of performance optimization becomes moot. However, it is desirable to share the stressed resources in a *fair* manner among the different customers. In this paper, we study fair resource allocation in multiclass parallel server queues during long periods of overload or temporary instability.

How does one define fairness in an overloaded system? Consider a system which is overloaded for a very long time window. During this time window, the system is essentially unstable, so that the total backlog grows without bound. However, it is plausible that some customer classes will enjoy a greater allocation of the stressed resources, enough such that the system will appear stable to these classes. This is unfair. Our *novel* notion of fairness requires that the backlog of the different customer classes will grow large according to a predetermined proportion, thereby proportionately distributing stress across customer classes. This is the same as requiring that the *backlog vector* will grow along a prespecified *direction*.

Working with this notion of fairness, our ultimate goal is to determine how to operate the overloaded system such that the total backlog is minimized subject to a fairness constraint. Our main result is that the MaxWeight scheduling algorithm, with carefully selected weight parameters, will achieve this goal asymptotically, as the duration of the overloaded period grows large. The MaxWeight policy selects, at each time point, a service configuration which maximizes the aggregate service rate weighted by the queue backlogs and a controllable weight vector (Tassiulas and Ephremides 1992, Mandelbaum and Stolyar 2004). MaxWeight has been shown to be throughput maximizing (see for example Tassiulas (1995), Tassiulas and Bhattacharya (2000), McKeown et al. (1999), Armony and Bambos (2003)). That is, as long as the system is stabilizable, it will be stable under MaxWeight.

Our analysis first shows that, under MaxWeight with arbitrary fixed weights, the backlog in an overloaded system will grow to infinity according to a well defined direction. We then proceed to show that this direction can be turned to any feasible fairness proportion by carefully selecting the weight parameters of MaxWeight. Finally, we establish that, in the limit, MaxWeight will minimize the total backlog among all algorithms that obtain this fairness direction.

The approach we take in our analysis is a trace-based approach (Loynes 1963, Armony and Bambos 2003, Ross and Bambos 2009) which analyzes individual traces of customers' arrival times and workload requests. To show that the backlog grows along a specific direction, we use *direct* geometric arguments that examine the dynamics of an actual trajectory of the backlogs and how it evolves over time. In particular, our statistical assumptions are extremely mild, and only require that the workload entering the system has a well defined long time average.

The rest of the paper is organized as follows: We conclude this introduction by surveying some closely related literature. In Section 2 we formally introduce our queueing model and the notion of fairness which we study. In Section 3 we show the existence of a unique limit for the queue backlogs when the system is unstable. In Section 4 we discuss how to control the backlog to achieve our desired fairness criterion. In Section 5 we show some numerical results to demonstrate the performance of our proposed algorithm in practice. Finally, we conclude in Section 6.

Related Literature

Stolyar (2004) analyzes the same type of parallel server queueing systems, under the assumption that the system is in heavy-traffic. That is, the system is stable but operates close to the boundary of the stability region. For such systems, the author shows that under MaxWeight and a complete resource pooling condition, the backlog vector will experience a state-space collapse, which implies that it will asymptotically follow a well defined direction. The author also shows that MaxWeight minimizes the workload, where the latter is defined as the total backlog weighted by the above direction. These results are similar to our convergence and backlog minimization results, but since the system considered in Stolyar (2004) is not overloaded, much stronger assumptions are required there–such as the Markov property and the complete resource pooling condition. Mandelbaum and Stolyar (2004) generalizes these results for convex holding cost functions and a corresponding generalized $c\mu$ rule.

More recently, Shah and Wischik in (Shah and Wischik 2011) study the asymptotic behavior of *fluid models* in overload under MaxWeight and other policies. Analogously to our result, they show that as time grows large, the fluid vector will approach a well defined direction. While the network structure and the set of policies that Shah and Wischik (2011) studies are more general than our setting, that paper focuses on *characterizing* the asymptotic direction of the backlog of the overloaded system. Our paper, on the other hand, focuses on *controlling* this direction to satisfy a *fairness* constraint and on minimizing the backlog subject to this constraint. Methodologically the two papers are also different; Shah and Wischik (2011) uses Lyapunov functions to prove convergence of the continuous *fluid* trajectory, while our analysis directly examines the dynamics of the *queueing trajectory*.

Also in the networked setting, Georgiadis and Tassiulas (2006) consider overloaded sensor networks where transmissions are scheduled in a distributed manner. Specifically, Georgiadis and Tassiulas (2006) show that a policy analogous to the Adaptive Back Pressure policy of Tassiulas (1995) (reference [20] in Tassiulas' paper) obtains, in the limit, a backlog vector which is the so-called "most balanced" among all feasible limits, in overload. Similar to Shah and Wischik (2011), the authors consider a fluid model rather than the direct *queueing trajectory* dynamics considered in this work.

Our concept of fairness in this paper is also somewhat unusual. Two commonly used fairness criteria are Max-Min and Proportional Fairness. A system is considered Max-Min fair if the utility of the user with the minimum utility is maximized. On the other hand, a system is considered to be proportionally fair if the amount of service resources a user is allocated corresponds to the proportion of anticipated resource consumption required by the user. These two notions of fairness can be tied together under the description of α-fairness with different values of α (Mo and Walrand 2000). There has been a substantial amount of work in the development of scheduling policies which ensure some fairness criteria (see Mo and Walrand (2000), Kelly and Williams (2004), Eryilmaz and Srikant (2005), Neely et al. (2005), Eryilmaz and Srikant (2006), Neely (2006), Bonald et al. (2006), Massoulie (2007) and related works). A common thread in those papers is an assumption of stability, whereas we consider fairness during periods of temporary instability. We define fairness by a set of ratios which specifies the desired proportion of the aggregate backlog each queue contributes. In Section 4.4 we elaborate more on the connection between the traditional notions of fairness and the one used here.

Perry and Whitt (2009, 2011) consider how to deal with unexpected overload. They consider overload in the setting of two initially separated service systems which share resources when one becomes overloaded. Using a fluid model, they consider how to share resources across the two systems in order to maintain a constant ratio between the two queues. Our setting is quite different as there is a single system with multiple parallel queues. However, we share a similar control goal of maintaining a constant ratio between queues, of which we may have more than two.

2. The Queueing Model

We consider a queueing system with Q queues and N service vectors. New jobs arrive to the system and are queued up to be served. The system administrator dynamically selects which service configuration to implement for service of available jobs in the queue. We consider how to make this selection in a 'fair' manner, where we will be more precise as to what we mean by fair in the coming discussion.

More formally, we consider a queueing system with Q parallel queues, indexed by $q \in \mathcal{Q} = \{1, 2, \ldots, Q\}$. Time is discrete and indexed by $t \in \mathbb{Z}_0^+$. Let $A_q(t)$ be the number of jobs arriving to queue q in time slot t. We assume that $0 \leq A_q(t) \leq \bar{A}_q$ for some arbitrarily large $\bar{A}_q > 0$. For each $q \in \mathcal{Q}$, we assume that

$$\lim_{t\to\infty} \frac{\sum_{s=0}^{t-1} A_q(s)}{t} = \rho_q \in (0, \infty) \tag{2.1}$$

is well-defined, positive, and finite. The arrival vector in time slot t is thus

$$A(t) = (A_1(t), A_2(t), \ldots, A_q(t), \ldots, A_Q(t)) \tag{2.2}$$

with corresponding long-term traffic load vector:

$$\rho = (\rho_1, \rho_2, \ldots, \rho_q, \ldots, \rho_Q) \tag{2.3}$$

Arriving jobs are buffered in their respective queues and are served in a first-come-first-served (FCFS) manner within each queue.

In each time slot, a service vector $S \in \mathcal{S} = \{S_1, S_2, \ldots, S_n, \ldots, S_N\}$ is selected to be used. Each service vector is a Q-dimensional vector:

$$S = (S_1, S_2, \ldots, S_q, \ldots, S_Q) \tag{2.4}$$

where $S_q \geq 0$ is the number of jobs removed from queue q in a single time slot when service configuration S is used.

The workload in queue q at time t is denoted by $X_q(t)$ with corresponding workload vector:

$$X(t) = (X_1(t), X_2(t), \ldots, X_q(t), \ldots, X_Q(t)) \tag{2.5}$$

This corresponds to the number of jobs in each queue in time-slot t.

Given workload vector $X(t)$ and service configuration $S(t)$, the number of jobs that are served and depart from queue q is:

$$D_q(t) = \min\{S_q(t), X_q(t)\} \tag{2.6}$$

where the minimum accounts for the fact that jobs can only be serviced if they are already waiting in the queue. Hence, if $D_q(t) = X_q(t) < S_q(t)$, there is some idle service provided by service vector $S(t)$ due to the lack of available jobs to be processed. The workload vector evolves as:

$$X(t+1) = X(t) + A(t) - D(t) \tag{2.7}$$

Assuming the queue begins empty at time $t = 0$, then the workload vector $X(t)$ is given by:

$$X(t) = \sum_{s=0}^{t-1} A(s) - \sum_{s=0}^{t-1} D(s) \tag{2.8}$$

Applications We note that many applications of interest can be modeled in this way. For example, in communication networks, the backlogs correspond to packets waiting to be transmitted on various wired or wireless links while the service vectors correspond to various packet switch configurations. In call centers, the backlogs correspond to the number of customers of various classes waiting to be served while the service vectors correspond to specific allocations of staff with differing skills to each customer class.

2.1. (In)Stability Region

Loosely speaking, we consider a system to be rate stable if the average job departure rate is equal to the average arrival rate. We define the stability region of a system, $\mathcal{P}$, such that if $\rho \in \mathcal{P}$, then there exists some policy which guarantees for each queue, q, that:

$$\lim_{t\to\infty} \frac{X_q(t)}{t} = 0 \tag{2.9}$$

The stability region can be characterized as:

$$\begin{aligned}\mathcal{P} &= \left\{ \rho \in \mathbb{R}_+^Q : \rho \leq \sum_{i=1}^{N} \alpha_i S_i \text{ for some } \alpha_i \geq 0, S_i \in \mathcal{S} \text{ such that } \sum_{i=1}^{N} \alpha_i = 1 \right\} \\ &= \left\{ \rho \in \mathbb{R}_+^Q : \langle \rho, \Delta v \rangle \leq \max_{S\in\mathcal{S}} \langle S, \Delta v \rangle \text{ for every } v \in \mathbb{R}_+^Q \right\}\end{aligned} \tag{2.10}$$

as defined in Ross and Bambos (2009). Hence, any system load which is dominated by a convex combination of service vectors is stablizable. Furthermore, stability is ensured by using this convex combination.

When the system load is outside of the stability region, then the system is unstable. Hence, if $\rho \notin \mathcal{P}$, then there does not exist any policy which guarantees that:

$$\lim_{t\to\infty} \frac{X_q(t)}{t} = 0 \tag{2.11}$$

and with probability 1,

$$\lim_{t\to\infty} \frac{X_q(t)}{t} > 0 \tag{2.12}$$

for some q (see Armony and Bambos (2003)). We refer to this as the *instability* region. Consider the following simple example of the stability and instability regions.

Example 1*Consider an example with two queues ($Q = 2$) and three ($N = 3$) service vectors:*

$$S_1 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, S_2 = \begin{pmatrix} 1 \\ 1.5 \end{pmatrix}, S_3 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \tag{2.13}$$

The stability region, $\mathcal{P}$, is given by the convex hull of S_1, S_2, S_3 (and their projections onto the axes). Note that S_3 is a non-essential *service vector, since it is dominated by a convex combination of S_1 and S_2 and its removal would not change $\mathcal{P}$. Any load vector ρ within the shaded region in Figure 1 is stabilizable; outside the region, is not. Hence, $\hat{\rho}$ is stabilizable while ρ' is not.*

The focus of this work is the understand and control the dynamics of systems which are temporarily overloaded over a long time horizon. To do this, we approximate the dynamics of a *temporarily* unstable system with a system which is *permanently* unstable ($\rho \notin \mathcal{P}$). When the period of overload is long enough, the dynamics of the two systems will be similar. Hence, our analysis focuses on the dynamics of unstable systems.

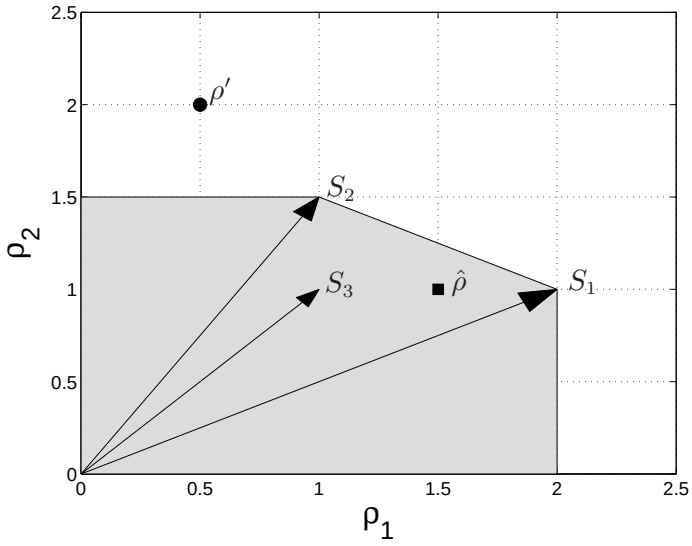


Figure 1 An example of the stability region for a 2-queue queueing system.

2.2. Fairness

The goal of this work is to determine a service discipline which allows us to fairly serve queues when the system is temporarily overloaded ($\rho \notin \mathcal{P}$). There are many different definitions of fairness (see for instance Mazumdar et al. (1991), Kelly et al. (1998), Mo and Walrand (2000)). In this paper, we focus on a notion of fairness where the workloads grow according to a fixed proportion. More formally, we assume we are given a set of ratios $\theta_q \geq 0, \sum_q \theta_q = 1$. These ratios specify the proportion of the aggregate workload which each queue contributes. Whenever $\rho \notin \mathcal{P}$, the goal is to control the workload such that for large t:

$$\frac{X_q(t)}{\sum_k X_k(t)} \approx \theta_q, \forall q \in \mathcal{Q} \tag{2.14}$$

More precisely, we want to control $X(t)$ such that:

$$\lim_{t\to\infty} \frac{X(t)}{t} = \eta \tag{2.15}$$

where η satisfies our fairness criterion:

$$\frac{\eta_i}{\sum_j \eta_j} = \theta_i, \forall i \tag{2.16}$$

θ defines a *direction* which we want the backlogs to grow; η includes the scale factor to specify how quickly they grow.

There may be many policies which achieve this fairness criterion and our goal is to find the policy which achieves it with the *minimal workload*. Let $X(t)$ correspond to the workload at time t given service vector $S(t)$ is used in time slot t. Our goal is to find $S(t)$ such that for large t:

$$\lim_{t\to\infty} \frac{X_q(t)}{\sum_k X_k(t)} = \theta_q, \forall q \in \mathcal{Q} \tag{2.17}$$

Furthermore, if there exists $X'(t)$ which uses $S'(t)$ in time slot t and $\lim_{t\to\infty} \frac{X'_q(t)}{\sum_k X'_k(t)} = \theta_q, \forall q \in \mathcal{Q}$, then

$$\lim_{t\to\infty} \frac{X(t)}{t} \leq \lim_{t\to\infty} \frac{X'(t)}{t} \tag{2.18}$$

We will go into more detail about this fairness criterion in Section 4. Further, we will show that this definition can be generalized to other fairness definitions. Our proposal is to use MaxWeight scheduling policies to achieve our fairness criterion Armony and Bambos (2003), Ross and Bambos (2009).

2.3. The MaxWeight Scheduling Policy

The stability region guarantees the existence of a stabilizing policy. We now briefly review a family of stabilizing policies, which we refer to as MaxWeight Scheduling. This is the special case of Projective Cone Scheduling (PCS) when projection matrices are diagonal. This family of policies is characterized by service cones wherein the set of workloads $X(t)$ under which service vector S_i is used forms a cone in the workload-space. We focus on this family of policies because they work 'well', i.e. are stabilizing when possible, they are simple to implement, and as we will see later, they are optimal in the sense that they obtain the minimum backlog (2.18) subject to the fairness criterion (2.17).

Let $\mathbf{B}$ be a symmetric, positive definite matrix with non-positive off-diagonal elements. Let $X = X(t)$ be the workload at time t. Then, the PCS algorithm selects service vector S^*:

$$S^* \in S^{\max}(X) = \arg\max_{S\in\mathcal{S}} \langle S, \mathbf{B}X\rangle = \arg\max_{S\in\mathcal{S}} X^T\mathbf{B}S \tag{2.19}$$

It is shown in Ross and Bambos (2009) that the PCS policy is rate stable for any $\rho \in \mathcal{P}$. This policy is also desirable from an implementation standpoint as it only requires current information (workload state) and does not require knowledge of the system load ρ.

We define the service cone C_i as the set of workload vectors, X, such that service vector S_i is used under the algorithm defined by positive definite matrix $\mathbf{B}$; i.e.:

$$C_i = C_i(\mathbf{B}) = \{X | S_i \in \arg\max_{S\in\mathcal{S}} \langle S, \mathbf{B}X\rangle\} \tag{2.20}$$

For the rest of the discussion, we will suppress the dependence of the service cones on the matrix $\mathbf{B}$.

Throughout this work, we will focus on the case where the matrix $\mathbf{B}$ is a diagonal matrix, $\mathbf{\Delta}$ (and positive-definite, hence, all its diagonal elements are positive). This specifies the set of algorithms to MaxWeight scheduling algorithms, which have been studied extensively in the past.

Assumption 2.1 $\mathbf{B} = \mathbf{\Delta}$ *is a diagonal, positive definite matrix, i.e.* $\mathbf{\Delta}_{qq} > 0$ *and* $\mathbf{\Delta}_{qq'} = 0, \forall q \neq q'$

While we are ultimately interested in the behavior of this queueing system when it is temporarily overloaded, our focus on MaxWeight scheduling policies ensures that the system is stabilized when possible. We will examine the asymptotic behavior of this class of policies and its implications in term of our fairness criterion. We will see that this family of algorithms allows us to satisfy certain fairness criterion by manipulating the $\mathbf{\Delta}$ matrix. Furthermore, we will see in Section 4.2 that the MaxWeight scheduling policies achieves the lowest workload which satisfies our fairness criterion. In order to see this, we must first understand how the $\mathbf{\Delta}$ matrix affects the asymptotic behavior of our queueing system.

3. Asymptotic Dynamics in Overload

Before we consider how to control the workload vector, $X(t)$, we first begin by building an understanding of its asymptotic dynamics given diagonal MaxWeight matrix $\mathbf{\Delta}$. We already know that if $\rho \in \mathcal{P}$, then the MaxWeight scheduling policy stabilizes the system and the workload is always finite. On the other hand, many real systems enter periods of instability where the system load is temporarily outside of the stability region. From Armony and Bambos (2003), when $\rho \notin \mathcal{P}$ the workload explodes: $X(t) \to \infty$. However, we are interested in finding out how exactly this happens. Is there a finite limit for $\lim_{t\to\infty} \frac{X(t)}{t}$? If so, what is it and what does it depend on? This section is devoted to answering these questions.

Suppose that the system operates in overload, in the sense that $\rho \notin \mathcal{P}$, where

$$\mathcal{P} = \left\{ \rho \in \mathbb{R}_+^Q : \langle \rho, \Delta v \rangle \leq \max_{S \in \mathcal{S}} \langle S, \Delta v \rangle \text{ for every } v \in \mathbb{R}^Q \right\}, \tag{3.1}$$

as defined in Ross and Bambos (2009). We consider the limit defined below:

$$H = \limsup_{t\to\infty} \left\langle \frac{X(t)}{t}, \mathbf{\Delta} \frac{X(t)}{t} \right\rangle \tag{3.2}$$

and select a convergent increasing unbounded subsequence $\{t_c\}$ on which the '$\limsup$' is attained[1] – hence,

$$\lim_{c\to\infty} \frac{X(t_c)}{t_c} = \eta \tag{3.3}$$

and

$$\lim_{c\to\infty} \left\langle \frac{X(t_c)}{t_c}, \mathbf{\Delta} \frac{X(t_c)}{t_c} \right\rangle = \langle \eta, \mathbf{\Delta}\eta \rangle = H. \tag{3.4}$$

Lemma 3.1*We have*

$$\rho \notin \mathcal{P} \implies \eta \neq 0 \tag{3.5}$$

[1] See footnote 12 of Ross and Bambos (2009) concerning why such a subsequence exists

PROOF: See the proof of Proposition 2.1 in Armony and Bambos (2003). In short, we have that

$$\rho \notin \mathcal{P} \implies \limsup_{t\to\infty} \frac{X_q(t)}{t} > 0 \text{ for some } q \in \mathcal{Q} \tag{3.6}$$

This in turn implies that:

$$\eta \neq 0 \text{ and, so } \limsup_{t\to\infty} \left\langle \frac{X(t)}{t}, \mathbf{\Delta}\frac{X(t)}{t} \right\rangle = \langle \eta, \Delta\eta \rangle = H > 0. \tag{3.7}$$

■

This result follows directly from the stability results of Armony and Bambos (2003), Ross and Bambos (2009) which say that when $\rho \in \mathcal{P}$ the time-scaled backlog goes to 0 as $t\to\infty$. When the system is not stable, the opposite occurs and there exists a non-zero sub-limit. Our key result in this section is that η is the **unique** limit of $\frac{X(t)}{t}$ over all possible arrival traces.

Theorem 3.1*When $\rho \notin \mathcal{P}$, we have*

$$\lim_{t\to\infty} \frac{X(t)}{t} = \eta \neq 0. \tag{3.8}$$

That is, the workload explodes on the same non-zero ray η on any arrival trace. Furthermore, η is defined as the unique solution to the following convex program:

$$\langle \eta, \mathbf{\Delta}\eta \rangle = \min_{\eta' \in \Psi(\rho,\mathcal{S})} \langle \eta', \mathbf{\Delta}\eta' \rangle \tag{3.9}$$

where

$$\Psi(\rho, \mathcal{S}) = \{\eta' : \eta' = (\rho - r)^+ \text{ with } r \in \mathcal{P}\} \tag{3.10}$$

and $\mathcal{P}$ is the stability region given by $\mathcal{S}$. Therefore, $r = \sum_{S\in\mathcal{S}} \alpha_S S$ with $\sum_{S\in\mathcal{S}} \alpha_S \leq 1$ and $\alpha_S \geq 0$ for each $S \in \mathcal{S}$, where $\mathcal{S}$ is the set of service vectors. Equivalently, η is the unique fixed point which satisfies $\eta = \rho - \sum_{S\in\mathcal{S}} \alpha_S S$ with $\sum_{S\in\mathcal{S}} \alpha_S = 1, \alpha_S > 0$ and

$$\langle \eta, \mathbf{\Delta}S_m \rangle \geq \langle \eta, \mathbf{\Delta}S_k \rangle \; \forall m \text{ such that } \alpha_m > 0 \tag{3.11}$$

PROOF: The proof will conclude at the end of Section 3. We will first show that the limit is unique for each individual arrival trace. We then extend our analysis to show the limit is the same for all arrival traces with identical time-average traffic load, ρ, which in turn, implies uniqueness over all arrival traces. We begin with a number of definitions and structural properties of the defined elements.

From Section V of Ross and Bambos (2009) on MaxWeight Scheduling cone geometry, recall that $\mathcal{C}_S = \{x \in \mathbb{R}^Q : \langle S, \mathbf{\Delta}x \rangle = \max_{S'\in\mathcal{S}} \langle S', \mathbf{\Delta}x \rangle\}$ is a cone, and when $X(t) \in \mathcal{C}_S^o$ (the interior of $\mathcal{C}_S$) the MaxWeight

scheduling policy will choose $S(t) = S$. Moreover, the *surrounding cone* of any non-zero vector η is the cone

$$\mathcal{C}(\eta) = \bigcup_{S \in \mathcal{S}^*(\eta) - \mathcal{S}^\dagger} C_S \tag{3.12}$$

where $S^*(\eta) = \text{argmax}_{S \in \mathcal{S}} \langle S, \mathbf{\Delta}\eta \rangle$ is the set of service vectors of that MaxWeight Scheduling would select for backlog η and $\mathcal{S}^\dagger$ is the set of *non-essential* ones (see Ross and Bambos (2009), end of Section IV). Loosely speaking, the non-essential service vectors are the one whose removal will not change the stability region. Hence, $\mathcal{C}(\eta)$ is the union of all cones corresponding to the set of service vectors which can be used if $X(t) = \eta$. We have

$$X(t) \in \mathcal{C}^o(\eta) \implies \langle S(t), \mathbf{\Delta}\eta \rangle = \max_{S \in \mathcal{S}} \langle S, \mathbf{\Delta}\eta \rangle \tag{3.13}$$

where $\mathcal{C}^o(\eta)$ is the interior of $\mathcal{C}(\eta)$. Define now

$$\mathcal{K}(\eta) = \{x \in \mathbb{R}^Q_{0+} : x_q > \max_{S \in \mathcal{S}} \{S_q\} \text{ for each } q \text{ with } \eta_q > 0\}, \tag{3.14}$$

which is the set of backlogs such that no queue q with $\eta_q > 0$ can be emptied in a single time slot. Note that $\mathcal{K}(\eta)$ is upward-scalable; indeed, $x \in \mathcal{K}(\eta)$ implies $\alpha x \in \mathcal{K}(\eta)$ for any scalar $\alpha > 1$. Note that when $X(t) \in \mathcal{K}(\eta)$ we have $X_q(t) > \max_{S \in \mathcal{S}} \{S_q\}$ for all $q \in \mathcal{Q}$ with $\eta_q > 0$, so $D_q(t) = \min\{X_q(t), S_q(t)\} = S_q(t)$. Therefore,

$$X(t) \in \mathcal{K}(\eta) \implies D_q(t) = S_q(t) \text{ for all } q \in \mathcal{Q} \text{ with } \eta_q > 0. \tag{3.15}$$

That is, all service capacity allocated at slot t to queue q with $\eta_q > 0$ is used; there is no idling in that time slot; and the departures from all queues q with $\eta_q > 0$ is exactly equal to the total service provided to that queue. Consider now the set

$$\mathcal{V}(\eta) = \mathcal{K}(\eta) \bigcap \mathcal{C}^o(\eta) \tag{3.16}$$

and note that it is upward-scalable, that is, $x \in \mathcal{V}(\eta)$ implies $\alpha x \in \mathcal{V}(\eta)$ for any scalar $\alpha > 1$. Thus, the set $\mathcal{V}(\eta)$ is 'cone-like'.

The main idea for the proof for the existence of our limit is that excursions away from η take a very long time–so long that the time-average properties we have for arrivals will become active. In particular, there will be some time after which all deviations of $\frac{X(t)}{t}$ away from η will remain in $\mathcal{V}(\eta)$. To show this, we will need to demonstrate certain properties of these excursions.

3.1. Structural Properties

We now identify a number of structural properties of η and related elements. The proofs of these properties can be found in the Appendix. These properties are important to characterizing deviations from η and showing our main result on the limit of $\frac{X(t)}{t}$.

Lemma 3.2*For every sequence $\{t'_c\}$ such that $t'_c < t_c$ and*

$$X(t) \in \mathcal{V}(\eta) \quad \textit{for every} \quad X(t) \in (t'_c, t_c] \tag{3.17}$$

for every c, we have

$$\left\langle \frac{X(t_c) - X(t'_c)}{t_c - t'_c}, \mathbf{\Delta}\eta \right\rangle = \left\langle \frac{\sum_{t=t'_c}^{t_c-1} A(t)}{t_c - t'_c}, \mathbf{\Delta}\eta \right\rangle - \max_{S \in \mathcal{S}} \langle S(t), \mathbf{\Delta}\eta \rangle . \tag{3.18}$$

During the interval $(t'_c, t_c]$, $X(t) \in \mathcal{K}(\eta)$, which means any service vector used in this interval will maximize the inner product: $\langle S(t), \mathbf{\Delta}\eta \rangle$. Furthermore, by definition of $\mathcal{K}(\eta)$, there is no idling for all q such that $\eta_q > 0$. This result simply accounts for the new jobs which arrive and the jobs which are serviced over the subset of queues with $\eta_q > 0$.

Lemma 3.3*For any increasing unbounded time sequences $\{t_n\}$ and $\{t'_n\}$, we have*

$$\lim_{n\to\infty} \frac{t_n - t'_n}{t_n} = \chi \in (0,1] \implies \lim_{n\to\infty} \frac{\sum_{t=t'_n}^{t_n-1} A(t)}{t_n - t'_n} = \rho \tag{3.19}$$

Therefore we can define the time-average workload over sub-intervals which grow linearly in time.

Lemma 3.4*For any increasing unbounded subsequence $\{t_m\}$ with $\lim_{m\to\infty} \frac{X(t_m)}{t_m} = \mu$, we have*

$$\langle \mu, \Delta\eta \rangle \geq \langle \rho, \mathbf{\Delta}\eta \rangle - \max_{S \in \mathcal{S}} \langle S, \mathbf{\Delta}\eta \rangle \tag{3.20}$$

Lemma 3.5*For any increasing unbounded subsequence $\{t_m\}$ with $\lim_{m\to\infty} \frac{X(t_m)}{t_m} = \mu$, we have*

$$\langle \mu, \mathbf{\Delta}\eta \rangle \geq \langle \eta, \mathbf{\Delta}\eta \rangle \implies \mu = \eta \tag{3.21}$$

Lemma 3.6*For every $\epsilon \in (0,1)$ we have*

$$-\left[\langle \rho, \mathbf{\Delta}\eta \rangle - \max_{S \in \mathcal{S}} \langle S, \mathbf{\Delta}\eta \rangle\right] \frac{\epsilon}{1-\epsilon} + \langle \eta, \mathbf{\Delta}\eta \rangle \frac{1}{1-\epsilon} \geq \langle \eta, \mathbf{\Delta}\eta \rangle \tag{3.22}$$

3.2. Uniqueness of limit $\lim_{\mathbf{t}\to\infty} \frac{\mathbf{X(t)}}{\mathbf{t}}$ on an individual arrival trace

In order to prove the uniqueness of the limit on a given arrival trace, we need one more definition and one more property. We shall begin by assuming that there is some other convergent subsequence $\{X(t_a)\}$ such that $\lim_{a\to\infty} \frac{X(t_a)}{t_a} = \psi \neq \eta$. Note that $\psi_q < \infty$ for all q. This is easy to see since $\psi_q = \lim_{a\to\infty} \frac{X_q(t_a)}{t_a} \leq \lim_{a\to\infty} \frac{A_q(t_a)}{t_a} = \rho_q < \infty$. We will eventually show that this subsequence does not exist.

Recall that we have subsequence $\{t_c\}$ which achieves the '$\limsup$' in (3.2):

$$\lim_{c\to\infty} \frac{X(t_c)}{t_c} = \eta \tag{3.23}$$

Define

$$s_c = \max\{t_a : t_a < t_c\} < t_c \tag{3.24}$$

Since s_c is a subsequence of t_a, $\lim_{c\to\infty} \frac{X(s_c)}{s_c} = \psi$. This is a subsequence of the deviations away from η. Given this definition of $\{s_c\}$, we can show the following property:

Lemma 3.7*We have that*

$$\liminf_{c\to\infty} \frac{t_c - s_c}{t_c} = \epsilon \in (0,1) \tag{3.25}$$

i.e. s_c grows nearly linearly with t_c. The proof of this Lemma can be found in the Appendix.

We are now prepared to show the uniqueness of the limit on an individual arrival trace. That is:

Proposition 3.1*There is no subsequence* $\{\mathbf{t_a}\}$ *with* $\lim_{\mathbf{a}\to\infty} \frac{\mathbf{X(t_a)}}{\mathbf{t_a}} = \psi \neq \eta$*. Therefore,*

$$\lim_{t\to\infty} \frac{X(t)}{t} = \eta \tag{3.26}$$

PROOF: Arguing by contradiction, assume that there is some other convergent subsequence $\{X(t_a)\}$ such that $\lim_{a\to\infty} \frac{X(t_a)}{t_a} = \psi \neq \eta$. We shall show that this is impossible. As stated before, note that $\psi_q < \infty$ for all q. Select now a subsequence of $\{t_c\}$ on which the '$\liminf$' is attained in (3.25), but keep the same indexing c of the original one for notational simplicity, hence,

$$\lim_{c\to\infty} \frac{t_c - s_c}{t_c} = \epsilon \in (0,1). \tag{3.27}$$

by Lemma 3.7; hence the length of time of the deviations grows linearly with t_c. Therefore, $\lim_{c\to\infty} \frac{t_c}{s_c} = \frac{1}{1-\epsilon}$ and $\lim_{c\to\infty} \frac{t_c - s_c}{s_c} = \frac{\epsilon}{1-\epsilon}$.

Applying Lemmas 3.2 and 3.3 with $\{t'_c\} = \{s_c\}$, dividing by $t_c - s_c$ and letting $c\to\infty$, we get

$$\lim_{c\to\infty} \left\langle \frac{X(t_c) - X(s_c)}{t_c - s_c}, \mathbf{\Delta}\eta \right\rangle = \langle \rho, \mathbf{\Delta}\eta \rangle - \max_{S\in\mathcal{S}} \langle S, \mathbf{\Delta}\eta \rangle \,. \tag{3.28}$$

Then, we can write

$$\begin{aligned}
\langle \psi, \mathbf{\Delta}\eta \rangle &= \lim_{c\to\infty} \left\langle \frac{X(s_c)}{s_c}, \mathbf{\Delta}\eta \right\rangle \\
&= \lim_{c\to\infty} \left\langle -\frac{X(t_c) - X(s_c)}{t_c - s_c} \, \frac{t_c - s_c}{s_c} + \frac{X(t_c)}{t_c} \, \frac{t_c}{s_c}, \mathbf{\Delta}\eta \right\rangle \\
&= -\left[\langle \rho, \mathbf{\Delta}\eta \rangle - \max_{S\in\mathcal{S}} \langle S, \mathbf{\Delta}\eta \rangle \right] \frac{\epsilon}{1-\epsilon} + \langle \eta, \mathbf{\Delta}\eta \rangle \, \frac{1}{1-\epsilon} \\
&\geq \langle \eta, \mathbf{\Delta}\eta \rangle
\end{aligned} \tag{3.29}$$

The last equality is due to Lemma 3.6. Therefore, $\langle \psi, \mathbf{\Delta}\eta \rangle \geq \langle \eta, \mathbf{\Delta}\eta \rangle$, which implies $\psi = \eta$ from Lemma 3.5. But this contradicts the assumption that $\psi \neq \eta$. This establishes the sought after contradiction. So for

each individual arrival trace, there exists a unique limit $\lim_{t\to\infty}\frac{X(t)}{t}=\eta$, which concludes the proof of the proposition. Moreover, since $\lim_{t\to\infty}\frac{X(t)}{t}=\eta$ this implies that there exists $t_o<\infty$ such that $X(t)$ is in $\mathcal{V}(\eta)$ for all $t>t_o$. ■

The main argument is essentially that no other sublimit can get very far from t_c and any excursion away from η is small. Therefore at large t, all subsequences are close enough to η to be in one of the neighboring cones, i.e. in $\mathcal{V}(\eta)$. We have now shown that there exists a unique limit, η, such that on a given arrival trace: $\lim_{t\to\infty}\frac{X(t)}{t}=\eta$. To complete the proof of Theorem 3.1, it remains to show that the limit, η, is independent of the particular arrival trace. To do this, we turn to characterizing η.

3.3. Characterizing the limit η

The purpose of this section is to characterize the limit η in terms of ρ and service vectors $\mathcal{S}$ to establish the independence of η on the individual arrival trace. Knowing that $\lim_{t\to\infty}\frac{X(t)}{t}=\eta$, we now turn to identifying a couple of the characteristic properties of η. The proofs of these Lemmas can be found in the Appendix.

Lemma 3.8*Every limit is a fixed point. That is,*

$$\eta=\lim_{t\to\infty}\frac{X(t)}{t}=\Big[\rho-\sum_{m=1}^{N}\alpha_m S_m\Big]^+ \tag{3.30}$$

for some $\alpha_m\geq 0,\sum_m\alpha_m=1$. *Furthermore,* $\alpha_m>0$ *implies that* $\eta\in C_{S_m}$.

Note that because $\alpha_m>0$ implies that $\eta\in C_{S_m}$, we know that η is in the intersection of all cones with $\alpha_m>0$. Hence, if there are multiple $\alpha_m>0$, then η is on the boarder of all cones with $\alpha_m>0$. Because η is a fixed point:

Lemma 3.9*The following equality holds:*

$$\langle\eta,\mathbf{\Delta}\eta\rangle=\langle\rho,\mathbf{\Delta}\eta\rangle-\max_{S\in\mathcal{S}}\langle S,\mathbf{\Delta}\eta\rangle \tag{3.31}$$

We are now in position to show that η depends only on the load vector, ρ, and available service vectors, $\mathcal{S}$. That is, η is independent of the particular arrival trace and is the solution of a simple convex program.

Proposition 3.2*The vector* $\eta=\lim_{t\to\infty}\frac{X(t)}{t}$ *is the unique minimizer of*

$$\langle\eta,\mathbf{\Delta}\eta\rangle=\min_{\eta'\in\Psi(\rho,\mathcal{S})}\langle\eta',\mathbf{\Delta}\eta'\rangle \tag{3.32}$$

where

$$\Psi(\rho,\mathcal{S})=\{\eta':\eta'=(\rho-r)^+ \text{ with } r\in\mathcal{P}\} \tag{3.33}$$

and $\mathcal{P}$ *is the stability region given by* $\mathcal{S}$. *Therefore,* $r=\sum_{S\in\mathcal{S}}\alpha_S S$ *with* $\sum_{S\in\mathcal{S}}\alpha_S\leq 1$ *and* $\alpha_S\geq 0$ *for each* $S\in\mathcal{S}$, *where* $\mathcal{S}$ *is the set of service vectors.*

PROOF: First we show that η is indeed the solution to (3.32). Then we show there is only one solution in order to conclude that η is unique.

From Lemma 3.8, we have that $\eta \in \Psi(\rho, \mathcal{S})$. Arbitrarily choose any vector

$$\bar{\eta} = \Big(\rho - \sum_{S\in\mathcal{S}} \alpha_S S\Big)^+, \text{ with } \sum_{S\in\mathcal{S}} \alpha_S \leq 1, \text{ and } \alpha_S \geq 0, S \in \mathcal{S}. \tag{3.34}$$

Projecting on $\mathbf{\Delta}\eta$ we get

$$\begin{aligned}
\langle \bar{\eta}, \mathbf{\Delta}\eta \rangle &= \left\langle \Big[\rho - \sum_{S\in\mathcal{S}} \alpha_S S\Big]^+, \mathbf{\Delta}\eta \right\rangle \\
&\geq \left\langle \rho - \sum_{S\in\mathcal{S}} \alpha_S S, \mathbf{\Delta}\eta \right\rangle \\
&= \langle \rho, \mathbf{\Delta}\eta \rangle - \sum_{S\in\mathcal{S}} \alpha_S \langle S, \mathbf{\Delta}\eta \rangle \\
&\geq \langle \rho, \mathbf{\Delta}\eta \rangle - \max_{S\in\mathcal{S}} \langle S, \mathbf{\Delta}\eta \rangle = \langle \eta, \mathbf{\Delta}\eta \rangle
\end{aligned} \tag{3.35}$$

The first inequality comes from the fact that $\mathbf{\Delta}_{qq} > 0$ and $\eta_q \geq 0$. The last equality comes from Lemma 3.9. Therefore, $\langle \bar{\eta}, \mathbf{\Delta}\eta \rangle \geq \langle \eta, \mathbf{\Delta}\eta \rangle$. This implies (recalling that $\mathbf{\Delta}$ is positive-definite) that

$$\begin{aligned}
0 &\leq \langle \bar{\eta} - \eta, \mathbf{\Delta}(\bar{\eta} - \eta) \rangle \\
&= \langle \bar{\eta}, \mathbf{\Delta}\bar{\eta} \rangle - 2\langle \bar{\eta}, \mathbf{\Delta}\eta \rangle + \langle \eta, \mathbf{\Delta}\eta \rangle \\
&\leq \langle \bar{\eta}, \mathbf{\Delta}\bar{\eta} \rangle - 2\langle \eta, \mathbf{\Delta}\eta \rangle + \langle \eta, \mathbf{\Delta}\eta \rangle \\
&= \langle \bar{\eta}, \mathbf{\Delta}\bar{\eta} \rangle - \langle \eta, \mathbf{\Delta}\eta \rangle,
\end{aligned} \tag{3.36}$$

so $\langle \bar{\eta}, \mathbf{\Delta}\bar{\eta} \rangle \geq \langle \eta, \mathbf{\Delta}\eta \rangle$ and η is the minimizer of $\langle \eta', \mathbf{\Delta}\eta' \rangle$.

We still need to prove that the minimizer η is unique. This is done by showing that 1) $\langle \eta, \mathbf{\Delta}\eta \rangle$ is strictly convex in η and 2) the set $\Psi(\rho, \mathcal{S})$ is convex–uniqueness will follow from convex programming theory. 1) It is trivial to show that $\langle \eta, \mathbf{\Delta}\eta \rangle$ is strictly convex in η since $\mathbf{\Delta} > 0$ is a positive definite matrix. 2) We now show that the set $\Psi \equiv \Psi(\rho, \mathcal{S})$ is convex. First, we see that for any $r \in \mathcal{P}$ and corresponding $x = (\rho - r)^+ \in \Psi$, there exists $\bar{x} = \rho - \bar{r} = (\rho - r)^+ = x$ with $\bar{r} \in \mathcal{P}$. Let $\bar{r}_k = \min(\rho_k, r_k) \leq r_k$. Since $\bar{r} \leq r \in \mathcal{P}$, then $\bar{x} \in \Psi$. Now, consider two vectors $x, x' \in \Psi$ with corresponding $r, r' \in \mathcal{P}$ such that $x = (\rho - r)^+$ and $x' = (\rho - r')^+$. What remains to be shown is that for any $a \in [0, 1]$, $ax + (1-a)x' \in \Psi$. Indeed, we have:

$$\begin{aligned}
ax + (1-a)x' &= a\bar{x} + (1-a)\bar{x}' \\
&= a(\rho - \bar{r}) + (1-a)(\rho - \bar{r}') \\
&= \rho - (a\bar{r} + (1-a)\bar{r}')
\end{aligned} \tag{3.37}$$

By the convexity of $\mathcal{P}$, we know that $a\bar{r} + (1-a)\bar{r}' \in \mathcal{P}$ and subsequently, $\rho - (a\bar{r} + (1-a)\bar{r}') \in \Psi$. This concludes the proof. ■

Based on the characterization of η as the solution to the convex program (3.32), via convex optimization theory, we can conclude that there is only one fixed point.

Lemma 3.10*There exists exactly one fixed point, η:*

$$\begin{aligned} \eta &= [\rho - \sum_m \alpha_m S_m]^+ \\ 0 &\leq \alpha_m \\ 1 &= \sum_m \alpha_m \\ \alpha_m > 0 &\implies \langle \eta, \mathbf{\Delta} S_m \rangle \geq \langle \eta, \mathbf{\Delta} S_k \rangle, \forall k \end{aligned} \tag{3.38}$$

We have just shown that on all arrival traces with system load ρ, $\lim_{t\to\infty} \frac{X(t)}{t} = \eta$, is unique. Furthermore, the limit, η, is identical across all such traces. η can be characterized as the unique solution to the convex program (3.9); equivalently, it is the unique fixed point of our system. This concludes the proof of Theorem 3.1.

To summarize, whenever the system is in overload the workload grows along a vector defined by η which is the solution of the convex program in Proposition 3.2. Moreover, η is *independent of the particular arrival trace*. From Lemma 3.8, we know that η is on the intersection of some set of cones. If there exists only one $\alpha_m > 0$, then $\eta = [\rho - S_m]^+$ and is in C_{S_m}. If there are multiple $\alpha_m > 0$, then $\eta = [\rho - \sum_m S_m]^+$ is on the boundary of the set of cones with $\alpha_m > 0$. We will utilize this information to control for our fairness criterion.

4. Fair Control via the $\mathbf{\Delta}$ matrix

We have now seen how the asymptotic behavior of the workload vector, $X(t)$, behaves given service vectors $\mathcal{S}$ and MaxWeight matrix $\mathbf{\Delta}$. In particular, when the system load is outside of the stability region, the workload will explode along a single vector. During a long period of temporary stress, the queueing system is effectively *unstable* during this window and the valuable service resources become strained. Under the MaxWeight scheduling policy, queues with exceptionally high load will starve resources from other, less stressed, queues. This begs the question of how to share resources in a fair manner when the system is unstable. Our goal in this section is to consider how to manipulate the $\mathbf{\Delta}$ matrix in order to ensure 'fairness' in this queueing system.

As we have discussed before, there are many different definitions of fairness. We focus on a notion of fairness where the workloads grow according to a fixed proportion. More formally, we assume we are given a set of ratios $\theta_q \geq 0, \sum_q \theta_q = 1$. Whenever $\rho \notin \mathcal{P}$, the goal is to control the workload such that:

$$\lim_{t\to\infty} \frac{X_q(t)}{\sum_k X_k(t)} = \theta_q, \forall q \in \mathcal{Q} \tag{4.1}$$

From Theorem 3.1, we know that $\lim_{t\to\infty} \frac{X(t)}{t} = \eta$ and we want:

$$\frac{\eta_q}{\sum_i \eta_i} = \theta_q, \forall q \tag{4.2}$$

When $\rho \in \mathcal{P}$, the system is stablizable, and so the workloads should remain finite. For large t:

$$\frac{X_q(t)}{t} \to 0, \forall q \tag{4.3}$$

We know that the MaxWeight scheduling policy guarantees the second criterion–it is a stabilizing policy (Ross and Bambos 2009). The focus of this section is to show that the MaxWeight scheduling policy also satisfies the first criterion under certain conditions via proper specification of the $\mathbf{\Delta}$ matrix.

4.1. Control via the Δ matrix

From Lemma 3.8, we know that η is on the intersection of the set of cones with $\alpha_m > 0$ in the definition of $\eta = (\rho - \sum_m \alpha_m S_m)^+$. This corresponds is a cone boundary if there are more than one $\alpha_m > 0$. This boundary depends on the $\mathbf{\Delta}$ matrix used in the MaxWeight scheduling policy. We now consider how we can choose the $\mathbf{\Delta}$ matrix to place cone boundaries and, subsequently, achieve the desired fairness criterion.

Lemma 4.1*In Q-dimensions, consider any M service vectors $S_{i_1}, S_{i_2}, \ldots, S_{i_M}$. Suppose there exists a diagonal positive definite matrix $\hat{\mathbf{\Delta}}$ and non-negative Q-dimensional vector $v \geq 0$ such that v is a* boundary vector *of the M cones, i.e. for each $k \in [1, M]$ and for all j:*

$$\left\langle v, \hat{\mathbf{\Delta}} S_{i_k} \right\rangle \geq \left\langle v, \hat{\mathbf{\Delta}} S_j \right\rangle \tag{4.4}$$

Then, for any non-negative vector η such that $\eta_q = 0$ if and only if $v_q = 0$, there exists a diagonal positive definite matrix $\mathbf{\Delta}$ such that for each $k \in [1, M]$ and all j:

$$\left\langle \eta, \mathbf{\Delta} S_{i_k} \right\rangle \geq \langle \eta, \mathbf{\Delta} S_j \rangle \tag{4.5}$$

i.e. the boundary between the M cones can be placed arbitrarily in $\mathbb{R}_+^Q$. This matrix is specified as:

$$\mathbf{\Delta}_{qq} = \begin{cases} \frac{\hat{\mathbf{\Delta}}_{qq} v_q}{\eta_q}, & v_q > 0; \\ 1, & v_q = 0. \end{cases} \tag{4.6}$$

This proof is given in the Appendix.

Because Lemma 4.1 holds for *any* positive definite diagonal matrix $\hat{\mathbf{\Delta}}$, it will hold for $\hat{\mathbf{\Delta}} = \mathbf{I}$. Furthermore, it is easy to verify condition (4.4) when $\hat{\mathbf{\Delta}} = \mathbf{I}$, as this results in taking simple inner products.

Corollary 4.1*The boundary, $\eta \geq 0$, between M cones defined by service vectors $S_{i_1}, S_{i_2}, \ldots, S_{i_M}$ can be placed arbitrarily in $\mathbb{R}_+^Q$ by using MaxWeight scheduling with matrix $\mathbf{\Delta}$ if there exists $v \geq 0$ on the boundary of the M cones induced by $\hat{\mathbf{\Delta}} = \mathbf{I}$:*

$$\left\langle v, S_{i_k} \right\rangle \geq \langle v, S_j \rangle, \forall k \in [1, M], \forall j \tag{4.7}$$

Furthermore, $v_q = 0$ if and only if $\eta_q = 0$. The required $\mathbf{\Delta}$ is:

$$\mathbf{\Delta}_{qq} = \begin{cases} \frac{v_q}{\eta_q}, & v_q > 0; \\ 1, & v_q = 0. \end{cases} \tag{4.8}$$

Under certain necessary and sufficient conditions we can arbitrarily place the cone boundary and we can then control the workload to explode along a desired vector defined by η.

Proposition 4.1*There exists MaxWeight matrix $\mathbf{\Delta}$ such that*

$$\lim_{t \to \infty} \frac{X(t)}{t} = \eta \tag{4.9}$$

if and only if the following conditions hold:

1. *$\eta = (\rho - \sum_m \alpha_m S_m)^+$ for some $\alpha_m \geq 0, \sum_m \alpha_m = 1$.*
2. *There exists $v \geq 0$ such that $\alpha_m > 0$ implies $\langle v, S_m \rangle \geq \langle v, S_k \rangle$ for all k. Furthermore, $v_q = 0$ if and only if $\eta_q = 0$.*

This allows us to characterize the *feasible* fairness criterion:

Corollary 4.2*Fairness criterion θ is feasible via MaxWeight Matching if and only if:*

1. *There exist M service vectors and $v \geq 0$, with $v_q = 0$ if $\theta_q = 0$, such that*

$$\left\langle v, S_{i_k} \right\rangle \geq \langle v, S_j \rangle, \forall k \in [1, M], \forall j \tag{4.10}$$

2. *For vectors S_{i_m} in 1, there exist $\alpha_m \geq 0$, $\sum_{m=1}^M \alpha_m = 1$ such that $\eta = (\rho - \sum_{m=1}^M \alpha_m S_{i_m})^+$. Additionally, for each $\alpha_m > 0$:*

$$\langle v, S_m \rangle \geq \langle v, S_k \rangle, \forall k \tag{4.11}$$

3. *η must satisfy:*

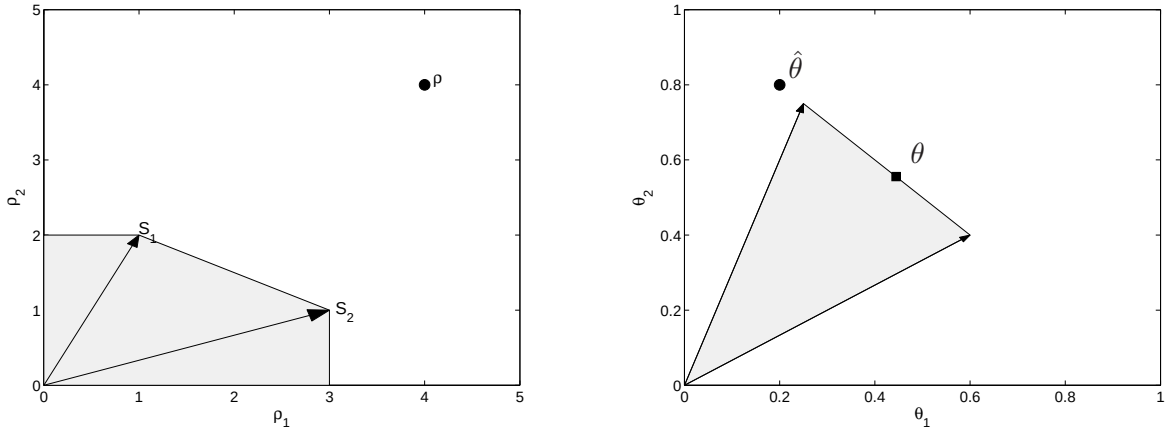
$$\theta_q = \frac{\eta_q}{\sum_k \eta_k} \tag{4.12}$$

This is a direct consequence of Proposition 4.1.

4.1.1. Feasible Criteria: An Example We now present an example with $N = 2$ service vectors and $Q = 2$ queues and specify the feasible criteria given $\rho = [4, 4]$, $S_1 = [1, 2]$ and $S_2 = [3, 1]$. It is easy to see that $v = [1, 2]$ satisfies Condition 1 in Corollary 4.2 as long as $\theta > 0$. Now, by Condition 2,

$$\begin{aligned}\eta &= (\rho - \alpha S_1 - (1-\alpha)S_2)^+ \\ &= \alpha(\rho - S_1) + (1-\alpha)(\rho - S_2) \\ &= \alpha \begin{bmatrix} 1 \\ 3 \end{bmatrix} + (1-\alpha) \begin{bmatrix} 3 \\ 2 \end{bmatrix}, \forall \alpha \in [0, 1] \end{aligned} \tag{4.13}$$

To find the feasible θ, we normalize η as in Condition 3. Hence, any $\theta = \alpha \begin{bmatrix} 1/4 \\ 3/4 \end{bmatrix} + (1-\alpha) \begin{bmatrix} 3/5 \\ 2/5 \end{bmatrix}, \alpha \in [0, 1]$ is feasible. In Figure 4.1.1, we can see the stability region and ρ, which is outside of the stability region. All fairness directions in the gray portion, such as θ, are feasible. Other fairness criteria, such as $\hat{\theta}$, are infeasible.



(a)Stability Region (b)Feasible Fairness Criteria

Figure 2 Feasible Fairness Criteria for ρ: $N = 2$ service vectors and $Q = 2$ queues.

4.1.2. An Infeasible Feasible Criterion One might naturally ask whether the set of feasible fairness directions according to MaxWeight Matching includes all feasible directions when a more general set of policies is allowed. It turns out the answer is no. We demonstrate this via an example where the fairness criterion is *not* achievable via MaxWeight Matching but there exists a policy which does achieve it. Consider a scenario with $Q = 3$ queues and $N = 3$ service vectors:

$$S_1 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, S_2 = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}, S_3 = \begin{pmatrix} 3/4 \\ 3/4 \\ 2 \end{pmatrix} \tag{4.14}$$

Let the fairness criteria and system load be:

$$\theta = \begin{pmatrix} 1/3 \\ 1/3 \\ 1/3 \end{pmatrix}, \rho = \begin{pmatrix} 13/8 \\ 13/8 \\ 5/2 \end{pmatrix} \tag{4.15}$$

It is easy to see that using a combination of all three service vectors, the fairness criteria can be achieved:

$$\eta = \left[\rho - \frac{1}{4}S_1 - \frac{1}{4}S_2 - \frac{1}{2}S_3\right]^+ = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \tag{4.16}$$

However, this fairness criteria *cannot* be met using MaxWeight Matching. In particular, by Conditions 1 and 2 of Corollary 4.2, there must exists some $v > 0$ (since $\theta > 0$) such that:

$$\langle v, S_1 \rangle = \langle v, S_2 \rangle = \langle v, S_3 \rangle \tag{4.17}$$

With some algebra, one can see that the v which satisfies (4.17) is

$$v = \gamma \begin{pmatrix} 2 \\ 2 \\ -1 \end{pmatrix} \tag{4.18}$$

for any γ. Hence, there does not exists $v > 0$ and Conditions 1 and 2 of Corollary 4.2 cannot be satisfied. This θ is not feasible via MaxWeight Matching.

We find that MaxWeight Matching allows us to control for a general, characterizable set of fairness criterion; however, there may be other policies which can achieve other criteria. Note that as long as MaxWeight policies can achieve the fairness criteria, it is guaranteed to achieve it in the most efficient manner by Theorem 4.1.

4.2. Workload Minimization

Recall that our ultimate goal is to minimize the long run average backlog (2.18) subject to the fairness criterion (2.17). So far we have established that, given a feasible fairness criterion θ, the MaxWeight scheduling policy will satisfy this criterion with the right choice of the matrix $\mathbf{\Delta}$. In particular, we have identified the 'direction' which the workload will grow as well as a methodology to control this direction. However, we have not specified the rate at which the workload will grow. We have used the MaxWeight scheduling policy to achieve our fairness criterion and justified the use of these algorithms because they are stabilizing when $\rho \in \mathcal{P}$, they are simple to implement, and they allow us to achieve our fairness criterion. We now discuss another feature of the MaxWeight scheduling policy which makes it highly desirable for our purposes. Namely, it achieves the *smallest* workload which satisfies our fairness criterion.

Theorem 4.1*Let $X(t)$ be the workload achieved under the MaxWeight scheduling policy. Let $\bar{X}(t)$ be the workload achieved under some other algorithm such that:*

$$\lim_{t\to\infty}\frac{X(t)}{t}=\eta,\lim_{t\to\infty}\frac{\bar{X}(t)}{t}=\bar{\eta} \tag{4.19}$$

where both achieve the desired fairness criterion:

$$\frac{\eta_q}{\sum_k\eta_k}=\frac{\bar{\eta}_q}{\sum_k\bar{\eta}_k}=\theta_q,\forall q \tag{4.20}$$

Then, η is the minimal *workload vector which can achieve the fairness criterion, θ:*

$$\eta\leq\bar{\eta} \tag{4.21}$$

The proof of this Theorem is given in the Appendix.

While there may be many algorithms which achieve our fairness criterion, θ, MaxWeight scheduling is efficient in the sense that no other algorithm has smaller backlogs. Note that our notion of efficiency is different than that from Shah and Wischik (2011). In their work, they show that MaxWeight is *not* efficient. However, they define efficiency as the total throughput of the system, i.e. they show that MaxWeight does not minimize $\sum_q X_q(t)$ over *all* possible directions. This is quite different from what we show, which is that, given a direction θ, then $\sum_q X_q(t)$ is minimized.

4.3. Robustness with Respect to ρ

Thus far, we have assumed that the load vector ρ is known. Under this assumption, we are able to select the necessary MaxWeight matrix, $\mathbf{\Delta}$, to achieve the desired fairness criterion, θ, as long as it is feasible. Now we suppose that ρ is unknown and examine whether we are still able to choose $\mathbf{\Delta}$ to achieve the desired fairness criterion. Throughout this discussion, we will assume that $N>1$; otherwise there is no control and $\eta=(\rho-S)^+$ for all ρ, irrespective of $\mathbf{\Delta}$.

Consider the following example with $N=3$ service vectors and $Q=2$ queues as depicted in Figure 3. Let

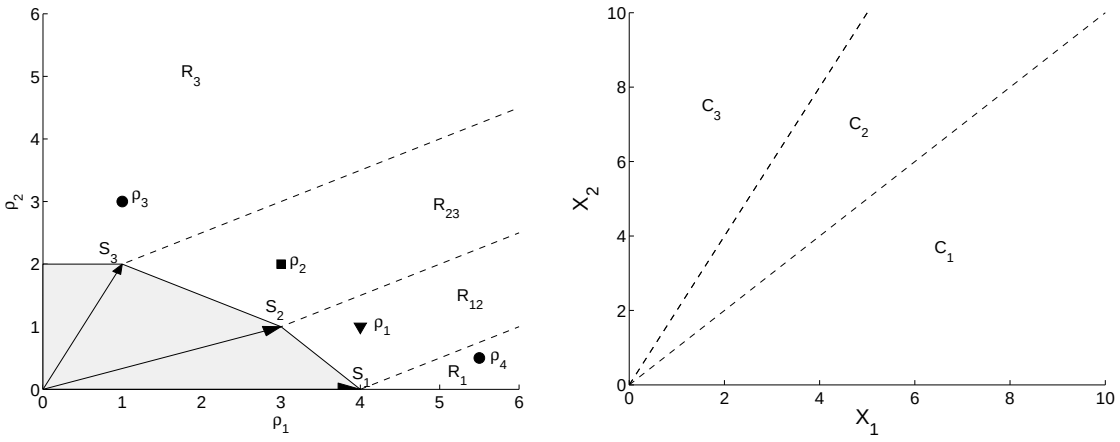
$$S_1=\begin{pmatrix}4\\0\end{pmatrix},S_2=\begin{pmatrix}3\\1\end{pmatrix},S_3=\begin{pmatrix}1\\2\end{pmatrix} \tag{4.22}$$

Suppose our fairness criteria is:

$$\theta=\begin{pmatrix}2/3\\1/3\end{pmatrix} \tag{4.23}$$

We consider 3 different load vectors, which are outside of the stability region:

$$\rho_1=\begin{pmatrix}4\\1\end{pmatrix},\rho_2=\begin{pmatrix}3\\2\end{pmatrix},\rho_3=\begin{pmatrix}1\\3\end{pmatrix},\rho_4=\begin{pmatrix}5\\.5\end{pmatrix} \tag{4.24}$$



(a)Stability Region (b)Scheduling cones

Figure 3 Stability and Cone regions for $N = 3$ service vectors and $Q = 2$ queues.

When the system load $\rho = \rho_3$ or ρ_4,Conditions 2 and 3 in Corollary 4.2 cannot be satisfied; hence, the fairness direction, θ, is infeasible and there does not exist a MaxWeight matrix to achieve it. With some algebra, we can see that the necessary MaxWeight matrix to achieve the fairness criteria depends on ρ:

$$\mathbf{\Delta}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}, \mathbf{\Delta}_2 = \begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix} \tag{4.25}$$

From Theorem 3.1, the workload is given by $\eta = [\rho - \sum_m \alpha_m S_m]^+$. Hence, to achieve the desired fairness criteria, the goal is to find a point on the boundary of the stability region such that subtracting that point from the system load, ρ, results in a vector which is consistent with the fairness direction θ. From Figure 3(a), we see that a point on the stability boundary which is given by the convex combination of S_1 and S_2 satisfies this constraint for ρ_1. The $\mathbf{\Delta}$ matrix skews the dimensions such that the boundary of interest is placed in the direction of the desired fairness criterion and makes this distance 'minimal' as in Proposition 3.2. Hence, the boundary vector of interest for ρ_1 is the boundary between cones 1 and 2. This boundary vector can be moved to the fairness direction θ by using $\mathbf{\Delta}_1$. Similarly, $\mathbf{\Delta}_2$ moves the boundary vector between cones 2 and 3 for ρ_2. This example shows that the boundary vector of interest and, subsequently, the necessary MaxWeight matrix $\mathbf{\Delta}$ depends on ρ.

Despite the preceding example, it is possible to select $\mathbf{\Delta}$ without precise knowledge of ρ. This ability depends on k, the number of subsets of service vectors $\{S_i\}$ of size greater than 1 which satisfy:

$$\langle v, S_{i_j} \rangle = \langle v, S_{i_n} \rangle > \langle v, S_m \rangle, \forall S_{i_j}, S_{i_n} \in \{S_i\} \text{ and } S_m \notin \{S_i\} \tag{4.26}$$

for some $v \geq 0$, $v \neq \mathbf{0}$. Hence, v is a *boundary vector* as it is a vector on the boundary between neighboring cones $\{S_i\}$. We refer to this boundary as a *relevant boundary*. Note that this boundary is potentially a

hyperplane of dimension greater than 1, so v may not be unique. Since $N > 1$, there exists at least one boundary which satisfies (4.26) ($k \geq 1$). Moreover, with N service vectors, there are $2^N - (N+1)$ subsets of size greater than 1. Clearly, some subsets may not satisfy (4.26), so $k < 2^N - (N+1)$. Our two cases are:

1. $[\mathbf{k = 1}]$ In this case, there is exactly one subset of service vectors with a relevant boundary. If the boundary of this subset is a hyperplane, then one can selected any boundary vector $v \geq 0$ (with $v_q = 0$ if and only if $\theta_q = 0$). Because there is only one boundary, we can specify the MaxWeight Matrix, $\mathbf{\Delta}$, for any fairness criterion, θ. As long as the system load ρ is such that θ satisfies Conditions 2 and 3 in Corollary 4.2, this $\mathbf{\Delta}$ matrix will control the backlogs to grow along θ. Therefore, we can choose $\mathbf{\Delta}$ without knowing ρ. ρ simply determines whether a fairness direction θ is achievable.

2. $[\mathbf{k > 1}]$ In this case, there are multiple (but finite) subsets of service vectors which satisfy (4.26). For each subset, $\mathcal{S}(y) = \{S_i\}$ ($y \in [1, k]$), let $v(y)$ be a non-negative boundary vector with $v(y)_q = 0$ if an only if $\theta_q = 0$. There is a $\mathbf{\Delta}(y)$ such that $\mathbf{\Delta}(y)_{qq} = v(y)_q/\theta_q$ will place the boundary of the $\mathcal{S}(y)$ cones at the direction specified by θ. If θ is feasible, there will be exactly one subset of service vectors $\mathcal{S}(y)$ for each ρ which satisfies Condition 2 of Corollary 4.2. There will be a range of ρ corresponding to each subset $\mathcal{S}(y)$, and subsequently $\mathbf{\Delta}(y)$. As long as our estimate of ρ is within the accuracy of these ranges, we can select the appropriate $\mathbf{\Delta}(y)$ to achieve the desired fairness criterion.

In our example at the beginning of this subsection, there were 2 boundaries of interest: $v_{12} = [1, 1]$ and $v_{23} = [1, 2]$ (see Figure 3). The boundary which matters depends on the system load, ρ, as well as the fairness direction, θ. For $\theta = [2/3, 1/3]$, if ρ is in the lower region, R_{12}, then the boundary vector of interest is v_{12}, between cones C_1 and C_2. If $\rho \in R_{23}$, then the boundary vector of interest is v_{23}, between cones C_2 and C_3. If $\rho \in R_1$ of $\rho \in R_3$, the fairness direction is infeasible. These regions can be determined for each subset of service vectors by solving for the set of ρ which satisfy Condition 2 of Corollary 4.2. As long as we can determine which region ρ resides, the MaxWeight Matrix $\mathbf{\Delta}$ can be specified without precise knowledge of ρ.

See Section 5 for a numerical example of how the state evolution depends on $\mathbf{\Delta}$ and ρ in these two cases.

4.4. Other Fairness Criteria

Thus far we have only considered fairness in terms of the ratios at which workloads will grow. As we have discussed, there are many different notions of fairness. In particular, proportional and max-min fairness are two widely used definitions of fairness. We now discuss how to extend our framework to these notions of fairness.

Max-Min Fairness In Max-Min fairness, the objective is to maximize the minimum utility. If utility is a decreasing function of workload, this would correspond to minimizing the maximum workload. Because

our system is unstable, the workload in each queue will grow without bound. However, we know that the time-normalized workload has a finite limit so that:

$$\lim_{t\to\infty} X(t) \approx \eta t \tag{4.27}$$

Therefore, to minimize the maximum workload, this would correspond to the objective:

$$\min_{\eta} \max_{q} \eta_q \tag{4.28}$$

Hence, we can select $\eta \in \Psi(\rho, \mathcal{S})$ which minimizes the maximum index to achieve Max-Min fairness.

Proportional Fairness In the traditional sense of Proportional fairness, the amount of service resources a queue is allocated corresponds to the proportion of anticipated resource consumption required by the queue, ρ_q. Hence, the fairness criterion would require that the average proportion of service rate to queue q, s_q, is proportional to ρ_q:

$$s_q = \frac{\rho_q}{\sum_k \rho_k} \tag{4.29}$$

This implies that, loosely speaking, $X_q(t) = [X_q(s) + \rho_q(t-s) - s_q S(t-s)]^+$, where $S \leq \sum_q \rho_q$ is the time-average aggregate service rate (if $S > \sum_q \rho_q$ then the system is stablizable). As $t \to \infty$, this implies that

$$\lim_{t\to\infty} \frac{X_q(t)}{t} = K\rho_q \tag{4.30}$$

for some constant $K < 1$. Hence, we can set $\eta = K\rho$ and achieve proportional fairness provided that η is a feasible fairness criterion.

Generally speaking, we can achieve any notion of fairness which can be characterized by a vector η which satisfies the constraints in Corollary 4.2. Note that unlike much of the conventional work on fairness, our framework presupposes that the queueing system is operating in an unstable regime.

5. Numerical Results

In this section, we present some numerical results to demonstrate the performance of MaxWeight Scheduling in a potential real system. We examine how the backlogs grow and approach the fairness criterion. All of our results are asymptotic results with $t\to\infty$. We can see through some numerical simulations how large t must be in practice to approach our asymptotic results.

To start we look at a system with two ($Q = 2$) queues and two ($N = 2$) service vectors given by the numerical examples in Section 4.3.

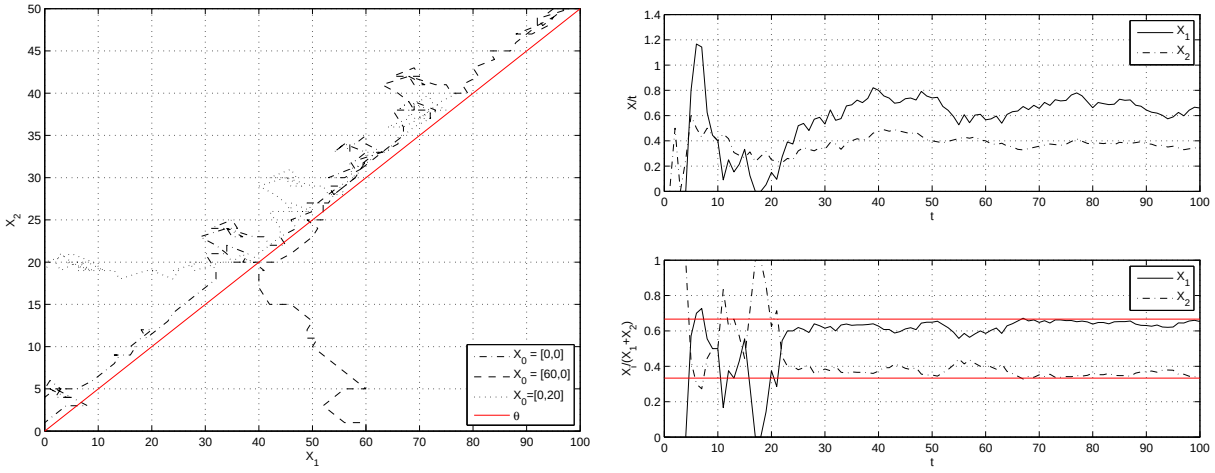
$$S_1 = \begin{pmatrix} 4 \\ 0 \end{pmatrix}, S_2 = \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \rho = \begin{pmatrix} 4 \\ 1 \end{pmatrix}, \theta = \begin{pmatrix} 2/3 \\ 1/3 \end{pmatrix} \tag{5.1}$$

With only $N = 2$ service vectors, there is only one boundary vector; ρ is such that the fairness criterion is feasible (as in Corollary 4.2), so the MaxWeight matrix $\mathbf{\Delta} = \mathbf{\Delta}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$ will achieve the fairness criterion. Our load vector $\rho = [4,1]^T \notin \mathcal{P}$. In each time slot, the number of jobs which arrive to queue 1 is uniformly distributed on $[0,8]$; for queue 2 it is uniformly distributed on $[0,2]$. In this case $\alpha_1 = 1/3, \alpha_2 = 2/3$ so that $\eta = (\rho - \alpha_1 S_1 - \alpha_2 S_2)^+ = [2/3, 1/3]^T$.

We consider how the workload vector grows for various initial conditions:

$$X(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, X(0) = \begin{pmatrix} 60 \\ 0 \end{pmatrix}, X(0) = \begin{pmatrix} 0 \\ 20 \end{pmatrix} \tag{5.2}$$

In Figure 4(a), we plot the trajectories of $X(t)$ for the different initial conditions, along with the line $X_1 = 2X_2$. We can see that all three trajectories converge to the desired fairness vector $\theta_1 = 2/3, \theta_2 = 1/3$. In Figure 4(b), we see the scaled backlogs, $\frac{X_i(t)}{t}$, and the relative backlog, $X_i(t)/\sum_j X_j(t)$, converge starting from initial condition $X(0) = [0,0]$. Moreover, we see that they quickly achieve the fairness criterion which is shown in red.



(a)Different initial conditions.

(b)Initial condition $X(0) = [0,0]$.

Figure 4 Dyanmics for $N = 2$ service vectors and $Q = 2$ queues.

When there are $N = 3$ service vectors, there are $k = 2$ subsets of service vectors and corresponding boundary vectors which satisfy (4.26).

$$S_1 = \begin{pmatrix} 4 \\ 0 \end{pmatrix}, S_2 = \begin{pmatrix} 3 \\ 1 \end{pmatrix}, S_3 = \begin{pmatrix} 0 \\ 4 \end{pmatrix}, \rho = \begin{pmatrix} 4 \\ 1 \end{pmatrix}, \theta = \begin{pmatrix} 2/3 \\ 1/3 \end{pmatrix} \tag{5.3}$$

Now there are two boundary vectors of interest: the one between C_1 and C_2 as well as the one between C_2 and C_3. The boundary which matters depends on ρ as described in Section 4.3. From that discussion, we know that $\mathbf{\Delta}_1$ and $\mathbf{\Delta}_2$ are the necessary MaxWeight matrices to move the boundary vectors between

C_1 and C_2 and between C_2 and C_3, respectively, to fairness direction θ. We refer to $\mathbf{\Delta}(\rho)$ as the necessary MaxWeight Matrix to achieve fairness direction θ, given system load, ρ. Let $\rho = \rho_1 = [4,1]^T$; we use $\mathbf{\Delta}_1$ as before and we can achieve the desired fairness criterion. On the other hand, if $\rho = \rho_2 = [3,2]^T$, $\mathbf{\Delta}_1$ will not achieve the desired fairness criterion, but matrix $\mathbf{\Delta}_2$ will. Hence:

$$\mathbf{\Delta}(\rho_1) = \mathbf{\Delta}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}, \mathbf{\Delta}(\rho_2) = \mathbf{\Delta}_2 = \begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix} \tag{5.4}$$

We can see in Figure 5 how the asymptotic dynamics of the queues depend on ρ and $\mathbf{\Delta}$. The fairness criterion is shown in red.

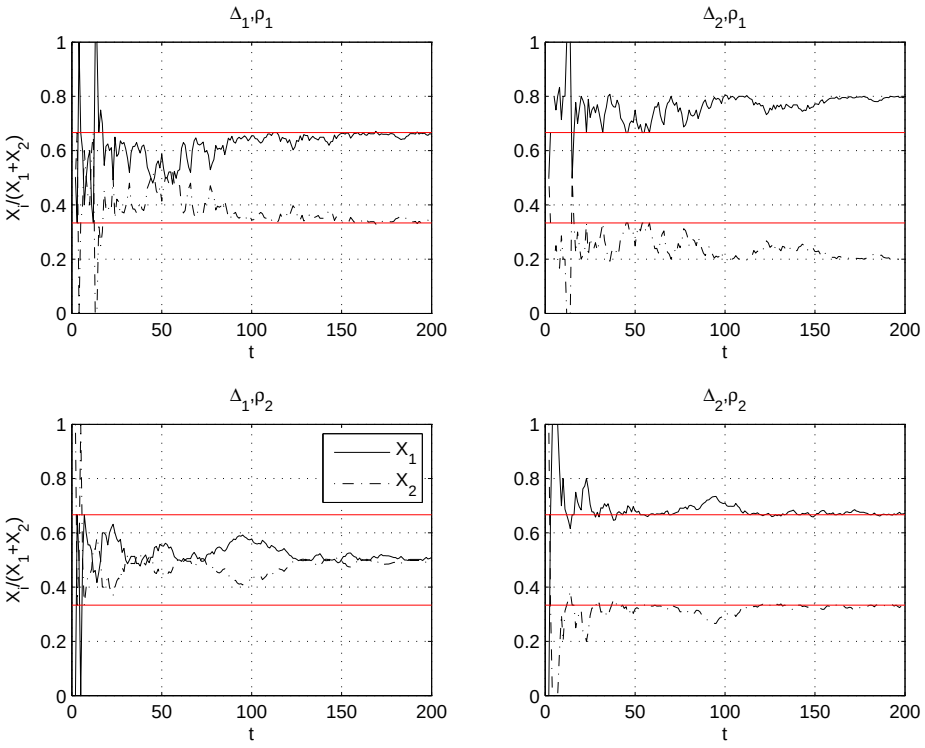


Figure 5 Dynamics of 2-queue queueing system with 3 service vectors.

In the next experiment, we consider a system with $Q = 3$ queues and $N = 3$ service vectors. Again there is only one $v \geq 0$ boundary vector to consider. As long as the fairness direction θ is feasible there is only one MaxWeight matrix $\mathbf{\Delta}$ (to a scale factor) to achieve it. Feasibility depends on ρ.

$$S_1 = \begin{pmatrix} 5 \\ 0 \\ 0 \end{pmatrix}, S_2 = \begin{pmatrix} 0 \\ 5 \\ 0 \end{pmatrix}, S_3 = \begin{pmatrix} 0 \\ 0 \\ 5 \end{pmatrix} \tag{5.5}$$

The desired fairness criterion is

$$\theta = \begin{pmatrix} 1/2 \\ 1/3 \\ 1/6 \end{pmatrix} \tag{5.6}$$

With some algebra, it is easy to see that as long as the load vector $\rho \notin \mathcal{P}$ is such that the fairness direction is feasible, then the necessary MaxWeight matrix, $\mathbf{\Delta}$, to achieve it is as follows:

$$\mathbf{\Delta} = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 6 \end{pmatrix} \tag{5.7}$$

The system load oscillates between being stable and unstable. Hence, there are *temporary periods of overload*.

$$\rho_{\text{stable}} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \rho_{\text{unstable}} = \begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix} \tag{5.8}$$

The system spends 500 time slots in the stable mode–$\rho = \rho_{\text{stable}}$–then switches to spend 500 time slots in the unstable mode–$\rho = \rho_{\text{unstable}}$. Arrivals to queue q in each time slot are uniformly distributed between $[0, 2\rho_q]$. When the system is in the stable mode, MaxWeight Scheduling should stablize the workload. When it is in the unstable mode, MaxWeight Scheduling should achieve the desired fairness criterion.

Figure 6 plots the scaled workload, $\frac{X(t)}{t}$, under this unstable/stable system. We can see that for the first unstable period ($t \in [0, 500]$), the fairness criterion is quickly achieved. However, on the next unstable period, the scaled backlogs ($X_i(t)/t$) do not appear to stabilize within the 500 epoch period. This is because the workload has been stablized during the stable period and we are scaling by the *total* time, not just the time starting from when we enter the period of instability. Hence, it may actually take a very long time before the scaled backlogs converge. One the other hand, the fairness criteria is quickly achieved. The third subfigure shows how the backlogs grow relative to each other: $\frac{X_i(t)}{\sum_j X_j(t)}$. We can see in this figure, the fairness criterion is achieved very quickly (plotted in red) during unstable periods. During stable periods this relative backlog is not very informative since all the backlogs will grow to zero.

While our fairness results for controlling the backlog were asymptotic, these numerical results suggest that the fairness criterion can be achieved very rapidly.

6. Discussion and Conclusion

In many real world systems, traffic load is unpredictable and often bursty in nature. In any finite window of time, the system may enter a period of temporary instability where the rate of incoming jobs is larger than the rate at which jobs can be serviced. During such periods of stress, it is natural to want to allocate limited service resources across various jobs classes in a fair manner.

In this work, we consider how to fairly serve under-provisioned queues. We focus on MaxWeight Scheduling policies because they are simple to implement and behave well during stable periods. In particular, MaxWeight policies guarantee finite backlogs when the system load is within the stability region. We find that, whenever the system is overloaded, the backlog approaches a straight line as the time window

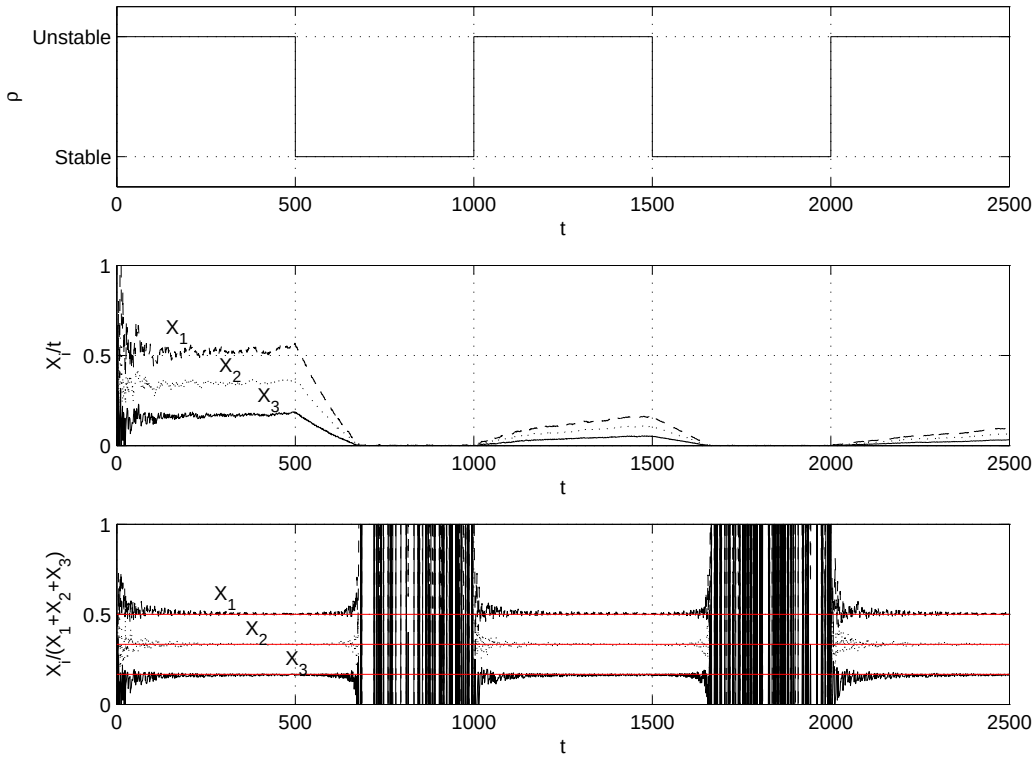


Figure 6 Dynamics of 3-queue queueing system oscillating between stable and unstable modes.

during which the system is overloaded increases. This straight line can be characterized as a fixed point, or equivalently, as the solution to a simple convex program. As such, it is straightforward to identify this line, as a function of the system parameters, the load vector ρ, and the MaxWeight matrix $\mathbf{\Delta}$.

Considering a fair allocation of resources in overloaded systems to be such that the backlog grows along a certain *direction* θ, we show how to choose the MaxWeight matrix $\mathbf{\Delta}$ to guarantee that the backlog will indeed grow along this direction, as long as this direction is feasible. Additionally, MaxWeight is shown to asymptotically minimize the backlog, among all other policies that achieve the same fairness criterion. As it turns out the choice of $\mathbf{\Delta}$ is robust with respect to the load vector ρ in the sense that for every direction θ, there exists a *partition* of the instability region into subsets, such that the same matrix $\mathbf{\Delta}$ will work for all load vectors in the same subset. In particular, it is sufficient to know what subset ρ belongs to, and it is unnecessary to know the exact value of ρ. Via numerical simulations we see that MaxWeight scheduling policy performs as expected; it achieves the fairness criterion during periods of overload and it stabilizes the queues otherwise.

Our analysis relies on geometrical arguments that are applied directly to individual backlog traces. In particular, no probabilistic assumptions are made. The only necessary assumption is that the arriving work-

load has a well defined long run average. This approach is also helpful in developing intuition with respect to the system dynamics.

This work can be extended in various directions. As demonstrated in Section 4.1.2, there exist some fairness criteria that are infeasible via MaxWeight Matching. So, one might consider whether other policies, such as Projective Cone Scheduling from Ross and Bambos (2009), would expand the set of feasible fairness criteria. In a similar vein, one might want to obtain other fairness criteria for unstable systems. Third, it may be possible to extend this work to networks of parallel queueing systems, by relying on results from Shah and Wischik (2011). Finally, while we have established convergence of the backlog vector under vary mild traffic conditions, if more restrictive assumptions are made (such as Markovian queues) one might be able to obtain results on the rate of convergence as well.

In this work we take a different view to traditional queueing. Namely, we focus on the *instability* region. While it is certainly desirable to operate systems within the stability region, there are many real world scenarios where this may not be possible. Input traffic may surge due to unplanned circumstances. Service resources may be reduced due to unavoidable accidents or catastrophes. During these periods of temporary instability it is often necessary to allocate limited resources in a fair manner. Once the system exits the window of stress, it will be stabilizable and the natural goals of throughput maximization and cost minimization can be restored.

References

Armony, M., N. Bambos. 2003. Queueing dynamics and maximal throughput scheduling in switched processing systems. *Queueing Systems* **44** 209–252.

Bonald, T., L. Massoulie, A. Proutiere, J. Virtamo. 2006. A queueing analysis of max-min fairness, proportional fairness and balanced fairness. *Queueing Systems* **53** 65–84.

Boyd, S., L. Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press.

Eryilmaz, A., R. Srikant. 2005. Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control. *IEEE Infocom*.

Eryilmaz, A., R. Srikant. 2006. Joint congestion control, routing, and mac for stability and fairness in wireless networks. *IEEE Journal on Selected Areas in Communications* **24** 1514–1524.

Georgiadis, Leonidas, Leandros Tassiulas. 2006. Optimal overload response in sensor networks. *IEEE Transactions on Information Theory* **52** 2684–2696.

Kelly, F. P., A.K. Maulloo, D.K.H. Tan. 1998. Rate control in communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society* **49** 237–252.

Kelly, F. P., R. J. Williams. 2004. Fluid model for a network operating under a fair bandwidth-sharing policy. *Annals of Applied Probability* **14** 1055–1083.

Loynes, R.M. 1963. The stability of a queue with non-independent interarrival and service times. *Proceedings of the Cambridge Philisophical Society* **58** 497–530.

Mandelbaum, A., S. Stolyar. 2004. Scheduling flexible servers with convex delay costs: Heavy-traffic optimality of the generalized cμ-rule. *Operations Research* **52**(6) 836–855.

Massoulie, L. 2007. Structural properties of proportional fairness: Stability and insensitivity. *Annals of Applied Probability* **17** 809–839.

Mazumdar, R., L. Mason, C. Douligeris. 1991. Fairness in network optimal flow control: optimality of product forms. *IEEE Transactions on Communications* **39** 775–782.

McKeown, N., A. Mekkittikul, V. Anantharam, J. Walrand. 1999. Achieving 100% throughput in an input-queued switch. *IEEE Transactions on Communications* **47** 1260–1267.

Mo, J., J. Walrand. 2000. Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking* **8** 556–567.

Neely, M. J. 2006. Super-fast delay tradeoffs for utility optimal fair scheduling in wireless networks. *IEEE Journal on Selected Areas in Communications* **24** 1489–1501.

Neely, M. J., E. Modiano, C. Li. 2005. Fairness and optimal stochastic control for heterogeneous networks. *IEEE Infocom*.

Perry, O., W. Whitt. 2009. Responding to unexpected overloads in large-scale service systems. *Management Science* **55** 1353–1367.

Perry, O., W. Whitt. 2011. A fluid approximation for service systems responding to unexpected overloads. *Operations Research* **to appear**.

Ross, K., N. Bambos. 2009. Projective cone scheduling (PCS) algorithms for packet switches of maximal throughput. *IEEE/ACM Transactions on Networking* **17**(3) 976–989.

Shah, D., D. Wischik. 2011. Fluid models of switched networks in overload Working paper.

Stolyar, S. 2004. Maxweight scheduling in a generalized switch: State space collapse and workload minimization in heavy traffic. *The Annals of Applied Probability* **14**(1) 1–53.

Tassiulas, L. 1995. Adaptive back-pressure congestion control based on local information. *IEEE Transactions on Automatic Control* **40** 236–250.

Tassiulas, L., P.P. Bhattacharya. 2000. Allocation of interdependent resources for maximal throughput. *Stochastic Models* **16** 27 – 48.

Tassiulas, L., A. Ephremides. 1992. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control* **37** 1936–1948.

Appendix

PROOF OF LEMMA 3.2:

We write (using similar arguments like in equations of A.22 to A.27 of Ross and Bambos (2009)),

$$\begin{aligned}
\langle X(t_c) - X(t'_c), \mathbf{\Delta}\eta \rangle &= \left\langle \sum_{t=t'_c}^{t_c-1} A(t), \mathbf{\Delta}\eta \right\rangle - \left\langle \sum_{t=t'_c}^{t_c-1} D(t), \mathbf{\Delta}\eta \right\rangle \\
&= \left\langle \sum_{t=t'_c}^{t_c-1} A(t), \mathbf{\Delta}\eta \right\rangle - \sum_{t=t'_c}^{t_c-1} \langle D(t), \mathbf{\Delta}\eta \rangle \\
&= \left\langle \sum_{t=t'_c}^{t_c-1} A(t), \mathbf{\Delta}\eta \right\rangle - \sum_{t=t'_c}^{t_c-1} \left[\sum_{q:\eta_q>0} D_q(t)\mathbf{\Delta}_{qq}\eta_q + \sum_{q:\eta_q=0} D_q(t)\mathbf{\Delta}_{qq}\eta_q \right] \\
&= \left\langle \sum_{t=t'_c}^{t_c-1} A(t), \mathbf{\Delta}\eta \right\rangle - \sum_{t=t'_c}^{t_c-1} \left[\sum_{q:\eta_q>0} S_q(t)\mathbf{\Delta}_{qq}\eta_q + \sum_{q:\eta_q=0} S_q(t)\, 0 \right] \\
&= \left\langle \sum_{t=t'_c}^{t_c-1} A(t), \mathbf{\Delta}\eta \right\rangle - \sum_{t=t'_c}^{t_c-1} \langle S(t), \mathbf{\Delta}\eta \rangle \\
&= \left\langle \sum_{t=t'_c}^{t_c-1} A(t), \mathbf{\Delta}\eta \right\rangle - \max_{S\in\mathcal{S}} \langle S(t), \mathbf{\Delta}\eta \rangle \, (t_c - t'_c), \qquad \text{(A1)}
\end{aligned}$$

To see the above steps, recall the following. First, $X(t) \in \mathcal{V}(\eta)$ for every $t \in (t'_c, t_c]$ and any c, by assumption. Therefore, $X(t) \in \mathcal{K}(\eta)$, hence, from (3.15) we get $D_q(t) = S_q(t)$ for $q \in \mathcal{Q}$ with $\eta_q > 0$, for every $t \in (t'_c, t_c]$ and any c. Moreover, $X(t) \in \mathcal{C}(\eta)$, hence, from (3.13) we get $\langle S(t), \mathbf{\Delta}\eta \rangle = \max_{S\in\mathcal{S}} \langle S, \mathbf{\Delta}\eta \rangle$, for all $t \in (t'_c, t_c]$ and any c. ■

PROOF OF LEMMA 3.3:

Note that $t'_n < t_n$ eventually (for any large n), expand the terms as follows:

$$\frac{\sum_{t=t'_n}^{t_n-1} A(t)}{t_n - t'_n} = \frac{\sum_{t=0}^{t_n-1} A(t)}{t_n - t'_n} - \frac{\sum_{t=0}^{t'_n-1} A(t)}{t_n - t'_n} = \frac{\sum_{t=0}^{t_n-1} A(t)}{t_n} \frac{t_n}{t_n - t'_n} - \frac{\sum_{t=0}^{t_n-1} A(t)}{t'_n} \frac{t'_n}{t_n - t'_n}, \qquad \text{(A2)}$$

and observe that letting $n \to \infty$ we get

$$\lim_{n\to\infty} \frac{\sum_{t=t'_n}^{t_n-1} A(t)}{t_n - t'_n} = \rho \, \frac{1}{\chi} - \rho \, (\frac{1}{\chi} - 1) = \rho, \qquad \text{(A3)}$$

since $\lim_{n\to\infty} \frac{\sum_{t=0}^{T} A(t)}{T} = \rho$. This completes the proof of the lemma. ■

PROOF OF LEMMA 3.4: We write

$$X(t_m) = \sum_{t=0}^{t_m-1} A(t) - \sum_{t=0}^{t_m-1} D(t) \qquad \text{(A4)}$$

and observe that $D_q(t) = \min\{S_q(t), X_q(t)\} \le S_q(t)$ for every $q \in \mathcal{Q}$, hence, $-D_q(t) \ge -S_q(t)$. Therefore, since $\mathbf{\Delta}$ is diagonal (with positive elements), we have $-\langle D(t), \mathbf{\Delta}\eta \rangle \ge \langle S(t), \mathbf{\Delta}\eta \rangle \ge -\max_{S\in\mathcal{S}} \langle S, \mathbf{\Delta}\eta \rangle$. Projecting on $\mathbf{\Delta}\eta$ we get

$$\langle X(t_m), \mathbf{\Delta}\eta \rangle \ge \left\langle \sum_{t=0}^{t_m-1} A(t), \mathbf{\Delta}\eta \right\rangle - \max_{S\in\mathcal{S}} \langle S, \mathbf{\Delta}\eta \rangle \, (t_m) \qquad \text{(A5)}$$

Dividing by t_m and letting $m \to \infty$, we get

$$\langle \mu, \mathbf{\Delta}\eta \rangle \geq \langle \rho, \mathbf{\Delta}\eta \rangle - \max_{S \in \mathcal{S}} \langle S, \mathbf{\Delta}\eta \rangle > 0 \tag{A6}$$

This completes the proof of the lemma. ■

PROOF OF LEMMA 3.5: Indeed (recalling that $\mathbf{\Delta}$ is positive-definite), we have

$$\begin{aligned} 0 \leq \langle \mu - \eta, \mathbf{\Delta}(\mu - \eta) \rangle &= \langle \mu, \mathbf{\Delta}\mu \rangle - 2 \langle \mu, \mathbf{\Delta}\eta \rangle + \langle \eta, \mathbf{\Delta}\eta \rangle \\ &\leq \langle \mu, \mathbf{\Delta}\mu \rangle - 2 \langle \eta, \mathbf{\Delta}\eta \rangle + \langle \eta, \mathbf{\Delta}\eta \rangle = \langle \mu, \mathbf{\Delta}\mu \rangle - \langle \eta, \mathbf{\Delta}\eta \rangle , \end{aligned} \tag{A7}$$

so $\langle \mu, \mathbf{\Delta}\mu \rangle \geq \langle \eta, \mathbf{\Delta}\eta \rangle$. But since $\langle \eta, \mathbf{\Delta}\eta \rangle = \limsup t \to \infty \left\langle \frac{X(t)}{t}, \mathbf{\Delta}\frac{X(t)}{t} \right\rangle$, we must have $\langle \mu, \mathbf{\Delta}\mu \rangle = \langle \eta, \mathbf{\Delta}\eta \rangle$, therefore, $\langle \mu - \eta, \mathbf{\Delta}(\mu - \eta) \rangle = 0$, which implies $\mu = \eta$. This completes the proof of the lemma. ■

PROOF OF LEMMA 3.6: Rewrite the inequality as $-\left[\langle \rho, \mathbf{\Delta}\eta \rangle - \max_{S \in \mathcal{S}} \langle S, \mathbf{\Delta}\eta \rangle\right] \epsilon + \langle \eta, \mathbf{\Delta}\eta \rangle \geq (1 - \epsilon) \langle \eta, \mathbf{\Delta}\eta \rangle$, since $1 - \epsilon > 0$. This is equivalent (since $\epsilon > 0$) to

$$\langle \eta, \mathbf{\Delta}\eta \rangle \geq \langle \rho, \mathbf{\Delta}\eta \rangle - \max_{S \in \mathcal{S}} \langle S, \mathbf{\Delta}\eta \rangle \tag{A8}$$

But this is true by Lemma 3.4 applied to the sequence $\{t_c\}$ with $\lim_{c \to \infty} \frac{X(t_c)}{t_c} = \eta$. This complete the proof of the lemma. ■

PROOF OF LEMMA 3.7: We have 2 cases to show since, by definition of $\{s_c\}$ we have that $0 \leq \frac{t_c - s_c}{t_c} \leq 1$. The first case is A) $\epsilon > 0$ and the second is B) $\epsilon < 1$.

A) We first show that $\epsilon > 0$. We start by showing that there is no increasing unbounded subsequence $\{t_b\}$ of $\{t_c\}$ such that $\lim_{b \to \infty} \frac{t_b - s_b}{t_b} = 0$, where $s_b = \max\{t_a < t_b\}$. Note that this also implies that $\lim_{b \to \infty} \frac{s_b}{t_b} = 1$. Arguing by contradiction, suppose it exists. Observe that for every $q \in \mathcal{Q}$ we have

$$-\bar{S}_q(t_b - s_b) \leq X_q(t_b) - X_q(s_b) \leq \bar{A}_q(t_b - s_b), \tag{A9}$$

where $\bar{A}_q < \infty$ is the maximum workload that can arrive in queue q in any time slot (see model in Ross and Bambos (2009) for assumption of boundedness) and $\bar{S}_q = \max_{S \in \mathcal{S}}\{S_q\} < \infty$ is the maximum workload that can be removed from queue q in any time slot. Dividing by t_b, letting $b \to \infty$, we get

$$\lim_{b \to \infty} \frac{X(t_b) - X(s_b)}{t_b} = 0 = \lim_{b \to \infty} \left[\frac{X(t_b)}{t_b} - \frac{X(s_b)}{s_b} \frac{s_b}{t_b} \right] = \eta - \psi \tag{A10}$$

which implies that $\eta = \psi$ and establishes the desired contradiction.

B) We still need to show that $\epsilon \neq 1$. Arguing by contradiction, suppose there exists a subsequence $\{t_i\}$ of $\{t_c\}$ (and corresponding subsequence $\{s_i\}$ of $\{s_c\}$) such that $\lim_{i \to \infty} \frac{t_i - s_i}{t_i} = 1$, hence, $\lim_{i \to \infty} \frac{s_i}{t_i} = 0$. Applying Lemmas 3.2 and 3.3 with $\{t'_i\} = \{s_i\}$

$$\lim_{i \to \infty} \left\langle \frac{X(t_i) - X(s_i)}{t_i - s_i}, \mathbf{\Delta}\eta \right\rangle = \langle \rho, \mathbf{\Delta}\eta \rangle - \max_{S \in \mathcal{S}} \langle S(t), \mathbf{\Delta}\eta \rangle \tag{A11}$$

It follows that

$$\begin{aligned}
\langle \eta, \mathbf{\Delta}\eta \rangle &= \lim_{i\to\infty} \left\langle \frac{X(t_i)}{t_i}, \mathbf{\Delta}\eta \right\rangle \\
&= \lim_{i\to\infty} \left\langle \frac{X(t_i)-X(s_i)}{t_i-s_i}\ \frac{t_i-s_i}{t_i} + \frac{X(s_i)}{s_i}\ \frac{s_i}{t_i}, \mathbf{\Delta}\eta \right\rangle \\
&= \lim_{i\to\infty} \left\langle \frac{X(t_i)-X(s_i)}{t_i-s_i}, \mathbf{\Delta}\eta \right\rangle \ \frac{t_i-s_i}{t_i} + \left\langle \frac{X(s_i)}{s_i}, \mathbf{\Delta}\eta \right\rangle \ \frac{s_i}{t_i} \\
&= \left[\langle \rho, \mathbf{\Delta}\eta \rangle - \max_{S\in\mathcal{S}} \langle S(t), \mathbf{\Delta}\eta \rangle \right] \cdot 1 + \langle \psi, \mathbf{\Delta}\eta \rangle \cdot 0 \\
&= \langle \rho, \mathbf{\Delta}\eta \rangle - \max_{S\in\mathcal{S}} \langle S(t), \mathbf{\Delta}\eta \rangle
\end{aligned} \tag{A12}$$

Now applying Lemma 3.4 on the subsequence $\{s_i\}$ with $\lim_{s_i\to\infty} \frac{X(s_i)}{s_i} = \psi$ we get

$$\langle \psi, \Delta\eta \rangle \geq \langle \rho, \mathbf{\Delta}\eta \rangle - \max_{S\in\mathcal{S}} \langle S, \mathbf{\Delta}\eta \rangle = \langle \eta, \mathbf{\Delta}\eta \rangle , \tag{A13}$$

using (A12). Hence, $\langle \psi, \Delta\eta \rangle \geq \langle \eta, \mathbf{\Delta}\eta \rangle$, which implies $\psi = \eta$ by Lemma 3.5. But this is impossible since by definition of subsequence $\{s_c\}$, $\psi \neq \eta$, which completes the proof of the lemma. ■

PROOF OF LEMMA 3.8: Consider a subsequence $\{t_n\}$ such that for each m:

$$\alpha_m = \lim_{n\to\infty} \frac{\sum_{t=0}^{t_n-1} \mathbf{1}_{\{S(t)=S_m\}}}{t_n} \tag{A14}$$

Note that by definition: $\alpha_m \in [0,1]$ and $\sum_m \alpha_m \leq 1$. Further, because $\rho \notin \mathcal{P}$, there exist q and $T < \infty$, such that $X_q(t) > 0$ for all $t > T$; hence, MaxWeight Scheduling policies will never idle for $t > T$ and $\sum_m \alpha_m = 1$.

We have for q such that $\eta_q > 0$:

$$\begin{aligned}
\eta_q &= \lim_{n\to\infty} \frac{X_q(t_n)}{t_n} \\
&= \lim_{n\to\infty} \frac{\sum_{t=0}^{t_n-1} \left[A_q(t) - D_q(t) \right]}{t_n} \\
&= \lim_{n\to\infty} \frac{X_q(t_o) + \sum_{t=t_o}^{t_n-1} \left[A_q(t) - \sum_m S_{m,q} \mathbf{1}_{\{S(t)=S_m\}} \right]}{t_n} \\
&= \rho_q - \sum_m \alpha_m S_{m,q}
\end{aligned} \tag{A15}$$

Where $t_o < \infty$ such that for all $t > t_o$, $X(t) \in \mathcal{V}(\eta)$. It's existence is given by Proposition 3.1. For q such that $\eta_q = 0$, we have:

$$\begin{aligned}
0 = \eta_q &= \lim_{n\to\infty} \frac{X_q(t_n)}{t_n} \\
&= \lim_{n\to\infty} \frac{\sum_{t=0}^{t_n-1} \left[A_q(t) - D_q(t) \right]}{t_n} \\
&= \lim_{n\to\infty} \frac{X_q(t_o) + \sum_{t=t_o}^{t_n-1} \left[A_q(t) - \sum_m \min\{X_q(t), S_{m,q}\} \mathbf{1}_{\{S(t)=S_m\}} \right]}{t_n}
\end{aligned}$$

$$\geq \lim_{n\to\infty} \frac{X_q(t_o) + \sum_{t=t_o}^{t_n-1} \left[A_q(t) - \sum_m S_{m,q}\mathbf{1}_{\{S(t)=S_m\}}\right]}{t_n}$$
$$= \rho - \sum_m \alpha_m S_{m,q} \tag{A16}$$

Which means that $\rho_q - \sum_m \alpha_m S_{m,q} \leq 0$ and

$$0 = \eta_q = \left[\rho_q - \sum_m \alpha_m S_{m,q}\right]^+, \tag{A17}$$

which gives us that $\eta = \left[\rho - \sum_m \alpha_m S_m\right]^+$.

Finally, we have to show that if $\alpha_m > 0$, then $\eta \in C_{S_m}$. We have seen that α_m is the proportion of time that service vector S_m is used under MaxWeight Scheduling once $X(t) \in \mathcal{V}(\eta)$ for all $t > t_o$. By Proposition 3.1, we know that t_o exists. By contradiction, suppose that $\eta \notin C_{S_m}$. This implies that there exists $m' \neq m$ such that $\langle \eta, \mathbf{\Delta} S_{m'}\rangle > \langle \eta, \mathbf{\Delta} S_m\rangle$. Since $\alpha_m > 0$, we must use S_m for some $t > t_o$. This contradicts the definition of $\mathcal{V}(\eta)$, which by (3.13) says that MaxWeight Scheduling would use $S_{m'}$ rather than S_m which would imply that $\alpha_m = 0$. Hence, if $\alpha_m > 0$, $\eta \in C_{S_m}$. ■

PROOF OF LEMMA 3.9: This follows from Lemma 3.8. Replacing $\eta = [\rho - \sum_{m=1}^N \alpha_m S_m]^+$ we have

$$\begin{aligned}
\langle \eta, \mathbf{\Delta}\eta\rangle &= \left\langle \left[\rho - \sum_{m=1}^N \alpha_m S_m\right]^+, \mathbf{\Delta}\eta \right\rangle \\
&= \sum_{q:\eta_q>0} \left[\rho - \sum_{m=1}^N \alpha_m S_m\right]_q \mathbf{\Delta}_{qq}\eta_q + \sum_{q:\eta_q=0} \left[\rho - \sum_{m=1}^N \alpha_m S_m\right]_q^+ \mathbf{\Delta}_{qq}\eta_q \\
&= \sum_{q:\eta_q>0} \left[\rho - \sum_{m=1}^N \alpha_m S_m\right]_q \mathbf{\Delta}_{qq}\eta_q + \sum_{q:\eta_q=0} \left[\rho - \sum_{m=1}^N \alpha_m S_m\right]_q \mathbf{\Delta}_{qq} 0 \\
&= \left\langle \rho - \sum_{m=1}^N \alpha_m S_m, \mathbf{\Delta}\eta \right\rangle \\
&= \langle \rho, \mathbf{\Delta}\eta\rangle - \sum_{m=1}^N \langle \alpha_m S_m, \mathbf{\Delta}\eta\rangle \\
&= \langle \rho, \mathbf{\Delta}\eta\rangle - \max_{S\in\mathcal{S}} \langle S, \mathbf{\Delta}\eta\rangle
\end{aligned} \tag{A18}$$

where the last equality follows from the fact that η is a fixed point. ■

PROOF OF LEMMA 3.10: This result follows from the KKT conditions for optimality of the convex program (3.9). Our goal is to show that if $\eta = (\rho - \sum_m \alpha_m S_m)^+$ is such that for all m with $\alpha_m > 0$, we have that $\langle \eta, \mathbf{\Delta} S_m\rangle \geq \langle \eta, \mathbf{\Delta} S_k\rangle$, then it is a solution to the convex program (3.9). The KKT conditions are necessary and sufficient for optimality if the objective function is differentiable and Slater's constraint is satisfied Boyd and Vandenberghe (2004). Our objective function is clearly differentiable.

Slater's condition says that there exists r, α such that $r < \sum_m \alpha_m S_m$, $r < \rho$, $\alpha_m > 0$ and $\sum_m \alpha_m = 1$. It is clear that $r = 0, \alpha_m = 1/M$ satisfies this condition. Since the KKT conditions are necessary and sufficient

in this case and there is only one solution, it will follow that there is exactly one fixed point if all fixed points satisfy the KKT conditions.

To examine the KKT conditions, we first rewrite the convex program in (3.9) as an equivalent convex program:

$$\begin{aligned} \min_{r,\alpha} \quad & \langle \rho - r, \mathbf{\Delta}(\rho - r) \rangle \\ s.t. \quad & r \leq \sum_m \alpha_m S_m \\ & r \leq \rho \\ & \alpha_m \geq 0, \forall m \\ & \sum_m \alpha_m = 1 \end{aligned} \tag{A19}$$

Let r^*, α_m^* (and correspondingly $\eta^* = \rho - r^* = [\rho - \sum_m \alpha_m^* S_m]^+$) be the solution to the preceding convex program. The KKT conditions for optimality say that all the constraints must be satisfied and:

$$\begin{aligned} \nabla \langle \rho - r, \mathbf{\Delta}(\rho - r) \rangle + \nabla \lambda'(r - \sum_m \alpha_m S_m) + \nabla \lambda''(r - \rho) - \nabla \lambda \alpha + \nabla v(\sum_m \alpha_m - 1) &= 0 \\ \lambda'_q(r_q - \sum_m \alpha_m (S_m)_q) &= 0 \\ \lambda''_q(r_q - \rho_q) &= 0 \\ \lambda_m \alpha_m &= 0 \\ \lambda, \lambda', \lambda'', v &\geq 0 \end{aligned} \tag{A20}$$

We look at the first condition:

$$\begin{aligned} \nabla_{r_q} : \; & 2\mathbf{\Delta}_{qq}(\rho - r^*)_q = \lambda'_q + \lambda''_q, \forall q \\ \nabla_{\alpha_m} : \; & \sum_q \lambda'_q (S_m)_q = v - \lambda_m \\ & \sum_q (2(\rho - r^*)_q \mathbf{\Delta}_{qq} - \lambda''_q)(S_m)_q = v - \lambda_m \\ & 2\langle \rho - r^*, \mathbf{\Delta} S_m \rangle = v + \sum_q \lambda''_q - \lambda_m \end{aligned} \tag{A21}$$

Now we show that for any fixed point, there exists $\lambda, \lambda', \lambda'', v$ which satisfy the KKT condition (A20). To do this, suppose we are given a fixed point $\eta = \rho - r = (\rho - \sum_m \alpha_m)^+$ with

$$\alpha_m > 0 \implies \langle \rho - r, \mathbf{\Delta} S_m \rangle \geq \langle \rho - r, \mathbf{\Delta} S_k \rangle \tag{A22}$$

We will construct non-negative Lagrange multiplies to satisfy the KKT conditions.

Consider q such that $\rho_q > r_q$. In order to satisfy the third constraint in (A20), $\lambda''_q = 0$. Now if $\rho_q \leq r_q$, in order to satisfy the first constraint in (A21) we have that $\lambda'_q + \lambda''_q = 0$. To ensure that the Lagrange multipliers are non-negative, we must have that $\lambda''_q = 0$ for all q. Subsequently:

$$\lambda'_q = 2\mathbf{\Delta}_{qq}(\rho - r)_q, \forall q \tag{A23}$$

Consider $\alpha_m > 0$. To satisfy the fourth constraint in (A20), $\lambda_m = 0$. Now to satisfy the second constraint in (A21):

$$0 \leq 2\left\langle \rho - r, \mathbf{\Delta} S_m \right\rangle = v, \forall m \text{ such that } \alpha_m > 0 \tag{A24}$$

which also satisfies the non-negativity of v. Consider $\alpha_m = 0$ and $\alpha_k > 0$. By the assumption that $\rho - r$ is a fixed point:

$$\begin{aligned} \left\langle \rho - r, \mathbf{\Delta} S_m \right\rangle &\leq \left\langle \rho - r, \mathbf{\Delta} S_k \right\rangle \\ &\Longrightarrow \frac{v - \lambda_m}{2} \leq \frac{v}{2} \\ &\Longrightarrow \lambda_m \geq 0 \end{aligned} \tag{A25}$$

Hence, we can satisfy the KKT conditions in (A20) with non-negative $\lambda, \lambda', \lambda'', v$. Since any fixed point satisfies the necessary and sufficient KKT conditions, all fixed points are solutions to the convex program. There is only one solution and so there is only one fixed point. ■

PROOF OF LEMMA 4.1: We show this via construction. Recall that $\hat{\mathbf{\Delta}}$ is diagonal: $\left\langle v, \hat{\mathbf{\Delta}} S \right\rangle = \sum_q v_q \hat{\mathbf{\Delta}}_{qq} S_q$. For $v_q > 0$

$$\mathbf{\Delta}_{qq} = \frac{\hat{\mathbf{\Delta}}_{qq} v_q}{\eta_q} \geq 0 \tag{A26}$$

where the positivity comes from the fact that each element is positive. If $v_q = 0$,

$$\mathbf{\Delta}_{qq} = 1 \tag{A27}$$

or some other arbitrary positive value.

Now, for any i_j ($j \in [1, m]$) and k the following holds:

$$\begin{aligned} \left\langle v, \hat{\mathbf{\Delta}} S_{i_j} \right\rangle &\geq \left\langle v, \hat{\mathbf{\Delta}} S_k \right\rangle \\ &\Rightarrow \sum_q \hat{\mathbf{\Delta}}_{qq} v_q (S_{i_j})_q \geq \sum_q \hat{\mathbf{\Delta}}_{qq} v_q (S_k)_q \\ &\Rightarrow \sum_q v_q \hat{\mathbf{\Delta}}_{qq} (S_{i_j})_q - \sum_q \eta_q \mathbf{\Delta}_{qq} (S_{i_j})_q - \sum_q \eta_q \mathbf{\Delta}_{qq} (S_k)_q \\ &\quad \geq \sum_q v_q \hat{\mathbf{\Delta}}_{qq} (S_k)_q - \sum_q \eta_q \mathbf{\Delta}_{qq} (S_{i_j})_q - \sum_q \eta_q \mathbf{\Delta}_{qq} (S_k)_q \\ &\Rightarrow \sum_q [(S_{i_j})_q (\hat{\mathbf{\Delta}}_{qq} v_q - \mathbf{\Delta}_{qq} \eta_q) - (S_k)_q \mathbf{\Delta}_{qq} \eta_q)] \\ &\quad \geq \sum_q [(S_k)_q (\hat{\mathbf{\Delta}}_{qq} v_q - \mathbf{\Delta}_{qq} \eta_q) - (S_{i_j})_q \mathbf{\Delta}_{qq} \eta_q)] \\ &\Rightarrow -\sum_q (S_k)_q \mathbf{\Delta}_{qq} \eta_q \geq -\sum_q (S_{i_j})_q \mathbf{\Delta}_{qq} \eta_q \\ &\Rightarrow \left\langle \eta, \mathbf{\Delta} S_{i_j} \right\rangle \geq \left\langle \eta, \mathbf{\Delta} S_k \right\rangle \end{aligned} \tag{A28}$$

■

PROOF OF PROPOSITION 4.1: Assume that Conditions 1 and 2 are satisfied. We show this implies there exists a $\mathbf{\Delta}$ such that $\lim_{t\to\infty} \frac{X(t)}{t} = \eta$. We first consider Condition 2. This says that $v \geq 0$ is the boundary of cones C_m where $\alpha_m > 0$ defined by $\hat{\mathbf{\Delta}} = \mathbf{I}$. By Lemma 4.1, we can construct a $\mathbf{\Delta}$ such that for all $\alpha_m > 0$:

$$\langle \eta, \mathbf{\Delta} S_m \rangle \geq \langle \eta, \mathbf{\Delta} S_k \rangle \tag{A29}$$

Now, in conjunction with Condition 1, we have that $\eta \in \Psi(\rho, \mathcal{S})$ is a fixed point. By Theorem 3.1, we have that

$$\lim_{t\to\infty} \frac{X(t)}{t} = \eta \tag{A30}$$

Now suppose there exists a $\mathbf{\Delta}$ such that $\lim_{t\to\infty} \frac{X(t)}{t} = \eta$. By Theorem 3.1, η is a (the only) fixed point and, hence, satisfies Condition 1. Now, we show that Condition 2 is satisfied by constructing the necessary $v \geq 0$. Similar to the construction of $\mathbf{\Delta}$ in the proof of Lemma 4.1 we can determine v–the boundary induced when $\hat{\mathbf{\Delta}} = \mathbf{I}$. That is:

$$v_q = \mathbf{\Delta}_{qq} \eta_q \tag{A31}$$

which equals 0 if and only if $\eta_q = 0$. This v_q satisfies Condition 2. ■

PROOF OF THEOREM 4.1: The proof is via contradiction. Since both MaxWeight Scheduling and this alternative algorithm achieve the fairness criterion, there exists some κ such that $\bar{\eta} = \kappa\eta$. Let's assume that η is *not* the smallest scaled workload that achieves the fairness criterion so that $\kappa < 1$.

Since

$$\lim_{t\to\infty} \frac{\bar{X}(t)}{t} = \kappa\eta \tag{A32}$$

for any $\epsilon > 0$, there exists T such that for all $t > T$:

$$\left| \kappa\eta_q - \frac{\bar{X}_q(t)}{t} \right| < \epsilon \tag{A33}$$

For any q such that $\eta_q > 0$, there exists some $T_q < \infty$ such that for all $t > T_q$, $\frac{\bar{X}_q(t)}{t} > 0$ and so after time T_q, $\bar{D}_q = \bar{S}_q(t)$, where $\bar{D}(t)$ and $\bar{S}(t)$ are the departures and service vector used in time slot t as defined in Section 2. Given $\epsilon > 0$, there exists T such that:

$$0 \leq \frac{\sum_{k=0}^{T-1} \bar{S}_q(k)}{T} - \frac{\sum_{k=0}^{T-1} \bar{D}_q(k)}{T} = \frac{\sum_{k=0}^{T_q-1} \bar{S}_q(k)}{T} - \frac{\sum_{k=0}^{T_q-1} \bar{D}_q(k)}{T} < \epsilon \tag{A34}$$

Define $\bar{\alpha}_m \geq 0$ be defined as:

$$\bar{\alpha}_m = \lim_{T\to\infty} \frac{\sum_{t=0}^{T-1} \mathbf{1}_{\{\bar{\mathbf{S}}(\mathbf{t})=\mathbf{S_m}\}}}{T} \tag{A35}$$

Note that $\sum_m \bar{\alpha}_m \leq 1$. $\bar{\alpha}_m$ is the proportion of time service vector S_m is used up to time t. If the limit in (A35) does not exists, we can take a sub-limit which is well-defined since $\bar{\alpha}_m \in [0, 1]$.

We have for each q such that $\eta_q > 0$:

$$\begin{aligned}
\left|\kappa\eta_q - (\rho - \sum_m \bar{\alpha}_m \bar{S}_m)_q^+\right| &\leq \left|\kappa\eta_q - \rho_q + (\sum_m \bar{\alpha}_m S_m)_q\right| \\
&= \left|\kappa\eta_q - \frac{\bar{X}_q(T)}{T} + \frac{\bar{X}_q(T)}{T} - \rho_q + \Big(\sum_m \bar{\alpha}_m S_m\Big)_q\right| \\
&\leq \left|\kappa\eta_q - \frac{\bar{X}_q(T)}{T}\right| + \left|\frac{\bar{X}_q(T)}{T} - \rho_q + \Big(\sum_m \bar{\alpha}_m S_m\Big)_q\right| \\
&< \epsilon + \left|\frac{\sum_{k=0}^{T-1} A_q(k)}{T} - \frac{\sum_{k=0}^{T-1} \bar{D}_q(k)}{T} - \rho_q + \Big(\sum_m \bar{\alpha}_m S_m\Big)_q\right| \\
&< \epsilon + \epsilon + \left|\frac{\sum_{k=0}^{T-1} \bar{D}_q(k)}{T} - \frac{\sum_{k=0}^{T-1} \bar{S}_q(k)}{T} + \frac{\sum_{k=0}^{T-1} \bar{S}_q(k)}{T} - \Big(\sum_m \bar{\alpha}_m S_m\Big)_q\right| \\
&< 2\epsilon + \left|\frac{\sum_{k=0}^{T-1} \bar{D}_q(k)}{T} - \frac{\sum_{k=0}^{T-1} \bar{S}_q(k)}{T}\right| + \left|\frac{\sum_{k=0}^{T-1} \bar{S}_q(k)}{T} - \Big(\sum_m \bar{\alpha}_m S_m\Big)_q\right| \\
&< 3\epsilon + \left|\frac{\sum_{k=0}^{T-1} \bar{S}_q(k)}{T} - \Big(\sum_m \bar{\alpha}_m S_m\Big)_q\right| \\
&< 4\epsilon \implies \kappa\eta_q = \rho_q - \sum_m \bar{\alpha}_m S_m
\end{aligned} \tag{A36}$$

The second inequality comes from the fact that $\kappa\eta$ is the limit of $\frac{\bar{X}(t)}{t}$. The third inequality comes from the definition of ρ as the long-term traffic load. The fourth inequality comes from (A35). The last equality comes from (A34).

Let $\psi = (\rho - \sum \bar{\alpha}_m S_m)^+$:

$$\begin{aligned}
\langle \kappa\eta, \mathbf{\Delta}\kappa\eta\rangle &= \langle \kappa\eta + \psi - \psi, \mathbf{\Delta}(\kappa\eta + \psi - \psi)\rangle \\
&= \langle \kappa\eta - \psi, \mathbf{\Delta}(\kappa\eta - \psi)\rangle + 2\langle \kappa\eta - \psi, \mathbf{\Delta}\psi\rangle + \langle \psi, \mathbf{\Delta}\psi\rangle \\
&\geq 2\langle \kappa\eta - \psi, \mathbf{\Delta}\psi\rangle + \langle \psi, \mathbf{\Delta}\psi\rangle \\
&\geq 2\langle \kappa\eta - \psi, \mathbf{\Delta}\psi\rangle + \langle \eta, \mathbf{\Delta}\eta\rangle \\
&= 2\sum_{q:\psi>0} (\kappa\eta_q - \psi_q)\mathbf{\Delta}_{qq}\psi_q + 2\sum_{q:\psi=0} (\kappa\eta_q - \psi_q)\mathbf{\Delta}_{qq}\psi_q + \langle \eta, \mathbf{\Delta}\eta\rangle \\
&= 2\sum_{q:\psi>0} 0\mathbf{\Delta}_{qq}\psi_q + 2\sum_{q:\psi=0} (\kappa\eta_q - \psi_q)\mathbf{\Delta}_{qq}0 + \langle \eta, \mathbf{\Delta}\eta\rangle \\
&= \langle \eta, \mathbf{\Delta}\eta\rangle \\
&\implies \langle \kappa\eta, \mathbf{\Delta}\kappa\eta\rangle > \langle \eta, \mathbf{\Delta}\eta\rangle \\
&\implies \kappa \geq 1
\end{aligned} \tag{A37}$$

where the second inequality comes from Proposition 3.2 and the fourth equality comes from(A36). We have a contradiction to the assumption that $\kappa < 1$ ■