

QUEUEING AND SCHEDULING IN RANDOM ENVIRONMENTS

Nicholas Bambos¹ and George Michailidis²

Abstract

We consider a processing system comprised of several parallel queues and a processor, which operates in a time-varying environment that fluctuates between various states or modes. The service rate at each queue depends on the processor bandwidth allocated to it, as well as the environment mode. Each queue is driven by a job traffic flow, which may also depend on the environment mode.

Dynamic processor scheduling policies are investigated for maximizing the system throughput, by adapting to queue backlogs and the environment mode. We show that allocating the processor bandwidth to the queues, so as to maximize the projection of the service rate vector to a linear function of the workload vector, can keep the system stable under the maximum possible traffic load.

The analysis of the system dynamics is first done under very general assumptions, addressing rate stability and flow conservation on individual traffic and environment evolution traces. The connection to stochastic stability is later discussed for stationary and ergodic traffic and environment processes. Various extensions to feed-forward networks of such nodes, the multi-processor case, etc. are also discussed. The approach advances the methodology of trace-based modelling of queueing structures.

Applications of the model include bandwidth allocation in wireless channels with fluctuating interference, allocation of switching bandwidth to traffic flows in communication networks with fluctuating congestion levels and various others.

Keywords: stability of queues, processing networks, dynamic scheduling, bandwidth allocation, computer networks, wireless networks.

¹bambos@stanford.edu; Department of Management Science & Engineering, and Department of Electrical Engineering, Stanford University, Stanford, CA 94305. Research supported in part by the National Science Foundation

²gmichail@umich.edu; Department of Statistics, The University of Michigan, Ann Arbor, MI. Research supported in part by the National Science Foundation.

1 Introduction - Processing Model and Summary of Results

Consider a processing system comprised of Q first-in-first-out (FIFO) infinite-capacity queues (buffers), indexed³ by $q \in \mathbf{Q} = \{1, 2, 3, \dots, Q\}$, and a single processor of some fixed total processing bandwidth (capacity), which without any loss of generality is taken to be 1 (or is scaled to 1, otherwise). Jobs arrive exogenously to individual buffers and are queued up until they are processed.

The system operates in a time-varying environment, which fluctuates between M distinct states or modes, indexed by $m \in \mathbf{M} = \{1, 2, \dots, M\}$. The state of the environment may affect the job traffic flow into each queue (arrival rate and mean size of incoming jobs), as well as the service rate that each queue receives per allocated processor bandwidth. Specifically, $r_m^q x^q$ is the service rate of queue q , when processing bandwidth x^q is allocated to it and the environment is in state m , subject to the feasibility constraint that $\sum_{q \in \mathbf{Q}} x^q = 1$. That is, the mode m determines the differential service rate r_m^q per unit of bandwidth (or service efficiency) realized at the queue.

How should the processor bandwidth be distributed to the various queues, given the system backlog and environment mode histories, so as to maximize the throughput? We focus on this question and study a family of processor schedules - called MaxProjection - which are shown to stabilize the system under the maximal possible traffic load.

The above canonical queueing/processing model finds some key *applications* in various communication technologies. In wireless data networking, for example, consider a tunable radio transmitter (at a base station) serving multiple receivers (mobiles) operating in orthogonal channels. The interference (environment mode) varies randomly in each channel and determines the effective transmission rate achievable in it. The dilemma of the transmitter (processor) is how to ‘divide its attention’ (allocate operational bandwidth) amongst the various traffic streams of packets/bits to be transmitted into their corresponding channels, given the current interference level per channel and packet backlog per traffic stream.

Another interesting application arises in high-speed communication networks, where there is competition for switching/forwarding bandwidth at each node (switch/router) amongst various traffic flows/sessions crossing it. Such traffic flows may encounter congestion in down-stream network nodes, where their packets may be dropped and have to be retransmitted, reducing the effective packet forwarding (service) rate on the flow. A high-level model of this scenario can be abstracted by considering the environment mode to be the congestion state of the network and its propensity to drop packet in the down-stream nodes of the flows (over an appropriate time scale). Packets of the various flows are queued up in the buffers of the node (switch) under consideration, where they compete over processing bandwidth. The issue is which flows to serve at each point in time, given their current packet backlogs at the switch queues and the congestion state of the network that the flows will encounter down-stream. Serving a flow that will see significant packet drop down-stream is not efficient, unless absolutely necessary because of high backlog of this flow at the node. We do not further elaborate on implementation issues, like up-stream signalling, network control time scales etc. focusing on the core model and its stability analysis.

³We employ the notation $\mathbb{Z}_+ = \{1, 2, 3, \dots\}$, $\mathbb{Z}_{0+} = \{0, 1, 2, 3, \dots\}$, $\mathbb{R} = (-\infty, +\infty)$, $\mathbb{R}_+ = (0, \infty)$, $\mathbb{R}_{0+} = [0, \infty)$.

In general, the model has applications in various processing situations, where the parallel queues and the processor can be considered as a small-scale *foreground* queueing structure of interest, which ‘floats’ in a random *background* environment capturing high operational complexities and random events of an overall highly-complex system. This is the case, for example, in the ‘caricature’ of a single network node (foreground structure) operating within a large-scale complex communication network (background environment). For such a perspective to be valid, it should be justifiable in the specific situation under consideration that variations of the background environment modulate the foreground structure, but the latter does not significantly affect the former, due to the massive scope and dynamic ‘degrees of freedom’ of the background system. Other applications of the model occur in the management of clusters of networked servers (server farm), manufacturing systems, and computing systems, where there is competition over resources in a time-varying environment.

Allocation of resources in randomly modulated environments has recently been studied in various forms within a Markovian context [17, 16, 10, 6, 19, 15, 14], as well as in a stationary ergodic one [3, 4] using sample-path analysis [8]. However, in studying the specific model discussed here, we employ a recently developed methodology [1] for modelling complex queueing structures and studying their stability on individual evolution traces, not associated with any particular probabilistic framework. The latter can be later imposed (as a more restrictive setup) to address more targeted distributional or statistical issues. The literature is further discussed in more specific contexts later in the paper.

The study of the system within the *trace-based modelling framework* reveals a rich geometric structure [5] associated with its dynamics and leads to the formulation and analysis of the throughput-maximizing MaxProjection processor schedule discussed below. The modelling framework is deployed in Section 1.1 and the main results and structure of the paper are presented in Section 1.2.

1.1 Model Structure and Assumptions

Let $e(t) \in \mathbf{M}$ be the environment mode at time t and $\mathcal{E} = \{e(t), t \in \mathbb{R}\}$ the **environment trace** of evolution. It is assumed that the percentage of time that the environment trace spends in each mode:

$$\pi_m = \lim_{t \rightarrow \infty} \frac{\int_0^t \mathbf{1}_{\{e(u)=m\}} du}{t} \quad (1.1)$$

is well defined (the limit⁴ exists) for every $m \in \mathbf{M}$. The symbol $\mathbf{1}_{\{\cdot\}}$ denotes the standard indicator function.

When the environment is in mode $m \in \mathbf{M}$, the front job in queue $q \in \mathbf{Q}$ receives service at rate $r_m^q x^q$, where $x^q \in [0, 1]$ is the (percentage of) processor bandwidth allocated to this queue and r_m^q is the service efficiency (speed) of the processor on this queue. Therefore, the *service rate vector* across all queues is $\mathbf{R}_m \vec{x}$, where $\mathbf{R}_m = \text{diag}\{r_m^1, r_m^2, \dots, r_m^q, \dots, r_m^Q\}$ is the diagonal $Q \times Q$ matrix having r_m^q as its q^{th} diagonal entry, and $\vec{x} = (x^1, x^2, \dots, x^q, \dots, x^Q)$ is the processor *bandwidth allocation vector* having x^q as its q^{th} element. As expected, $\sum_{q \in \mathbf{Q}} x^q = 1$, since the total bandwidth (scaled to 1) is allocated to the queues. In principle, a schedule may allocate positive bandwidth to an empty queue and have it stay idle; however,

⁴Note that relation (1.1) is automatically valid when the environment trace \mathcal{E} is a sample path of a random process which is stationary and ergodic with respect to time-shifts $\theta_z \{e(t), t \in \mathbb{R}\} = \{e(t - z), t \in \mathbb{R}\}, z \in \mathbb{R}$.

the particular schedule studied later assigns zero bandwidth to empty queues, in the presence of non-empty ones. We define the space of all possible bandwidth allocation vectors by:

$$\mathbf{X} = \{(x^1, x^2, \dots, x^q, \dots, x^Q) \in \mathbb{R}_{0+}^Q : \sum_{q \in \mathbf{Q}} x^q = 1, \text{ with } x^q \in [0, 1] \text{ for each } q \in \mathbf{Q}\}. \quad (1.2)$$

To conclude, when the environment mode is $m \in \mathbf{M}$, the service rate vector is $\mathbf{R}_m \vec{x}$, controlled by the bandwidth allocation vector $\vec{x} \in \mathbf{X}$.

Let $t_j^q \in \mathbb{R}$ be the arrival time of the j^{th} job to arrive to queue $q \in \mathbf{Q}$ and $\sigma_j^q \in \mathbb{R}_+$ its associated service (processing) time requirement. The latter implies that if this job were to be served at *constant* rate r , its service time would be σ_j^q/r . The **traffic trace** into queue $q \in \mathbf{Q}$ is $\mathcal{T}^q = \{(t_j^q, \sigma_j^q), j \in \mathbb{Z}\}$. Equivalently,

$$\mathcal{T}^q = \{s^q(t), t \in \mathbb{R}\}, \quad (1.3)$$

where $s^q(t)$ is the rate at which service requirement (workload) arrives to queue q at time t . Note⁵ that $s^q(t)$ is zero between job arrival times and has an infinite-jump or δ -function of magnitude σ_j^q at time t_j^q in the previous setup. It is assumed that between any two finite times, only a finite amount of work may arrive, and only a finite number of δ -jumps (job arrivals) may occur. More importantly, it is assumed that the traffic intensity of \mathcal{T}^q or mean rate $\rho^q = \lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t s^q(u) du$ is well defined⁶ and positive for each $q \in \mathbf{Q}$. Hence, the *traffic load vector* of the system is:

$$\vec{\rho} = (\rho^1, \rho^2, \dots, \rho^q, \dots, \rho^Q) = \lim_{t \rightarrow \infty} \left\{ \frac{\int_0^t \vec{s}(u) du}{t} \right\} > \vec{0}, \quad (1.4)$$

where⁷ $\vec{s}(t) = (s^1(t), s^2(t), \dots, s^q(t), \dots, s^Q(t))$ is the (instantaneous) traffic rate vector. It is interesting to note that $\vec{s}(t)$ may be modulated by the environment and actually be:

$$\vec{s}(t) = \sum_{m \in \mathbf{M}} \vec{s}_m(t) \mathbf{1}_{\{e(t)=m\}}, \quad (1.5)$$

where $\vec{s}_m(t)$ is the rate vector of a set of Q flows that are driving the queues when the environment is in state $m \in \mathbf{M}$. Indeed, there may actually be different sets of flows driving the queues in different environment modes in various applications.

Finally, we introduce the **control trace** $\mathcal{C} = \{\vec{x}(t), t \in \mathbb{R}_+\}$, where $\vec{x}(t)$ is the bandwidth allocation vector used at time t to distribute the processor bandwidth to the queues. Define now $W^q(t)$ to be the workload (total service requirement of all jobs) in queue $q \in \mathbf{Q}$ at time t , when the system operates under a chosen control trace $\mathcal{C} = \{\vec{x}(t), t \in \mathbb{R}_+\}$, given fixed traffic and environment traces \mathcal{T} and \mathcal{E} correspondingly. This is easily seen to evolve (for $z, t \in \mathbb{R}$ with $z < t$) according to the integral equation:

$$W^q(t) = W^q(z) + \int_z^t s^q(u) du - \sum_{m \in \mathbf{M}} \int_z^t \mathbf{1}_{\{e(u)=m\}} \mathbf{1}_{\{W^q(u) > 0\}} r_m^q x^q(u) du. \quad (1.6)$$

⁵We prefer the $\mathcal{T}^q = \{s^q(t), t \in \mathbb{R}\}$ formulation of the traffic trace here, because we aim to later expand the nature of $s^q(t)$ to include both δ -functions and finite non-negative values between them.

⁶Note that (1.4) is automatically well defined if each traffic trace \mathcal{T}^q is a sample path of a random process which is stationary and ergodic with respect to time-shifts $\theta_z \{s^q(t), t \in \mathbb{R}\} = \{s^q((t-z)), t \in \mathbb{R}\}, z \in \mathbb{R}$.

⁷We denote by $\vec{0}$ the vector having all its components equal to zero and interpret vector inequalities as holding component-wise.

Note that the $\mathbf{1}_{\{W^q(u)>0\}}$ term suppresses the service rate when the queue is empty. There is heavy entanglement between the above equations across various queues $q \in \mathbf{Q}$ occurring via the control $\vec{x}(u)$, which has to satisfy $\sum_{q \in \mathbf{Q}} x^q(u) = 1$ at all times. Being simpler to work with vector variables in what follows, we write the family of equations for various $q \in \mathbf{Q}$ in the concise vector form:

$$\vec{W}(t) = \vec{W}(z) + \int_z^t \vec{s}(u) du - \int_z^t \mathbf{I}\{\vec{W}(u)\} \mathbf{R}(u) \vec{x}(u) du, \quad (1.7)$$

where $\mathbf{R}(u) = \sum_{m \in \mathbf{M}} \mathbf{1}_{\{e(u)=m\}} \mathbf{R}_m$ is the service efficiency matrix imposed by the environment at time u , and $\mathbf{I}\{\vec{W}(u)\}$ is the $Q \times Q$ diagonal matrix with $\mathbf{1}_{\{W_u^q > 0\}}$ as its q^{th} entry. The above integral equation can correspondingly be written in a differential form:

$$\frac{d\vec{W}(t)}{dt} = \vec{s}(t) - \mathbf{I}\{\vec{W}(t)\} \mathbf{R}(t) \vec{x}(t). \quad (1.8)$$

Note that this is a non-linear (because of $\mathbf{I}\{\vec{W}\}$) differential equation with time-varying parameters. The fact that $\vec{s}(t)$ has δ -jumps makes its analysis quite challenging. By convention, we take $\vec{W}(t)$ and $\mathbf{R}(t)$ to be right-continuous and have left limits. Since $\vec{s}(u)$ has δ -jumps at job arrivals, one must be careful to define/interpret the integral $\int_z^t \vec{s}(u) du$ as being calculated in $(z, t]$ or $\int_{z^+}^{t^+} \vec{s}(u) du$. That is, if $\vec{s}(u)$ has a δ -jump at z , then this is not included in the integral value; but if it has one at t , then that is included.

1.2 The Maximal Rate Projection (MaxProjection) Schedule - Summary of Results

We aim to study the dynamics of the workload vector $\vec{W}(t) = (W^1(t), W^2(t), \dots, W^q(t), \dots, W^Q(t))$ at large times, for given traffic and environment traces, and chosen control ones. It turns out that the following set of traffic load vectors:

$$\Phi_{\mathcal{E}} = \left\{ \vec{\rho} > \vec{0} : \langle \vec{\rho}, \hat{\eta} \rangle < \sum_{m \in \mathbf{M}} \pi_m \max_{\vec{x} \in \mathbf{X}} \langle \mathbf{R}_m \vec{x}, \hat{\eta} \rangle \text{ for every unit vector } \hat{\eta} \geq \vec{0} \right\} \quad (1.9)$$

(vector inequalities are considered component-wise) plays a pivotal role in characterizing the asymptotic behavior of the workload vector at large times. Note that $\Phi_{\mathcal{E}}$ depends on the environment trace \mathcal{E} via the quantities π_m and \mathbf{R}_m . Observe also that $\Phi_{\mathcal{E}}$ is defined via a ‘polar characterization’ (by sweeping over all directions defined by the unit vector $\hat{\eta}$) and is actually a convex polyhedron. Denote by $\bar{\Phi}_{\mathcal{E}}$ the closure of $\Phi_{\mathcal{E}}$ in the standard Euclidean topology.

It is later shown (in Section 3) that if $\vec{\rho} \notin \bar{\Phi}_{\mathcal{E}}$ then $\limsup_{t \rightarrow \infty} \frac{W^q(t)}{t} > 0$ for at least one $q \in \mathbf{Q}$, for any control trace \mathcal{C} . Therefore, at least one queue blows up eventually and the system is unstable, no matter what control $\vec{x}(t), t \in \mathbb{R}_+$ we utilize. The lack of stabilizing control traces when $\vec{\rho} \notin \bar{\Phi}_{\mathcal{E}}$ raises the question whether special control traces can be synthesized by applying adaptive control policies to keep the system stable when $\vec{\rho} \in \bar{\Phi}_{\mathcal{E}}$. This key issue is addressed below.

We introduce a family of backlog-responsive control policies, called **MaxProjection**, which schedule the processor effort $\vec{x} \in \mathbf{X}$ depending of the current system backlog \vec{W} and the current environment mode

m , by maximizing the projection of the service rate vector $\mathbf{R}_m \vec{x}$ on the vector $\mathbf{A} \vec{W}$ or:

$$\boxed{\vec{x}_m\{\vec{W}\} = \left\{ \vec{x} \in \mathbf{X} : \langle \mathbf{R}_m \vec{x}, \mathbf{A} \vec{W} \rangle \text{ is maximal} \right\}}, \quad (1.10)$$

where $\mathbf{A} = \text{diag}\{a^1, a^2, \dots, a^q, \dots, a^Q\}$ is any *diagonal* $Q \times Q$ matrix of *positive* real numbers (hence, \mathbf{A} is non-singular, self-adjoint, and positive-definite). This defines a family of scheduling policies, parameterized by the Q elements of the matrix \mathbf{A} . We can rewrite the MaxProjection control (1.10) in the form:

$$\vec{x}_m\{\vec{W}\} = \left\{ \vec{x} \in \mathbf{X} : \sum_{q \in \mathbf{Q}} r_m^q x^q a^q W^q \text{ is maximal} \right\}, \quad (1.11)$$

when the workload is $\vec{W} \geq \vec{0}$ and the environment is in state $m \in \mathbf{M}$. Note that since $\sum_{q \in \mathbf{Q}} x^q = 1$ the MaxProjection schedule (1.11) reduces to the following simple algorithm. When the environment mode is $m \in \mathbf{M}$ and the workload vector is \vec{W} :

- If there is a *single* queue for which the maximal value $\max_{q \in \mathbf{Q}} \{r_m^q a^q W^q\}$ is attained, then the whole processor bandwidth is allocated to that queue.
- If there is *more than one* queue for which the maximal value $\max_{q \in \mathbf{Q}} \{r_m^q a^q W^q\}$ is attained, then the processor bandwidth is split across only those maximal product queues proportionally to $1/d[r_m^q]^2$.

This is explained in Section 2 in more detail, where the rich geometry of the MaxProjection schedule is probed and analyzed. It is seen that its dynamics are dominated by a time-varying attractor, which shifts in the workload space following the changes of the environment mode.

In Section 3, it is shown that $\vec{\rho} \in \Phi_{\mathcal{E}}$ implies $\lim_{t \rightarrow \infty} \frac{\vec{W}(t)}{t} = 0$ under the MaxProjection schedule. Thus, by continuously adapting to the current backlog and environment mode, this schedule generates a feasible control trace that stabilizes the system when $\vec{\rho} \in \Phi_{\mathcal{E}}$. Indeed, $\lim_{t \rightarrow \infty} \frac{\vec{W}(t)}{t} = 0$ guarantees that the system is rate-stable, that is, the job departure rate per flow is equal to the arrival one and, hence, there is flow conservation through the queueing system [1, 8]. This is established under very general conditions on individual traffic and environment traces, allowing even for positive rate vector $\vec{s}(t)$ between consecutive δ -jumps (job arrivals).

The connection to stochastic stability is discussed in Section 4, by introducing a probabilistic (stationary ergodic) framework to model the traffic and environment processes. It is shown that the workload process has a key monotonicity property which allows the use of Loynes' method [12] to construct a stationary operational regime (steady-state) of the system.

Key extensions of the model are discussed in Section 5, specifically: 1) multiple processors, 2) continuous environment modes, and 3) feed-forward networks of modulated nodes of the above type.

Generalized MaxProjection schedules - beyond those with simple diagonal matrix \mathbf{A} - are considered in Section 6. Finally, some remarks and comments on future research are made in Section 7.

2 Geometry and Dynamics of the MaxProjection Schedule

In this section we investigate the dynamics of the system operating under the MaxProjection schedule. In particular, we explore its interesting ‘geometry’ to be leveraged later in the proofs.

Let us consider the evolution of the system in a fixed time interval (z, z') where no job arrival occurs and the environment mode does not change. That is, we assume throughout this section that $\vec{s}(t) = \vec{0}$ and $\mathbf{R}(t) = \mathbf{R}_m$ for all $t \in (z, z')$ and study how $\vec{W}(t)$ evolves under the MaxProjection schedule. It turns out that $\vec{W}(t)$ is attracted towards conic hyperplanes of progressively lower dimension, being pulled towards the ray attractor (1-dimensional cone)

$$\vec{V}_m = \left(\frac{1}{r_m^1 a^1}, \frac{1}{r_m^2 a^2}, \dots, \frac{1}{r_m^q a^q}, \dots, \frac{1}{r_m^Q a^Q} \right), \quad (2.1)$$

onto which it eventually collapses and ‘slides’ towards $\vec{0}$. This evolution occurs when the interval (z, z') is long enough to allow the system to drain to $\vec{0}$ when starting from $\vec{W}(z)$. In general, however, as the system evolves over a long time interval job arrivals will ‘kick’ the workload vector around and/or environment mode shifts to m' will activate different ray attractors $\vec{V}_{m'}$ and pull the workload vector in different directions. This complicated behavior is analyzed below.

Recall that when the workload is $\vec{W} \neq \vec{0}$ and the environment mode is $m \in \mathbf{M}$, the MaxProjection schedule chooses the bandwidth allocation vector maximizing the product in the parentheses below:

$$\vec{x}_m\{\vec{W}\} = \left\{ \vec{x} \in \mathbf{X} : \sum_{q \in \mathbf{Q}} r_m^q x^q a^q W^q \text{ is maximal} \right\}. \quad (2.2)$$

Hence, if the product $r_m^q a^q W^q$ for some specific queue $q \in \mathbf{Q}$ is strictly larger than that of any other one, then all the processor bandwidth will be allocated to that queue and $\vec{x}\{\vec{W}\} = (0, 0, \dots, 0, 1, 0, \dots, 0)$ with a single 1 appearing at the q^{th} position. In general, however, the product $r_m^q a^q W^q$ will be maximal for several queues concurrently and the processor bandwidth will have to be split proportionally amongst them.

Note that $\vec{x}\{\vec{W}\} = \vec{x}\{\gamma\vec{W}\}$ for every $\gamma \in (0, \infty)$ from (2.2) or that the bandwidth allocation vector is invariant with respect to scalings of the workload vector. This indicates that the workload space can be partitioned into cones, where the bandwidth allocation vector is constant. This is indeed demonstrated below.

2.1 Active Queues and Bandwidth Allocation

Let us define $\mathcal{Q}_m\{\vec{W}\} = \{q \in \mathbf{Q} : x_m^q\{\vec{W}\} > 0\}$ to be the set of active queues - that is, those where non-zero processor bandwidth is allocated under the MaxProjection schedule - when the environment is in state m and the workload vector is \vec{W} . Thus, if we define

$$\mathcal{D}_m\{\vec{W}\} = \max_{b \in \mathbf{Q}} \{r_m^b a^b W^b\} \quad (2.3)$$

we see that we must have $r_m^q a^q W^q = \mathcal{D}_m\{\vec{W}\} > r_m^{q'} a^{q'} W^{q'}$ for each $q \in \mathcal{Q}_m\{\vec{W}\}$ and $q' \in \mathbf{Q} - \mathcal{Q}_m\{\vec{W}\}$.

How should the processor bandwidth be allocated to active queues? To address this issue consider an interval (T, T') where the set of active queues remains invariant and equal to $\mathcal{Q} = \mathcal{Q}_m\{\vec{W}(t)\}$ for all $t \in (T, T')$. Note that (T, T') is a subset of the initial interval (z, z') , hence, $\vec{s}(t) = 0$ and $\mathbf{R}(t) = \mathbf{R}_m$ throughout (T, T') too. For each $q \in \mathcal{Q}$ and $q' \in \mathbf{Q} - \mathcal{Q}$, we must have

$$r_m^q a^q W^q(t) = \mathcal{D}_m\{\vec{W}(t)\} > r_m^{q'} a^{q'} W^{q'}(t) \quad (2.4)$$

for all $t \in (T, T')$. Consider now the situation where $\vec{x}\{\vec{W}(t)\}$ is also *constant* throughout (T, T') despite the fact that $\vec{W}(t)$ itself is changing. Since the workload process is right-continuous at T , using (1.7) we can write

$$W^q(t) = W^q(T) - r_m^q x_m^q \{\vec{W}(t)\} (t - T) \quad (2.5)$$

for each $q \in \mathcal{Q}$ and $t \in (T, T')$. Substituting back to (2.4) we get for each $q \in \mathcal{Q}$ that

$$\mathcal{D}_m\{\vec{W}(t)\} = r_m^q a^q \left(W^q(T) - r_m^q x_m^q \{\vec{W}(t)\} (t - T) \right) \quad (2.6)$$

for all $t \in (T, T')$. Equivalently, noting that $\mathcal{D}_m\{\vec{W}(T)\} = r_m^q a^q W^q(T)$, we can write

$$\mathcal{D}_m\{\vec{W}(t)\} = \mathcal{D}_m\{\vec{W}(T)\} - [r_m^q]^2 a^q x_m^q \{\vec{W}(t)\} (t - T), \quad (2.7)$$

hence,

$$[r_m^q]^2 a^q x_m^q \{\vec{W}(t)\} (t - T) = \mathcal{D}_m\{\vec{W}(T)\} - \mathcal{D}_m\{\vec{W}(t)\}, \quad (2.8)$$

for all $q \in \mathcal{Q}$ and $t \in (T, T')$. Note that the right-hand side is independent of the queue $q \in \mathcal{Q}$. In order for this ‘balance’ to be maintained across all queues in \mathcal{Q} throughout (T, T') and $\vec{x}\{\vec{W}(t)\}$ to be independent of t in this time interval (as considered above), we must have

$$x_m^q \{\vec{W}\} = \begin{cases} \frac{1}{[r_m^q]^2 a^q} \frac{1}{N_m\{\vec{W}\}}, & q \in \mathcal{Q}_m\{\vec{W}\} \\ 0, & q \in \mathbf{Q} - \mathcal{Q}_m\{\vec{W}\} \end{cases} \quad (2.9)$$

where the normalizing constant is

$$N_m\{\vec{W}\} = \sum_{q \in \mathcal{Q}_m\{\vec{W}\}} \frac{1}{[r_m^q]^2 a^q} \quad (2.10)$$

We have obtained the above formulas for the bandwidth allocation vector induced by the MaxProjection schedule, by considering that the environment mode m and the set of active queues $\mathcal{Q} = \mathcal{Q}_m\{\vec{W}(t)\}$ stay the same throughout the interval (T, T') . The emerging picture regarding the geometry and dynamics of the system becomes clear in the next section.

2.2 The Hierarchical Cone Structure of the MaxProjection Schedule

For each $\mathcal{Q} \subseteq \mathbf{Q}$ and $m \in \mathbf{M}$, define $\mathcal{C}_m^{\mathcal{Q}}$ to be the set of workload vectors for which only queues $q \in \mathcal{Q}$ receive service under the MaxProjection schedule when the environment mode is m , while the queues

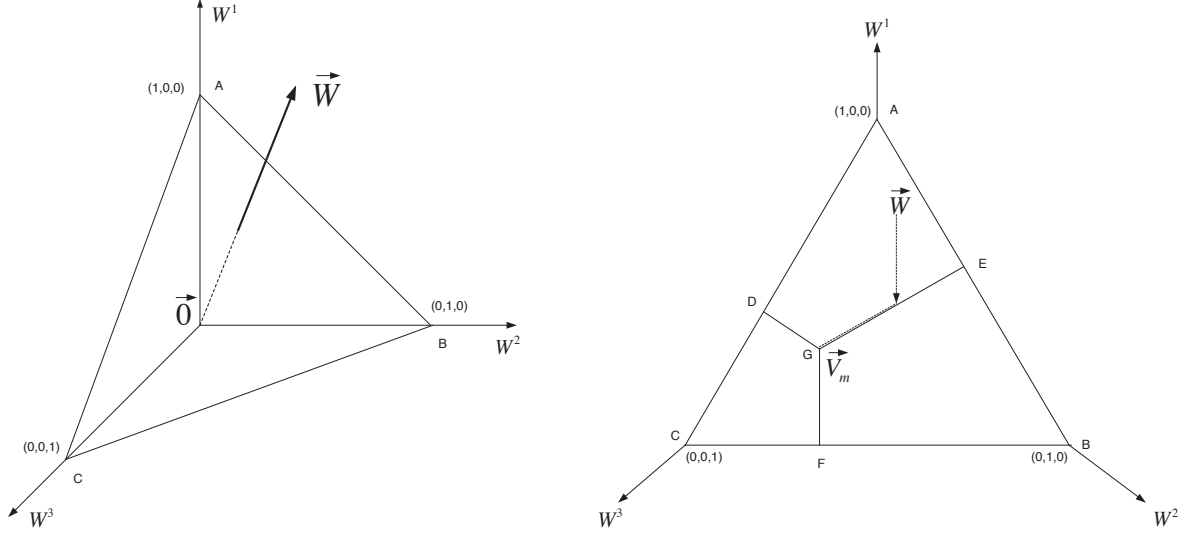


Figure 1: The evolution of the workload vector $\vec{W}(t)$ in a simple example of three queues, when the environment mode is m and the ray attractor \vec{V}_m is active. A) The left graph shows the three-dimensional workload space and defines the plane (triangle) A-B-C which helps us visualize the evolution of $\vec{W}(t)$ by marking the intersection of each vector (ray) with this plane. B) In the right graph, we represent each vector in the workload space by its intersection point with the A-B-C plane (triangle). The attractor \vec{V}_m corresponding to \mathbf{R}_m for the environment mode m is represented by G. According to the discussion in Section 2.2, there are $\binom{3}{1} = 3$ three-dimensional cones A-E-G-D/ $\vec{0}$, B-E-G-F/ $\vec{0}$, C-F-G-D/ $\vec{0}$, and $\binom{3}{2} = 3$ two-dimensional cones G-D/ $\vec{0}$, G-E/ $\vec{0}$, G-F/ $\vec{0}$, and $\binom{3}{3} = 1$ one-dimensional cone G/ $\vec{0}$, which is actually the ray attractor \vec{V}_m . On the right graph we also observe the evolution (dashed line) of the workload $\vec{W}(t)$, given that it starts from \vec{W} , evolves according to the MaxProjection schedule, and the environment stays long enough in mode m so that the described evolution is completed. Note that $\vec{W}(t)$ evolves in the three-dimensional cone A-E-G-D/ $\vec{0}$, being attracted by the two-dimensional cone G-E/ $\vec{0}$ until it ‘collapses’ into the latter. Then, it moves in the G-E/ $\vec{0}$ cone, being attracted by the one-dimensional cone G/ $\vec{0}$ and eventually collapsing onto that. It will stay on G/ $\vec{0}$ while contracting towards $\vec{0}$. Note that, in general, before this evolution has been completed new jobs may arrive and the environment mode may change causing a new cone structure and attractors to appear and drive the evolution of the workload state.

$q' \in \mathbf{Q} - \mathcal{Q}$ receive no service. Specifically,

$$\mathcal{C}_m^{\mathcal{Q}} = \left\{ \vec{W} \geq \vec{0} : r_m^q a^q W^q = \max_{b \in \mathbf{Q}} \{r_m^b a^b W^b\} > r_m^{q'} a^{q'} W^{q'} \text{ for all } q \in \mathcal{Q} \text{ and } q' \in \mathbf{Q} - \mathcal{Q} \right\} \quad (2.11)$$

or (equivalently)

$$\mathcal{C}_m^{\mathcal{Q}} = \left\{ \vec{W} \geq \vec{0} : x_m^q \{\vec{W}\} > 0 \text{ for all } q \in \mathcal{Q} \text{ and } x^{q'} \{\vec{W}\} = 0 \text{ for all } q' \in \mathbf{Q} - \mathcal{Q} \right\} \quad (2.12)$$

Note that $\mathcal{C}_m^{\mathcal{Q}}$ is a cone because $\vec{W} \in \mathcal{C}_m^{\mathcal{Q}}$ automatically implies that $\gamma \vec{W} \in \mathcal{C}_m^{\mathcal{Q}}$ for any $\gamma \in (0, \infty)$. Since $\mathcal{Q} \subseteq \mathbf{Q}$, there are $2^{\mathcal{Q}}$ cones $\mathcal{C}_m^{\mathcal{Q}}$ for each $m \in \mathbf{M}$. They can be classified according to their dimensionality in the following hierarchy:

- There are Q cones corresponding to singletons $\mathcal{Q} = \{q\}$, $q \in \mathbf{Q}$. Each of these cones

$$\mathcal{C}_m^{\{q\}} = \left\{ \vec{W} \geq \vec{0} : \left\{ r_m^q a^q W^q > r_m^{q'} a^{q'} W^{q'} \text{ for all } q' \in \mathbf{Q} - \{q\} \right\} \right\} \quad (2.13)$$

is Q -dimensional.

- There are $\binom{Q}{2}$ cones generated by 2-queue sets $\mathcal{Q} = \{q_1, q_2\} \subseteq \mathbf{Q}$. For any $q_1, q_2 \in \mathbf{Q}$, the cone

$$\mathcal{C}_m^{\{q_1, q_2\}} = \left\{ \vec{W} \geq \vec{0} : r_m^{q_1} a^{q_1} W^{q_1} = r_m^{q_2} a^{q_2} W^{q_2} > r_m^{q'} a^{q'} W^{q'} \text{ for all } q' \in \mathbf{Q} - \{q_1, q_2\} \right\} \quad (2.14)$$

is $(Q - 1)$ -dimensional.

- In general, there are $\binom{Q}{k}$ cones generated by k -queue sets $\mathcal{Q} = \{q_1, q_2, \dots, q_k\} \subseteq \mathbf{Q}$. For each $q_1, q_2, \dots, q_k \in \mathbf{Q}$, the cone

$$\begin{aligned} \mathcal{C}_m^{\{q_1, q_2, \dots, q_k\}} = \left\{ \vec{W} \geq \vec{0} : r_m^{q_1} a^{q_1} W^{q_1} = r_m^{q_2} a^{q_2} W^{q_2} = \dots = r_m^{q_k} a^{q_k} W^{q_k} > \right. \\ \left. > r_m^{q'} a^{q'} W^{q'} \text{ for all } q' \in \mathbf{Q} - \{q_1, \dots, q_k\} \right\} \end{aligned} \quad (2.15)$$

is $(Q - k + 1)$ -dimensional. Indeed, note that the k equality constraints reduce the dimension of the cone set by $k - 1$.

- Finally, there is a single cone involving all queues in \mathbf{Q} , that is,

$$\mathcal{C}_m^{\mathbf{Q}} = \{ \vec{W} \geq \vec{0} : r_m^1 a^1 W^1 = r_m^2 a^2 W^2 = \dots = r_m^q a^q W^q = \dots = r_m^Q a^Q W^Q \}. \quad (2.16)$$

This is a 1-dimensional cone. Actually, $\mathcal{C}_m^{\mathbf{Q}} = \gamma \vec{V}_m$ for $\gamma \geq 0$.

Note that the above cones form a disjoint partitioning of the workload space for each $m \in \mathbf{M}$. That is, $\mathcal{C}_m^{\mathcal{Q}} \cap \mathcal{C}_m^{\mathcal{Q}'} = \emptyset$ for all $\mathcal{Q}, \mathcal{Q}' \subseteq \mathbf{Q}$ with $\mathcal{Q}' \neq \mathcal{Q}$, and

$$\bigcup_{\mathcal{Q} \subseteq \mathbf{Q}} \mathcal{C}_m^{\mathcal{Q}} = \mathbb{R}_{0+}^{\mathbf{Q}} \quad (2.17)$$

For different $m \in \mathbf{M}$ we get a different disjoint partitioning of the workload space $\{\mathcal{C}_m^{\mathcal{Q}}, \mathcal{Q} \subseteq \mathbf{Q}\}$. Figure 1 shows the cone structure for the simple case of three queues.

Let us now compute the MaxProjection bandwidth allocation vector $\vec{x}_m\{\vec{W}\}$ when the environment mode is $m \in \mathbf{M}$ and the workload vector \vec{W} belongs to a cone $\mathcal{C}_m^{\mathcal{Q}}$. Note first that

$$\mathcal{Q}_m\{\vec{W}\} = \mathcal{Q} \text{ for all } \vec{W} \in \mathcal{C}_m^{\mathcal{Q}} \quad (2.18)$$

We can then see from (2.9) that we can rewrite the bandwidth allocation vectors as follows. Note first that

$$\boxed{\vec{W} \in \mathcal{C}_m^{\mathcal{Q}} \Rightarrow x_m^q\{\vec{W}\} = X_m^q(\mathcal{Q})}, \quad (2.19)$$

where

$$\boxed{X_m^q(\mathcal{Q}) = \begin{cases} \frac{1}{[r_m^q]^2 a^q} \frac{1}{N_m(\mathcal{Q})}, & q \in \mathcal{Q} \\ 0, & q \in \mathbf{Q} - \mathcal{Q} \end{cases}} \quad (2.20)$$

and the normalizing constant is

$$\boxed{N_m(\mathcal{Q}) = \sum_{q \in \mathcal{Q}} \frac{1}{[r_m^q]^2 a^q}} \quad (2.21)$$

Thus, the MaxProjection bandwidth allocation vector is actually constant within each cone \mathcal{C}_m^Q . Indeed, writing $\vec{X}_m(Q) = (X_m^1(Q), X_m^2(Q), \dots, X_m^q(Q), \dots, X_m^Q(Q))$ we see that

$$\boxed{\vec{x}_m\{\vec{W}\} = \sum_{Q \subseteq \mathbf{Q}} \vec{X}_m(Q) \mathbf{1}_{\{\vec{W} \in \mathcal{C}_m^Q\}}} \quad (2.22)$$

where $\vec{x}_m\{\vec{W}\} = (x_m^1\{\vec{W}\}, x_m^2\{\vec{W}\}, \dots, x_m^q\{\vec{W}\}, \dots, x_m^Q\{\vec{W}\})$ for any workload vector $\vec{W} \in \mathbb{R}_{0+}^Q$ and mode $m \in \mathbf{M}$.

2.3 Shifting Workload Attractors of the MaxProjection Schedule

From the above analysis, we see that the environment mode defines a set of cones, which have progressively lower dimensionality. Actually, $(N - 1)$ -dimensional cones appear as boundaries between N -dimensional cones. As the workload vector evolves within a N -dimensional cone, it gets attracted by some boundary $(N - 1)$ -dimensional one and eventually collapses onto it. This is repeated until the workload state gets attracted and collapses onto the attractor ray

$$\vec{V}_m = \left(\frac{1}{r_m^1 a^1}, \frac{1}{r_m^2 a^2}, \dots, \frac{1}{r_m^q a^q}, \dots, \frac{1}{r_m^Q a^Q} \right), \quad (2.23)$$

when the environment mode is $m \in \mathbf{M}$. Indeed, given that the environment stays at mode m long enough and no job arrival occurs, the direction of the workload vector $\vec{W}(t)$ gradually converges towards that of \vec{V}_m until they become identical at some finite time. Then, the workload vector gradually recedes on the ray \vec{V}_m until it hits 0. In reality, this evolution is interrupted and the workload vector diverted by job arrivals and environmental mode changes which cause the attractor $\vec{V}_{e(t)}$ to shift and pull the workload towards it.

3 Rate-Stability and Flow Conservation under MaxProjection Schedules

In this section, we address the stability/throughput issue of the system. We employ the very general (yet practical) concept of rate-stability, which implies that on each traffic flow the average job departure rate is equal to the average job arrival one. That is, there is flow conservation across the queueing structure, and no flow deficit appears at the output due to (linear in time) accumulation of jobs in it. A sufficient condition for the system to be rate-stable is $\lim_{t \rightarrow \infty} \frac{\vec{W}(t)}{t} = \vec{0}$, as discussed in [1, 8] in a general context.

We investigate stability under the mildest possible assumptions on the traffic trace $\mathcal{T} = \{\vec{s}(t), t \in \mathbb{R}_+\}$, which go beyond the queueing system and address the more general problem of stable solutions of the integral and differential equations (1.7) and (1.8) correspondingly. Specifically, we assume that:

1. The function $\vec{s}(t)$ has δ -jumps corresponding to job arrivals (a finite number of them in any finite interval). In the pure queueing model, $\vec{s}(t) = \vec{0}$ between consecutive δ -jumps.

2. However, in this section we allow $\vec{s}(t)$ to possibly take positive values between any two successive δ -jumps, generalizing the model beyond the initial queueing context.

Of course, we still assume that (1.4) is satisfied. We start by considering conditions under which the system goes unstable. Recall that $\bar{\Phi}_{\mathcal{E}}$ is the closure of the set defined in (1.9).

Proposition 3.1 (Unstable Traffic Traces) For any arbitrarily fixed environment trace \mathcal{E} defining $\bar{\Phi}_{\mathcal{E}}$ and any traffic trace \mathcal{T} with load vector $\vec{\rho}$, we have

$$\vec{\rho} \notin \bar{\Phi}_{\mathcal{E}} \Rightarrow \liminf_{t \rightarrow \infty} \left\langle \frac{\vec{W}(t)}{t}, \hat{\eta}_* \right\rangle > 0, \quad (3.1)$$

for some unit vector $\hat{\eta}_* \in \mathbb{R}_+^Q$, under any feasible control trace $\mathcal{C} = \{\vec{x}(t), t \in \mathbb{R}_+\}$. Consequently,

$$\vec{\rho} \notin \bar{\Phi}_{\mathcal{E}} \Rightarrow \limsup_{t \rightarrow \infty} \frac{W_t^q}{t} > 0 \quad (3.2)$$

for at least one queue $q \in \mathbf{Q}$, irrespectively of what bandwidth allocation schedule is used. Therefore, when $\vec{\rho} \notin \bar{\Phi}_{\mathcal{E}}$ the system is essentially unstable.

Proof: Since $\vec{\rho} \notin \bar{\Phi}_{\mathcal{E}}$ there must be some unit vector $\hat{\eta}_*$ (which depends on $\vec{\rho}$) such that

$$\langle \vec{\rho}, \hat{\eta}_* \rangle > \sum_{m \in \mathbf{M}} \pi_m \max_{\vec{x}_m \in \mathbf{X}} \langle \mathbf{R}_m, \hat{\eta}_* \rangle \quad (3.3)$$

Projecting (1.7) on $\hat{\eta}_*$ (and suppressing $\mathbf{1}_{\{W^q(u) > 0\}}$ and $W^q(0)$ to obtain inequality), we get

$$\left\langle \vec{W}(t), \hat{\eta}_* \right\rangle \geq \left\langle \int_0^t \vec{s}(u) du, \hat{\eta}_* \right\rangle - \sum_{m \in \mathbf{M}} \int_0^t \langle \mathbf{R}_m \vec{x}(u), \hat{\eta}_* \rangle \mathbf{1}_{\{e(u)=m\}} du, \quad (3.4)$$

which implies that

$$\left\langle \vec{W}(t), \hat{\eta}_* \right\rangle \geq \left\langle \int_0^t \vec{s}(u) du, \hat{\eta}_* \right\rangle - \sum_{m \in \mathbf{M}} \max_{\vec{x} \in \mathbf{X}} \langle \mathbf{R}_m \vec{x}, \hat{\eta}_* \rangle \int_0^t \mathbf{1}_{\{e(u)=m\}} du. \quad (3.5)$$

Dividing both sides by $t \rightarrow \infty$ and using (1.4) and (1.1), we get

$$\liminf_{t \rightarrow \infty} \left\langle \frac{\vec{W}(t)}{t}, \hat{\eta}_* \right\rangle \geq \langle \vec{\rho}, \hat{\eta}_* \rangle - \sum_{m \in \mathbf{M}} \max_{\vec{x} \in \mathbf{X}} \langle \mathbf{R}_m \vec{x}, \hat{\eta}_* \rangle \pi_m > 0 \quad (3.6)$$

which is positive because of the special (3.3) inequality for the $\hat{\eta}_*$ unit vector. As a result, it automatically follows that we should have $\limsup_{t \rightarrow \infty} \frac{W_t^q}{t} > 0$ for at least one queue $q \in \mathbf{Q}$. ■

Given that for $\vec{\rho} \notin \bar{\Phi}_{\mathcal{E}}$ the system is unstable and the workload can grow linearly in time under any processor schedule, the question that naturally arises is whether for $\vec{\rho} \in \bar{\Phi}_{\mathcal{E}}$ the workload may only grow sub-linearly in time under the MaxProjection schedule. That is, whether MaxProjection can maintain rate-stability and conserve the flow across the system, when the load vector is within the alleged stability region. This is indeed established below. The proof uses a methodological framework initially developed in [1] for a different queueing structure. We refer to the previous paper for details on this framework, and focus below on addressing the unique aspects of the arguments needed for the stability analysis pursued here.

Proposition 3.2 (Rate-Stable Traffic Traces under MaxProjection) When the system operates under the MaxProjection schedule on any arbitrarily fixed environment trace \mathcal{E} defining $\Phi_{\mathcal{E}}$ and any traffic trace \mathcal{T} with $\vec{\rho} \in \Phi_{\mathcal{E}}$, we have

$$\lim_{t \rightarrow \infty} \frac{\vec{W}(t)}{t} = \vec{0}, \quad (3.7)$$

that is, $\vec{W}(t)$ can only grow sub-linearly with time. Thus, the system is rate stable.

Proof: The proof primarily reflects the *geometry* of the system induced under the MaxProjection schedule. It is supported by some core analytic arguments and is deployed in steps, as follows.

Step 1 (The Assumption to Be Contradicted) Showing that $\lim_{t \rightarrow \infty} \frac{\vec{W}(t)}{t} = \vec{0}$ is equivalent to showing that $\lim_{t \rightarrow \infty} \left\langle \frac{\vec{W}(t)}{t}, \mathbf{A} \frac{\vec{W}(t)}{t} \right\rangle = 0$, since \mathbf{A} is diagonal with positive elements. Arguing by contradiction, assume that

$$\limsup_{t \rightarrow \infty} \left\langle \frac{\vec{W}(t)}{t}, \mathbf{A} \frac{\vec{W}(t)}{t} \right\rangle = \beta > 0. \quad (3.8)$$

Let $\{t_\ell\}_{\ell=1}^{\infty}$ be an increasing unbounded time sequence on which

$$\lim_{\ell \rightarrow \infty} \frac{\vec{W}(t_\ell)}{t_\ell} = \vec{\xi} \neq \vec{0} \quad (3.9)$$

and the previous limit supremum (3.8) is attained, hence, $\left\langle \vec{\xi}, \mathbf{A} \vec{\xi} \right\rangle = \beta$. We shall show that this assumption eventually contradicts the fact that β is the limit supremum defined above.

The existence of the sequence $\{t_\ell\}_{\ell=1}^{\infty}$ can be guaranteed as follows. First, choose a time sequence $\{t_n\}_{n=1}^{\infty}$ on which the limit supremum (3.8) is attained. Now consider the sequence $\{\frac{\vec{W}(t_n)}{t_n}\}_{n=1}^{\infty}$. Note that by (1.7) we have $\vec{0} \leq \vec{W}(t) \leq \int_0^t \vec{s}(u) du$ and so (dividing by t and letting $t \rightarrow \infty$) we get

$$\vec{0} \leq \liminf_{t \rightarrow \infty} \frac{\vec{W}(t)}{t} \leq \limsup_{t \rightarrow \infty} \frac{\vec{W}(t)}{t} \leq \vec{\rho} \quad (3.10)$$

Hence, choosing any $\theta > 0$ and defining $\vec{\theta}$ to be the Q -dimensional vector with all its components equal to θ , we have that $\frac{\vec{W}(t)}{t} \in \{\vec{y} \in \mathbb{R}_{0+}^Q : \vec{0} \leq \vec{y} \leq \vec{\rho} + \vec{\theta}\}$ eventually (for all t greater than some t_θ which depends on θ). The set $\{\vec{y} \in \mathbb{R}_{0+}^Q : \vec{0} \leq \vec{y} \leq \vec{\rho} + \vec{\theta}\}$ is bounded, so the sequence $\{\frac{\vec{W}(t_n)}{t_n}\}_{n=1}^{\infty}$ is eventually bounded and must have a convergent subsequence (see [11]). The latter is the sequence $\{t_\ell\}_{\ell=1}^{\infty}$ we are seeking.

Step 2 (The $\vec{\xi}$ -Surrounding Cone) Given any arbitrarily fixed $\epsilon > 0$, define now the following cone of workload vectors for each $m \in \mathbf{M}$:

$$\mathbf{C}_m^\epsilon(\vec{\xi}) = \left\{ \vec{W} \geq \vec{0} : \max_{\vec{x} \in \mathbf{X}} \left\langle \mathbf{R}_m \vec{x}, \mathbf{A} \vec{\xi} \right\rangle - \epsilon \leq \left\langle \mathbf{I}\{\vec{W}\} \mathbf{R}_m \vec{x}_m \{\vec{W}\}, \mathbf{A} \vec{\xi} \right\rangle \right\} \quad (3.11)$$

where $\vec{x}_m \{\vec{W}\} = \left\{ \vec{x} \in \mathbf{X} : \left\langle \mathbf{R}_m \vec{x}, \mathbf{A} \vec{W} \right\rangle \text{ is maximal} \right\}$ according to the MaxProjection schedule (1.10). The set $\mathbf{C}_m^\epsilon(\vec{\xi})$ is a cone because both $\mathbf{I}\{\gamma \vec{W}\} = \mathbf{I}\{\gamma \vec{W}\}$ and $\vec{x}_m \{\gamma \vec{W}\} = \vec{x}_m \{\gamma \vec{W}\}$ are scale invariant with respect to $\gamma \in (0, \infty)$. Note that $\vec{\xi} \in \mathbf{C}_m^\epsilon(\vec{\xi})$ because

$$\left\langle \mathbf{I}\{\vec{\xi}\} \mathbf{R}_m \vec{x}_m \{\vec{\xi}\}, \mathbf{A} \vec{\xi} \right\rangle = \left\langle \mathbf{R}_m \vec{x}_m \{\vec{\xi}\}, \mathbf{A} \vec{\xi} \right\rangle = \max_{\vec{x} \in \mathbf{X}} \left\langle \mathbf{R}_m \vec{x}, \mathbf{A} \vec{\xi} \right\rangle. \quad (3.12)$$

The first above equality is due to the fact that $\mathbf{I}\{\vec{\xi}\}\mathbf{R}_m\vec{x}_m\{\vec{\xi}\} = \mathbf{R}_m\vec{x}_m\{\vec{\xi}\}$ because \mathbf{R}_m is diagonal and $x_m^q\{\vec{\xi}\} = 0$ for $\xi^q = 0$. The second equality is due to the definition of the MaxProjection schedule.

Since the vector $\vec{\xi} \neq \vec{0}$ belongs to each cone \mathbf{C}_m^ϵ for various $m \in \mathbf{M}$, the intersection of all these cones must be a non-empty cone. Therefore, define

$$\mathbf{C}_*^\epsilon(\vec{\xi}) = \bigcap_{m \in \mathbf{M}} \mathbf{C}_m^\epsilon(\vec{\xi}) \neq \emptyset \quad (3.13)$$

Actually, $\mathbf{C}_*^\epsilon(\vec{\xi})$ is a Q -dimensional cone (as opposed to a lower dimensional one). Indeed, note that there are positive vectors $\vec{W} > \vec{0}$ satisfying

$$\max_{\vec{x} \in \mathbf{X}} \langle \mathbf{R}_m \vec{x}, \mathbf{A} \vec{\xi}^\lambda \rangle - \epsilon \leq \langle \mathbf{R}_m \vec{x}_m(\vec{W}), \mathbf{A} \vec{\xi}^\lambda \rangle = \langle \mathbf{I}\{\vec{W}\} \mathbf{R}_m \vec{x}_m(\vec{W}), \mathbf{A} \vec{\xi}^\lambda \rangle \quad (3.14)$$

for all $m \in \mathbf{M}$, given an arbitrarily fixed $\epsilon > 0$. Note that $\mathbf{I}\{\vec{W}\}$ is simply the identity matrix for $\vec{W} > \vec{0}$. The directions (rays) of these vectors $\vec{W} > \vec{0}$ are perturbations of the direction of the vector $\vec{\xi}$, for which (3.14) holds for all $m \in \mathbf{M}$ by construction. Figure 2 shows a ‘visualization’ of the cones $\mathbf{C}_m^\epsilon(\vec{\xi})$ and $\mathbf{C}_*^\epsilon(\vec{\xi})$ for the simple case of a system with three queues. Note the form of the cone $\mathbf{C}_*^\epsilon(\vec{\xi})$ when $\vec{\xi}$ is on a boundary ‘wall’ or ‘corner’ of the workload space \mathbb{R}_+^Q (as in Fig. 2.D).

We conclude that, for each arbitrarily fixed $\epsilon > 0$, we have $\vec{\xi} \in \mathbf{C}_*^\epsilon(\vec{\xi})$ and the cone $\mathbf{C}_*^\epsilon(\vec{\xi})$ is a fully Q -dimensional one, as opposed to having lower dimension. Moreover,

$$\vec{W} \in \mathbf{C}_*^\epsilon(\vec{\xi}) \Rightarrow \max_{\vec{x} \in \mathbf{X}} \langle \mathbf{R}_m \vec{x}, \mathbf{A} \vec{\xi}^\lambda \rangle - \epsilon \leq \langle \mathbf{I}\{\vec{W}\} \mathbf{R}_m \vec{x}_m(\vec{W}), \mathbf{A} \vec{\xi}^\lambda \rangle \quad (3.15)$$

for all $m \in \mathbf{M}$. Actually, the reverse is also true, since this is essentially the defining property of this cone.

Step 3 (The Cone Entry Times) Since $\lim_{\ell \rightarrow \infty} \frac{\vec{W}(t_\ell)}{t_\ell} = \vec{\xi} \neq \vec{0}$, we have $\vec{W}(t_\ell) \approx \vec{\xi} t_\ell$ at large times. Given any arbitrarily fixed ϵ , the workload vector $\vec{W}(t_\ell)$ will eventually be in the cone $\mathbf{C}_*^\epsilon(\vec{\xi})$ for all ℓ greater than some ℓ_*^ϵ (which gets larger when ϵ gets smaller). Define now z_ℓ to be the last time before t_ℓ that $\vec{W}(t)$ crosses from outside of cone $\mathbf{C}_*^\epsilon(\vec{\xi})$ to the inside, that is:

$$z_\ell = \inf \left\{ z < t_\ell : \vec{W}(t) \in \mathbf{C}_*^\epsilon(\vec{\xi}) \text{ for every } t \in (z, t_\ell] \right\} \quad (3.16)$$

which implies that

$$\vec{W}(z_\ell^-) \notin \mathbf{C}_*^\epsilon(\vec{\xi}), \text{ but } \vec{W}(t) \in \mathbf{C}_*^\epsilon(\vec{\xi}) \text{ for every } t \in [z_\ell, t_\ell]. \quad (3.17)$$

If $\vec{W}(t)$ has been in the cone $\mathbf{C}_*^\epsilon(\vec{\xi})$ throughout the interval $(0, t_\ell]$ then we set $z_\ell = 0$ by convention.

Observe now that the length $|\vec{W}(t_\ell) - \vec{W}(z_\ell^-)|$ is greater than the minimal distance of $\vec{W}(t_\ell)$ from the boundary of the $\mathbf{C}_*^\epsilon(\vec{\xi})$ cone, since at time z_ℓ^- the workload $\vec{W}(z_\ell^-)$ is outside of the cone $\mathbf{C}_*^\epsilon(\vec{\xi})$. (Note that in the special case where $\vec{\xi}$ is on the workload space boundary, like in Figure 2.D, we should exclude the common boundary of the cone and the workload space in the previous minimal distance calculation.) The previous observation implies that

$$0 < \liminf_{\ell \rightarrow \infty} \frac{|\vec{W}(t_\ell) - \vec{W}(z_\ell^-)|}{|\vec{W}(t_\ell)|}, \quad (3.18)$$

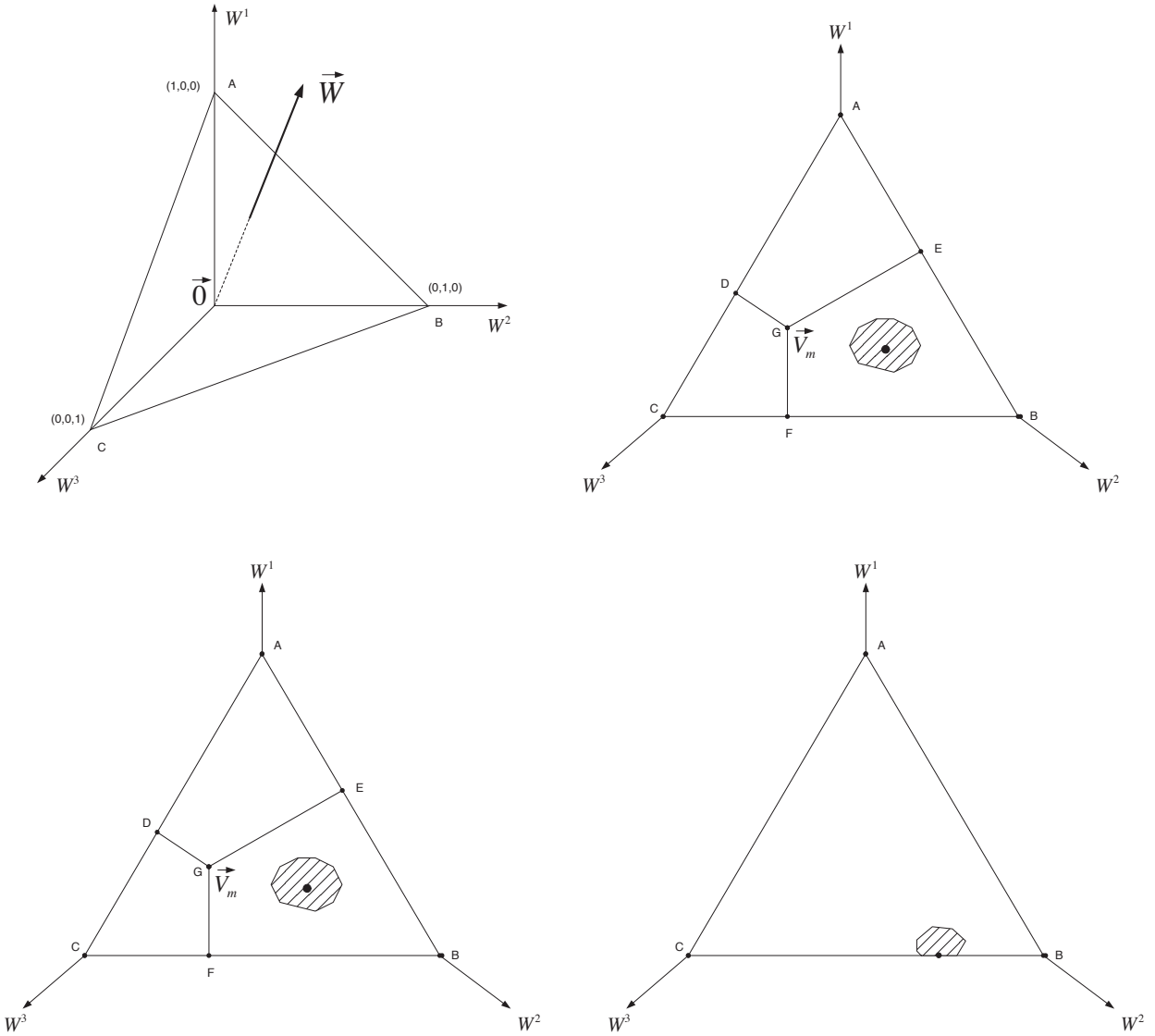


Figure 2: **(The $\vec{\xi}$ -Surrounding Cone)** Continuing with the three-queue example of Figure 1, we consider here the $\vec{\xi}$ -surrounding cones $C_m^\epsilon(\vec{\xi})$ and $C_*^\epsilon(\vec{\xi})$. A) The upper-left graph shows the three-dimensional workload space and defines the A-B-C plane which helps us visualize the position of vectors in following graphs. On this A-B-C plane we mark the intersection of each vector/ray in the workload space with this plane. B) In the upper-right graph, the footprint of the $C_m^\epsilon(\vec{\xi})$ cone on the A-B-C plane is represented by the shaded area, around the vector $\vec{\xi}$ represented by the dark dot. Shown also is the attractor \vec{V}_m and the corresponding cone partitioning of the workload space. C) In the lower-left graph, the footprint of the $C_*^\epsilon(\vec{\xi})$ cone on the A-B-C plane is represented by the shaded area, around the vector $\vec{\xi}$ represented by the dark dot. D) In the lower-right graph, the footprint of the $C_*^\epsilon(\vec{\xi})$ cone on the A-B-C plane is represented by the shaded area, in the extreme case where the vector $\vec{\xi}$ (represented by the dark dot) is on the boundary of the workload space.

since $\vec{W}(t_\ell) \approx \vec{\xi} t_\ell \rightarrow \infty$ for $\ell \rightarrow \infty$ and $C_*^\epsilon(\vec{\xi})$ is a Q -dimensional cone in \mathbb{R}_{0+}^Q . But $\vec{W}(t_\ell) - \vec{W}(z_\ell^-) \leq \int_{z_\ell^-}^{t_\ell} \vec{s}(u) du$ from (1.7), so we get

$$0 < \liminf_{\ell \rightarrow \infty} \frac{|\vec{W}(t_\ell) - \vec{W}(z_\ell^-)|}{|\vec{W}(t_\ell)|} \leq \limsup_{\ell \rightarrow \infty} \frac{\int_{z_\ell^-}^{t_\ell} \vec{s}(u) du}{|\vec{\xi}| t_\ell}. \quad (3.19)$$

The reason why we are using z_ℓ^- instead of z_ℓ in the previous expressions is that $\vec{W}(z_\ell)$ may enter into C_*^ϵ with a jump (at a job arrival) coming from $\vec{W}(t_\ell^-)$ which lies outside the cone by construction.

Now observe that, if $\lim_{\ell \rightarrow \infty} \frac{t_\ell - z_\ell}{t_\ell} = 0$ (or equivalently $\lim_{\ell \rightarrow \infty} \frac{z_\ell}{t_\ell} = 1$), then $\frac{\int_{z_\ell^-}^{t_\ell} \vec{s}(u) du}{t_\ell} = 0$. Indeed, using (1.4) we get

$$\frac{\int_{z_\ell^-}^{t_\ell} \vec{s}(u) du}{t_\ell} = \frac{\int_0^{t_\ell} \vec{s}(u) du}{t_\ell} - \frac{\int_0^{z_\ell^-} \vec{s}(u) du}{z_\ell} \times \frac{z_\ell}{t_\ell} \rightarrow \vec{\rho} - \vec{\rho} \times 1 = 0 \quad (3.20)$$

as $\ell \rightarrow \infty$. But that would contradict (3.19) by violating its left inequality, since its rightmost term would be squeezed to zero. From the above thread of arguments we see that $\liminf_{\ell \rightarrow \infty} \frac{t_\ell - z_\ell}{t_\ell} = \delta > 0$. This implies that there exists an increasing subsequence $\{t_k\}_{k=1}^\infty$ of $\{t_\ell\}_{\ell=1}^\infty$ such that

$$\boxed{\lim_{k \rightarrow \infty} \frac{t_k - z_k}{t_k} = \delta > 0} \quad (3.21)$$

for some positive δ . A ‘visualization’ of the geometric rationale behind the previous arguments is provided in Figure 3.

Step 4 (Workload Evolution at Cone Entry Points) Consider now the evolution of the workload in the interval $(z_b, t_b]$, while it is floating in the cone C_m^ϵ . We have:

$$\vec{W}(t_k) - \vec{W}(z_k) = \int_{z_k}^{t_k} \vec{s}(u) du - \sum_{m \in \mathbf{M}} \int_{z_k}^{t_k} \mathbf{1}_{\{e(u)=m\}} \mathbf{I}\{\vec{W}(u)\} \mathbf{R}_m \vec{x}_m \{\vec{W}(u)\} du \quad (3.22)$$

Projecting on the vector $\mathbf{A}\vec{\xi}$, we get

$$\langle \vec{W}(t_k) - \vec{W}(z_k), \mathbf{A}\vec{\xi} \rangle = \left\langle \int_{z_k}^{t_k} \vec{s}(u) du, \mathbf{A}\vec{\xi} \right\rangle - \sum_{m \in \mathbf{M}} \int_{z_k}^{t_k} \mathbf{1}_{\{e(u)=m\}} \left\langle \mathbf{I}\{\vec{W}(u)\} \mathbf{R}_m \vec{x}_m \{\vec{W}(u)\}, \mathbf{A}\vec{\xi} \right\rangle du. \quad (3.23)$$

Now, since $\vec{W}(u) \in C_*^\epsilon$ for all $u \in (z_k, t_k]$ by construction, we have from (3.15) that

$$\max_{\vec{x} \in \mathbf{X}} \langle \mathbf{R}_m \vec{x}, \mathbf{A}\vec{\xi} \rangle - \epsilon \leq \left\langle \mathbf{I}\{\vec{W}(u)\} \mathbf{R}_m \vec{x}_m \{\vec{W}(u)\}, \mathbf{A}\vec{\xi} \right\rangle \quad (3.24)$$

for all $u \in (z_k, t_k]$. Therefore, we get from (3.23) that

$$\left\langle \vec{W}(t_k) - \vec{W}(z_k), \mathbf{A}\vec{\xi} \right\rangle \leq \left\langle \int_{z_k}^{t_k} \vec{s}(u) du, \mathbf{A}\vec{\xi} \right\rangle - \sum_{m \in \mathbf{M}} \max_{\vec{x} \in \mathbf{X}} \langle \mathbf{R}_m \vec{x}, \mathbf{A}\vec{\xi} \rangle \int_{z_k}^{t_k} \mathbf{1}_{\{e(u)=m\}} du + \epsilon. \quad (3.25)$$

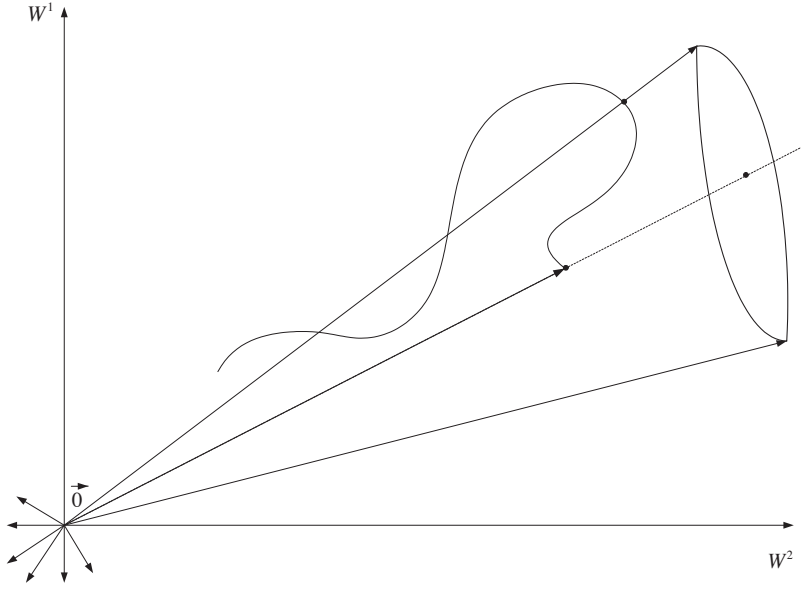


Figure 3: **(Workload Evolution in the $C_*^c(\vec{\xi})$ Cone)** Referring to steps 3 and 4 of the proof of Proposition 3.2, this graph presents a ‘visualization’ of the $\vec{\xi}$ -surrounding cone and a ‘caricature’ of the workload trajectory, as it enters the cone for the last time before t and drifts in it throughout the $(z, t]$ time interval (for appropriate subsequences of the times z and t). The multiple arrows at the origin represent the multi-dimensional nature of the graph and cone.

Dividing by $t_k - z_k$ and letting $k \rightarrow \infty$, we get

$$\limsup_{b \rightarrow \infty} \left\langle \frac{\vec{W}(t_k) - \vec{W}(z_k)}{t_k - z_k}, \mathbf{A}\vec{\xi} \right\rangle \leq \langle \vec{\rho}, \mathbf{A}\vec{\xi} \rangle - \sum_{m \in \mathbf{M}} \pi_m \max_{\vec{x} \in \mathbf{X}} \langle \mathbf{R}_m \vec{x}, \mathbf{A}\vec{\xi} \rangle + \epsilon. \quad (3.26)$$

Indeed, note that that from (1.4) and (3.21) we have

$$\frac{\int_{z_k}^{t_k} \vec{s}(u) du}{t_k - z_k} = \frac{\int_0^{t_k} \vec{s}(u) du}{t_k} \frac{t_k}{t_k - z_k} - \frac{\int_0^{z_k} \vec{s}(u) du}{z_k} \frac{z_k}{t_k - z_k} \rightarrow \vec{\rho} \frac{1}{\delta} - \vec{\rho} \left(\frac{1}{\delta} - 1 \right) = \vec{\rho}, \quad (3.27)$$

as $k \rightarrow \infty$. Similarly, from (1.1) and (3.21) we have

$$\frac{\int_{z_k}^{t_k} \mathbf{1}_{\{e(u)=m\}} du}{t_k - z_k} = \frac{\int_0^{t_k} \mathbf{1}_{\{e(u)=m\}} du}{t_k} \frac{t_k}{t_k - z_k} - \frac{\int_0^{z_k} \mathbf{1}_{\{e(u)=m\}} du}{z_k} \frac{z_k}{t_k - z_k} \rightarrow \pi_m \frac{1}{\delta} - \pi_m \left(\frac{1}{\delta} - 1 \right) = \pi_m, \quad (3.28)$$

as $k \rightarrow \infty$, for every $m \in \mathbf{M}$.

The key inequality (3.26) implies that when $\vec{\rho} \in \Phi_{\mathcal{E}}$ (as assumed here) and ϵ is small enough, we can get the right-hand-side to be *negative*. Therefore, choosing a small enough ϵ , we can get

$$\limsup_{b \rightarrow \infty} \left\langle \frac{\vec{W}(t_k) - \vec{W}(z_k)}{t_k - z_k}, \mathbf{A}\vec{\xi} \right\rangle \leq -\gamma, \quad (3.29)$$

where

$$0 < \gamma = \sum_{m \in \mathbf{M}} \pi_m \max_{\vec{x} \in \mathbf{X}} \langle \mathbf{R}_m \vec{x}, \mathbf{A}\vec{\xi} \rangle - \langle \vec{\rho}, \mathbf{A}\vec{\xi} \rangle - \epsilon. \quad (3.30)$$

Choose now an increasing unbounded subsequence $\{t_c\}_{c=1}^\infty$ of $\{t_k\}_{k=1}^\infty$ on which the previous limit supremum is attained. Then, recalling (3.21) we see that

$$\lim_{c \rightarrow \infty} \left\langle \frac{\vec{W}(t_c) - \vec{W}(z_c)}{t_c}, \mathbf{A}\vec{\xi} \right\rangle \times \lim_{c \rightarrow \infty} \frac{t_c}{t_c - z_c} \leq -\gamma, \quad (3.31)$$

so (since $\{t_c\}_{c=1}^\infty$ is a subsequence of $\{t_b\}_{b=1}^\infty$)

$$\lim_{c \rightarrow \infty} \left\langle \frac{\vec{W}(t_c) - \vec{W}(z_c)}{t_c}, \mathbf{A}\vec{\xi} \right\rangle \leq -\gamma\delta. \quad (3.32)$$

Since $\lim_{c \rightarrow \infty} \frac{\vec{W}(t_c)}{t_c} = \vec{\xi}$ is convergent (because $\{t_c\}_{c=1}^\infty$ is a subsequence of $\{t_b\}_{b=1}^\infty$ and of $\{t_a\}_{a=1}^\infty$) we now get from (3.31) the left inequality below:

$$\lim_{c \rightarrow \infty} \left\langle \frac{\vec{W}(t_c)}{t_c}, \mathbf{A}\vec{\xi} \right\rangle + \gamma\delta \leq \liminf_{c \rightarrow \infty} \left\langle \frac{\vec{W}(z_c)}{t_c}, \mathbf{A}\vec{\xi} \right\rangle \leq \liminf_{c \rightarrow \infty} \left\langle \frac{\vec{W}(z_c)}{z_c}, \mathbf{A}\vec{\xi} \right\rangle. \quad (3.33)$$

The right inequality is due to the fact that $z_c < t_c$ for all c . Choosing an increasing unbounded subsequence $\{t_d\}_{d=1}^\infty$ of $\{t_c\}_{c=1}^\infty$ on which the previous limit infimum is attained, we get

$$\lim_{d \rightarrow \infty} \left\langle \vec{\xi}, \mathbf{A}\vec{\xi} \right\rangle + \gamma\delta \leq \lim_{d \rightarrow \infty} \left\langle \frac{\vec{W}(z_d)}{z_d}, \mathbf{A}\vec{\xi} \right\rangle \quad (3.34)$$

Step 5 (Establishing the Contradiction) Finally, we consider an increasing unbounded subsequence $\{t_e\}_{e=1}^\infty$ of $\{t_d\}_{d=1}^\infty$ such that

$$\lim_{e \rightarrow \infty} \frac{\vec{W}(z_e)}{z_e} = \vec{\zeta} \quad (3.35)$$

and from (3.34)

$$\left\langle \vec{\xi}, \mathbf{A}\vec{\xi} \right\rangle + \gamma\delta \leq \left\langle \vec{\zeta}, \mathbf{A}\vec{\xi} \right\rangle \quad (3.36)$$

The existence of such a subsequence can be established by arguing as in (3.10).

Note that from (3.36) we see that $\vec{\zeta}$ cannot be equal to $\vec{\xi}$, so $\vec{\zeta} - \vec{\xi} \neq \vec{0}$. Therefore, because \mathbf{A} is positive-definite, we get $0 < \left\langle \vec{\zeta} - \vec{\xi}, \mathbf{A}(\vec{\zeta} - \vec{\xi}) \right\rangle$, and by expanding it we get

$$0 < \left\langle \vec{\zeta}, \mathbf{A}\vec{\zeta} \right\rangle - \left\langle \vec{\xi}, \mathbf{A}\vec{\zeta} \right\rangle - \left\langle \vec{\zeta}, \mathbf{A}\vec{\xi} \right\rangle + \left\langle \vec{\xi}, \mathbf{A}\vec{\xi} \right\rangle. \quad (3.37)$$

Now, because \mathbf{A} is self-adjoint we get $\left\langle \vec{\xi}, \mathbf{A}\vec{\zeta} \right\rangle = \left\langle \mathbf{A}\vec{\xi}, \vec{\zeta} \right\rangle = \left\langle \vec{\zeta}, \mathbf{A}\vec{\xi} \right\rangle$ (the last equality following from the fact that the inner product is symmetric in its arguments), hence, from (3.37) we have

$$2 \left\langle \vec{\zeta}, \mathbf{A}\vec{\xi} \right\rangle < \left\langle \vec{\zeta}, \mathbf{A}\vec{\zeta} \right\rangle + \left\langle \vec{\xi}, \mathbf{A}\vec{\xi} \right\rangle. \quad (3.38)$$

But from (3.36) we have $\left\langle \vec{\xi}, \mathbf{A}\vec{\xi} \right\rangle < \left\langle \vec{\zeta}, \mathbf{A}\vec{\xi} \right\rangle$ and substituting in (3.38) we get

$$\left\langle \vec{\xi}, \mathbf{A}\vec{\xi} \right\rangle < \left\langle \vec{\zeta}, \mathbf{A}\vec{\zeta} \right\rangle, \quad (3.39)$$

which contradicts the assumption that $\langle \vec{\xi}, \mathbf{A}\vec{\xi} \rangle = \beta$ is the limit supremum considered at the beginning in (3.8). Indeed, on the sequence $\{t_e\}_{e=1}^{\infty}$ a higher value would be attained, as shown in (3.39). This completes the proof of the lemma. \blacksquare

The previous proposition establishes rate-stability of the queueing structure and flow conservation [1, 8] for any $\vec{\rho} \in \Phi_{\mathcal{E}}$ under MaxProjection. Therefore, since the structure is essentially unstable for any processor schedule when $\vec{\rho} \notin \bar{\Phi}_{\mathcal{E}}$, we can say that the MaxProjection schedule *maximizes the throughput* of the processing system.

4 Stochastic Stability under MaxProjection

In this section we turn our attention to a more restrictive - but perhaps more traditional - form of stability, which arises when a full probabilistic framework is superposed on the structure. It turns out that this system possesses an important monotonicity property, which allows the use of the powerful Loynes method [12] for constructing a stationary regime, when the system is modelled within a stationary ergodic framework. The discussion below provides the connection of the trace-based analysis of the previous section to traditional notions of stochastic stability.

Let us start by introducing some necessary additional assumptions that establish a general probabilistic framework within which we can address issues of stochastic stability. We assume that there is some probability space (Ω, \mathcal{F}, P) where 1) the environment fluctuation process $\mathbf{E} = \{\mathbf{e}(t), t \in \mathbb{R}\}$ and 2) the stochastic traffic flows $\mathbf{T}^q = \{s^q(t), t \in \mathbb{R}\}$ for $q \in \mathbf{Q}$ are defined. The trace $\mathcal{E} = \{e(t), t \in \mathbb{R}\}$ can now be viewed as a sample path of the stochastic process \mathbf{E} . Similarly, the trace $\mathcal{T}^q = \{s^q(t), t \in \mathbb{R}\}$ can be viewed as a single sample path of the stochastic process \mathbf{T}^q for each $q \in \mathbf{Q}$. We impose the following restrictive assumptions:

1. For each $q \in \mathbf{Q}$, the function $s^q(t)$ has a finite number of job arrivals (δ -jumps) in any finite time interval, while $s^q(t) = 0$ between consecutive job arrivals, almost surely. Thus, \mathbf{T}^q is basically a random marked point process [2, 7].
2. For each $q \in \mathbf{Q}$, the stochastic process \mathbf{T}^q is stationary and ergodic with respect to time shifts $\theta_z\{s^q(t), \mathbb{R}\} = \{s^q(t - z), t \in \mathbb{R}\}$, for all $z \in \mathbb{R}$. As a result, the condition (1.4) is guaranteed by Birkoff's ergodic theorem [13, 18].
3. In any finite time interval the function $e(t)$ has a finite number of jumps, which correspond to changes of the environment mode $m \in \mathbf{M}$. Thus, \mathbf{E} is a simple random marked point process [2, 7].
4. The process \mathbf{E} is stationary and ergodic with respect to time shifts $\theta_z\{e(t), \mathbb{R}\} = \{e(t - z), t \in \mathbb{R}\}$, for all $z \in \mathbb{R}$. As a result, the condition (1.1) is guaranteed by Birkoff's ergodic theorem [13, 18].

Under the above conditions, we can construct a stationary workload regime of the system operating under the MaxProjection schedule. We first establish below a key monotonicity property of the workload vector, which is later leveraged in the Loynes' construction [12] of the stationary regime.

For clarity, we adopt the following notation in this section. Let $W^q(t; z, \vec{w})$ denote the workload of queue $q \in \mathbf{Q}$ at time t , given that system started at time $z < t$ with initial workload \vec{w} and operates under the MaxProjection schedule. The following proposition then holds.

Proposition 4.1 (Workload Monotonicity) For any fixed $s, t \in \mathbb{R}$ with $s < t$ and any initial workloads $\vec{0} \leq \vec{w}_1$ and $\vec{0} \leq \vec{w}_2$, we have that

$$\vec{w}_1 \leq \vec{w}_2 \Rightarrow \vec{W}(t; s, \vec{w}_1) \leq \vec{W}(t; s, \vec{w}_2) \quad (4.1)$$

almost surely. Therefore, the workload is a path-wise increasing function of its initial value.

Proof: On arbitrarily fixed traffic and environment sample paths, compare at each point in time within the interval $(s, t]$ the evolution of two copies of the system, \mathcal{S}^1 with initial workload \vec{w}_1 and \mathcal{S}^2 with initial workload \vec{w}_2 . Due to the nature of the traffic and environment traces and the structure of the MaxProjection policy, it can be easily seen that we can partition $(s, t]$ into a union of disjoint intervals $(T_k, T_{k+1}]$ with $s = T_0 < T_1 < T_2 < \dots < T_k < T_{k+1} < \dots < T_{K-1} < T_K = t$, such that for any $k \in \{1, 2, \dots, K\}$ the following are true:

1. There is no job arrival in any queue in (T_k, T_{k+1}) .
2. There is no change of environment mode in (T_k, T_{k+1}) .
3. The set \mathcal{Q}_k^1 of queues receiving non-zero processor bandwidth (hence, having non-zero service rate) under MaxProjection in system \mathcal{S}^1 remains invariant throughout (T_k, T_{k+1}) . The same holds for the set \mathcal{Q}_k^2 defined analogously for \mathcal{S}^2 . Note that in general $\mathcal{Q}_k^1 \neq \mathcal{Q}_k^2$.

The epochs T_k correspond to occurrences (possibly simultaneous) of one or more events of the following types: 1) job arrival, 2) change of mode, 3) change in the set of queues receiving service under MaxProjection in \mathcal{S}^1 or \mathcal{S}^2 or both.

In order to prove the proposition it suffices to show that it holds in any arbitrarily chosen interval $(T_k, T_{k+1}]$ and then apply induction on consecutive intervals. The reason is that for any intermediate time $z \in (s, t]$ we have:

$$\vec{W}(t; s, \vec{w}) = \vec{W}(t; z, \vec{W}(z; s, \vec{w})) \quad (4.2)$$

for any initial workload \vec{w} , as can be easily deduced from the structure of the system. Because of this property we can simply prove the proposition by induction on consecutive intervals of the type (T_k, T_{k+1}) defined above. We proceed in this direction below.

Working in an arbitrarily chosen (T_k, T_{k+1}) interval, let $\vec{v}_1 = \vec{W}(T_k; s, \vec{w}_1)$ be the workload of system \mathcal{S}^1 at epoch T_k and $\vec{v}_2 = \vec{W}(T_k; s, \vec{w}_2)$ that of \mathcal{S}^2 . We show below that:

$$\vec{v}_1 \leq \vec{v}_2 \Rightarrow \vec{W}(z; T_k, \vec{v}_1) \leq \vec{W}(z; T_k, \vec{v}_2) \quad (4.3)$$

for every $z \in (T_k, T_{k+1})$. Since throughout (T_k, T_{k+1}) both sets of queues \mathcal{Q}_k^1 and \mathcal{Q}_k^2 (receiving service in \mathcal{S}^1 and \mathcal{S}^2 correspondingly) do not change, we examine the following three cases:

1. For every queue $q \notin \mathcal{Q}_k^1 \cup \mathcal{Q}_k^2$ we have that

$$W^q(z; T_k, \vec{v}_1) = v_1^q \leq v_2^q = W^q(z; T_k, \vec{v}_2) \quad (4.4)$$

for all $z \in (T_k, T_{k+1})$. The reason is that no queue in $q \notin \mathcal{Q}_k^1 \cup \mathcal{Q}_k^2$ receives service, hence, their workloads remain unchanged in this time interval.

2. Consider now the case where $\mathcal{Q}_k^1 \subseteq \mathcal{Q}_k^2$, and choose a queue $q_o \in \mathcal{Q}^1$ (hence, $q_o \in \mathcal{Q}_2$ also). Observe that according to the MaxProjection schedule - using (2.5), (2.9), (2.10) - we get

$$W^{q_o}(z; T_k, \vec{v}_1) = v_1^{q_o} - \frac{t - T_k}{r_m^{q_o} N^1} \leq v_2^{q_o} - \frac{t - T_k}{r_m^{q_o} N^2} = W^{q_o}(z; T_k, \vec{v}_2) \quad (4.5)$$

for all $z \in (T_k, T_{k+1})$, because $v_1^{q_o} \leq v_2^{q_o}$ and the normalizing constants obey the inequality:

$$N^1 = \sum_{q \in \mathcal{Q}_k^1} \frac{1}{[r_m^q]^2 a^q} \leq \sum_{q \in \mathcal{Q}_k^2} \frac{1}{[r_m^q]^2 a^q} = N^2 \quad (4.6)$$

since $\mathcal{Q}_1 \subseteq \mathcal{Q}_2$, so N^2 includes more terms than N^1 . That is, $W^{q_o}(z; T_k, \vec{v}_1)$ in \mathcal{S}^1 recedes faster than $W^{q_o}(z; T_k, \vec{v}_2)$ in \mathcal{S}^2 and, hence, remains below the latter. Finally, since the environment mode m and the sets \mathcal{Q}_k^1 and \mathcal{Q}_k^2 stay invariant in (T_k, T_{k+1}) , we have

$$r_m^q a^q W^q(z; T_k, \vec{v}_1) \leq r_m^{q_o} a^{q_o} W^{q_o}(z; T_k, \vec{v}_1) \leq r_m^{q_o} a^{q_o} W^{q_o}(z; T_k, \vec{v}_2) = r_m^q a^q W^q(z; T_k, \vec{v}_2) \quad (4.7)$$

for all $q \in \mathcal{Q}_k^2 = \mathcal{Q}_k^1 \cup \mathcal{Q}_k^2$. The left inequality and right equality follow from the nature of the MaxProjection schedule, while the middle inequality holds because of (4.5). From the left inequality and right equality, cancelling out the r_m^q and a^q , we get the required result in this case.

3. Finally, consider the case where $\mathcal{Q}_k^1 - \mathcal{Q}_k^2$ is non-empty and choose a queue $q_* \in \mathcal{Q}_k^1 - \mathcal{Q}_k^2$. Observe that for all $z \in (T_k, T_{k+1})$ we have

$$W^{q_*}(z; T_k, \vec{v}_1) \leq W^{q_*}(z; T_k, \vec{v}_2), \quad (4.8)$$

because queue $q_* \in \mathcal{Q}_k^1 - \mathcal{Q}_k^2$ receives no service in \mathcal{S}^2 although it does in \mathcal{S}^1 . Therefore, its workload in \mathcal{S}^1 recedes, while it remains unchanged in \mathcal{S}^2 . Then, since the environment mode m and the sets \mathcal{Q}_k^1 and \mathcal{Q}_k^2 stay invariant in (T_k, T_{k+1}) , we see that

$$r_m^q a^q W^q(z; T_k, \vec{v}_1) \leq r_m^{q_*} a^{q_*} W^{q_*}(z; T_k, \vec{v}_1) \leq r_m^{q_*} a^{q_*} W^{q_*}(z; T_k, \vec{v}_2) < r_m^q a^q W^q(z; T_k, \vec{v}_2) \quad (4.9)$$

for every $q \in \mathcal{Q}_k^1 \cup \mathcal{Q}_k^2$ and every $z \in (T_k, T_{k+1})$. The left and right inequalities follow from the nature of the MaxProjection schedule, while the middle one holds because of (4.8). From the left and right inequalities, cancelling out the r_m^q and a^q , we get the required result in this case.

This completes the proof of (4.3). At time T_{k+1} any combination of the following events may occur: 1) a change in the environment mode, 2) a change of \mathcal{Q}_k^1 or \mathcal{Q}_k^2 or both, and 3) job arrivals to one or more queues. Since the workload process is right-continuous and has left limits, the inequality (4.3) extends to T_{k+1} , thus, holding for every $z \in (T_k, T_{k+1}]$. Inductively applying (4.3) on consecutive intervals, we obtain the desired result, which completes the proof of the proposition. \blacksquare

The previous proposition establishes the domination property of MaxProjection, that is: if a system starts with component-wise larger workload than another at some time, it will always have higher workload. This key property is leveraged in a Loynes' construction of a stationary regime, as follows.

On any fixed sample path of the environment and traffic processes, consider the evolution of two copies of the system operating under MaxProjection. The first system starts empty at time τ' and the second one starts empty at time $\tau \leq \tau'$. Consider now the workload vector $\vec{W}(t; \tau', \vec{0})$ of the first system at time t and also the workload vector $\vec{W}(t; \tau, \vec{0})$ of the second system correspondingly, when $\tau \leq \tau' \leq t$. We have,

$$\vec{W}(t; \tau', \vec{0}) \leq \vec{W}(t; \tau', \vec{w}) = \vec{W}(t; \tau, \vec{0}) \quad (4.10)$$

for every $\tau \leq \tau' \leq t$, where $\vec{w} = \vec{W}(\tau'; \tau, \vec{0})$. The left inequality is obtained from Proposition 4.1 and the fact that $\vec{0} \leq \vec{w}$. The right equality is obtained by the fact that $\vec{W}(t; \tau', \vec{w}) = \vec{W}(t; \tau', \vec{W}(\tau'; \tau, \vec{0})) = \vec{W}(t; \tau, \vec{0})$ for all $\tau \leq \tau' \leq t$, due to structure of the system.

From inequality (4.10) we see that for any fixed time t the vector $\vec{W}(t; \tau, \vec{0})$ increases component-wise as τ decreases, when the system starts empty (on any fixed sample path, almost surely). Therefore, the limit

$$\lim_{\tau \rightarrow -\infty} \vec{W}(t; \tau, \vec{0}) = \vec{W}_*(t) \quad (4.11)$$

is well-defined for every $t \in \mathbb{R}$ almost surely, hence, the stochastic process $\{\vec{W}_*(t), t \in \mathbb{R}\}$ is well-defined. Moreover, note that

$$\theta_z \vec{W}(t; \tau, \vec{0}) = \vec{W}(t + z; \tau + z, \vec{0}) \quad (4.12)$$

for all $z \in \mathbb{R}$ path-wise. Indeed, recall that θ_z shifts each sample path backwards by z , so $\theta_z \vec{W}(t; \tau, \vec{0})$ is the workload computed on the shifted path in the interval $[\tau, t]$. This is the same as keeping the sample path fixed and shifting the workload computation window to $[\tau + z, t + z]$. Taking the limits in (4.11) as $\tau \rightarrow -\infty$ and using (4.12) we get

$$\theta_z \vec{W}_*(t) = \vec{W}_*(t + z) \quad (4.13)$$

for every $z \in \mathbb{R}$. This implies that the process $\{\vec{W}_*(t), t \in \mathbb{R}\}$ is stationary and ergodic with respect to time shifts θ_z , since the environment and traffic processes are such on the probability space (Ω, \mathcal{F}, P) .

The previous discussion provides the basis for investigating how the finite-dimensional distributions of the workload process converge at large times. We discuss below the case of two-dimensional distributions, but clearly the discussion extends to higher dimensional cylinder sets. Let $\mathcal{B}_1, \mathcal{B}_2$ be arbitrarily chosen Borel sets in \mathbb{R}_{0+}^Q and T_1, T_2 arbitrarily chosen times. Then, we have

$$\boxed{\lim_{t \rightarrow \infty} P[\vec{W}(t + T_1; 0, \vec{0}) \in \mathcal{B}_1, \vec{W}(t + T_2; 0, \vec{0}) \in \mathcal{B}_2] = P[\vec{W}_*(T_1) \in \mathcal{B}_1, \vec{W}_*(T_2) \in \mathcal{B}_2]}. \quad (4.14)$$

This limit is obtained by the following steps. First, from (4.12) we have $\theta_z \vec{W}(t + T_1; 0, \vec{0}) = \vec{W}(t + T_1 + z; z, \vec{0})$ and $\theta_z \vec{W}(t + T_2; 0, \vec{0}) = \vec{W}(t + T_2 + z; z, \vec{0})$ for every $z \in \mathbb{R}$, so we can set $z = -t$ to get

$$\theta_{-t} \vec{W}(t + T_1; 0, \vec{0}) = \vec{W}(T_1; -t, \vec{0}) \quad \text{and} \quad \theta_{-t} \vec{W}(t + T_2; 0, \vec{0}) = \vec{W}(T_2; -t, \vec{0}) \quad (4.15)$$

Now, since the environment and traffic processes are stationary and ergodic with respect to time-shifts θ , we get for $z = -t$ that (first equality below):

$$\begin{aligned} P[\vec{W}(t + T_1; 0, \vec{0}) \in \mathcal{B}_1, \vec{W}(t + T_2; 0, \vec{0}) \in \mathcal{B}_2] &= P[\theta_{-t}\vec{W}(t + T_1; 0, \vec{0}) \in \mathcal{B}_1, \theta_{-t}\vec{W}(t + T_2; 0, \vec{0}) \in \mathcal{B}_2] \\ &= P[\vec{W}(T_1; -t, \vec{0}) \in \mathcal{B}_1, \vec{W}(T_2; -t, \vec{0}) \in \mathcal{B}_2] \end{aligned} \quad (4.16)$$

The second equality is due to (4.15). But from (4.11) we see that $\lim_{t \rightarrow \infty} \vec{W}(T_1; -t, \vec{0}) = \vec{W}_*(T_1)$ and $\lim_{t \rightarrow \infty} \vec{W}(T_2; -t, \vec{0}) = \vec{W}_*(T_2)$ almost surely, so

$$\lim_{t \rightarrow \infty} P[\vec{W}(T_1; -t, \vec{0}) \in \mathcal{B}_1, \vec{W}(T_2; -t, \vec{0}) \in \mathcal{B}_2] = P[\vec{W}_*(T_1) \in \mathcal{B}_1, \vec{W}_*(T_2) \in \mathcal{B}_2] \quad (4.17)$$

From (4.16) and (4.17), we immediately get (4.11). Note that the rationale used for getting the result for two-dimensional distributions extends directly to any finite-dimensional ones.

The above discussion shows that the workload of the system starting empty at time zero will stochastically converge to the stationary regime $\{\vec{W}_*(t), t \in \mathbb{R}\}$. However, one should notice that although $\lim_{\tau \rightarrow -\infty} \vec{W}(t; \tau, \vec{0}) = \vec{W}_*(t)$ is well-defined, it may actually be finite or infinite. If $\vec{\rho} \in \Phi_{\mathcal{E}}$, we expect that $\vec{W}_*(t) < \infty$ almost surely (for all $t \in \mathbb{R}$), so the system is stable. Alternatively, if $\vec{\rho} \in \bar{\Phi}_{\mathcal{E}}$, we expect that $\vec{W}_*(t) = \infty$ almost surely (for all $t \in \mathbb{R}$), so the system is unstable. We do not pursue any further the issue of stochastic stability, since the analysis here is done under rather restrictive assumptions, compared to trace-based stability studied in the previous section. Instead, we turn our attention to some important generalizations of the results below.

5 Model Extensions

An important strong point of the trace-based stability analysis pursued in Section 3 is that it can be directly extended to more general models of queueing systems and networks, as follows.

5.1 The Multi-Processor Case

Up to now, we have assumed that there is only one processor operating on the queues. Consider now the case where there are K distinct processors, indexed by $k \in \mathbf{K} = \{1, 2, \dots, K\}$. Without any loss of generality, each processor's total service capacity is scaled to 1. We provide a brief sketch of how the results extend to the multi-processor case below. Let $\mathbf{R}_m^k = \text{diag}\{r_m^{1,k}, r_m^{2,k}, \dots, r_m^{q,k}, \dots, r_m^{Q,k}\}$ be the diagonal matrix (with positive elements) of differential service rates per unit of bandwidth of the k^{th} processor on the various queues, when the environment is in state $m \in \mathbf{M}$. If \vec{x}^k is the bandwidth allocation vector of the k^{th} processor, then $\sum_{k \in \mathbf{K}} \mathbf{R}_m^k \vec{x}^k$ is the service rate vector of the system.

The MaxProjection schedule is now applied to each processor $k \in \mathbf{K}$ individually, choosing its bandwidth allocation vector by

$$\vec{x}_m^k \{\vec{W}\} = \left\{ \vec{x} \in \mathbf{X} : \left\langle \mathbf{R}_m^k \vec{x}, \mathbf{A}\vec{W} \right\rangle \text{ is maximal} \right\} \quad (5.1)$$

where \mathbf{A} is the diagonal positive matrix used before. This choice of bandwidth allocation vectors maximizes

$$\left\langle \sum_{k \in \mathbf{K}} \mathbf{R}_m^k \vec{x}^k, \mathbf{A} \vec{W} \right\rangle = \sum_{k \in \mathbf{K}} \left\langle \mathbf{R}_m^k \vec{x}^k, \mathbf{A} \vec{W} \right\rangle, \quad (5.2)$$

that is, the projection of the total service rate on the scaled workload vector $\mathbf{A} \vec{W}$. The stability region of the multi-processor system becomes

$$\Phi_{\mathcal{E}} = \left\{ \vec{\rho} > \vec{0} : \langle \vec{\rho}, \hat{\eta} \rangle < \sum_{k \in \mathbf{K}} \sum_{m \in \mathbf{M}} \pi_m \max_{\vec{x} \in \mathbf{X}} \left\langle \mathbf{R}_m^k \vec{x}, \hat{\eta} \right\rangle \text{ for every unit vector } \hat{\eta} \geq \vec{0} \right\}. \quad (5.3)$$

and the per-processor MaxProjection schedule (5.1) will stabilize the system when $\vec{\rho} \in \Phi_{\mathcal{E}}$. The proof of this fact proceeds along the lines of the proof of Proposition 3.2.

5.2 Continuous Environment Modes

Let us again consider the case of a single processor and sketch out how the model would be extended when the environment modes m are continuous, say, $m \in \mathbf{M} = (0, 1]$. We assume that the following limit

$$\pi(a, b] = \lim_{t \rightarrow \infty} \frac{\int_0^t \mathbf{1}_{\{e(u) \in (a, b]\}} du}{t} \quad (5.4)$$

exists for any interval $(a, b] \subseteq \mathbf{M} = (0, 1]$ and $\pi(a, b]$ forms a measure on \mathbf{M} . Then, the stability region becomes:

$$\Phi_{\mathcal{E}} = \left\{ \vec{\rho} > \vec{0} : \langle \vec{\rho}, \hat{\eta} \rangle < \int_{\mathbf{M}} \max_{\vec{x} \in \mathbf{X}} \left\langle \mathbf{R}_m \vec{x}, \hat{\eta} \right\rangle d\pi \text{ for every unit vector } \hat{\eta} \geq \vec{0} \right\}. \quad (5.5)$$

Proving rate stability of the system operating under the MaxProjection schedule (1.10) when $\vec{\rho} \in \Phi_{\mathcal{E}}$ would proceed along the lines of the proof of Proposition 3.2. However, it is not clear how stochastic stability would be treated in this case, since the proof of Proposition 4.1 would clearly collapse.

5.3 Feed-Forward Networks of Modulated Nodes

The trace-based modelling approach is most appropriate for studying networks of interacting nodes, operating in random environments. When a job completes service at a node, it joins a queue at a down-stream one. It is assumed that the network is feed-forward or acyclic, that is, nodes can be classified in levels, such that job-routes visit progressively higher-level nodes. Each node is of the type studied before and the MaxProjection schedule is applied locally at each node. There is flow conservation through each rate-stable node, preventing any flow deficit at its output.

The interesting fact is that distributed application of the MaxProjection schedule at each node maximizes the overall network throughput. The network extension framework is a direct amendment of that found in[1] (for a node of different nature and structure, of course). Therefore, we do not replicate it here.

6 Generalized MaxProjection Schedules

Finally, it is interesting to discuss how MaxProjection can be extended to a more general family of schedules that stabilize the system when $\vec{\rho} \in \Phi_{\mathcal{E}}$. Recall that in the original MaxProjection schedule (1.10) the matrix $\mathbf{A} = \text{diag}\{a^1, a^2, \dots, a^q, \dots, a^Q\}$ is diagonal and $a^q > 0$ for each $q \in \mathbf{Q}$. What should be the structure of a more general $Q \times Q$ matrix \mathbf{H} , such that the bandwidth allocation vectors

$$\vec{x}_m\{\vec{W}\} = \left\{ \vec{x} \in \mathbf{X} : \langle \mathbf{R}_m \vec{x}, \mathbf{H} \vec{W} \rangle \text{ is maximal} \right\}, \quad (6.1)$$

maximize the throughput? To address this question we consider the proof of Proposition 3.2 and see for which matrices \mathbf{H} it would still go through.

We expect the schedule (6.1) to have the following basic behavior. If for some queue q the workload $W^q \rightarrow \infty$ increases unboundedly while the workloads $W^p < \infty$ for all other queues $p \in \mathbf{Q} - \{q\}$ remain finite and bounded, we expect that the generalized MaxProjection schedule (6.1) will gradually switch all the processor bandwidth to queue q under stress and $x^q \rightarrow 1$. Consider now the matrix $\mathbf{H} = \{h(i, j), i, j \in \mathbf{Q}\}$ and write the expression to be maximized in (6.1) as:

$$\langle \mathbf{R}_m \vec{x}, \mathbf{H} \vec{W} \rangle = \sum_{i \in \mathbf{Q}} r_m^i x^i \left(\sum_{j \in \mathbf{Q}} h(i, j) W^j \right). \quad (6.2)$$

In order for $x^q \rightarrow 1$ as $W^q \rightarrow \infty$, we must eventually have $r_m^q \left(\sum_{j \in \mathbf{Q}} h(q, j) W^j \right) > r_m^p \left(\sum_{j \in \mathbf{Q}} h(p, j) W^j \right)$ for every $p \in \mathbf{Q} - \{q\}$ as $W^q \rightarrow \infty$ and $W^p < \infty$. A sufficient condition for this to be true is that for every fixed $q \in \mathbf{Q}$ we have:

$$h(q, q) > h(p, q) \text{ for every } p \in \mathbf{Q} - \{q\}. \quad (6.3)$$

That is, each diagonal element of \mathbf{H} dominates any off-diagonal one on the same column. For example, any non-singular matrix \mathbf{H} with positive diagonal elements and negative off-diagonal ones would suffice.

In order for the last step of the proof of Proposition 3.2 to go through, we also need \mathbf{H} to be symmetric and positive-definite. We can expect that generalized MaxProjection schedules based on such matrices \mathbf{H} provide rate-stability of the system when $\vec{\rho} \in \Phi_{\mathcal{E}}$. This family of stabilizing schedules deserves further study. We do not elaborate more on it here, since it is a subject of current further research.

7 Conclusions and Final Remarks

The problem of allocating processor bandwidth to parallel queues has been addressed, when environment fluctuations affect or modulate the service rate per unit bandwidth per queue (and perhaps even the job arrival rates, structure etc.) The stability region of such systems has been characterized. It has been shown that there is a family of processor schedules, which maximize the throughput by inducing the maximal projection of the instantaneous service rate vector on the workload vector (or its linear transformation) of the system. Rate-stability (or structural trace-stability) has been established under very general assumptions on

the traffic and environment traces, absent any probabilistic structure. Stochastic stability has been discussed under more restrictive assumptions of stationarity and ergodicity.

It should be noted that the provided stability analysis addresses - beyond the immediate queueing system under consideration - the more general stability problem of the deterministic (or stochastic) controlled differential equation

$$\frac{d\vec{W}(t)}{dt} = \vec{s}(t) - \mathbf{I}\{\vec{W}(t)\}\mathbf{R}(t)\vec{x}(t), \quad (7.1)$$

which is non-linear (because of $\mathbf{I}\{\vec{W}\}$) and has time-varying structure. The key challenges faced are that 1) the function $\vec{s}(t)$ has δ -jumps and 2) we want to characterize the stability status of the differential equation through highly non-localized (global) rate characterizations of the traffic (1.4) and environment (1.1) traces. The established workload monotonicity property provides the connection to stochastic stability. It is not clear how to treat such differential equations in general under such broad assumptions.

Several additional research threads are currently being pursued, including generalizing the MaxProjection schedule to broader families of bandwidth allocation schemes, applying the trace-based modelling methodology to other more general queueing network structures, etc.

Acknowledgments: The authors would like to thank an anonymous referee whose comments improved the presentation of the manuscript.

References

- [1] Armony, M. and N. Bambos (2001). Queueing Dynamics and Maximal Throughput Scheduling in Switched Processing Systems. To appear in Queueing Systems: Theory and Applications.
- [2] Baccelli, F. and P. Bremaud (1994). Elements of Queueing Theory. Springer, New York.
- [3] Bambos, N. and G. Michailidis (1995). On the Stationary Dynamics of Parallel Queues with Random Server Connectivities. Proceedings of 34th Conference on Decision and Control (CDC), pp. 3638-3643, New Orleans, LA.
- [4] Bambos, N. and G. Michailidis (2001). Queueing Networks of Random Link Topology; Stationary Dynamics of Maximal Throughput Schedules. Technical Report NetLab-2001-10/02, Stanford University, October 2001. Submitted for publication.
- [5] Bambos, N. and G. Michailidis (2001). Processor Scheduling in Fluctuating Environments. Adaptive Bandwidth Allocation for Throughput Maximization. Technical Report SU NETLAB-2001-11/01, Engineering Library, Stanford University, Stanford
- [6] Bambos, N. and G. Michailidis (2002). On Parallel Queueing with Random Server Connectivity and Routing Constraints. Probability in Engineering and Informational Sciences, v.16, pp. 185-203.
- [7] Brandt, A., Franken., P. and B. Lisek (1990). Stationary Stochastic Models. Wiley.
- [8] M. El-Taha and S. Stidham (1999). Sample-Path Analysis of Queueing Systems. Kluwer, Boston.

- [9] Horn, R.A. and C.R. Johnson (1985). *Matrix Analysis*. Cambridge University Press.
- [10] Lott, C. and D. Teneketzis (2000). On the Optimality of an Index Rule in Multichannel Allocation for Single-Hop Mobile Networks with Multiple Service Rates. *Probability in Engineering and Informational Sciences*, v. 14, pp. 259-297.
- [11] Loomis, L. and S. Sternberg (1990). *Advanced Calculus*. Revised Edition. Jones-Bartlett.
- [12] Loynes, R.M. (1962). The Stability of a Queue with Non-independent Inter-arrival and Service Times. *Proc. Cambridge Philosophical Society*, v. 58, 497-520.
- [13] Petersen, K. (1983). *Ergodic Theory*. Cambridge University Press.
- [14] Shakkottai, S. and Stolyar, A.L. (2000), Scheduling for Multiple Flows Sharing a Time-Varying Channel: The Exponential Rule. Preprint.
- [15] Stolyar, A.L. (2001). MaxWeight Scheduling in a Generalized Switch: State Space Collapse and Equivalent Workload Minimization under Complete Resource Pooling. Preprint.
- [16] Tassiulas, L. and A. Ephremides (1993). Dynamic Server Allocation to Parallel Queues with Randomly Varying Connectivity. *IEEE Trans. on Info. Theory*, v. 39, pp. 466-478.
- [17] Tassiulas, L. (1997). Scheduling and Performance Limits of Networks with Constantly Changing Topology. *IEEE Trans. on Info. Theory*, v. 43, pp. 1067-1073.
- [18] Walter, K. (1981). *Ergodic Theory*. Springer.
- [19] Wasserman, K. and T.L. Olsen (2001). On Mutually Interfering Parallel Servers subject to External Disturbances. *Operations Research*, v. 49, pp. 700-709.