Report Number: WUCS-89-22

1989-01-01

Resource Allocation for Markovian Queueing Networks: The Partial Information Case

Andreas D. Bovopoulos

In this paper a resource allocation algorithm is presented for Markovian queueing networks operating under state dependent routing and flow control. The state of the network is described by the total number of packets in the network. In addition, in this paper, a new proof based on feasible direction techniques is presented for a classical result concerning Jacksonian networks. Specifically, the result states that for a Jacksonian network whose Norton equivalent is a concave increasing function with respect to the number of packets in the network, the optimal flow control is a window flow control with the random point,... Read complete abstract on page 2.

Resource Allocation for Markovian Queueing Networks: The Partial Information Case

Andreas D. Bovopoulos

Complete Abstract:

In this paper a resource allocation algorithm is presented for Markovian queueing networks operating under state dependent routing and flow control. The state of the network is described by the total number of packets in the network. In addition, in this paper, a new proof based on feasible direction techniques is presented for a classical result concerning Jacksonian networks. Specifically, the result states that for a Jacksonian network whose Norton equivalent is a concave increasing function with respect to the number of packets in the network, the optimal flow control is a window flow control with the random point, if it exists, at the end of the window. The result is proven for two distinct optimization criteria.

RESOURCE ALLOCATION FOR MARKOVIAN QUEUEING NETWORKS: THE PARTIAL INFORMATION CASE

Andreas D. Bovopoulos

WUCS-89-22

Department of Computer Science
Washington University
Campus Box 1045
One Brookings Drive
Saint Louis, MO 63130-4899

Resource Allocation for Markovian Queueing Networks: The Partial Information Case

Andreas D. Bovopoulos
Department of Computer Science
Washington University, St. Louis, MO 63130-4899
e-mail: andreas@patti.wustl.edu

ABSTRACT

In this paper a resource allocation algorithm is presented for Markovian queueing networks operating under state dependent routing and flow control. The state of the network is described by the total number of packets in the network. In addition, in this paper, a new proof based on feasible direction techniques is presented for a classical result concerning Jacksonian networks. Specifically, the result states that for a Jacksonian network whose Norton equivalent is a concave increasing function with respect to the number of packets in the network, the optimal flow control is a window flow control with the random point, if it exists, at the end of the window. The result is proven for two distinct optimization criteria.

Index Terms: flow control, norton equivalent, optimization, partial information, queueing networks, resource allocation, routing.

Resource Allocation for Markovian Queueing Networks: The Partial Information Case

1. Introduction

In [BOV98b] the optimal resource allocation for a Markovian network was derived under the condition that the controller had complete information about the state of the network. In this paper resource allocation algorithms for a Markovian network are derived under the assumption that at any given moment the network controller has only knowledge about the total number of packets in the network. In practice this information can be easily obtained using acknowledgment packets.

Each of the problems analyzed in this paper is reduced to a *centralized* optimization problem. The network controller has only *partial information* about the activities inside the network. In [BOV89a], [BOV89b] and [BOV89c], the same problem is investigated but with a controller using different information patterns.

The first problem that is analyzed is the derivation of the optimal flow control problem of a product form network under two distinct constraint optimization criteria. We present a new proof of the classical result proven in [LAZ83], that the optimal flow control of a Jacksonian network whose Norton equivalent [WAL88] is a concave increasing function with respect to the number of packets in it, is a window flow control with the random point, if it exists, at the end of the window.

In this paper we also derive an algorithm for the derivation of good state dependent routing and flow control parameters under two different optimization criteria. Specifically, although a network operating under state dependent routing does not have a product form solution, we approximate it with one that has a product form solution. This approximation makes the analysis of the network tractable.

This paper is organized as follows. In Section 2, the problem formulation is introduced. In Sections 3 and 4, the optimal flow control problem is analyzed under two distinct optimization criteria. In Section 5 properties of the optimal flow control are proven. In Section 6, a resource allocation algorithm that computes good routing and flow control parameters under two distinct optimization criteria is derived. In Section 7, the algorithm introduced in Section 6, is applied to a particular example.

2. The Statement of the Problem

Each of the I processors of a queueing network has an infinite number of buffers and serves packets with an exponential service rate. Let μ^i be the service rate of the i^{th} processor, $i \in I$. The routing and flow control parameters are a function of the number of packets in the network. Let $\mathbf{R}_{\mathrm{k}} = [r_{\mathrm{k}}^{ij}]$ be the $(I+1) \times (I+1)$ routing matrix $(0 \leq i \leq I,\ 0 \leq j \leq I)$ when there are k packets in the network. In this notation, packets join the network at node i with probability r_{k}^{0i}. Upon completion of service at node i, packets leave the network with probability r_{k}^{i0} or are routed from node i to node j with probability r_{k}^{ij}. Let λ_k refer to the arrival rate when there are k packets in the network. The evolution of the queueing network is described by the stochastic process

$$\mathbf{Q}_t^* = (Q_t^1, \cdots, Q_t^I) \quad ,$$

where Q_t^i refers to the number of packets at node i, $1 \leq i \leq I$. The state space of the system is given by

$$E^* = \{\mathrm{k} = (k_1, \cdots, k_I) | 0 \leq k_i, \quad i = 1, 2, \cdots, I\} \quad .$$

In what follows, the states of $\mathbf{Q}_t^*$ will be aggregated to form a new state space. A new process Q_t is defined by

$$Q_t = Q_t^1 + \cdots + Q_t^I \quad .$$

The state space of Q_t is given by

$$E = \{k_1 + \cdots + k_I | 0 \leq k_i, \quad i = 1, 2, \cdots, I\} \quad .$$

Definition 1. $\lambda = (\lambda_k),\ k \in E$, denotes the control.
Definition 2. The class of controls $\lambda = (\lambda_k),\ k \in E$, that satisfies the peak constraint

$$0 \leq \lambda_k \leq c \quad ,$$

is called *admissible.*

The decision concerning the controller's policy requires the introduction of an optimization criterion that is based on the information available to the controller. In this paper two different criteria are utilized.

First Criterion: ([LAZ83]) Maximize the throughput of the network so that the expected time delay of a packet in the network does not exceed an upper bound:

$$\max_{EQ - TE\gamma \leq 0} E\gamma \quad . \tag{2.1}$$

Observe that the time delay constraint is not written in the form $\frac{EQ}{E\gamma} \leq T$, since $\frac{EQ}{E\gamma}$ is not defined if $\lambda_k = 0$, for all $k \in E$. Instead, using Little's formula, the criterion is written in the form $EQ - TE\gamma \leq 0$, where $EQ - TE\gamma$ is a continuous differentiable function with respect to the arrival rates.

Second Criterion: Minimize the expected time delay of a packet in the network so that the expected throughput does not fall bellow a lower bound Γ:

$$\min_{E\gamma \geq \Gamma} E\tau \tag{2.2}$$.

This criterion was first introduced in [BOV87]. Its analysis in the subsequent sections is the first systematic investigation of its properties.

3. Optimal Flow Control under the First Optimization Criterion

In this section a new proof is presented of the classical result proven in [LAZ83], that the optimal flow control of a Jacksonian network whose Norton equivalent [WAL88] is a concave increasing function with respect to the number of packets in it, is a window flow control with the random point, if it exists, at the end of the window.

Let the $1 \times (I+1)$ matrix $\Theta \stackrel{def}{=} [\theta^0 \; \theta^1 \; \cdots \; \theta^I]$ be the solution of the traffic flow equations

$$\Theta = \Theta \mathbf{R} \quad ,$$

where $\theta^0 = 1$.

The service rate of the process Q_t is the conditional service rate of the network given that there are Q_t packets in the network [HSI86]. For product form networks this is given by the well known Norton equivalent [WAL88]. Let

$$g_k \stackrel{def}{=} \sum_{k_1+k_2+\cdots+k_I=k} \prod_{j=1}^{I} \left(\frac{\theta^j}{\mu^j} \right)^{k_j} \quad , \tag{3.1}$$

for all l, $l \geq 0$, where $0 \leq k_i$, for $i = 1, \cdots, I$, and for all k, $k \geq 1$. If k is the total number of packets in processors $1, 2, \cdots, I$, then the Norton equivalent, symbolized by ν_k, is given by

$$\nu_k \stackrel{def}{=} \frac{g_{k-1}}{g_k} \quad , \tag{3.2}$$

for all k, $k \geq 1$.

In this section, we assume that the routing parameters $\mathbf{R}$ are fixed. Therefore the problem of optimal resource allocation under the first optimization criterion is reduced to a flow control problem.
For the solution of the flow control problem *a prime nonlinear optimization methodology* is used ([LUE84]). The idea behind such an approach is to start from an initial feasible point and to move to a better one. We prove that such an approach is highly effective in the context of flow control.

Observe that when $\lambda_0 = 0$, the corresponding operating point is feasible because $EQ - TE\gamma = 0$. The current feasible point is $\lambda_k = 0$ for $k \geq 0$. The next operating point should be derived by the solution of the following optimization problem:

$$\max_{\lambda_0}(\frac{\lambda_0}{\nu_1}) \tag{3.3}$$

under the constraints

$$(1 - T\nu_1)(\frac{\lambda_0}{\nu_1}) \leq 0 \quad , \tag{3.4}$$

and

$$0 \leq \lambda_0 \leq c \quad . \tag{3.5}$$

In order for a nonzero feasible solution of the problem to exist, $1 - T\nu_1 \leq 0$. If the previous inequality is not fulfilled, then the initial feasible point of the optimization problem is the optimal point as well.

Let us assume in the sequel that $T \geq \frac{1}{\nu_1}$. From Equations 3.3-3.5 it is concluded that the next feasible point is $\lambda_0 = c$, and $\lambda_k = 0$, for $k \geq 1$. Using feasible direction techniques, a better operating point can be derived from this feasible point. The next feasible point can be derived by formulating a linear program using the methodology introduced in [BOV85], [BOV89b], and by permitting at most two packets to enter the network. In this case the variables λ_k for $k = 0$ and $k = 1$ are unknowns, and the initial feasible point is the state $(\lambda_0, \lambda_1) = (c, 0)$. Such an approach leads to the optimal solution of the problem.

Let the probability that there are k packets in the network be given by p_k. Let r_k^1 refer to the probability that a new incoming packet joins the network, while there are k packets in the network. Further, let r_k^0 refer to the probability that a new incoming packet is rejected when there are k packets in the network. Obviously,

$$r_k^0 + r_k^1 = 1 \ .$$

Let $x_k \stackrel{\text{def}}{=} p_k r_k^1$ and $y_k \stackrel{\text{def}}{=} p_k r_k^0$.

$E\gamma_N$ and $E\tau_N$ denote the average throughput and average time delay, respectively, of the network given that at most N packets can be in the network at any given moment. Thus

$$E\gamma_N \;=\; \sum_{k=1}^{N} p_k \nu_k \;=\; \sum_{k=1}^{N} (x_k + y_k)\, \nu_k \quad , \tag{3.6}$$

and

$$EQ_N \;=\; \sum_{k=1}^{N} p_k k \;=\; \sum_{k=1}^{N} (x_k + y_k)\, k \quad . \tag{3.7}$$

The global balance equations (GBEs) are given by the following equations:

$$p_k r_k^1 c \;=\; p_{k+1} \nu_{k+1} \quad ,$$

or, equivalently,

$$x_k c \;=\; (x_{k+1} + y_{k+1}) \nu_{k+1} \;\; \textit{for all } k, \;\; 0 \leq k \leq N-1 \quad . \tag{3.8}$$

Proposition 3.1 *The optimal flow control parameters λ_k, $k \in E$, are given by the equations:*

$$\lambda_k \;=\; c\left(1 - \frac{x_k}{x_k \;+\; y_k}\right) \quad , \tag{3.9}$$

for all i, $1 \leq i \leq I$, where (x_k, y_k), $k \in E$, is the solution of the following iterative algorithm:

Step 0: *N=1.*

Iteration:

Step 1: *For the current value of N, solve the following linear optimization problem:*

$$max \sum_{k=1}^{N} (x_k + y_k)\, \nu_k \tag{3.10}$$

under the following constraints:

$$\sum_{k=1}^{N} (x_k + y_k)\, k \leq T \sum_{k=1}^{N} (x_k + y_k)\, \nu_k \quad , \tag{3.11}$$

$$x_k c = (x_{k+1} + y_{k+1}) \nu_{k+1} \quad for\ 0 \leq k \leq N-1 \quad ,$$

$$\sum_{k=0}^{k=N} (x_k + y_k) = 1 \quad , \tag{3.12}$$

where

$$x_k \geq 0 \ and\ y_k \geq 0 \ for\ 0 \leq k \leq N \quad . \tag{3.13}$$

Step 2: *If* $E\gamma_N = E\gamma_{N-1}$, *stop; the derived flow control is optimal. Else,* $N := N + 1$, *and repeat all the steps of the iteration, using the optimal solution of the linear program as the initial feasible point of the next iteration.*

Proof: This proposition is an application of Proposition 4.2. of [BOV89b].

4. Optimal Flow Control under the Second Optimization Criterion

As mentioned above, there are cases for which we require the minimization of the time delay such that the throughput is greater than or equal to a given lower bound, referred to as Γ. In the sequel we present a methodology that reduces the derivation of the optimal flow control policy to a linear optimization problem.

We introduce the following transformations:

$$x_k^* \stackrel{\text{def}}{=} g x_k \quad for\ all\ k, \ \ 0 \leq k \leq N \quad , \tag{4.1}$$

and

$$y_k^* \stackrel{\text{def}}{=} g y_k \quad for\ all\ k, \ \ 0 \leq k \leq N \quad , \tag{4.2}$$

where

$$g \stackrel{\text{def}}{=} \frac{1}{\sum_{k=1}^{N} (x_k + y_k) \nu_k} \quad . \tag{4.3}$$

From (4.3) we see that

$$\sum_{k=1}^{N} (x_k^* + y_k^*) \nu_k = 1 \quad . \tag{4.4}$$

Equation 3.6 for the throughput then takes the form

$$E\gamma_N \;=\; \frac{1}{g}\sum_{k=1}^{N}(x_k^*+y_k^*)\,\nu_k = \left(\sum_{k=1}^{N}(x_k^*+y_k^*)\,\nu_k\right)\left(\sum_{l=0}^{N}x_l^*+y_l^*\right)^{-1} \quad . \tag{4.5}$$

The expected time delay is given by

$$E\tau_N = \sum_{k=1}^{N}(x_k^*+y_k^*)\,k \quad , \tag{4.6}$$

and the GBEs (3.8) become

$$x_k^* c \;=\; \left(x_{k+1}^* \;+\; y_{k+1}^*\right)\nu_{k+1} \quad \mathit{for\ all}\ k,\; 0 \le k \le N-1 \quad . \tag{4.7}$$

Proposition 4.1 *The optimal flow control parameters $\lambda_k, k \in E$, are given by the equations:*

$$\lambda_k \;=\; c\left(1-\frac{x_k^*}{x_k^*+y_k^*}\right) \quad , \tag{4.8}$$

where (x_k^*, y_k^*), $k \in E$, is the solution of the following iterative algorithm:

Step 0: *N=1.*

Iteration:

Step 1:

$$min\sum_{k=1}^{N}(x_k^* \;+\; y_k^*)\,k \tag{4.9}$$

under the following constraints:

$$\sum_{k=1}^{N}(x_k^*+y_k^*)\,\nu_k \;=\; 1 \quad , \tag{4.10}$$

$$1 \ge \Gamma\sum_{k=0}^{N}\left(x_{k+1}^* \;+\; y_{k+1}^*\right) \quad , \tag{4.11}$$

$$x_k^* c = \left(x_{k+1}^* + y_{k+1}^*\right) \nu_{k+1} \tag{4.12}$$

for all k, $0 \le k \le N-1$,

where

$$x_k^* \geq 0 \text{ and } y_k^* \geq 0 \text{ for } 0 \leq k \leq N \quad . \tag{4.13}$$

Step 2: *If the linear program has a feasible solution; stop. The derived flow control is optimal. Else, $N := N+1$, and repeat all the steps of the iteration.*

5. Properties of the Optimal Flow Control Policy under the Two Criteria

In this section we present structural properties of the optimal solution of the two linear programs presented in Sections 3 and 4.

Proposition 5.1. *The optimal solution of each of the two linear programs contains at most one random point. Hence the optimal solution of each linear program is of the form:*

$$\lambda_k = \begin{cases} c & \text{if } 0 \leq k \leq L-1 \text{ and } k \neq m \\ 0 < \lambda_m \leq c & \text{if } k{=}m \\ 0 & \text{if } L \leq k \leq N-1 \end{cases}$$

Proof : The proof of this proposition is identical to the proof of Proposition 4.3 in [BOV89b].

■

In the sequel a proof is presented [WEI82] of the fact that with concave increasing service rates (as in the case of Jacksonian networks [SHA86],[WAL88],) the optimal flow control under the first and second optimization criteria is of a window type with the random point, if it exists, at the end of the window. In [LAZ83] this property was proven only under the first optimization criterion.

Proposition 5.2. *If ν_k is a concave increasing function with respect to k, the optimal flow control under the first and second optimization criteria is of window type with the random point, if it exists, corresponding to the last packet of the window. Thus, the optimal flow control is of the form*

$$\lambda_k = \begin{cases} c & \text{if } 0 \le k \le L-2 \\ 0 < \lambda_{L-1} \le c & \text{if } k = L-1 \\ 0 & \text{if } L \le k \quad . \end{cases}$$

Proof : Let us assume that under the current policy N packets can enter into the network. Let $(\lambda_0, \cdots, \lambda_{N-1})$ and $(\lambda_0^*, \cdots, \lambda_{N-1}^*)$ correspond to two different control policies. We show how to choose $c \ge \lambda_m^* > \lambda_m$, $\lambda_{m+1}^* < \lambda_{m+1}$, and $\lambda_k^* = \lambda_k$ for all k, $k \ne m$ and $k \ne m+1$, such that $E\gamma_N^* \ge E\gamma_N$ and $E\tau_N^* \le E\tau_N$. Observe that

$$(p_0^*, \cdots, p_N^*) = ((1-\delta)p_0, \cdots, (1-\delta)p_{m-1}, (1+\epsilon)p_m, (1-n)p_{m+1}, \cdots, (1-n)p_N)$$

We choose δ, ϵ, and n, such that

$$\sum_{k=0}^{N} p_k \;=\; 1$$

and

$$EQ_N^* \;=\; EQ_N \quad .$$

In other words,

$$\delta \sum_{k=0}^{m-1} p_k \;+\; n \sum_{k=m+1}^{N} p_k = \epsilon p_m$$

and

$$\delta \sum_{k=0}^{m-1} kp_k \;+\; n \sum_{k=m+1}^{N} kp_k \;=\; \epsilon m p_m \quad .$$

It is easy to see that if $\epsilon > 0$, then $\delta > 0$ and $n > 0$. Letting

$$\alpha_k = \begin{cases} \frac{\delta p_k}{\epsilon p_m} & \text{if } 0 \le k < m \\ \frac{n p_k}{\epsilon p_m} & \text{if } m < k \le N \end{cases} ,$$

we have $\alpha_k > 0$, for $k \ne m$ with

$$\sum_{k \ne m} \alpha_k = 1$$

and

$$\sum_{k \neq m} k\alpha_k \ = \ m \quad .$$

By the concavity of ν_k with respect to k,

$$\sum_{k \neq m} \nu_k \alpha_k \leq \nu_m \quad .$$

Thus,

$$E\gamma_N^* \geq E\gamma_N \quad ,$$

and

$$E\tau_N^* \leq E\tau_N \quad .$$

Furthermore, because $n > 0$,

$$p_N^* < p_N \quad . \tag{5.1}$$

■

The feasible direction technique is greatly simplified by the use of structural properties regarding the next feasible point. If ν_k is a concave increasing function of k, the optimal feasible point which permits L packets to enter the network is of the form

$$\lambda_k \ = \ \begin{cases} c & \text{for } 0 \leq k \leq L-2 \\ 0 < \lambda_{L-1} \leq c & \text{for } k = L-1 \end{cases}$$

under the first and second optimization criteria. In this case the following simplified iterative algorithms can be introduced.

Iterative Algorithm for the First Optimization Criterion

Step 0: *$L := 1$. Set $\lambda_0 := c$ and $\lambda_k := 0$ for all k, $k \geq 1$. Check to see whether $E\tau_1 <= T$. If yes, continue to Step 1. Otherwise stop; no packets can enter into the network.*

Step 1: *$L := L+1$. Set $\lambda_k := c$ for all k, $0 \leq k \leq L-1$. Check whether $E\tau_L \leq T$. If yes, repeat Step 1. Else, find the exact rate (which is between 0 and c) with which the last packet should be accepted and which results in*

$EQ - TE\gamma = 0$; the resulting flow control is the optimal flow control; thus stop.

Iterative Algorithm for the Second Optimization Criterion

Step 0: $L := 0$. *Set* $\lambda_k := 0$ *for all* k, $k \geq 0$.

Step 1: $L := L+1$. *Set* $\lambda_k := c$, *for all* k, $0 \leq k \leq L-1$. *If* $E\gamma_L \geq \Gamma$, *go to Step 2; else, repeat Step 1.*

Step 2: *If* $E\gamma = \Gamma$, *the resulting flow control is the optimal flow control; thus stop. Else, find the value of* λ_{L-1} *(which is between 0 and c) which results in* $E\gamma = \Gamma$; *the resulting flow control is the optimal flow control; stop.*

In the last part of this section we would like to comment on the complexity of the introduced algorithms.

Observe that each time we solve the linear program, we use the solution of the previous step as an initial feasible point. The proximity of such a point to the solution of the current linear program results in an accelerated execution of the linear programs. This systematic approach to optimal flow control requires obviously less computation than any other approach. Since the dimensionality of the problem is kept at a minimum, this iterative algorithm is optimal for the solution of the above class of problems.

For the case in which the service rate is a concave increasing function with respect to the number of packets in the network, *there is a one-to-one mapping between the flow control solution given under the first and the second optimization criteria.*

The optimization problems

$$\max_{E\tau \leq T} E\gamma$$

and

$$\min_{E\gamma \geq \Gamma} E\tau \quad ,$$

have the same class of window flow control solutions. If the window size is the same, it is easy to see that

$$\Gamma T = EQ \quad .$$

6. Resource Allocation Algorithm Based on Product Form Approximation

We approximate the original network with one that has a product form solution.

6.1. State Dependent Routing

For fixed, state independent routing, the Norton equivalent ν_k of the network analyzed is a concave increasing function with respect to the number of packets k [SHA86], [WAL88].

The near optimal routing inside the network is computed by maximizing the value of the Norton equivalent of the approximate product form network for each state in the state space E.

Let R_k^* be the routing that maximizes the value ν_k, for all k, $k \geq 1$. R_k^* is the near optimal state dependent routing of the original network. Let $(\nu_k)_{R_k^*}$ be the value of the ν_k for the routing parameters R_l^* for all $k \geq 1$.

Proposition 6.1

(i) $(\nu_k)_{R_k^*}$ *is an increasing function of* k *all* k, $k \geq 1$.

(ii) $\frac{k}{(\nu_k)_{R_k^*}}$ *is an increasing function of* k, *for all* k, $k \geq 1$.

(iii) $\min_{R_k} \nu_k$ *is a concave increasing function of* k, *for all* k, $k \geq 1$.

Proof:

(i)

$$(\nu_k)_{R_k^*} \leq (\nu_{k+1})_{R_k^*} \leq (\nu_{k+1})_{R_{k+1}^*}$$

Thus, $(\nu_k)_{R_k^*}$ is an increasing function of k.

(ii) Let us assume that $\frac{k}{(\nu_k)_{R_k^*}} > \frac{k+1}{(\nu_{k+1})_{R_{k+1}^*}}$. It is known [WAL88] that $\frac{k}{(\nu_k)_{R_{k+1}^*}} \leq \frac{k+1}{(\nu_{k+1})_{R_{k+1}^*}}$. The previous two relations suggest that $\frac{k}{(\nu_k)_{R_k^*}} > \frac{k}{(\nu_k)_{R_{k+1}^*}}$. The last relation furthermore implies that $(\nu_k)_{R_k^*} < (\nu_k)_{R_{k+1}^*}$, which is incorrect. Thus, $\frac{k}{(\nu_k)_{R_k^*}}$ *is an increasing function of* k, *for* $k = 1, 2, \cdots, N$.

(iii) The proof of this statement is straightforward.

■

Thus the near optimal flow control of a network operating under the worst possible routing is a window flow control with a random point, if it exists, at the end of the window. This statement relates to the lower bound of the performance of the network.

Two issues should be analyzed before the introduction of algorithms designed to improve the routing and flow control of the network.

(i) *What is the effect of an increase of the value of* ν_k*?* In Proposition 10.2 it is proven that an increase in ν_k results in a decrease in the expected number of packets EQ. Furthermore if λ_k is a decreasing function of the number of packets in the system, then $E\gamma$ is increasing, and consequently $E\tau$ is decreasing.

(ii) *What is the effect of an increase in* λ_k*, the rate with which packets enter in the network?* In Proposition 10.1 it is proven that an increase in λ_k results in an increase in the expected number of packets EQ. Furthermore if ν_k is an increasing function of the number of packets in the system, then $E\gamma$ is increasing. If $\frac{k}{\nu_k}$ is increasing, $E\tau$ is increasing as well.

Because $(\nu_k)_{R_k^*}$ and $\frac{k}{(\nu_k)_{R_k^*}}$ are both increasing functions of k for $k \geq 1$, the optimal flow control is given by

$$\lambda_k = \begin{cases} c & \text{if } 0 \leq k \leq L-1, \quad k \neq m \\ 0 < \lambda_m \leq c & \text{if } k = m \\ 0 & \text{if } L \leq k \end{cases} .$$

If $(\nu_k)_{R_k^*}$ is a concave increasing function with respect to k, then the random point, if it exists, is at the end of the window [LAZ83].

6.2. Resource Allocation under the Two Optimization Criteria

From Proposition 10.2(i) and (iv), it is known that any increase in the value of the Norton equivalent leads to an increase in the throughput of the network and a decrease in the expected time delay of the packets. It it thus advantageous to operate the network under the routing parameters which result in the maximum value of the Norton equivalent $(\nu_k)_{R_k}$ for all k, $k = 1, 2, \cdots, L$. The maximization of the value of the Norton equivalent is a convex nonlinear optimization problem that can be solved using a Flow Deviation Algorithm [KOB83].

In this subsection, combining results presented above, we develop resource allocation algorithms that evaluate a network's state dependent routing and flow control parameters under the first and second optimization criteria.

Iterative Algorithm :
Let $L(j)$ be the maximum i for which $\lambda_i \neq 0$ (*i.e.*, let $L(j)$ be the window size after the j^{th} iteration).

Let $E\gamma(j)$ be the expected throughput when the network is subject to the routing parameters and flow control derived during the j^{th} iteration.

Step 0 : Set $L(0) := 0$, $L(1) := 1$, *and* $n := 1$.

n$^{\text{th}}$ Iteration :

Step 1 : Using the Flow Deviation Algorithm [KOB83], for each value of k *such that* $L(n-1) < k \leq L(n)$, *find the optimal routing parameters* R_k^* *which maximize the Norton equivalent* ν_k, *by solving the following convex nonlinear optimization problem:*

$$\min \frac{1}{E\gamma(k)} \quad ,$$

under the constraints:

$$\sum_{j \in IN(i)} \theta^j \;=\; \sum_{j \in OUT(i)} \theta^j \quad ,$$

for every $i = 1, \cdots, I$, *where :*

IN(i) is the set of channels incoming to node i of the forward network,

OUT(i) is the set of channels outgoing from node i of the forward network.

Step 2: For the routing parameters computed in Step 1, update the flow control policy using the linear program presented in the Sections 4, 5 and 6 of the present paper. The solution of the linear program gives the new flow control of window size $L(n+1)$.

Step 3 : If $L(n+1) = L(n)$, *go to Step 4. Else,* $n := n+1$, *go to Step 1.*

Step 4 : For the routing and flow control parameters computed above, solve the global balance equations. From the equilibrium probabilities, compute the expected throughput and the expected time delay of the packets in the network.

7. Applications

In this section the resource allocation algorithm introduced in Subsection 6.2 is applied and thoroughly examined in an example modeling a network of parallel processors.

Let us assume that we wish to derive good state dependent routing and flow control parameters of a network of parallel processors depicted in Figure 7.1, in order to maximize the network throughput. The service rates of the processors are $\mu^1 = 2\ packets/sec$, $\mu^2 = 1\ packet/sec$, and $\mu^3 = .5\ packets/sec$. Packets arrive into the network with state dependent arrival rate λ_l, where $0\ packets/sec \leq \lambda_l \leq 8\ packets/sec$.

The flow control parameters are computed by first approximating the service rate of the network when k packets are in it, by $(\nu_k)_{R_k^*}$, for all values of k, $k \geq 1$. Step 2, of the algorithm then computes an approximate throughput and expected

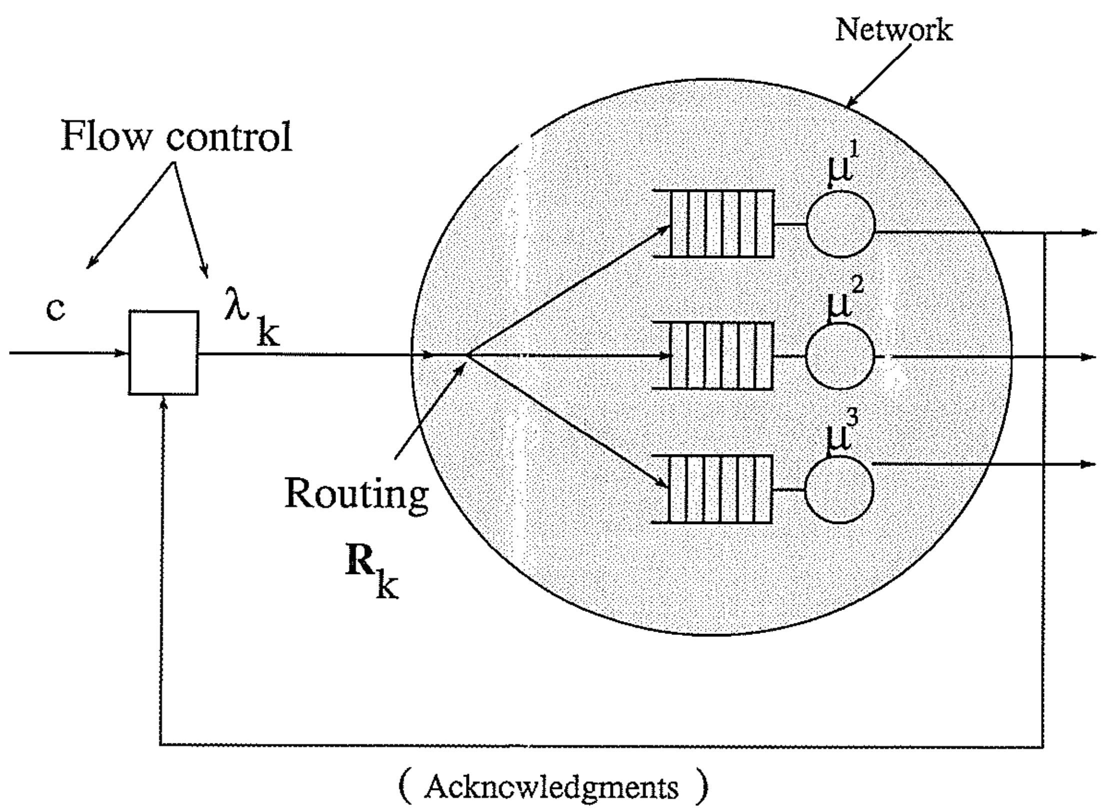

FIGURE 7.1. A network of three parallel processors

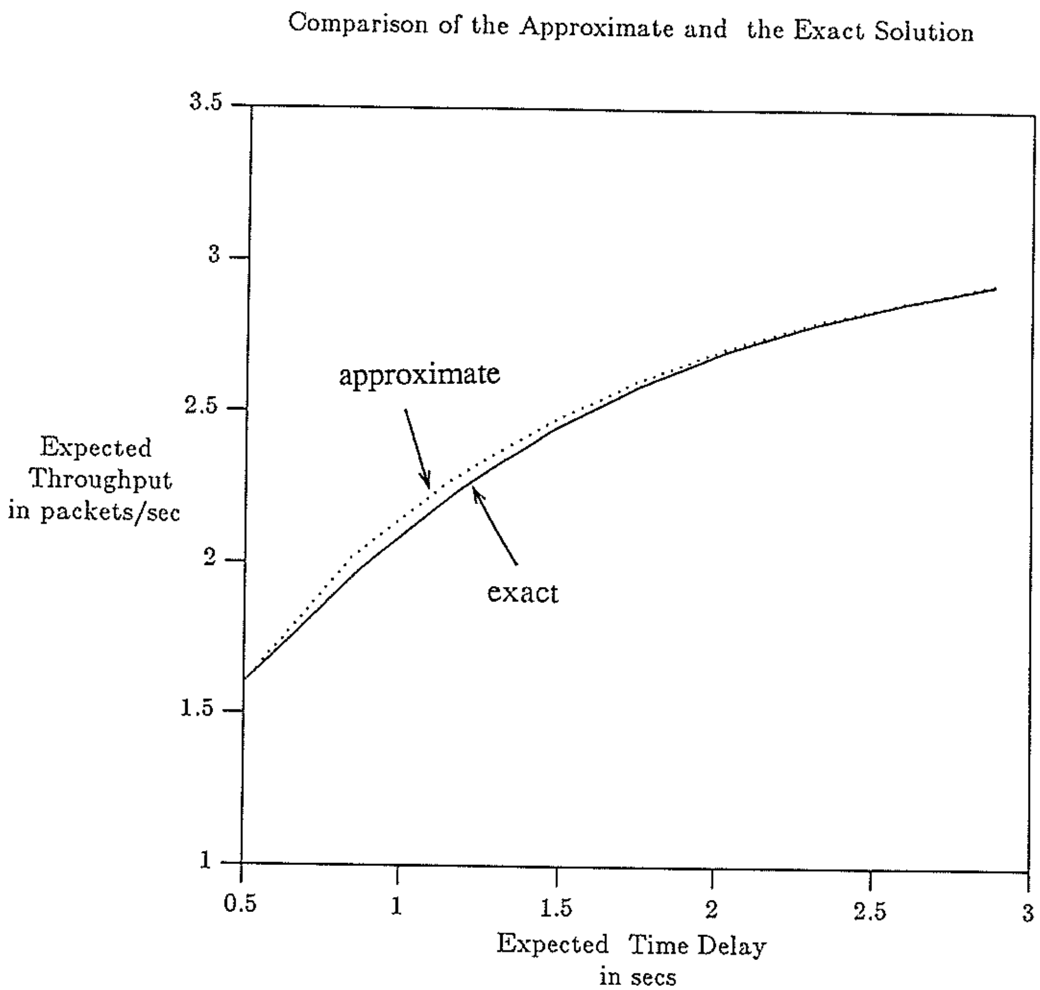

FIGURE 7.2. Expected throughput versus expected time delay.

time delay of the packets in the network. The dotted curve of Fig. 7.2, shows this approximated network behavior. This curve is computed by applying the algorithm introduced in Subsection 6.2 up to Step 3. By ignoring Step 4 of the algorithm, the complexity of the algorithm is substantially reduced. The solid curve of Figure 7.2, shows the exact (through the application of Step 4 of the algorithm) network performance. Notice that as the network load increases the relative error between the two curves decreases.

8. Conclusions

A study of the resource allocation problem for Markovian queueing networks operating under state dependent routing and flow control has been presented.

In addition, a new proof based on feasible direction techniques has been presented for a classical result concerning Jacksonian networks. Specifically, the result states that for a Jacksonian network whose Norton equivalent is a concave increasing function with respect to the number of packets in the network, the optimal flow control is a window flow control with the random point, if it exists, at the end of the window. The result has been proven for two distinct optimization criteria.

9. References

[BOV85] Bovopoulos, A. D. and Lazar, A. A., "Optimal Routing and Flow Control of a Network of Parallel Processors with Individual Buffers," *Proceedings of the 23rd Annual Allerton Conference on Communication, Control and Computing,* University of Illinois at Urbana-Champaign, October 2-4, 1985, pp. 564-573.

[BOV87] Bovopoulos, A. D. and Lazar, A. A., "Optimal Load Balancing for Markovian Queueing Networks," *Proceedings of the* 30^{th} *Midwest Symposium on Circuits and Systems,* Syracuse University, August 17-18,1987, pp. 1428-1436.

[BOV89a] Bovopoulos, A. D. and Lazar, A. A., "Load Balancing Algorithms for Jacksonian Networks with Acknowledgement Delays", *Proceedings of IEEE INFOCOM'89,* Ottawa, Canada, April 25-27, 1989

[BOV89b] Bovopoulos, A. D. and Lazar, A. A., "Optimal Resource Allocation for Markovian Queueing Networks: The Complete Information Case," Technical Report WUCS-89-21, Department of Computer Science, Washington University.

[BOV89c] Bovopoulos, A. D., "On the Effect of Delayed Feedback Information on Network Performance," Technical Report WUCS-89-23, Department of Computer Science, Washington University.

[HSI86] Hsiao, M. T. and Lazar, A. A., "Norton's Equivalent Revisited," *Proceedings of the Twentieth Annual Conference on Information Sciences and Systems,* Princeton University, Princeton, NJ, March, 1986.

[JAC57] Jackson, J. R., "Networks of Waiting Lines," *Operat. Res.,* Vol. 5, 1957, pp. 518-521.

[KOB83] Kobayashi, H. and Gerla, M., "Optimal Routing in Closed Queueing Networks," *ACM Transactions on Computer Systems*, Vol. 1, No. 4, Nov. 1983, pp. 294-310.

[LAZ83] Lazar, A. A., "Optimal Flow Control of a Class of Queueing Networks in Equilibrium," *IEEE Transactions on Automatic Control,* Vol. AC-28, No. 11, November 1983, pp. 1001-1007.

[LUE84] Luenberger, D. G., *Linear and Nonlinear Programming*, 2nd edition, Addison-Wesley Publishing company, 1984.

[SHA86] Shantikumar, J. G. and Yao, D. D., "The effect of increasing service rates in a closed queueing network," *Journal of Applied Probability,* Vol. 23, 1986, pp. 474-483.

[WAL88] Walrand, J., *An Introduction to Queueing Networks,* Prentice Hall, New Jersey, 1988.

[WEI82] Weiss, G., Personal Communication, April 1982.

10. Appendix

Monotonicity Properties for a Controlled Finite Birth-Death Process

In order to advance our understanding about the behavior of a single-class network subject to state dependent flow control, we analyze the behavior of the closed network depicted in Fig. 10.1. If k is the total number of packets in the upper processor, the upper and the lower processors serve the packets with state dependent rates μ_k and λ_k respectively. Let $E\gamma_N$ be the throughput of this network. Let EQ_N^c and $E\tau_N^c$ be the expected number of packets and the expected time delay of the packets, respectively, in the lower processor of the network shown in Fig. 10.1. Similarly, let EQ_N and $E\tau_N$ be the expected number of packets and the expected time delay of the packets, respectively, in the upper processor of the network depicted in Fig. 10.1. Observe that

$$E\gamma_N = \frac{EQ_N}{E\tau_N} = \frac{EQ_N^c}{E\tau_N^c} = \frac{N}{E\tau_N + E\tau_N^c}$$

Proposition 10.1

(i) *If μ_k is an increasing function of k, the expected throughput $E\gamma_N$ is increasing in λ_k^* for $k = 0, 1, \cdots, N-1$.*

(ii) *The expected number of packets EQ_N is increasing in λ_k^* for $k = 0, 1, \cdots, N-1$.*

(iii) *The expected number of packets EQ_N^c is decreasing in λ_k^* for $k = 0, 1, \cdots, N-1$.*

(iv) *If μ_k is an increasing function of k, the expected time delay of the packets in the controller $E\tau_N^c$ is decreasing in λ_k^* for $k = 0, 1, \cdots, N-1$.*

(v) *If $\frac{k}{\mu_k}$ is an increasing function of k, the expected time delay $E\tau_N$ is increasing in λ_k^* for $k = 0, 1, \cdots, N-1$.*

Proof :

Let $\mathbf{p}^* \stackrel{\text{def}}{=} (p_0^*, \cdots, p_N^*)$ correspond to the control $\lambda^* \stackrel{\text{def}}{=} (\lambda_0^*, \cdots, \lambda_N^*)$, and let

$\mathbf{p}^{\#} \stackrel{\text{def}}{=} (p_0^{\#}, \cdots, p_N^{\#})$ correspond to the control $\lambda^{\#} \stackrel{\text{def}}{=} (\lambda_0^{\#}, \cdots, \lambda_N^{\#})$. Let us assume that $\lambda_i^{\#} \geq \lambda_i^*$ for all $i \in E_1$. Then

$$\frac{p_k^*}{p_{k-1}^*} = \frac{\lambda_{k-1}^*}{\mu_k} \leq \frac{\lambda_{k-1}^{\#}}{\mu_k} = \frac{p_k^{\#}}{p_{k-1}^{\#}} \quad ,$$

for $k = 1, \cdots, N-1$, from which it follows that

$$\sum_{i=k}^{N} p_i^* \leq \sum_{i=k}^{N} p_i^{\#} \quad .$$

Since the μ_k are increasing,

$$\sum_{k=1}^{N} \mu_k p_k^* \leq \sum_{k=1}^{N} \mu_k p_k^{\#} \quad ,$$

which completes the proof of (i).

(ii) $EQ_N = \sum_{k=1}^{N} kp_k^*$. The arguments of (i) hold here if k is substituted for μ_k.

(iii) $EQ_N^c = N - EQ_N$. The statement is then true because of (ii).

(iv) This statement holds because $E\tau_N^c = \frac{EQ_N^c}{E\gamma_N}$.

(v) $E\tau_N$ is a weighted average of $\frac{k}{\mu_k}$, for $k = 1, \cdots, N$, with weights $q_k^* = \frac{\mu_k p_k^*}{\sum_{i=1}^{N} \mu_i p_i^*}$. The arguments of (i) hold if q_k^* is substituted for p_k^*, and the statement follows.

■

In a similar way we can prove the following relations.

Proposition 10.2

(i) If λ_k is a decreasing function of k, the expected throughput $E\gamma_N$ is increasing in μ_k^* for $k = 0, 1, \cdots, N-1$.

(ii) The expected number of packets EQ_N^c is increasing in μ_k^* for $k = 0, 1, \cdots, N-1$.

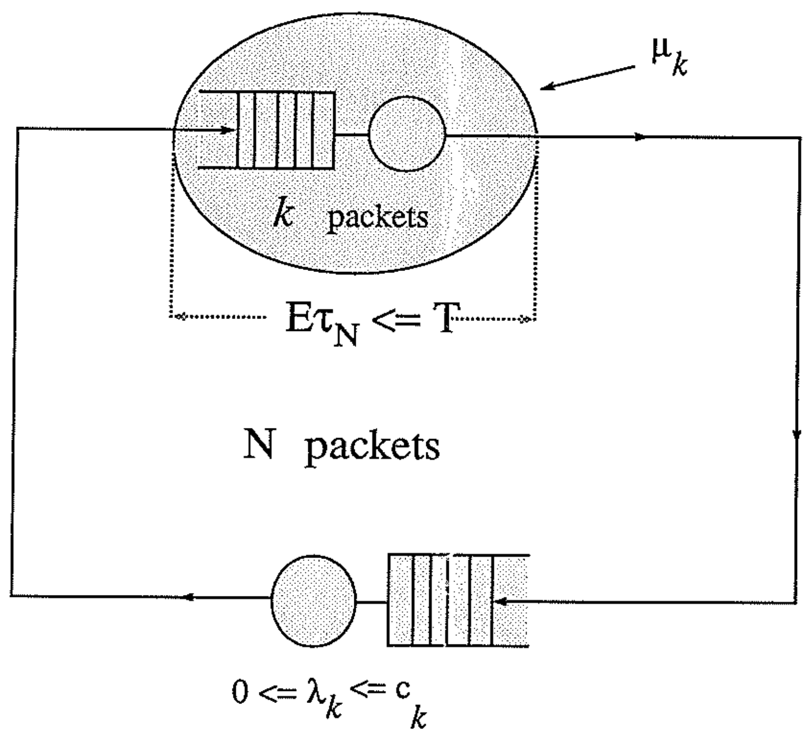

FIGURE 10.1. A single-class network which can contain at most N packets, subject to state dependent flow control.

(iii) The expected number of packets EQ_N *is decreasing in* μ_k^* *for* $k = 0, 1, \cdots, N-1$.

(iv) If λ_k *is a decreasing function of* k*, the expected time delay of the packets in the network* $E\tau_N$ *is decreasing in* μ_k^* *for* $k = 0, 1, \cdots, N-1$.

(v) If $\frac{N-k}{\lambda_k}$ *is a decreasing function of* k*, the expected time delay* $E\tau_N^c$ *is increasing in* μ_k^* *for* $k = 0, 1, \cdots, N-1$.

Sharon,

Hi. The CICS sponsors have asked Dr. Ball for copies of all the papers we have published, and I have been given the enviable job of collecting three copies of each of these, tracking down the authors, etc. For the tech reports, Dr. Ball wants "official" copies, so the sponsors will know they are tech reports.

Here are the tech report numbers that Dr. Ball and I need for these CICS sponsors. We need three copies of each report. There's no hurry as to when we get them, but within 3-4 weeks or so...

Year	Numbers needed
89	1, 2, 3, 5, 6, 8, 9, 12, 16, 24, 26, 28, 30, 31, 41, 45, 47, 49
88	11, 16, 19, 21, 22, 24, 25, 26
87	14, 15

Let me know if you have any questions. This is the last time anyone will need copies of all these, since CICS will keep a copy of each of these in our own files, for future demands from sponsors.

Dave